



OPEN

Facilitating time series classification by linear law-based feature space transformation

Marcell T. Kurucz^{1,2✉}, Péter Pósfay¹ & Antal Jakovác¹

The aim of this paper is to perform uni- and multivariate time series classification tasks with linear law-based feature space transformation (LLT). First, LLT is used to separate the training and test sets of instances. Then, it identifies the governing patterns (laws) of each input sequence in the training set by applying time-delay embedding and spectral decomposition. Finally, it uses the laws of the training set to transform the feature space of the test set. These calculation steps have a low computational cost and the potential to form a learning algorithm. For the empirical study of LLT, a widely used human activity recognition database called AReM is employed. Based on the results, LLT vastly increases the accuracy of traditional classifiers, outperforming state-of-the-art methods after the proposed feature space transformation is applied. The fastest error-free classification on the test set is achieved by combining LLT and the k-nearest neighbor (KNN) algorithm while performing fivefold cross-validation.

In machine learning and data mining, time series classification (TSC) is one of the most challenging tasks^{1,2}; it aims to assign all unlabeled time series to one of the predefined categories. TSC is widely studied across various domains, including activity recognition^{3–6}, biology^{7–10}, and finance^{11–14}, but its growing popularity is primarily due to the rapidly increasing amount of temporal data collected by widespread sensors¹⁵. Accordingly, with the proliferation of the Internet of Things and big data technologies, TSC has become increasingly crucial. A TSC task can be defined as a uni-^{16,17} or multivariate^{18,19} problem, depending on whether one or more values are observed at a particular point in time. In recent decades, a number of methods have been proposed to address both problems and can be divided into feature-based and distance-based approaches (see, e.g.,²⁰). The main difference between the two is that the former approach contains a feature extraction phase before the classification stage, while the latter performs classification based on a suitable (dis)similarity measure of the series. Hereinafter, we focus only on the methods and algorithms of the feature-based approach that are used during the feature extraction and classification phases.

Some methods for feature extraction include basic statistical methods^{21–23} and spectral-based methods, such as discrete Fourier transform (DFT)²⁴ or discrete wavelet transform (DWT)^{25,26} methods, in which features of the frequency domain are considered. Others are based on singular value decomposition (SVD)²⁷, where eigenvalue analysis is applied to the dimensional reduction of the feature set. Furthermore, there are some model-based methods that are mainly used to capture information about the dynamic behavior of the investigated series. Within this group, the different versions of the autoregressive (AR) integrated moving average (ARIMA) model²⁸ are widely applied (see, e.g.,^{5,29–31}). In most of these works, the coefficients of a fitted AR model are applied as features^{30,31}, or they are used to build a more complex generative model²⁹.

After the time series data are transformed into feature vectors, they are classified by conventional classifiers such as logistic regression (LR)^{32,33}, decision tree (DT)³⁴, random forest (RF)³⁵, k-nearest neighbor (KNN)³⁶, support vector machine (SVM)^{37,38}, relevance vector machine (RVM)³⁹, and Gaussian process (GP)⁴⁰ classifiers. Alternatively, deep neural networks (DNNs) can be applied to automatically compile the feature space before classification^{41–43}.

The aim of this paper is to find a solution for uni- and multivariate TSC tasks with a linear law-based feature space transformation (LLT) approach that can be applied between the feature extraction and classification phases. LLT is based on an idea borrowed from physics, the so-called existence of conserved quantities. In general, these are quantities that remain constant for the whole system if some basic conditions are fulfilled. The conservation of energy and momentum are perhaps the most known examples, but in theory infinitely many conserved quantities

¹Department of Computational Sciences, Wigner Research Centre for Physics, 29-33 Konkoly-Thege Miklós Street, Budapest 1121, Hungary. ²Institute of Data Analytics and Information Systems, Corvinus University of Budapest, 8 Fővám Square, Budapest 1121, Hungary. ✉email: kurucz.marcell@wigner.hu

i	Input 1	Input 2	...	Input m	Output
1	$x_t^{1,1}$	$x_t^{1,2}$...	$x_t^{1,m}$	y^1
2	$x_t^{2,1}$	$x_t^{2,2}$...	$x_t^{2,m}$	y^2
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
n	$x_t^{n,1}$	$x_t^{n,2}$...	$x_t^{n,m}$	y^n

Table 1. Internal data structure of the TSC problem.

can be constructed for a given system, although not all of them represent new information. LLT searches for conserved quantities in time series data and characterizes them with vectors (hereinafter referred to as laws).

To facilitate TSC tasks, LLT, which is a data-driven approach, identifies a set of laws related to each class and then applies these sets to transform the time series data of a new instance. Laws related to each class compete to reveal the class of the new time series; however, only the laws corresponding to the correct class can transform it close to the null vector. In essence, this transformation generates a new feature space utilizing the linear laws belonging to classes. The resulting features are expected to be more selective for the given TSC task. Similarly to principal component analysis (PCA)^{44,45}, LLT is based on spectral decomposition. However, in contrast to PCA, LLT focuses on the most invariant property of the time series data instead of the most variable property.

In practice, first, LLT separates the training and test sets of the instances. Then, it identifies the governing patterns (laws) of each input sequence in the training set by applying time-delay embedding and spectral decomposition. Finally, it uses the laws of the training set to transform the feature space of the test set. These calculation steps have a low computational cost and the potential to form a learning algorithm.

Generally, raw time series contain various types of distortion, which makes their comparison and classification difficult⁴⁶. The common types of distortion are noise, offset translation, amplitude scaling, longitudinal scaling, linear drift, and discontinuities⁴⁷. In addition to its low computational cost, the main advantage of LLT is that it is robust to noise and amplitude scaling by definition. Discontinuities, provided they rarely occur, do not alter the resulting laws significantly. Offset transformation and linear drift can also be easily adapted, and the laws are very generalizable, i.e., if the training set contains instances with specific offset or linear drift values, the laws will be able to treat any others. Furthermore, it becomes more and more resistant to the other distortions mentioned above as the number (not the rate) of (atypical) instances increases in the training set. The reason is that the transformation is applied by the most suitable law per class. Thus, if the training set contains instances with a similar set of distortions as the new instance, the transformation is applied based on their laws. For the same reason, the generated feature set, and thus any learning algorithm based on it, is largely immune to catastrophic forgetting^{48–50} when the training set is extended.

For the empirical study of LLT, a widely used human activity recognition database called AREM is applied. This database contains temporal data from a wireless sensor network worn by users who were asked to perform seven different activities. The related TSC task is the identification of these activities based on the noisy sensor data generated by the users. Based on the results, the accuracy of traditional classifiers are greatly increased using LLT, outperforming state-of-the-art methods after the proposed feature space transformation is applied. Due to its robustness against a significant amount of noise (which is a characteristic of the HAR databases⁵¹), fast and error-free classification is achieved based on the combination of the KNN algorithm. (The working paper of this study is available at SSRN⁵²).

The rest of this paper is organized as follows. In the Methods section, the general TSC problem as well as the LLT algorithm are introduced. In the Experimental setup section, the AREM database and the applied classification algorithms are presented. In the “Results and discussion” section, the results of the classification obtained with and without LLT are compared and discussed. Finally, in the Conclusions and future work section, conclusions are provided, and future research directions are discussed.

Methods

Time series classification problem. The TSC problem discussed in this paper is based on the following data structure. We consider input data as $X = \{X_t \mid t \in \{1, 2, \dots, k\}\}$ sets (time series), where t denotes the observation times. The internal structure of these input data is $X_t = \{x_t^{i,j} \mid i \in \{1, 2, \dots, n\}, j \in \{1, 2, \dots, m\}\}$, where i identifies the instances and j indexes the different input series belonging to a given instance. The output is a vector $y \in \{1, 2, \dots, c\}$ identifying the classes (c) of instances ($y = \{y^i \in \mathbb{R} \mid i \in \{1, 2, \dots, n\}\}$). Our goal is to predict the y values (classes) from the X input data.

For the sake of transparency, the above-mentioned data structure is illustrated in Table 1.

Linear laws of time series. Before we describe how to transform the original feature space to facilitate the TSC task, the concept of linear laws⁵³ is introduced. Here, we only summarize the most important results needed to understand the logic behind LLT. The detailed derivations and proofs can be found in⁵⁴. First, let us consider a generic time series z_t , where $t \in \{1, 2, \dots, k\}$ denotes the time. The l th order ($l \in \mathbb{Z}^+$ and $l < k$) time-delay embedding⁵⁵ of the given series is obtained as follows:

Source	$x_t^{1,1}$	$x_t^{2,1}$...	$x_t^{\tau,1}$	$x_t^{1,2}$...	$x_t^{\tau,m}$
Laws	$v^{1,1}$	$v^{2,1}$...	$v^{\tau,1}$	$v^{1,2}$...	$v^{\tau,m}$

Table 2. Linear laws of the training set.

Source	$x_t^{\tau+1,1}$	$x_t^{\tau+2,1}$...	$x_t^{n,1}$	$x_t^{\tau+1,2}$...	$x_t^{n,m}$
Matrices	$S^{\tau+1,1}$	$S^{\tau+2,1}$...	$S^{n,1}$	$S^{\tau+1,2}$...	$S^{n,m}$

Table 3. S matrices of the test set.

$$A = \begin{pmatrix} z_1 & z_2 & \dots & z_l \\ z_2 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ z_{k-1} & \dots & \dots & z_k \end{pmatrix}. \tag{1}$$

A linear function that maps the rows of the matrix A to zero is called a (linear) law. An exact linear law can be represented by a vector v that fulfills the following condition:

$$\|Av\| = 0. \tag{2}$$

It can be seen that using normal quadratic norm finding v is equivalent to finding the eigenvector corresponding to the zero eigenvalue of the symmetric matrix S , which is defined as⁵⁴:

$$S = A^T A. \tag{3}$$

This method can be thought of as complementary to PCA, where we are looking for the largest eigenvalues (representing the largest change in the data). Here, we search for the smallest (preferably zero) eigenvalue, which represents the direction in which the data change the least. In practice, the linear law, v , maps the learning set close to zero:

$$Sv \approx 0, \tag{4}$$

where 0 is a null column vector with l elements, v is a column vector with l elements and $v \neq 0$. To find the v coefficients of Eq. (4), first, we perform eigendecomposition on the S matrix. Then, we select the eigenvector that is related to the smallest eigenvalue. Finally, we apply this eigenvector as v coefficients, and hereinafter, we refer to it as the linear law of z_t .

Linear law-based feature space transformation. To facilitate the classification task presented in the time series classification problem section, we transform the original feature space using linear laws.

First, instances (i) are divided into training ($tr \in \{1, 2, \dots, \tau\}$) and test ($te \in \{\tau + 1, \tau + 2, \dots, n\}$) sets in such a way that the classes of instances are balanced in the two sets. For the sake of transparency, we assume that the original arrangement of the instances in the dataset (see Table 1) satisfies this condition for the tr and te sets. Then, we calculate the linear law (see v in Eq. 4) of all input series in the training set ($x_t^{1,1}, x_t^{2,1}, \dots, x_t^{\tau,m}$), which results in $\tau \times m$ laws (eigenvectors) in total. To do this, we create an $A^{tr,j}$ matrix for all instances in the training set (tr) and all input series (j). We calculate a separate $S^{tr,j}$ matrix for each component of this index pair (see Eqs. 1–3). Then, we use the eigenvectors related to their smallest eigenvalues as laws (see Eq. 4). As a result of these steps, the dataset presented in Table 2 is obtained.

We remark that the linear laws (eigenvectors) contained by Table 2 are column vectors with l elements. For the sake of transparency, we introduce V^j matrices with $l \times \tau$ dimensions that group linear laws based on the related input series ($V^j = [v^{1,j}, v^{2,j}, \dots, v^{\tau,j}]$). Although we have not indexed which input series belongs to which class, because the instances belonging to each class are balanced in the tr and te sets, they are also balanced in the V^j matrices. Consequently, the laws contained in each V^j matrix are derived from instances that cover all classes together ($V^j = \{V_1^j, V_2^j, \dots, V_c^j\}$, where V_c^j denotes the laws of the training set related to the input series j and the class c).

As a next step, we calculate $S^{te,j}$ matrices (see Eq. 3) from the input series of the test instances in the test set. For one instance, m matrices are calculated (one for each input series, see Table 3).

We left-multiply the V^j matrices obtained from the training set by the $S^{te,j}$ matrices of the test set related to the same input variable ($S^{\tau+1,1} V^1, S^{\tau+1,2} V^2, \dots, S^{n,m} V^m$). Each of these products results in a matrix that inherits the dimensions of the V^j matrices ($l \times \tau$). In this step, the laws contained in the V^j matrices provide an estimate of whether the $S^{te,j}$ matrices of the test set are related to the same class as them. More specifically, only those columns of the $S^{te,j} V^j$ matrices are close to the null vector, and they have a relatively small variance, where the classes of the corresponding training and testing data match.

Then, we reduce the dimension of the resulting matrices by using an f function. This function selects the column vectors with the smallest variance from the $S^{te,j} V^j$ matrices for each class as follows:

te	Input 1	Input 2	...	Input m	Output
$\tau + 1$	$f(S^{\tau+1,1} V^1)$	$f(S^{\tau+1,2} V^2)$...	$f(S^{\tau+1,m} V^m)$	$y^{\tau+1}$
$\tau + 2$	$f(S^{\tau+2,1} V^1)$	$f(S^{\tau+2,2} V^2)$...	$f(S^{\tau+2,m} V^m)$	$y^{\tau+2}$
\vdots	\vdots	\vdots	...	\vdots	\vdots
n	$f(S^{n,1} V^1)$	$f(S^{n,2} V^2)$...	$f(S^{n,m} V^m)$	y^n

Table 4. Feature space of the test set generated based on the linear laws of the training set.

$$S^{te,j} V^j = \{S^{te,j} V_1^j, S^{te,j} V_2^j, \dots, S^{te,j} V_c^j\},$$

$$f(S^{te,j} V^j) = \{o_1^{te,j}, o_2^{te,j}, \dots, o_c^{te,j}\}, \quad (5)$$

For example, the series $o_c^{te,j}$ is a column vector of the $S^{te,j} V^j$ matrix that has the minimum variance. Thus, after this step, the transformed feature space of the test set has $((n - \tau)l) \times (mc)$ dimensions without the output variable (see Table 4).

Finally, to facilitate cross-validation and performance measurements, we calculate the arithmetic mean and variance of each transformed time series (see the g_1 and g_2 functions in Eq. 6). Both cases result in $m \times c$ scalars for each instance of the test set (one scalar for each input series of each class). This final transformation reduces the dimension of the $f(S^{te,j} V^j)$ matrices (see Eq. 5) as follows:

$$g_1(f(S^{te,j} V^j)) = \{\text{Mean}(o_1^{te,j}), \text{Mean}(o_2^{te,j}), \dots, \text{Mean}(o_c^{te,j})\},$$

$$g_2(f(S^{te,j} V^j)) = \{\text{Var}(o_1^{te,j}), \text{Var}(o_2^{te,j}), \dots, \text{Var}(o_c^{te,j})\}. \quad (6)$$

These means and variances are used as features to predict the class of instances in the test set (te). That is, the classes are predicted based on $\{\text{Mean}(o_1^{te,1}), \text{Mean}(o_2^{te,1}), \dots, \text{Mean}(o_c^{te,1}), \text{Mean}(o_1^{te,2}), \dots, \text{Mean}(o_c^{te,m})\}$ in the first case and $\{\text{Var}(o_1^{te,1}), \text{Var}(o_2^{te,1}), \dots, \text{Var}(o_c^{te,1}), \text{Var}(o_1^{te,2}), \dots, \text{Var}(o_c^{te,m})\}$ in the second.

In summary, we applied the linear laws of the training set to transform the feature space of test instances to facilitate their classification. If the instances of classes show similarity, the transformed test instances belonging to the same class will have values close to zero in the same places since the laws related to the same class will transform them close to null vectors (see Eq. 4).

Experimental setup

This section describes the employed database and presents the classifier algorithms applied to the original and transformed feature spaces.

Human activity recognition dataset. In this paper, the database called the Activity Recognition system based on Multisensor data fusion (AReM)⁵⁶ is employed (freely available at: <https://archive.ics.uci.edu/ml/datasets/Activity+Recognition+system+based+on+Multisensor+data+fusion+%28AReM%29>, retrieved: 20 September 2022).

This database was compiled by using three wireless beacons that implement the IEEE 802.15.4 standard. These beacons were attached to a user's chest and both ankles, and the user performed seven different activities, including cycling, lying down, sitting, standing, and walking, as well as two types of bending ($y \in \{1, 2, \dots, 7\}$). The signal strength of the beacons, which was measured in a unit called the received signal strength (RSS), decreased in proportion to the distance between the beacons and the applied wireless scanners.

The RSS values of these beacons were sampled at a frequency of 20 Hz. Every 50 ms, the scanner recorded an RSS value from each of the three sensors. Finally, the mean and variance of the RSS values accumulated every 250 ms were computed for each beacon, resulting in 6 features (time series; $m = 6$). Every user performed a specific activity for 2 min, and 480 consecutive values were generated for each of these series ($k = 480$). With the exception of the two different types of bending, 15 such datasets were recorded from each activity performed by different users. From the type 1 and type 2 bending activities, 7 and 6 datasets were collected, respectively ($n = 88$).

The multivariate TSC task associated with the AReM database is to predict the type of activities based on the 6 features generated by the three wireless beacons. Before attempting to solve this task, we randomly divide the 88 instances ($5 \times 15 + 7 + 6$ datasets) into training (tr) and test (te) sets. After this step, approximately 53.4% of the instances belonging to each activity are included in the test set, while all other instances are part of the training set. Within each category, the test set contained the following data (see the AReM⁵⁶ database):

- Bending 1: 2, 3, 4, 6.
- Bending 2: 2, 3, 5.
- Cycling: 1, 2, 5, 7, 9, 10, 14, 15.
- Lying: 1, 2, 6, 9, 11, 12, 14, 15.
- Sitting: 4, 7, 8, 9, 10, 11, 12, 15.
- Standing: 2, 3, 6, 7, 8, 12, 13, 14.
- Walking: 1, 3, 4, 7, 8, 9, 11, 13.

(a) Based on mean values					(b) Based on variances				
	Applied feature space					Applied feature space			
	Original		Transformed by LLT			Original		Transformed by LLT	
	Accuracy (%)	Time (s)	Accuracy (%)	Time (s)		Accuracy (%)	Time (s)	Accuracy (%)	Time (s)
Ensemble	74.5	150.7	100.0	135.3	Ensemble	68.1	172.2	91.5	213.4
KNN	85.1	54.4	100.0	42.5	KNN	80.9	63.8	100.0	59.7
DT	74.5	28.8	87.2	39.0	DT	57.4	43.6	85.1	44.6
SVM	83.0	121.2	89.4	151.2	SVM	80.9	149.9	91.5	195.2

Table 5. Classification results. Remarks: fivefold cross-validation and 30-step Bayesian hyperparameter optimization were applied. The total number of observations ($n - \tau$) was 47 in all four cases. In the LLT case, the calculation was based on the absolute value of the means—see subfigure (a). Confusion matrices and the details of hyperparameter optimization can be found in the Supplementary information. The calculations were performed using a computer with an Intel Core i7-6700K CPU processor running at 4.00 GHz with 16 GB RAM.

Applied classifiers. After transforming the feature space of the test data by using the linear laws of the training set (see the linear law-based feature space transformation and human activity recognition dataset sections), we compare the accuracy and calculation time of four different classifiers on both the original and the transformed feature spaces. The applied algorithms are the ensemble, KNN, DT, and SVM algorithms, which are used with fivefold cross-validation and 30-step Bayesian hyperparameter optimization. The classification task was performed in the Classification Learner App of MATLAB (details of the applied methods and the classification process can be found at <https://www.mathworks.com/help/stats/classificationlearner-app.html>, retrieved: 20 September 2022).

Results and discussion

From the original six features contained in the AReM database, first, we generated a new feature space with 42 features by using the LLT algorithm (6 features per class). To facilitate cross-validation and performance measurements, we calculated the mean and variance of each transformed time series (see Eq. 6). The means and variances in the transformed case were calculated from the 30 long time series. We then solved the classification problem of mean values and variances based on both the original and transformed feature spaces by using four different classifiers (see the “Applied classifiers” section). The results of the calculations are shown in Table 5 (for the sake of comparison, in the nontransformed case, we performed calculations based on the mean and variance of each input series of the original feature space).

Table 5 shows that the accuracy of the original feature space was approximately 78.1% on average (79.3% for the mean and 76.8% for the variance), while after the transformation, we obtained approximately 93.1% accurate classification (93.1% and 92.0%, respectively). This increase in performance was associated with only a relatively small increase in training time (9.8% on average). Furthermore, the fastest (42.5 s and 59.7 s, respectively) error-free classifications were achieved by combining the LLT and KNN algorithms, which outperformed the state-of-the-art methods. Furthermore, with this algorithm, we obtained a shorter calculation time in the case of the transformed feature space than in the original case. Moreover, the hyperparameters of this algorithm conspicuously converged very rapidly to their optimum (see the Supplementary information), which may further decrease the required optimization time.

In comparison with the recently published results related to the AReM database, Vydia and Sasikumar (2022)⁵⁷ achieved a maximum classification accuracy of 99.63% by using the DWT along with the entropy features from empirical mode decomposition (EMD) and four different classifiers. While their method is superior to several state-of-the-art machine learning techniques⁵⁷, it is slightly less accurate than the combination of the LLT algorithm with the KNN or ensemble classifiers.

An additional advantage of the LLT algorithm is that it has a low computational cost—the transformation of the original feature space took only approximately 1.8 sec (the calculations were performed using a computer with an Intel Core i7-6700K CPU processor running at 4.00 GHz with 16 GB RAM). Additionally, LLT has the potential to form a learning algorithm by continuously improving the set of laws applied during the transformation.

Conclusions and future work

In this paper, the LLT algorithm, which aims to facilitate uni- and multivariate TSC tasks, was introduced. LLT has a low computational cost and the potential to form a learning algorithm. For its empirical study, we applied a widely used multisensor-based human activity recognition dataset called AReM. Based on the results, LLT vastly increased the accuracy of traditional classifiers, which outperformed state-of-the-art methods after the proposed feature space transformation. The fastest error-free classification on the test set was achieved by the combination of the LLT and KNN algorithms while performing 5-fold cross-validation and 30-step Bayesian hyperparameter optimization. In this case, the hyperparameters conspicuously converged very rapidly to their optimum, which may further decrease the required optimization time.

Our future research will focus on the application of the proposed feature space transformation for portfolio selection and heart disease classification based on ECG signals. Additionally, we will develop R and Python

packages to facilitate the use of LLT. Further studies could also focus on how LLT can be applied as a part of a learning algorithm in which the set of laws used for the feature space transformation is continuously improved. Finally, it may be worthwhile to examine how LLT can be integrated into the framework of neural networks.

Data availability

The Activity Recognition system based on Multisensor data fusion (AReM)⁵⁶ is freely available at: <https://archive.ics.uci.edu/ml/datasets/Activity+Recognition+system+based+on+Multisensor+data+fusion+%28AReM%29>, retrieved: 20 September 2022.

Received: 20 July 2022; Accepted: 19 October 2022

Published online: 27 October 2022

References

- Lines, J. & Bagnall, A. Time series classification with ensembles of elastic distance measures. *Data Min. Knowl. Discov.* **29**, 565–592 (2015).
- Wang, T. *et al.* Adaptive feature fusion for time series classification. *Knowl. Based Syst.* **243**, 108459 (2022).
- Mocanu, D. C. *et al.* Factored four way conditional restricted Boltzmann machines for activity recognition. *Pattern Recognit. Lett.* **66**, 100–108 (2015).
- Karim, F., Majumdar, S., Darabi, H. & Harford, S. Multivariate LSTM-FCNS for time series classification. *Neural Netw.* **116**, 237–245 (2019).
- Wang, J., Chen, Y., Hao, S., Peng, X. & Hu, L. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognit. Lett.* **119**, 3–11 (2019).
- Yang, C., Jiang, W. & Guo, Z. Time series data classification based on dual path CNN-RNN cascade network. *IEEE Access* **7**, 155304–155312 (2019).
- Schäfer, P. & Leser, U. Multivariate time series classification with weasel+ muse. arXiv preprint [arXiv:1711.11343](https://arxiv.org/abs/1711.11343) (2017).
- Rajan, D. & Thiagarajan, J. J. A generative modeling approach to limited channel ECG classification. in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 2571–2574 (IEEE, 2018).
- Elsayed, N., Maida, A. S. & Bayoumi, M. An analysis of univariate and multivariate electrocardiography signal classification. in *2019 18th IEEE International Conference On Machine Learning and Applications (ICMLA)*. 396–399 (IEEE, 2019).
- Bock, C., Moor, M., Jutzeler, C. R. & Borgwardt, K. Machine learning for biomedical time series classification: From shapelets to deep learning. *Artif. Neural Netw.* 33–71 (2021).
- Kwon, D.-H., Kim, J.-B., Heo, J.-S., Kim, C.-M. & Han, Y.-H. Time series classification of cryptocurrency price trend based on a recurrent LSTM neural network. *J. Inf. Process. Syst.* **15**, 694–706 (2019).
- Fons, E., Dawson, P., Zeng, X.-j., Keane, J. & Iosifidis, A. Evaluating data augmentation for financial time series classification. arXiv preprint [arXiv:2010.15111](https://arxiv.org/abs/2010.15111) (2020).
- Feo, G., Giordano, F., Niglio, M. & Parrella, M. L. Financial time series classification by nonparametric trend estimation. in *Methods and Applications in Fluorescence*. 241–246 (Springer, 2022).
- Assis, C. A., Machado, E. J., Pereira, A. C. & Carrano, E. G. Hybrid deep learning approach for financial time series classification. *Rev. Bras. Comput. Appl.* **10**, 54–63 (2018).
- Marussy, K. & Buza, K. Success: A new approach for semi-supervised classification of time-series. in *International Conference on Artificial Intelligence and Soft Computing*. 437–447 (Springer, 2013).
- Sun, J. *et al.* Univariate time series classification using information geometry. *Pattern Recognit.* **95**, 24–35 (2019).
- del Campo, F. A. *et al.* Auto-adaptive multilayer perceptron for univariate time series classification. *Exp. Syst. Appl.* **181**, 115147 (2021).
- Baydogan, M. G. & Runger, G. Learning a symbolic representation for multivariate time series classification. *Data Min. Knowl. Discov.* **29**, 400–422 (2015).
- Ruiz, A. P., Flynn, M., Large, J., Middlehurst, M. & Bagnall, A. The great multivariate time series classification bake off: A review and experimental evaluation of recent algorithmic advances. *Data Min. Knowl. Discov.* **35**, 401–449 (2021).
- Susto, G. A., Cenedese, A. & Terzi, M. Time-series classification methods: Review and applications to power systems data. in *Big Data Application in Power Systems*. 179–220 (2018).
- Rodríguez, J. J., Alonso, C. J. & Boström, H. Boosting interval based literals. *Intell. Data Anal.* **5**, 245–262 (2001).
- Rodríguez, J. J. & Alonso, C. J. Interval and dynamic time warping-based decision trees. in *Proceedings of the 2004 ACM Symposium on Applied Computing*. 548–552 (2004).
- Rodríguez, J. J. & Alonso, C. J. Support vector machines of interval-based features for time series classification. in *International Conference on Innovative Techniques and Applications of Artificial Intelligence*. 244–257 (Springer, 2004).
- Rafiei, D. & Mendelzon, A. Querying time series data based on similarity. *IEEE Trans. Knowl. Data Eng.* **12**, 675–693. <https://doi.org/10.1109/69.877502> (2000).
- Yan, L., Liu, Y. & Liu, Y. Application of discrete wavelet transform in shapelet-based classification. *Math. Probl. Eng.* **19**, 2020 (2020).
- Lim, S. E. & Na, J. Time series classification using wavelet transform. *Korean Data Inf. Sci. Soc.* **32**, 943–952 (2021).
- Zheng, L., Wang, Z., Liang, J., Luo, S. & Tian, S. Effective compression and classification of ECG arrhythmia by singular value decomposition. *Biomed. Eng. Adv.* **2**, 100013 (2021).
- Asteriou, D. & Hall, S. G. Arima models and the Box-Jenkins methodology. *Appl. Econ.* **2**, 265–286 (2011).
- Sykacek, P. & Roberts, S. J. Bayesian time series classification. *Adv. Neural Inf. Process. Syst.* **14**, 179 (2001).
- Garrett, D., Peterson, D. A., Anderson, C. W. & Thaut, M. H. Comparison of linear, nonlinear, and feature selection methods for EEG signal classification. *IEEE Trans. Neural Syst. Rehabil. Eng.* **11**, 141–144 (2003).
- He, Z.-Y. & Jin, L.-W. Activity recognition from acceleration data using AR model representation and SVM. in *2008 International Conference on Machine Learning and Cybernetics*. Vol. 4. 2245–2250. <https://doi.org/10.1109/ICMLC.2008.4620779> (2008).
- Fang, Z., Wang, P. & Wang, W. Efficient learning interpretable shapelets for accurate time series classification. in *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. 497–508 (IEEE, 2018).
- Dempster, A., Petitjean, F. & Webb, G. I. Rocket: Exceptionally fast and accurate time series classification using random convolutional kernels. *Data Min. Knowl. Discov.* **34**, 1454–1495 (2020).
- Li, G., Yan, W. & Wu, Z. Discovering shapelets with key points in time series classification. *Exp. Syst. Appl.* **132**, 76–86 (2019).
- Zagorecki, A. A versatile approach to classification of multivariate time series data. in *2015 Federated Conference on Computer Science and Information Systems (FedCSIS)*. 407–410 (IEEE, 2015).
- Mello, C. E., Carvalho, A. S., Lyra, A. & Pedreira, C. E. Time series classification via divergence measures between probability density functions. *Pattern Recognit. Lett.* **125**, 42–48 (2019).
- ALTobi, M. A. S., Bevan, G., Wallace, P., Harrison, D. & Ramachandran, K. Fault diagnosis of a centrifugal pump using MLP-GABP and SVM with CWT. *Eng. Sci. Technol. Int. J.* **22**, 854–861 (2019).

38. Raghu, S. *et al.* Performance evaluation of DWT based sigmoid entropy in time and frequency domains for automated detection of epileptic seizures using svm classifier. *Comput. Biol. Med.* **110**, 127–143 (2019).
39. Al-Ghraibah, A., Boucheron, L. & McAteer, R. An automated classification approach to ranking photospheric proxies of magnetic energy build-up. *Astron. Astrophys.* **579**, A64 (2015).
40. Boone, K. Avocado: Photometric classification of astronomical transients with gaussian process augmentation. *Astron. J.* **158**, 257 (2019).
41. Zheng, Y., Liu, Q., Chen, E., Ge, Y. & Zhao, J. L. Time series classification using multi-channels deep convolutional neural networks. in *International Conference on Web-Age Information Management*. 298–310 (Springer, 2014).
42. Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L. & Muller, P.-A. Deep learning for time series classification: a review. *Data Min. Knowl. Discov.* **33**, 917–963 (2019).
43. Wang, J. & Tang, S. Time series classification based on ARIMA and Adaboost. in *MATEC Web of Conferences*. Vol. 309. 03024 (EDP Sciences, 2020).
44. Pearson, K. Liii. On lines and planes of closest fit to systems of points in space. *Lond. Edinb. Dublin Philos. Mag. J. Sci.* **2**, 559–572 (1901).
45. Hotelling, H. Analysis of a complex of statistical variables into principal components. *J. Educ. Psychol.* **24**, 417 (1933).
46. Sharabiani, A. *et al.* Efficient classification of long time series by 3-D dynamic time warping. *IEEE Trans. Syst. Man Cybern. Syst.* **47**, 2688–2703 (2017).
47. Keogh, E. J. & Pazzani, M. J. An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. *KDD* **98**, 239–243 (1998).
48. McCloskey, M. & Cohen, N. J. Catastrophic interference in connectionist networks: The sequential learning problem. in *Psychology of Learning and Motivation*. Vol. 24. 109–165 (Elsevier, 1989).
49. Robins, A. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connect. Sci.* **7**, 123–146 (1995).
50. French, R. M. Catastrophic forgetting in connectionist networks. *Trends Cognit. Sci.* **3**, 128–135 (1999).
51. Ullah, I. & Youn, H. Y. Efficient data aggregation with node clustering and extreme learning machine for WSN. *J. Supercomput.* **76**, 10009–10035 (2020).
52. Kurbucz, M. T., Pósfay, P. & Jakovác, A. Facilitating time series classification by linear law-based feature space transformations. in *SSRN*. 4161139 (2022).
53. Jakovác, A., Kurbucz, M. T. & Pósfay, P. Reconstruction of observed mechanical motions with artificial intelligence tools. *New J. Phys.* (2022).
54. Jakovác, A. Time series analysis with dynamic law exploration. <https://doi.org/10.48550/ARXIV.2104.10970> (2021).
55. Takens, F. Dynamical systems and turbulence. *Lect. Notes Math.* **898**, 366 (1981).
56. Palumbo, F., Gallicchio, C., Pucci, R. & Micheli, A. Human activity recognition using multisensor data fusion based on reservoir computing. *J. Ambient Intell. Smart Environ.* **8**, 87–107 (2016).
57. Vidya, B. & Sasikumar, P. Wearable multi-sensor data fusion approach for human activity recognition using machine learning algorithms. *Sens. Actuators A Phys.* **341**, 113557 (2022).

Acknowledgements

The authors would like to thank András Telcs and Zoltán Somogyvári for their valuable comments and suggestions. The authors appreciate the support of Eötvös Loránd Research Network. The research was supported by the Ministry of Innovation and Technology NRD Office within the framework of the MILAB Artificial Intelligence National Laboratory Program. A.J. and M.T.K. received support from the Hungarian Scientific Research Fund (OTKA/NKFIH) under contracts No. K123815 and PD142593, respectively.

Author contributions

A.J., M.T.K., and P.P. conceptualized the study and contributed to the design of the proposed model. M.T.K. implemented the algorithm and performed the analysis. All authors wrote and reviewed the manuscript.

Funding

Open access funding provided by ELKH Wigner Research Centre for Physics.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1038/s41598-022-22829-2>.

Correspondence and requests for materials should be addressed to M.T.K.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2022