Information Fusion 104 (2024) 102159

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

MultiBEATS: Blocks of eigenvalues algorithm for multivariate time series dimensionality reduction☆

Aurora González-Vidal, Antonio Martinez-Ibarra *, Antonio F. Skarmeta

Department of Information and Communication Engineering, University of Murcia, Murcia, 30100, Spain

ARTICLE INFO

Keywords:
Multivariate time series
Time series representation
Seq2VAR
Machine learning classification

ABSTRACT

Multivariate Time Series are sequences of observations taken from multiple sources. The proliferation of environments in which data is collected by means of sensors and the adoption of data-based services reveals the importance of their efficient analysis. In smart environments, the analysis of the raw Multivariate Time Series is cumbersome. The algorithms are heavy, slow, and fail to extract all the knowledge available. In that sense, representation techniques reduce the data points but maintain the inner patterns so that the information present in data persists. There exist several approaches to represent Univariate Time Series, but for the multivariate case they are scarce. We have adapted the existent univariate representation algorithm BEATS, which is based on Discrete Cosine transformation and the extraction of eigenvalues to multivariate scenarios and we have called it MultiBEATS. Our method allows flexibility with regard to data reduction and does not require labelled data. To investigate the efficacy of MultiBEATS, we selected 8 open multivariate time series datasets and compared the running time and accuracy of their classification using raw data and MultiBEATS transformed data. We have also compared our results with the representation algorithm Seq2VAR, based on autoencoders, for having a baseline. The MultiBEATS transformation reduces the execution time of classification algorithms, has improved the results in accuracy in 2 of the datasets and matches a third one. With regards to the baseline, MultiBEATS is faster than Seq2VAR and more accurate in 7 out of the 8 datasets. In conclusion, MultiBEATS is a promising approach for efficiently reducing multivariate time series data, therefore helping decision-making in dynamic smart environments.

1. Introduction

In Internet of Things (IoT) deployments and smart environments, where operational decisions can be made based on contextual and on-the-go data generated by sensors, actuators, and crowdsensing, multivariate data are commonly found. Multivariate data consists of a sequence of observations taken from multiple sources. These data streams are most often in the form of a key–vector pair with a number of observations taken at the same timestamp, each associated with a different sensor or actuator monitoring a single system. These can be considered a collection of temporally correlated univariate data streams that provide a complete view of the system being monitored and can be defined as a Multivariate Time Series (MTS).

Given that nowadays we are able to collect huge amounts of data by means of such ubiquitous systems, we encounter serious problems in managing data of such volume, velocity, and variety, and many times with missing values [1]. This has given birth to the Big Data paradigm, which relates to the techniques and technologies that aim to solve operational problems related to data of such characteristics [2]. Several approaches can be taken for aiding the storage, management, and analysis of the data and, to what volume refers, representation of data becomes crucial for Machine Learning (ML) and Artificial Intelligence (AI) applications [3]. Data gathering is only the first step to providing new intelligent services and improving the existing ones. The collected data, when fused with different sources and provided with a context, can be turned into information and then obtain knowledge by means of Ml and AI techniques. Such knowledge makes it possible to create those new services in different verticals including smart buildings [4], smart agriculture [5], and cybersecurity [6]. In those scenarios, we will typically encounter MTS that enclose the complex dynamic of the systems. By analysing these multivariate time series, we can gain

☆ This work forms part of the ThinkInAzul programme and was supported by MCIN with funding from European Union NextGenerationEU (PRTR-C17.I1) and by Comunidad Autónoma de la Región de Murcia - Fundación Séneca. It was partially funded by the HORIZON-MSCA-2021-SE-01-01 project Cloudstars (g.a. 101086248) as well as by projects GEMINI TED2021-129767B-I00 and PERSEO PDC2021-121561-I00 both financed by MCIN/AEI/10.13039/501100011033 and the European Union NextGenerationEU/PRTR.

* Corresponding author.
E-mail address: antonio.martinezi@um.es (A. Martinez-Ibarra).

https://doi.org/10.1016/j.inffus.2023.102159
Received 2 May 2023; Received in revised form 22 November 2023; Accepted 23 November 2023
Available online 25 November 2023

insights about the patterns, trends, and anomalies that may be present in the data, and use this information to make informed decisions and optimize various processes.

Traditional methods require domain experts to manually engineer features, which can be time-consuming. In contrast, representation learning seeks to automatically generate features that describe the underlying data [7]. More specifically dimensionality reduction, which involves generating compressed representations of data in order to extract the most relevant information [8] will be the scope of this work. Dimensionality reduction by means of representation techniques is especially crucial in the case of Multivariate Time Series (MTS), which often contain a large number of variables and time steps that can result in redundant information. Dimensionality reduction techniques applied in representation learning can help streamline the analysis and processing of MTS data. In this work, we are introducing MultiBEATS, a MTS dimensionality reduction algorithm inspired by the Univariate Time Series (UTS) representation algorithm named BEATS and we are proving its efficiency by comparing the classification results, accuracy and time, of several labelled real-life MTS datasets before and after the dimensionality reduction carried out by MultiBEATS. We compare with other representation algorithms for MTS, named Seq2VAR, as well. Our contributions are summarized as follows:

- Generalization of a segmentation and representation UTS algorithm for the case of MTS
- Creation of an algorithm that does not require labelled data and that maintains the MTS structure reducing their volume, therefore, it does not restrict the applications.
- Comprehensive evaluation to demonstrate the effectiveness in terms of accuracy and processing time of the proposed approach
- Comparison with the state-of-the-art MTS representation algorithm

The structure of the paper is as follows: Section 2 describes the background needed for our work and analyses other works related to the representation of MTS, especially the one with which we will compare MultiBEATS. Then, in 3 we provide a detailed description of our MTS dimensionality reduction algorithm called MultiBEATS. Section 4 describes the results of the classification approach that has been selected to test MultiBEATS. Finally, Section 5 concludes the paper and opens up the following lines of research.

2. Background and related work

The MultiBEATS algorithm is the multivariate version of our algorithm called BEATS (Blocks of Eigenvalues Algorithm for Time Series Segmentation) [9]. We are going to evaluate its performance by applying the state-of-the-art classification algorithm for MTS named Long Short Term Memory Fully Convolutional Network (LSTM-FCNs) to the transformed series and also to compare it with the representation strategy called Seq2VAR.

In that sense, together with the related work for MTS dimensionality representation, which includes a section about Seq2VAR, we find it interesting to provide some background on BEATS and univariate dimensionality reduction techniques in general together with an explanation on the LSTM-FCNs MTS classification algorithm.

2.1. Univariate dimensionality reduction techniques: BEATS and others

Given that the relationships among the variables in MTS are difficult to obtain accurately, and the high dimensionality of the variables present in MTS data mining, more research reports towards dimensionality reduction have been conducted on Univariate Time Series (UTS) than MTS [10].

Feature selection [11] and feature representation, discrete wavelet transformation [12], shape space representation [13], piecewise linear approximation [14] and symbolic approximation [15] or even using Lagrangian multiplier [16] have been used in the past to aggregate and represent UTS.

In our previous work [9], where the algorithm BEATS is presented, a more thorough review of UTS segmentation and representation methods is present.

Our BEATS methodology involves dividing time series data into blocks, which can either overlap or be non-overlapping. Each of these blocks represents a subset of the larger data structure, and BEATS synthesizes the information contained within each block independently. By doing so, BEATS is able to reduce the number of data points while still retaining the essential characteristics of the original data set, without sacrificing important information. Such synthesis is carried out by using matrix-based data aggregation, Discrete Cosine Transform (DCT), and eigenvalues characterization of the time series data.

Our transformation inherited the properties of the Karhunen–Loeve Transform (KLT) which decorrelates a signal sequence as is said to be an optimal transform because:

- It completely decorrelates the signal in the transform domain,
- It minimizes the MSE in bandwidth reduction or data compression,
- It contains the most variance (energy) in the fewest number of transform coefficients, and
- It minimizes the total representation entropy of the sequence.

In such work, we highlighted the benefits of BEATS compared with the -at the time- state-of-the-art segmentation and representation UTS algorithms. We assessed and compared the efficacy of reducing the datasets with BEATS for classification and clustering tasks, by analysing the metrics of the transformed datasets and comparing them with the metrics obtained with the raw data.

2.2. MTS dimensionality reduction and representation

Some representation methods for MTS have been thought of only for classification purposes. It is the case of Symbolic Representation for Multivariate Time Series (SMTS) [17]. The process of SMTS involves using a random forest on a MTS to divide it into leaf nodes, where each leaf node is represented by a word to create a codebook. Another random forest is then utilized with the codebook to classify the mts by utilizing each of the words. Learned Pattern Similarity (LPS) [18] is a similar model that learns the dependency patterns or “autopatterns” of time series data by modelling the relationships between its segments. This is achieved through a segmentation approach that is conceptually related to multiple lag values for autocorrelation. The resulting segments are represented as matrices, and a tree-based learning strategy for regression is utilized to discover the dependency structure among them. Subsequently, a bag-of-words (BoW) representation is generated for each time series, which encodes the learned dependency patterns and these words are used with a similarity measure to classify the unknown MTS. Finally, in order to find shapelets, that are short discriminative subsequence of time series, in MTS ShapeNet was proposed in [19]. The candidate shapelets of MTS may come from different variables of different lengths and thus cannot be directly compared. To address this challenge, ShapeNet embeds shapelet candidates of different lengths into a unified space for shapelet selection.

The Auto-Regressive (AR) kernel [20], is a method for measuring the similarity between MTS and later on classify them according to that. Another notable approach is the Auto-Regressive Forests for MTS Modelling (mv-ARF) [21], which utilizes a tree ensemble trained on different time lags. Additionally, WEASEL + MUSE [22] extracts a multivariate feature vector by applying a classical bag-of-patterns approach on each variable using various sliding window sizes to capture discrete features, words, and pairs of words. Non-discriminative features are subsequently removed using a Chi-squared test for feature selection.

Finally, a logistic classifier is used to classify the time series based on the final feature vector.

The complex pairwise dependencies among multivariate variables can be described through advanced graph-based methodologies. In such approach, each variable is treated as a node within the graph framework, and the relationships between variables are represented as edges. In [23], they introduce a method termed Multivariate Time Series Classification with Variational Graph Pooling (MTPool), that aims to derive a representation of Multivariate Time Series (MTS) data. The MTPool methodology initially constructs a graph structure based on the data, capturing spatial and temporal features of the MTS. Subsequently, a temporal convolution module is employed to further enhance the model's understanding of temporal relationships. To ensure permutation invariance, maintain input correlation, and facilitate dimensionality adjustability, a pooling layer that encompasses an "encoder–decoder" architecture, facilitating the generation of centroids related to input data is incorporated. The coarsened graph-level representations obtained through this process serve as input features for a differentiable classifier.

Let us now review more general approaches, where a representation model does not depend on the ML problem to be solved. It is the case of the temporal kernelized autoencoder (TKAE) model [24] that learns compressed representations of MTS. Such approach utilizes a deep Autoencoder with recurrent layers, employing a bidirectional recurrent neural network to capture time dependencies and generate a fixed-size vector representation of the input MTS. A dense nonlinear layer is used to merge the final states of the forward and backward networks, reducing the dimensionality. They show the benefit of the kernel alignment for learning representations in classification settings with missing data and build a one-class classifier for anomaly detection. To avoid biases from imputation, they train the architecture with a loss function that maintains pairwise similarities in the learned representations, even when dealing with missing data, through a kernel alignment procedure that aligns dot products of the representations with a kernel similarity defined in the input space.

In [10] the authors propose the MTS dimensionality reduction algorithm called Piecewise representation based on Principal Component Analysis (PPCA). PPCA segments multivariate time series into several sequences, calculates the covariance matrix for each of them in terms of the variables, and employs PCA to obtain the principal components in an average covariance matrix.

Seq2VAR [25] can be considered a more sophisticated method to learn representations of MTS and it is the method we will compare MultiBEATS with.

2.2.1. Baseline method for comparison: Seq2VAR

The observations of causal interactions between dynamic variables that are Time Series are typically modelled using vector linear autoregression (VAR). While VAR captures the causal and dynamic information in the data, it lacks a representation inference mechanism [26]. Neural networks, on the other hand, are capable of inference and can therefore be used for representation learning.

Latent variable models, such as variational autoencoders (VAE), can capture relationships among variables that are indirectly inferred through a mathematical model. However, their powerful decoder may not necessarily require meaningful latent representations to recover data. Neural relational inference (NRI) [27] is a specialized VAE architecture that explicitly captures variable interactions to explain data dynamics and it inspires the encoder from the Seq2VAR model, while VAR contributes to the design for the decoder.

The Seq2VAR model has several advantages: it represents MTS samples as the parameters of a VAR to prevent information vanishing; it uses binary sampling to build sparse representations, allowing the interpretation of non-zero entries as Granger causalities; and it implements a relational inference neural network (RINN) to improve generalization capacity. The RINN takes a sample X(s) as input and outputs a tensor $\hat{A}(s)$ [27], which is the latent representation of the MTS.

Seq2VAR can modify the encoding posterior distribution and regularization based on the task of interest. Stochastic gates are also set for each sample to obtain explicit sparsity. In addition to that, the model imposes the constraint of symmetry on the tensor, which is particularly important for physical links between variables.

For the experiments presented in their paper, two versions of Seq2VAR were trained: dense with no penalty and binary, which applies a binary mask to the encoder and referred to as binary-Seq2VAR.

The first experiment tries to assess Seq2VAR in a binary linear setting. It takes a stationary 1-order linear autoregressive model, where the linear transition matrix is a permutation matrix, and shows that the binary sampling approach is relevant for Seq2VAR model and that the inference mechanism of the model is shared by all samples and integrates the noise signal by seeing numerous noisy examples. It also shows that when noise is low, both Seq2VAR and binary-Seq2VAR offer almost perfect graph discovery. The second experiment tries to assess Seq2VAR in a physical setting by evaluating the capacity to find Granger causality graph hidden in physical data. Granger causality is a statistical concept used to determine whether one time series can predict another time series and it can be represented as a directed graph, where nodes represent the time series variables (A and B), and directed arrows indicate the flow of influence. If an arrow goes from A to B, it suggests that A Granger-causes B. The physical system consists of 10 balls and springs, each connected to the other and moving in 2D space. The classes are defined by sampling interaction graphs and two datasets are proposed, one with identical rigidity for all springs and one with variable rigidity. It is found that Seq2VAR gives good results for both homogeneous and heterogeneous rigidity graphs.

In conclusion, the Seq2VAR model is a powerful tool for time series classification, particularly in the context of multivariate time series. It addresses key shortcomings of traditional vector linear autoregression models, such as the lack of a representation inference mechanism, by leveraging the power of neural networks. Additionally, the model's ability to build sparse representations and capture Granger causalities through binary sampling, as well as the implementation of a relational inference neural network, make it a versatile and effective approach for a range of applications. The experiments presented demonstrate the model's efficacy in both binary linear and physical settings, validating its potential for practical use in time series classification.

2.3. Multivariate time series classification algorithms

There exist specific algorithms for MTS representation applied to classification problems. In our experimental framework, for evaluating the goodness of MultiBEATS, we have used a classification scenario to evaluate the gains and losses in accuracy and in processing time. Before selecting the classification strategy, we have reviewed the literature. We have reviewed the latest tendencies that consist of using deep learning and the inclusion of attention mechanisms [28].

Many of the approaches employed for MTSC are conversions of models originally designed for univariate data to handle the multivariate case. Neural networks are a natural example of this. There are currently many new deep learning architectures being proposed for time series classification. For example in [29] the multilevel discrete wavelet decomposition is used to decompose an MTS into a group of sub-MTS so as to extract multilevel time–frequency representations. Then, a deep CNN is trained for each level to learn level-specific nonlinear features.

As a special line of research in deep learning for MTS classification, attention mechanisms allow neural networks to focus on different parts of the input data when making predictions. This is especially useful when dealing with sequences of data, like sentences in natural language processing which was their original intention. Instead of blindly processing the entire sequence at once, attention mechanisms enable

the model to dynamically emphasize certain parts of the input while generating or predicting output. This can be as well utilized for time series. One of the key advantages of attention mechanisms is their ability to capture long-range dependencies in sequences and relate different elements to each other.

One example of the use of attention mechanisms for MTS is how in [30] the authors propose integrating Transformer-inspired methods into Multi-Task Sequence Classification to tackle challenges such as the identification of patterns that exist between distant data points in the series. Transformer-like models combine encoders and decoders, leveraging self-attention to capture context and mitigate gradient vanishing. The authors use Swin Transformer to address complexities, feature interactions, and long-range dependencies in MTSC. Swin Transformer introduces hierarchical feature maps and a window partitioning scheme to manage computational load, along with a shifted windowing mechanism for enhanced classification. In this same line, a Cross Attention Stabilized Fully Convolutional Neural Network is proposed in [31], where a temporal attention mechanism to extract long- and short-term memories across all time steps is followed by a variable attention step designed to select relevant variables at each time step.

2.3.1. Multivariate LSTM-FCNs for time series classification

To perform the classification of multivariate time series and to test how the application of MultiBEATS affects the performance in terms of accuracy and time, we have used a state-of-the-art method that consists of transforming the existing univariate time series classification models, the Long Short Term Memory Fully Convolutional Network (LSTM-FCN) and Attention LSTM-FCN (ALSTM-FCN), into a multivariate time series classification model by augmenting the fully convolutional block with a squeeze-and-excitation block to further improve accuracy [32]. LSTMs are a kind of Recurrent Neural Networks (RNN) that is able to address the vanishing gradient problem. LSTMs can learn temporal dependencies and the model can be designed with multiple input and output nodes, each corresponding to a different variable in the multivariate time series. However, long-term dependencies of long sequences are challenging to learn using LSTMs. In that sense, an attention mechanism can be used. The attention mechanism is designed to improve the performance of LSTMs by allowing them to selectively focus on parts of the input that are most relevant to the current output. We have chosen this classification model because it outperformed most state-of-the-art models in the datasets that we are going to use while requiring minimum preprocessing. A very decisive fact is also that the models are highly efficient at test time and small enough to deploy on memory-constrained systems.

Within the LSTM-FCN network there are several parameters that we can tune to optimize the model's performance. We have explored the values of the number of nodes in the Attention LSTM layer and its dropout rate, the first with possible values of 8 or 16 and the second with rates of 0.25 or 0.5, and the number of filters in the two unidimensional convolution layers, with values of 64, 128 or 256. By applying cross-validation with a grid composed of those hyperparameters we are able to select the configuration that suits best each dataset and avoid issues like overfitting or overall poor model performance. Other parameters that can be tuned in this type of network are the learning rate, the number of epochs, and the batch size.

3. MultiBEATS

This section describes our proposed algorithm and discusses its mathematical and analytical background. We present MultiBEATS and illustrate its various components and steps. As it will be shown, the idea of our algorithm is to extract relevant patterns of the MTS and represent the data using a smaller amount of datapoints. In order to validate our algorithm, we will use such patterns in a classification task, but it could be applied as well for clustering purposes. Predictive scenarios are out of the scope of MultiBEATS and would need further developments.

3.1. MultiBEATS construction

The Discrete Cosine Transform (DCT) provides significant dimensionality reduction for time series, improving accuracy in time series classification and clustering.

Our method, which is multivariate, will use the two-dimensional Discrete Cosine Transform (2D-DCT). The 2D-DCT is a linear transform that is used to convert a two-dimensional signal or image from the spatial domain to the frequency domain. Usually, 2D-DCT is used to represent images as the weighted sums of cosines having different horizontal and vertical frequencies. The 2D-DCT plays a key role in image compression, as it can efficiently represent image information with fewer coefficients than the original pixel values, and we are translating this property for the compression of any multivariate time series scenario.

The general equation for a 2D (N by M matrix) DCT is defined by the following equation: let N be the number of time series (rows) and M be their timesteps or variables (columns), then see Eq. (1) given in Box I.

The first step of MultiBEATS consists of separating the time series into blocks by using sliding windows of different sizes. Sliding windows is a technique that involves breaking up an input data sequence into a series of overlapping segments called *windows*, and processing each segment separately. For us, the segments will be matrices of $N \times M$ observations that can be understood as blocks. Those blocks will be reduced independently by applying 2D DCT, computing the eigenvectors of the resulting transformation, and projecting the data of the block into those for obtaining the representation. In that sense, the size of the window and the overlap, which depends directly on how many data points we want our sequence of blocks to overlap by, will determine the reduction percentage. As an example, let us say the time series have 40 timestamps. By selecting *slide* = 10 and *window* = 20, we will obtain 3 blocks of data with the indexes 1 to 20, 11 to 30 and 21 to 40. Then, having computed the eigenvectors and projected the data over them, we select the number of features that we want per block. For example, if we decide to keep 2 features, then the final data will result in 6 observations, having reduced the dataset in a 85% $(1 - \frac{6}{40})$. More generally, since the TS for the same dataset can have different lengths, we have computed the percentage of reduction using the maximum length.

$$\%\text{of reduction} = 1 - \frac{\text{final TS length}}{\max(\text{TS length})}.$$

The size of the 2D DCT transformed matrix will be the same as the size of the original one, which means that the matrix might not be squared. A variation of the standard eigenvalue and eigenvector computation is known as the singular value decomposition (SVD). The SVD is a factorization of a rectangular matrix A into the product of three matrices: $A = U\Sigma V^T$, where U and V are orthogonal matrices (meaning their columns are mutually orthogonal unit vectors) and Σ is a diagonal matrix. The diagonal elements of Σ are called the singular values of A and represent the square roots of the eigenvalues of $A^T A$ (or equivalently, of AA^T). The columns of U and V are the left and right eigenvectors, respectively.

3.2. MultiBEATS usage for classification

The lower-dimensional representation of MTS that MultiBEATS provide can be used for any analytical purposes. Once MultiBEATS is applied to the MTS, the sole modification compared to the raw data relates to the feature dimension. In that sense, the same algorithms, strategies, and tools, in general, can be used to process the series as long as those methods do not present restrictions on the needs of series

$$F(u,v) = \left(\frac{2}{N}\right)^{\frac{1}{2}} \left(\frac{2}{M}\right)^{\frac{1}{2}} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \Lambda(i)\Lambda(j) \cos\left[\frac{\pi u}{2N}(2i+1)\right] \cos\left[\frac{\pi v}{2M}(2j+1)\right] f(i,j), \text{ where } \Lambda(x) = \begin{cases} \frac{1}{\sqrt{2}} & \text{x} = 0 \\ 1 & \text{otherwise} \end{cases} \quad (1)$$

Box I.

Table 1
Datasets description.

Dataset	Num of classes	Num of variables/TS	TS length	Num of samples	Train-Test split	Source
Arabic Digits (AD)	10	13	4–93	8800	75–25	Dua and Graff [34]
AUSLAN (AU)	95	22	45–136	2565	44–66	Dua and Graff [34]
Character Trajectories (CT)	20	3	109–205	2858	10–90	Dua and Graff [34]
CMUSubject16 (CMU16)	3	62	109–205	58	25–75	CMU [35]
ECG	2	2	39–152	200	50–50	RT [36]
Japanese Vowels (JV)	9	12	7–29	640	42–58	Dua and Graff [34]
Libras	15	2	45	360	50–50	Dua and Graff [34]
Wafer	2	6	104–198	1194	25–75	RT [36]

length. In our case, in order to be able to quantify the information loss, if any, we have decided to use a supervised machine learning scenario since accuracy is easy to compute and understand. In that sense, we used labelled multivariate datasets that were ready for classification tasks using both the raw data and the MultiBEATS transformed one, having in that way a way to measure the gains in time and sometimes in accuracy or to quantify the losses in a way that a potential user could decide according to their system's resources what to do in terms of data dimensionality reduction.

4. Experiments

In this section, we present the three sets of experiments conducted on several open MTS labelled datasets, which are also introduced. All scripts and data needed to perform the experiments are publicly available for reproducibility purposes [33].

The accuracy of the classification LSTM-FCN strategy using both the raw data and the MultiBEATS transformations as input was calculated 10 times since we considered this a fairer situation compared to setting a fixed seed given the libraries that were used. Such results were averaged and also the standard deviation of the accuracy is presented.

The experiments were run on a machine with O.S. Ubuntu 20.04 with a AMD Ryzen 5 5500U with Radeon Graphics processor and 14 GB of RAM, MultiBEATS transformations were performed in R 4.1.2 and the classification algorithms and Seq2VAR were programmed in Python 3.8. For more details about the implementation, please see the GitHub repository [33].

4.1. Datasets

In the past, access to annotated MTS datasets was often limited to certain research groups or organizations due to data privacy concerns or other restrictions. However, with the growing interest in machine learning and data science, there has been an increased demand for large, high-quality datasets that can be used for training and testing machine learning models. As a result, there is now more of a push to make these types of datasets publicly available, and there are efforts underway to create and curate annotated multivariate time series datasets that can be shared with the wider research community. It is the case of the archive of Mustafa Baydogan [17], that has become widely used in the community for benchmarking MTS classification. The archive consists of 15 datasets in "mat" format. This is because the sequence length of each instance in the same dataset can be in arbitrary lengths as shown in the column *TS length* of the Table 1. In such a Table, we show the characteristics of the datasets that have been chosen for testing our algorithm, and we can see the variety in the number of classes, in the number of variables, namely, the number of Time Series that compound each observation and in the number of samples that those datasets contain. A brief description of each of them is as follows:

- Arabic Digits (AD) dataset contains time series of 13 Frequency Cepstral Coefficients (MFCCs) had taken from 44 males and 44 females Arabic native speakers between the ages 18 and 40 to represent ten spoken Arabic digits, from 0 to 9.
- AUSLAN (AU) dataset contains 27 examples of each of 95 Australian Sign Language signs that were captured from a native signer. A two hand system having two Fifth Dimension Technologies (5DT) gloves and two Ascension Flock-of-Birds magnetic position trackers, one attached to each hand was used.
- Character Trajectories (CT) dataset contains labelled samples of pen tip trajectories recorded whilst writing individual characters. All samples are from the same writer. Only characters that are written without lifting the pen during the writing process, also referred to as a single pen-down segment, were chosen for consideration.
- CMUSubject16 (CMU16) dataset is a collection of 3 different motions capture recordings (jump, run, walk) taken from a particular subject over 58 trials. Such a dataset belongs to a greater experiment on several other subjects that performed other kinds of motions including interactions with the environment or dancing. This dataset is part of the Carnegie Mellon University Graphics Lab Motion Capture Database.
- ECG dataset comprises a collection of time series data sets where each file contains the sequence of measurements recorded by one electrode during one heartbeat. Each heartbeat has an assigned classification of normal or abnormal. All abnormal heartbeats are representative of a cardiac pathology known as supraventricular premature beat.
- Japanese Vowels (JV) dataset contains multivariate time series data where nine male speakers uttered two Japanese vowels /ae/ successively. Here, one utterance by a speaker forms a time series whose length is in the range 7–29 and each point of a time series is of 12 features (12 coefficients). The goal of this classification dataset is to classify the speakers.
- LIBRAS, the Brazilian sign language, is the subject of this dataset featuring 15 distinct classes, each containing 24 instances. These classes represent various hand movement types within LIBRAS, captured as two-dimensional curves in videos. These videos showcase Libras performances by four different individuals across two sessions, with each video lasting about 7 s. Video pre-processing involves time normalization, selecting 45 frames via uniform

Table 2
LSTM-FCN classification results using raw data vs. using MultiBEATS (MBEATS).

Dataset	Reduction (%)	Parameters (nfeat, SL, WD)	Hyperparameters (nodes1, dropout1, filters1, filters2)	Accuracy (mean(sd))	Execution time + Reduction time (s)
AD raw	0	–	{16, 0.5, 256, 128}	**0.988** (0.001)	893.5
AD MBEATS R1	33.3	{2,3,3)	{8, 0.25, 128, 128}	0.954 (0.003)	357.6 + 38.0 = 395.6
AD MBEATS R2	61.1	{2,4,5}	{8, 0.25, 128, 128}	0.947 (0.002)	230.9 + 22.9 = **253.8**
AU raw	0	–	{16, 0.5, 256, 128}	**0.939** (0.005)	94.9
AU MBEATS R1	31.6	{2,4,5}	{16, 0.5, 128, 256}	0.924 (0.010)	22.3 + 20.6 = 142.9
AU MBEATS R2	44.0	{2,5,6}	{8, 0.25, 128, 256}	0.931 (0.005)	101.0 + 17.0 = **118.0**
CT raw	0	–	{8, 0.5, 64, 128}	**0.978** (0.003)	**31.6**
CT MBEATS R1	33.3	{3,4,15}	{16, 0.25, 64, 256}	0.953 (0.005)	38.6 + 41.2 = 79.8
CT MBEATS R2	42.9	{3,5,12}	{16, 0.5, 64, 256}	0.947 (0.004)	29.4 + 27.2 = 56.6
CMU16 raw	0	–	{8, 0.5, 64, 64}	**1.000** (0.000)	47.2
CMU16 MBEATS R1	23.4	{3,4,35}	{8, 0.25, 64, 64}	**1.000** (0.000)	34.5 + 64.2 = 98.7
CMU16 MBEATS R2	46.8	{3,6,10}	{8, 0.25, 64, 64}	0.966 (0.000)	29.4 + 6.3 = **35.7**
ECG raw	0	–	{16, 0.5, 64, 64}	0.837 (0.023)	**8.6**
ECG MBEATS R1	33.2	{2,3,8}	{16, 0.5, 64, 128}	0.823 (0.025)	7.8 + 2.7 = 10.5
ECG MBEATS R2	49.6	{2,4,5}	{16, 0.5, 128, 256}	**0.860** (0.000)	11.3 + 0.8 = 12.1
JV raw	0	–	{8, 0.5, 128, 164}	**0.981** (0.005)	13.3
JV MBEATS R1	29.3	{2,3,3}	{16, 0.5, 64, 128}	0.965 (0.005)	11.1 + 0.9 = **12.0**
JV MBEATS R2	46.1	{2,3,7}	{8, 0.5, 128, 256}	0.961 (0.010)	12.3 + 0.8 = 13.1
Libras raw	0	–	{16, 0.5, 256, 256}	**0.828** (0.006)	16.4
Libras MBEATS R1	20.3	{2,3,3}	{16, 0.5, 128, 256}	0.774 (0.003)	10.7 + 1.5 = **12.2**
Libras MBEATS R2	46.7	{2,3,10}	{16, 0.5, 256, 256}	0.793 (0.020)	11.3 + 1.1 = 12.4
Wafer raw	0	–	{8, 0.5, 256, 64}	0.916 (0.036)	73.0
Wafer MBEATS R1	21.3	{2,2,40}	{16, 0.25, 64, 64}	0.942 (0.044)	20.7 + 60.1 = 80.8
Wafer MBEATS R2	32,1	{2,3,3}	{8, 0.5, 128, 128}	**0.956** (0.050)	33.0 + 12.1 = **45.1**

distribution. Within each frame, centroid pixels of segmented hand objects form discrete curves, normalized in a unitary space

- Wafer dataset relates to the field of semiconductor microelectronics manufacturing. It encompasses a set of inline process control measurements that are gathered from different sensors throughout the manufacturing process of silicon wafers for semiconductor production. Each dataset within the wafer database corresponds to the measurements recorded by a single sensor during the processing of an individual wafer using a specific tool. These measurements are categorized into two classes: normal and abnormal.

4.2. Results

4.2.1. Raw vs. MultiBEATS

Firstly, we have compared the accuracy and execution time of LSTM-FCN classification using the raw dataset and the MultiBEATS dataset applied with two different reduction percentages, denoted by R1 and R2 in Table 2. It can be seen in the same table, Table 2, that we have included a hyperparameter search step for having a more fair comparison between the different approaches. In that sense, times are not fully comparable since the configuration of the neural network affects the number of computations needed and therefore the time. For the MultiBEATS transformed data, we have reported two distinct temporal measures. Firstly, we specify the duration necessary to execute the data transformation process. After that, we present the time required for conducting the classification training and inference and also show the added time. The segregation of these timings is deliberate, as it offers more information for certain scenarios. Specifically, it may be advantageous to perform data transformation before storage, thus conserving storage space. Subsequently, this transformed data can be retrieved and utilized for classification tasks, resulting in reduced computational overhead compared to utilizing raw data and in the majority of the cases losing little accuracy or even improving the results. The execution times are always reduced with one exception, that is the CT dataset, since the optimal number of filter2 and nodes1 for MultiBEATS are twice the ones for the raw data. Execution times are reduced by an average of 36% and total times are reduced by an average of 17%, while accuracy is reduced only by a 1.5%, showing the potential of MultiBEATS.

4.2.2. Raw vs. MultiBEATS for increasing reduction percentages

We have run several configurations of MultiBEATS in order to see how the reduction of features affects accuracy. In Fig. 1, we can observe the following according to accuracy and feature reduction respectively. As it can be observed in the graphs, the percentage of reductions do not fall exactly on the same number for all datasets (see x axis). This is due to the fact that the slide and window sizes are the customizable parts so we were not able to obtain an exact reduction but we consider that this also aids the visualization of the results. Reduction percentages were calculated comparing the length of the original series and the length of the MultiBEATS ones.

In the case of the Arabic Digits dataset, the best accuracy is obtained with the raw data. However, the differences in accuracy with MultiBEATS are between a 2.3% and a 4%. Also, we were unable to test MultiBEATS scenarios with a window or slide size greater than 4 since the minimum length of the time series was exactly that number. For that reason, there were fewer tests run in the Arabic Digits dataset.

In the case of the AUSLAN dataset, we find that the accuracy of the raw data is lower than in all of the MultiBEATS cases. More specifically, reductions at 12% and 40% present the highest accuracy which is 93%. This is due to the fact that there have been some tests returning a very low accuracy for the raw dataset, as can be understood by the great standard deviation shown in the picture. The accuracy of MultiBEATS cases is more stable in the AUSLAN dataset and, as always, the number of features, namely the observations in the time series, is reduced.

In the case of the CharacterTrajectories dataset, although the raw data reaches the highest accuracy, that is 97.6%, we see that with MultiBEATS a 96% is reached in several scenarios, being the one with a reduction of the 71.3% of the data one of the most interesting ones if we analyse the running time too.

In the case of CMUSubject16, as already seen in Table 2, we obtained 100% of accuracy using MultiBEATS in two cases (reductions of 10 and 30.5%), that is a greater number than with the raw data.

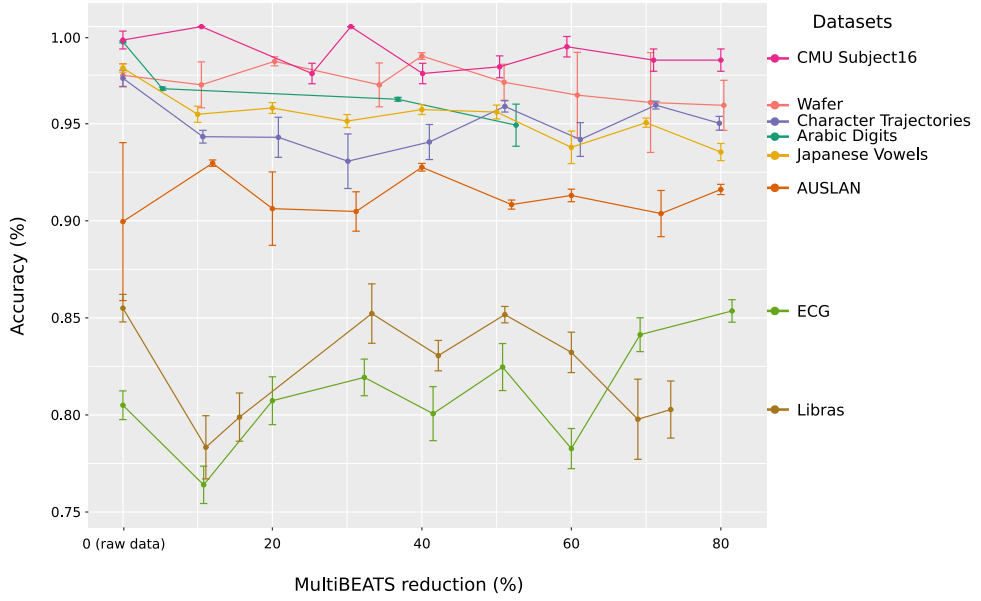

Fig. 1. Evolution of accuracy compared to MultiBEATS reduction percentages.

Table 3
RF classifications results using raw data vs. Seq2VAR and derivates vs. MultiBeats.

Dataset	Best model (-, Var or Seq2VAR or B-Seq2VAR, {nfeat, SL, WD})	Hyperparameters ({n_estimators, max_depth, max_features})	Accuracy	Execution time (s)
AD raw	–	{None, 'sqrt', 100}	**0.968** (0.003)	5.8
AD comparison	Seq2VAR	{5, 'sqrt', 100}	0.875 (0.020)	1863.1
AD MultiBEATS	{2,3,3}	{15, 'sqrt', 100}	0.910 (0.002)	5.0 + 38.0 = 43.0
AU raw	–	{None, 1.0, 80}	**0.904** (0.003)	1.14
AU comparison	Seq2VAR	{15, 'sqrt', 80}	0.820 (0.029)	982.1
AU MultiBEATS	{2,4,5}	{15, 1.0, 80}	0.818 (0.010)	2.14 + 20.60 = 22.74
CT raw	–	{15, 'log2', 80}	**0.960** (0.002)	0.14
CT comparison	Seq2VAR	15, 'sqrt', 40	0.844 (0.022)	10.5
CT MultiBEATS	3,4,15	15, 'sqrt', 80	0.952 (0.001)	0.20 + 41.20 = 41.4
CMU16 raw	–	15, 'log2', 20	0.931 (0.060)	0.02
CMU16 comparison	B-Seq2VAR	15, 'log2', 40	**1.000** (0.000)	240.7
CMU16 MultiBEATS	3,5,10	5, 'log2', 20	0.828 (0.069)	0.03 + 32.20 = 32.23
ECG raw	–	None, 'log2', 20	0.813 (0.032)	0.02
ECG comparison	Var	5, 'log2', 20	0.760 (0.144)	2.27
ECG MultiBEATS	2,3,8	15, 'sqrt', 40	**0.843** (0.015)	0.04 + 2.70 = 2.74
JV raw	–	None, 'sqrt', 100	**0.950** (0.003)	0.16
JV comparison	Seq2VAR	15, 'log2', 20	0.892 (0.049)	59.9
JV MultiBEATS	2,3,7	15, 'log2', 80	0.908 (0.005)	0.09 + 1.00 = 1.09
Libras raw	–	None, 1.0, 100	0.696 (0.016)	0.45
Libras comparison	Var	5, 'sqrt', 80	0.585 (0.056)	4.2
Libras MultiBEATS	2,2,16	None, 'log2', 40	**0.781** (0.016)	0.05 + 1.80 = 1.85
Wafer raw	–	15, 'sqrt', 20	0.969 (0.004)	0.03
Wafer comparison	Var	5, 'log2', 20	0.972 (0.005)	19.6
Wafer MultiBEATS	2,3,3	None, 'log2', 80	**0.981** (0.001)	0.11 + 12.10 = 12.21

In the case of ECG, MultiBEATS also beats the scenario where raw data is used several times. Even more interesting, we can see that using the maximum reduction that we tried, that is an 81.5% and which means that the maximum length of the time series was reduced from 152 to 28, we obtained the maximum accuracy which is a 85.53%.

In the case of Japanese Vowels, MultiBEATS was not able to get a higher accuracy than using the raw data, that was 97.86%. The highest accuracy is obtained with a reduction of 20% and 40%, reaching a 95.75%.

For the Libras dataset, in the case of a 34.3% of reduction, the accuracy is the same as using the raw data until the thousands and presents a lower standard deviation. The rest of the MultiBEATS cases present lower accuracy but not so far.

Finally, in the case of the Wafer dataset, we can see that the classification model trained with data reduced with MultiBEATS obtains a greater accuracy than with the raw data, especially in the cases of 20.3 and 40% reduction, reaching 98.2 and 98.5% respectively against the 97% obtains using the raw data.

4.2.3. Comparison with Seq2VAR

In order to implement the different methods from this paper: VAR, Seq2VAR and B-Seq2VAR, we used the publicly available code of

the repository from the author [37]. In the study that Seq2VAR is presented, the used classification algorithm is Random Forest (RF). However, RF is primarily designed for handling tabular data, where each instance has a fixed set of features or attributes. Therefore, RF is not inherently designed to directly handle multivariate time series (MTS) data and the authors adopt the possibility of flattening the data by eliminating one of their dimensions. For example, for a given scenario with K MTS, each composed of N time series of length M, the transformation results in K UTS, each of them with length N×M, effectively simplifying the problem's multivariate nature and risking the loss of correlation between series and temporal information loss. In order to test the results, we have applied their methodology to all the datasets that are analysed here, since their approach was done under synthetic data and it was not reproducible to the best of our knowledge. As mentioned in Section 2.2.1, they tried 3 different models: Var, Seq2VAR and B-Seq2VAR, and the best results have been summarized by us in Table 3. In such a table we can see the accuracy and processing time results using RF with raw data, comparing with the algorithms proposed in the Seq2VAR study, showing only the best one of the 3 cases (see *Best model* column) and with the MultiBEATS reduction too. In the case of Seq2VAR the time is incremented for all cases, so our algorithm is more efficient in that sense.

To optimize the model's performance, a hyperparameter search has been done with each RF algorithm for each dataset. The parameters chosen to tune via cross-validation are *n_estimators* which is the number of trees in the forest ranging from 20 to 100 in steps of 20, *max_depth* that is the maximum depth of the tree and goes from 5 to 15 in steps of 5 plus *None* with which the nodes are expanded until all leaves are pure, and finally, *max_features* that are the number of features to consider when looking for the best split and can take the value *sqrt*, *log2* or *1.0* (equivalent to *max_features*).

As for accuracy, we see that MultiBEATS is the best one for ECG, Libras and Wafer, being CMUsubject16 the only dataset that using Seq2VAR methods and RF overcomes the rest. Random Forest is, in general, a much faster method than neural networks. In that sense, we can extract from this that when data is not abundant and we do not plan to use neural networks as the classification algorithm, using the raw data might be faster than transforming the data. We can highlight as well that the comparison regarding transformation times between our proposal and the benchmarked one, Seq2VAR, is evidently in MultiBEATS favour.

4.2.4. Discussion of the results: MultiBEATS characteristics

In general, the reasons why a dimensionality reduction method works better/worse for a particular dataset depend on various factors, including the inherent characteristics of the dataset and the properties of the dimensionality reduction technique. For MultiBEATS, we can say that it is particularly effective at representing signals with a relatively low-frequency content. If the datasets contain signals with characteristics that align with the strengths of DCT, our method will be better as well as datasets with strong temporal dependencies and patterns that are suited for sliding windows techniques. In order to verify such assumptions, further analysis needs to be done in the future with synthetic datasets. Nevertheless, through the experiments, we have observed that regarding accuracy MultiBEATS excels using ECG and Wafer, and for CMUSubject16 and Libras it also works well. Those datasets present a lower number of classes, in comparison with the rest and the length of the time series is longer as well. As for what reduction in time refers to, we can appreciate that a higher window size turns into a higher reduction time (see Table 2). In Table 3 we can see that the only case where Seq2VAR is better than MultiBEATS is with the CMUdataset, however, with a different configuration and other algorithms we saw that both raw and MultiBEATS data get the maximum accuracy for such a dataset. In terms of strength, we can think that since Seq2VAR uses binary sampling to build sparse representations can be robust to noise. By interpreting non-zero entries in the binary-sampled representations as Granger causalities, Seq2VAR effectively reduces noise by focusing on the most relevant causal relationships among variables. In that sense, since the datasets we tested are from a curated repository, probably further analysis in controlled environments could be realized. What is indisputable is the fact that the time that it takes to reduce the data with Seq2VAR is much greater than with MultiBEATS and that is a clear disadvantage regardless of the characteristics of the data.

5. Conclusions and future work

In this manuscript, we present an analysis of our proposed MultiBEATS technique for representing time series data. We demonstrate that applying MultiBEATS can significantly reduce the length of time series, which in turn leads to faster processing time in machine learning algorithms and potentially higher accuracy in classification for some cases, indicating a more effective representation of the data in terms of information compression and knowledge preservation.

In future studies, we plan to explore the impact of varying window size configurations on the reduction percentage, which can help us better understand the effects of slide parameter changes on the results. Additionally, we aim to test our methodology in other areas such as clustering and anomaly detection and test MultiBEATS and other dimensionality reduction techniques using synthetic datasets with well-known and fully controlled characteristics in what refers to signal-to-noise ratio, temporal dependencies, global and local patterns, data distribution and geometry in order to better assess the suitability of our dimensionality reduction technique. Our ultimate goal is to integrate MultiBEATS into *tiny* machine learning applications, where resource constraints make data reduction a critical concern. We will also focus on generating a package and an API that is able to receive MTS and outputs the transformed MultiBEATS MTS for their immediate use. We will investigate as well the possibility of adapting MultiBEATS for MTS predictive scenarios, and we will assess its usability in forecasting future values and trends in time series data, further broadening its applicability and potential impact across diverse fields.

CRediT authorship contribution statement

Aurora González-Vidal: Conceptualization, Funding acquisition, Methodology, Supervision, Visualization, Writing – original draft, Writing – review & editing. **Antonio Martinez-Ibarra:** Data curation, Software, Validation, Writing – original draft. **Antonio F. Skarmeta:** Funding acquisition, Investigation, Resources, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Aurora Gonzalez-Vidal reports financial support was provided by European Union NextGenerationEU. Aurora Gonzalez-Vidal reports financial support was provided by Comunidad Autónoma de la Región de Murcia - Fundación Séneca. Antonio F. Skarmeta reports financial support was provided by European Commission H2020 PHOENIX (g.a. 893079). Antonio F. Skarmeta reports financial support was provided by European Commission Horizon2.5 MASTER- PIECE project (g.a. 101096836).

Data availability

I have shared a link to a github repository which will provide access to all data and code.

References

[1] A. González-Vidal, P. Rathore, A.S. Rao, J. Mendoza-Bernal, M. Palaniswami, A.F. Skarmeta-Gómez, Missing data imputation with bayesian maximum entropy for internet of things applications, IEEE Internet Things J. 8 (21) (2020) 16108–16120.

[2] M.V. Moreno, F. Terroso-Sáenz, A. González-Vidal, M. Valdés-Vela, A.F. Skarmeta, M.A. Zamora, V. Chang, Applicability of big data techniques to smart cities deployments, IEEE Trans. Ind. Inform. 13 (2) (2016) 800–809.

[3] Y. Li, S. Wang, Q. Tian, X. Ding, Feature representation for statistical-learning-based object detection: A review, Pattern Recognit. 48 (11) (2015) 3542–3559.

[4] A. Gonzalez-Vidal, J. Mendoza-Bernal, S. Niu, A.F. Skarmeta, H. Song, A transfer learning framework for predictive energy-related scenarios in smart buildings, IEEE Trans. Ind. Appl. (2022).

[5] A. González-Vidal, J. Mendoza-Bernal, A.P. Ramallo, M.Á. Zamora, V. Martínez, A.F. Skarmeta, Smart operation of climatic systems in a greenhouse, Agriculture 12 (10) (2022) 1729.

[6] E.M. Campos, P.F. Saura, A. González-Vidal, J.L. Hernández-Ramos, J.B. Bernabé, G. Baldini, A. Skarmeta, Evaluating federated learning for intrusion detection in internet of things: Review and challenges, Comput. Netw. 203 (2022) 108661.

[7] Y. Bengio, A. Courville, P. Vincent, Representation learning: A review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–1828.

[8] W. Jia, M. Sun, J. Lian, S. Hou, Feature dimensionality reduction: a review, Complex Intell. Syst. 8 (3) (2022) 2663–2693.

[9] A. Gonzalez-Vidal, P. Barnaghi, A.F. Skarmeta, Beats: Blocks of eigenvalues algorithm for time series segmentation, IEEE Trans. Knowl. Data Eng. 30 (11) (2018) 2051–2064.

[10] X. Wan, H. Li, L. Zhang, Y.J. Wu, Dimensionality reduction for multivariate time-series data mining, J. Supercomput. 78 (7) (2022) 9862–9878.

[11] A. Gonzalez-Vidal, F. Jimenez, A.F. Gomez-Skarmeta, A methodology for energy multivariate time series forecasting in smart buildings based on feature selection, Energy Build. 196 (2019) 71–82.

[12] T. Tamanna, M.A. Rahman, S. Sultana, M.H. Haque, M.Z. Parvez, Predicting seizure onset based on time-frequency analysis of EEG signals, Chaos Solitons Fractals 145 (2021) 110796.

[13] A.S. Gezawa, Z.A. Bello, Q. Wang, L. Yunqi, A voxelized point clouds representation for object classification and segmentation on 3D data, J. Supercomput. 78 (1) (2022) 1479–1500.

[14] Y. Xie, S. Liu, S. Huang, H. Fang, M. Ding, C. Huang, T. Shen, Local trend analysis method of hydrological time series based on piecewise linear representation and hypothesis test, J. Clean. Prod. 339 (2022) 130695.

[15] K. Feasel, Symbolic aggregate approximation (SAX), in: Finding Ghosts in Your Data: Anomaly Detection Techniques with Examples in Python, Springer, 2022, pp. 293–309.

[16] R. Rezvani, P. Barnaghi, S. Enshaeifar, A new pattern representation method for time-series data, IEEE Trans. Knowl. Data Eng. 33 (7) (2019) 2818–2832.

[17] M.G. Baydogan, G. Runger, Learning a symbolic representation for multivariate time series classification, Data Min. Knowl. Discov. 29 (2015) 400–422.

[18] M.G. Baydogan, G. Runger, Time series representation and similarity based on local autopatterns, Data Min. Knowl. Discov. 30 (2016) 476–509.

[19] G. Li, B. Choi, J. Xu, S.S. Bhowmick, K.-P. Chun, G.L.-H. Wong, Shapenet: A shapelet-neural network approach for multivariate time series classification, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, 2021, pp. 8375–8383.

[20] M. Cuturi, A. Doucet, Autoregressive kernels for time series, 2011, arXiv preprint arXiv:1101.0673.

[21] K.S. Tuncel, M.G. Baydogan, Autoregressive forests for multivariate time series modeling, Pattern Recognit. 73 (2018) 202–215.

[22] P. Schäfer, U. Leser, Multivariate time series classification with weasel+ MUSE, 2017, arXiv preprint arXiv:1711.11343.

[23] Z. Duan, H. Xu, Y. Wang, Y. Huang, A. Ren, Z. Xu, Y. Sun, W. Wang, Multivariate time-series classification with hierarchical variational graph pooling, Neural Netw. 154 (2022) 481–490.

[24] F.M. Bianchi, L. Livi, K.Ø. Mikalsen, M. Kampffmeyer, R. Jenssen, Learning representations of multivariate time series with missing data, Pattern Recognit. 96 (2019) 106973.

[25] E. Pineau, S. Razakarivony, T. Bonald, Seq2var: multivariate time series representation with relational neural networks and linear autoregressive model, in: Advanced Analytics and Learning on Temporal Data: 4th ECML PKDD Workshop, AALTD 2019, WÜRzburg, Germany, September 20, 2019, Revised Selected Papers 4, Springer, 2020, pp. 126–140.

[26] H.Y. Toda, P.C. Phillips, Vector autoregression and causality: a theoretical overview and simulation study, Econ. Rev. 13 (2) (1994) 259–285.

[27] T. Kipf, E. Fetaya, K.-C. Wang, M. Welling, R. Zemel, Neural relational inference for interacting systems, in: International Conference on Machine Learning, PMLR, 2018, pp. 2688–2697.

[28] A.P. Ruiz, M. Flynn, J. Large, M. Middlehurst, A. Bagnall, The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances, Data Min. Knowl. Discov. 35 (2) (2021) 401–449.

[29] Z. Chen, Y. Liu, J. Zhu, Y. Zhang, R. Jin, X. He, J. Tao, L. Chen, Time-frequency deep metric learning for multivariate time series classification, Neurocomputing 462 (2021) 221–237.

[30] R. Chen, X. Yan, S. Wang, G. Xiao, DA-net: Dual-attention network for multivariate time series classification, Inform. Sci. 610 (2022) 472–487.

[31] Y. Hao, H. Cao, A new attention mechanism to classify multivariate time series, in: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, 2020.

[32] F. Karim, S. Majumdar, H. Darabi, S. Harford, Multivariate LSTM-FCNs for time series classification, Neural Netw. 116 (2019) 237–245.

[33] A. Gonzalez-Vidal, Multibeats repository, 2023, Accessed: May 2, 2023, https://github.com/auroragonzalez/multiBEATS.

[34] D. Dua, C. Graff, UCI machine learning repository, 2017, URL http://archive.ics.uci.edu/ml.

[35] CMU, Carnegie mellon university graphics lab motion capture database, 2012, URL , Funding from NSF grant number EIA-0196217.

[36] O. RT, 2012, URL http://www.cs.cmu.edu/~bobski/.

[37] E. Pineau, Seq2VAR-multivariate-time-series-representation-learning, 2021, Accessed: April 27, 2023, https://github.com/edouardpineau/Seq2VAR-multivariate-time-series-representation-learning.

Aurora González-Vidal graduated in Mathematics from the University of Murcia in 2014. In 2015 she got a fellowship to work in the Statistical Division of the Research Support Service, where she specialized in Statistics and Data Analysis. Afterward, she studied a Big Data Master. In 2019, she got a Ph.D. in Computer Science. Currently, she is a postdoctoral researcher at the University of Murcia. She has collaborated in several national and European projects such as ENTROPY, IoTCrawler, and DEMETER. Her research covers machine learning in IoT-based environments, missing values imputation, and time-series segmentation. She is the president of the R Users Association UMUR.

Antonio Martínez Ibarra studied Physics at the University of Murcia, graduating in 2021. The following year he got a Masters degree in Big Data from the same university. His master's thesis focused on the field of energy efficiency under the umbrella of the European project PHOENIX. Currently, he is working as a researcher at the University of Murcia on the project PERSEO, with tasks focused on machine and deep learning applied to agriculture, marine sciences, and energy efficiency.

Antonio F. Skarmeta received the M.S. degree in Computer Science from the University of Granada and B.S. (Hons.) and the Ph.D. degrees in Computer Science from the University of Murcia Spain. Since 2009 he is Full Professor at the same department and University. Antonio F. Skarmeta has worked on national and international research projects in networking, security and IoT area, like SEMIRAMIS, SMARTIE, SOCIOTAL, IoT6 ANASTACIA, CyberSec4Europe. His main interest is in the integration of security services, identity, IoT and Smart Cities. He has been heading of the research group ANTS since its creation on 1995. He has published over 200 international papers.