Benchmarking Classifiers on Medical Datasets of UEA Archive

Rufat Babayev
Fraunhofer ITEM
Hannover, Germany
rufat.babayev@item.fraunhofer.de

Lena Wiese
Fraunhofer ITEM
Hannover, Germany
lena.wiese@item.fraunhofer.de

ABSTRACT

Time Series Classification (TSC) includes generation of classifier models for discrete labeled time series data containing real-valued measurements of different variables collected in a temporal order. Over the last years, several TSC algorithms have been proposed both in the traditional machine learning and deep learning domains which have shown remarkable enhancement over the previously published state-of-the-art methods. General emphasis has been based on univariate TSC (UTSC), where a time series containing measurements of a single variable is associated with a class label. In contrast, the medical domain has been more focused on mutlivariate TSC (MTSC) (where multiple variables are associated with a label) considering the availability of popular publicly available medical datasets such as PhysioNet [24] and MIMIC-III [13]. These datasets are fairly complex having high missing rate and unequal length time series. In comparison, UEA archive includes 8 medical datasets having equal length time series without any missing values which makes the comparison of algorithms straightforward. The direct (dimension independent) technique to MTSC is to apply univariate classifiers on the dimensions individually. We compare recent bespoke MTSC algorithms to the dimension independent techniques on 8 datasets from UEA archive. The results show that dimension independent techniques with/without the application of Principal Component Analysis (PCA) have comparable or better scores in some configurations.

CCS CONCEPTS

• **Computing methodologies → Machine learning**.

KEYWORDS

time series classification, evaluating classifiers, multivariate time series, UEA archive, principal component analysis

ACM Reference Format:
Rufat Babayev and Lena Wiese. 2021. Benchmarking Classifiers on Medical Datasets of UEA Archive. In *Proceedings of AI Health WWW 2021: International Workshop on AI in Health: Transferring and Integrating Knowledge for Better Health (AI Health WWW 2021).* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

AI Health WWW 2021, April 19, 2021, Ljubljana, Slovenia

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Time Series Classification (TSC) is a type of supervised machine learning where attributes of the input vector have ordered and real-valued entries. This makes time series data different from what the traditional algorithms are designed for: traditional algorithms can miss temporal order if they are fed with such kind of data. Over the last years, many TSC algorithms have been proposed which have shown remarkable improvement over the previously published methods [4, 22].

Over the years, the main emphasis has been on univariate time series classification, where each series contains records of a single variable and a class label. However, it is more common to notice MTSC tasks, (especially in the medical domain) where time series contains records of multiple variables and associated label. For example, activity recognition, measurements based on EEG, ECG and health monitoring are all multivariate by nature. Nevertheless, the general focus in TSC field has been on the univariate case. For example, sktime [17] which is a popular unified framework for machine learning with time series, contains significantly more univariate algorithms than the multivariate ones. The sktime framework is designed to work with datasets from UCR [9] and UEA [3] archives where the former is a resource for UTSC, and the latter is for MTSC. The 2018 versions of archives contain 128 and 30 datasets respectively. According to our findings, there are 22 medical datasets in UCR archive and 14 medical datasets in UEA archive. Since, the datasets both in UCR and UEA are equal length and do not have missing values, making comparison of algorithms on them becomes straightforward without additional processing.

Using sktime, we compare recent MTSC algorithms to the direct dimension independent transformations of univariate classifiers for 8 medical datasets from UEA archive. More formally, two MTSC algorithms are compared to five transformations. Moreover, we apply PCA [28] on the same datasets to see the effect of feature-reduction on the classification performance. PCA is a technique that is used to explain the variance-covariance structure of a set of features through linear combinations [28]. In both cases, the results show that the dimension independent transformations demonstrate comparable or similar classification performance on various configurations.

2 RELATED WORK

One of the well-known applications of univariate time series classification is the usage of different distance measures. In this respect, [2, 20] thoroughly reviewed different measures in the domain of UTSC. Through kNN or SVM, these distance measures can be utilized for TSC. The Python-based sktime library and its Java counterpart tsml [6] provide a lot of distance measures and TSC algorithms in this respect. Moreover, a lot of research has been carried out

for multivariate time series classification using data from the UEA archive [3–5, 22].

The closest to our paper is [22] which uses the approach to build an ensemble of univariate classifiers over multiple dimensions for MTSC. They compare the classification performance of the ensemble classifier to deep learning models, traditional algorithms using dynamic time warping (DTW) and combined methods on the UEA archive of multivariate time series datasets [3].

In contrast, our paper is focused on medical datasets. Instead of building an ensemble of univariate classifiers, we directly apply univariate classifiers over dimensions, compare them to bespoke MTSC algorithms and investigate an influence of PCA on classification results. We not only assess the accuracy score, but also other necessary metrics such as f1, recall and AUROC (AUC) score.

To the best of our knowledge, our paper is the pioneering one which is focused on medical datasets from UEA archive using the transformations of univariate classifiers for MTSC.

3 PRELIMINARIES

In this section, we present mathematical notations for a multivariate (multidimensional) time series and briefly discuss background for MTSC.

Following the notations from [8], we specify a multivariate time series with D variables (a.k.a, a D-dimensional time series) of length T as $X = (x_1, x_2, \ldots, x_T)^\intercal \in \mathbb{R}^{T \times D}$, where $\forall t = \{1, 2, \ldots, T\}$, $x_t \in \mathbb{R}^D$ is a vector which represents the t-th measurements (observations) of all variables and x_t^d is the observation of d-th variable of x_t. In this paper, we focus on time series classification to predict a label $l_n \in \{1, \ldots, L\}$ for each of N multivariate time series collected in a dataset $\mathcal{D}$, where $\mathcal{D} = \{(X_n)\}_{n=1}^N$ and $X_n = \left[x_1^{(n)}, x_2^{(n)}, \ldots, x_{T_n}^{(n)}\right]$.

The extra complexity for MTSC is that differential patterns may be dependent on the dimension interactions, not just correlation, or the size of the data may conceal such patterns. With this in mind, MTSC algorithms can be classified in the same manner as UTSC algorithms; e.g. distance-based, shapelet-based, dictionary-based, interval-based and deep neural networks. Distance based methods are primarily focused on dynamic time warping (DTW) [11] or its modifications. A straightforward method is 1-nearest neighbors with DTW distance which is a popular benchmark. This can be extended towards MTSC by adapting it over dimensions.

The interval-based methods that can be adapted for MTSC include forests of decision trees such as Time Series Forest (TSF) [10] and Random Interval Spectral Forest (RISF) [16]. The effective dictionary based methods that can be adapted for MTSC are Contract Bag Of Symbolic Fourier Approximation Symbols (cBOSS) [19, 23] and Word Extraction for time series classification (WEASEL) [22]. There is also a bespoke MTSC algorithm which is a combination of WEASEL and Multivariate Unsupervised Symbols and dErivatives (MUSE) called WEASEL+MUSE or simply MUSE [22]. Finally, a bespoke shapelet-based method for MTSC is MrSEQL (Multiple Representation Sequence Learner) [15] which utilizes the combination of linear models and a symbolic sequence learning algorithm.

4 EMPIRICAL EVALUATION

We evaluate the performance of the classification on multivariate time series data using several experimental configurations. We evaluate our models for different settings such as for different datasets, PCA transformations of these datasets, different adaptations of univariate classifiers and bespoke MTSC algorithms.

4.1 Datasets and Task Description

The UCR (University of California Riverside) archive was first released in 2002 with 16 datasets. It was gradually extended with more datasets and the 2018 version of the UCR archive contains a broader scope of cases, including variable length time series, but it is still a resource for UTSC problems.

Like the univariate counterpart, the UEA archive was a result of a partnership between the academic staff at the University of East Anglia (UEA) and the University of California, Riverside (UCR) [3]. In comparison to UCR, it contains only multivariate time series datasets. The 2018 version of the UEA archive has 30 datasets with a broad scope of problems, dimensions (variables) and time series lengths. That version contains the data which are formatted to be of equal length, includes no missing values and provides train/test splits.

In our study we explicitly selected datasets related to the medical domain from the UEA archive. They are multivariate time series datasets. The descriptions of the datasets are as follows:

Epilepsy [14] - The data were constructed with healthy participants simulating the class activities. Data was collected from 6 participants using a 3D accelerometer on the dominant wrist whilst conducting 4 different activities [3].

FingerMovements [7] - This dataset consists of 500 milliseconds intervals of EEG recordings, 130 milliseconds prior to the moment a key is pressed by a subject. A single subject, sitting in a normal position at keyboard was asked to type characters using only the index and pinky fingers. [4]. There are two classes: left and right.

HandMovementDirection [25] - The data set contains directionally modulated MEG activity that was recorded while subjects performed wrist movements in four different directions. Brain activity during wrist movements was recorded with MEG at 625 Hz from two healthy, right-handed subjects.

Heartbeat [1] - This dataset is derived from the PhysioNet/CinC Challenge 2016. Heart sound recordings were sourced from several contributors around the world, collected in either a clinical or non-clinical environment, from both healthy subjects and pathological patients [3].

BasicMotions [12] - The dataset was created as part of a student project where four students performed four activities while wearing a smart watch. The watch collected 3D accelerometer and 3D gyroscope data. The dataset contains four classes: walking, resting, running and badminton [3].

SelfRegulationSCP1 [26] - It is obtained from the dataset Ia of BCI II competition which consists of "*Self-regulation of Slow Cortical Potentials*". The dataset is provided by University of Tuebingen. The data were taken from a healthy subject. The subject was asked to move a cursor up and down on a computer screen, while his cortical potentials were taken.

SelfRegulationSCP2 [27] - It is obtained from the dataset Ib of BCI II competition which consists of "*Self-regulation of Slow Cortical Potentials*". The dataset is provided by University of Tuebingen. The data were taken from an artificially ventilated ALS patient. The subject was asked to move a cursor up and down on a computer screen, while his cortical potentials were taken.

EyesOpenShut [29] - The problem is to detect whether a person's eyes are open or shut based on a 1 second reading of an EEG This is a reformulation of the data on the UCI archive[1].

Characteristic properties of the datasets are given in Table 1.

Table 1: Properties of the datasets. Train refers to train set size, Test to test set size, Dim to the number of dimensions (variables), Len to the length of time series (number of time steps), #class. to the number of classes (labels) and Type to the type of the dataset (HAR - Human Acitivity Recognition, EEG - Electroencephalogram).

Dataset	Train	Test	Dim	Len	#class.	Type
Epilepsy	137	138	3	207	4	HAR
FingerMovements	316	100	28	50	2	EEG
HandMovementDirection	160	74	10	400	4	EEG
Heartbeat	204	205	61	405	2	AUDIO
BasicMotions	40	40	6	100	4	HAR
SelfRegulationSCP1	268	293	6	896	2	EEG
SelfRegulationSCP2	200	180	7	1152	2	EEG
EyesOpenShut	56	42	14	128	2	EEG

We discarded 6 other medical datasets: EigenWorms, MotorImagery and FaceDetection, AtrialFibrillation, StandWalkJump, ERing. They either have big train/test size and length which makes TSC algorithms (especially MTSC algorithms) run very slowly or have small train/test size which are not relevant for PCA transformation.

We performed either binary or multiclass classification on these datasets and reported the results of different metrics. The class imbalance of the datasets are given in Table 2. Imbalances are in the acceptable range, therefore, we did not apply any resampling or any imbalanced learning strategy.

Table 2: Class imbalance of the datasets.

Dataset	Training set imbalance
Epilepsy	24.82% - 27.01% - 26.28% - 21.9%
FingerMovements	50.32% - 49.68%
HandMovementDirection	25% - 25% - 25% - 25%
Heartbeat	27.94% - 72.06%
BasicMotions	25% - 25% - 25% - 25%
SelfRegulationSCP1	50.37% - 49.63%
SelfRegulationSCP2	50% - 50%
EyesOpenShut	58.93% - 41.07%

[1]https://archive.ics.uci.edu/ml/datasets/EEG+Eye+State

4.2 Machine Learning Approaches

In our experiments, we used the column ensembling strategy from sktime library through **ColumnEnsembleClassifier** class. This strategy enables univariate classifiers to be applied to the multivariate time series data where in reality, each dimension of a multivariate time series is a univariate time series. The main point here is that a multivariate time series is split into several univariate ones and a classifier is applied to each of the dimensions (variables, columns) independently, then predictions are aggregated. We call such univariate classifiers as *dimension independent* classifiers or *adapted* classifiers for MTSC. Such classifiers can also be called *adaptations* or *transformations* for short.

We tested the following univariate classifiers in our experiments which are adapted for MTSC:

TimeSeriesForest (TSF) - Implementation of Deng's Time Series Forest using intervals. Number of trees in the forest is 100.

ContractableBOSS (cBOSS) - dictionary based cBOSS classifier based on Symbolic Fourier Approximation (SFA) transform [19, 23]. In SFA, a time series is approximated using the truncated Fourier transform. This classifier improves the ensemble structure of the original BOSS algorithm.

WEASEL (W-EL) - dictionary based classifier based on SFA transform, BOSS and linear regression.

KNeighborsTimeSeriesClassifier (kNN) - KNN time series classification built on sklearn [21] KNeighborsClassifier. We explicitly used 1-nearest neighbors with DTW distance which is a well-known benchmark for TSC tasks.

RandomIntervalSpectralForest (RISF) - Implementation of Deng's Time Series Forest, with minor changes. It is still an interval-based tree classifier. As compared to TSF, it uses a single interval for each tree, and utilizes spectral features rather than summary statistics. Selected number of trees in the forest is 100.

In our experiments we also tested two bespoke multivariate classifiers:

MrSEQLClassifier (MrSEQL) - is a classifier which trains linear classification models (logistic regression) with features extracted from multiple symbolic representations of time series (SAX, SFA). For MTSC, MrSEQL extracts features from each dimension of the data independently.

MUSE (MUSE) - multivariate dictionary based classifier based on SFA transform and dictionaries. It is a multivariate extension of WEASEL.

All classifiers are run with user-provided and/or default parameters and random seed of 1 for reproducibility.

5 INTERPRETATION OF RESULTS

Results are generated for different classifiers and datasets. In Table 1, various properties of the datasets are given. In terms of properties, datasets range from long (SelfRegulationSCP2) to short (FingerMovements) in length, from small (BasicMotions, EyesOpenShut) to moderate (the rest of the datasets) on the number of samples and by the number of classes (either binary or multiclass) and types of the datasets. This span of the properties provides certain configurations to test the classifiers.

The sktime is sklearn compatible, therefore, all scores are obtained using *average* parameter with the value of "*macro*", AUC

Table 3: Performance of classifiers in Epilepsy, FingerMovements, HandMovementDirection and Heartbeat datasets.

	Epilepsy				FingerMovements				HandMovementDirection				Heartbeat			
Class.	Acc.	F1	AUC	Rec.	Acc.	F1	AUC	Rec.	Acc.	F1	AUC	Rec.	Acc.	F1	AUC	Rec.
TSF	0.8985	0.8980	0.9840	0.9	0.52	0.5198	0.5548	0.5206	**0.3918**	**0.3694**	**0.6581**	**0.3821**	0.7219	0.5442	0.6330	0.5539
RISF	0.9565	0.9569	0.9982	0.9588	0.51	0.5059	0.5046	0.5120	0.2162	0.1942	0.4770	0.1952	**0.7658**	0.5665	0.6847	0.5789
kNN	0.7028	0.6963	0.7984	0.6977	**0.59**	**0.5849**	**0.5880**	**0.5880**	0.2432	0.2447	0.5023	0.2535	0.6926	0.4511	0.4959	0.4959
cBOSS	0.9710	0.9718	0.9998	0.9729	0.5	0.4949	0.4993	0.5022	0.2702	0.2537	0.5011	0.3023	0.7073	0.4142	0.4916	0.4898
W-EL	0.9710	0.9718	0.9948	0.9729	0.55	0.5488	0.5554	0.5492	0.2837	0.2574	0.4834	0.2619	0.7170	0.5871	0.6896	0.5829
MrSEQL	0.9927	0.9921	1.0	0.9926	0.56	0.5592	0.5682	0.5610	0.1486	0.1401	0.4841	0.1428	0.7317	0.4842	**0.7910**	0.5283
MUSE	**1.0**	**1.0**	**1.0**	**1.0**	0.57	0.5689	0.5850	0.5712	0.2432	0.2142	0.5736	0.2511	0.7365	**0.5957**	0.7715	**0.5910**

Table 4: Performance of classifiers in BasicMotions, SelfRegulationSCP1, SelfRegulationSCP2 and EyesOpenShut datasets.

	BasicMotions				SelfRegulationSCP1				SelfRegulationSCP2				EyesOpenShut			
Class.	Acc.	F1	AUC	Rec.	Acc.	F1	AUC	Rec.	Acc.	F1	AUC	Rec.	Acc.	F1	AUC	Rec.
TSF	**1.0**	**1.0**	**1.0**	**1.0**	0.7185	0.7041	0.9297	0.7185	0.4944	0.4909	0.4958	0.4944	0.4523	0.4367	0.5306	0.4523
RISF	**1.0**	**1.0**	**1.0**	**1.0**	0.6666	0.6426	0.8881	0.6666	**0.5222**	**0.5221**	**0.5676**	**0.5222**	**0.5714**	**0.5178**	**0.5748**	**0.5714**
kNN	0.5	0.4538	0.6666	0.5	0.6814	0.6754	0.8088	0.6814	0.4888	0.4848	0.4888	0.4888	0.4523	0.4445	0.4523	0.4523
cBOSS	**1.0**	**1.0**	**1.0**	**1.0**	0.6444	0.6140	0.9278	0.6444	0.4722	0.4722	0.4807	0.4722	0.5	0.3713	0.6122	0.5
W-EL	0.925	0.9236	0.98	0.925	0.7703	0.7624	0.9538	0.7703	0.5055	0.5048	0.5307	0.5055	0.4761	0.4296	0.5623	0.4761
MrSEQL	0.95	0.9494	0.9975	0.95	0.8777	0.8716	0.9938	0.8777	0.5055	0.4973	0.4612	0.5055	0.5	0.3333	0.5056	0.5
MUSE	**1.0**	**1.0**	**1.0**	**1.0**	**0.9703**	**0.9705**	**0.9982**	**0.9703**	0.5166	0.4276	0.5350	0.5166	0.5	0.4631	0.5374	0.5

score is additionally obtained using *multi_class* parameter being "*ovo*" (one vs. one).

We report the results obtained from the datasets in Table 3 and Table 4.

On the Epilepsy dataset, MUSE outperforms all other classifiers. For this dataset, MrSEQL performs closest to MUSE. The dimension independent classifier closest to MUSE in terms of performance is cBOSS. Overall, every classifier has high performance in this dataset, since this dataset is fairly straightforward having only three dimensions (variables).

In comparison to Epilepsy, bespoke MTSC algorithms do not outperform all dimension independent classifiers in FingerMovements dataset. Despite of the fact that, FingerMovements dataset is the second among the datasets in terms of dimensions (28), dimension independent kNN (1NN) with DTW distance performs better than any other dimension independent approaches and bespoke MTSC algorithms. In this dataset, the closest in performance to 1NN is MUSE.

On the HandMovementDirection dataset, the adapted TSF classifier outperforms all others and no other classifier comes closer to it. The number of dimensions in HandMovementDirection dataset is 10 and length of the dataset (400) is moderate and bigger than the length of Epilepsy dataset.

However, on the Heartbeat dataset which has highest number of dimensions (61) and the length similar to HandMovementDirection dataset, different classifiers performed the highest on different metrics. It is not possible to tell which classifier is the optimal for this dataset. On this dataset, MUSE has the highest F1 and Recall scores, and the second highest Accuracy and AUC scores. But, there are dimension independent classifiers (RISF, WEASEL) which perform similarly to MUSE as well. Note that, type of this dataset is AUDIO.

In Table 4, the results for the last 4 datasets are reported. BasicMotions dataset has smallest train/test size. It is the second among the datasets in terms of dimensions (6) and the length (100). Therefore, multiple dimension independent classifiers and bespoke MTSC algorithm MUSE have the same scores. Since the type of the dataset is HAR, we notice a similar phenomenon as we noticed on the Epilepsy dataset.

SelfRegulationSCP1 and SelfRegulationSCP2 are also EEG datasets and they have highest length among the datasets we examine. In the former dataset, MUSE significantly outperforms all other classifiers. On the latter dataset, RISF outperforms all others, while MUSE becomes the closest to RISF. SelfRegulationSCP2 is the dataset with the highest length, therefore, the performance of RISF as a dimension independent classifier is significant here.

EyesOpenShut is the last EEG dataset with the second smallest train/test size and the third smallest length (128). Here, RISF significantly outperforms all other classifiers including MUSE.

The results obtained from FingerMovements, HandMovementDirection, SelfRegulationSCP2 and EyesOpenShut datasets demonstrate the effectiveness of dimension independent approaches in higher dimensional and long multivariate time series data; especially the EEG data.

6 PRINCIPAL COMPONENT ANALYSIS

To further examine the dimension independence, time series length and their influence on the classification performance, we applied Prinicipal Component Analysis (PCA) to the datasets. To do this, we use **sktime.transformations.panel.pca.PCATransformer** class that applies PCA on univariate time series data. It provides a simple wrapper around **sklearn.decomposition.PCA** class. In the context of sktime library, there is no PCA transformer that can

apply transformation directly on the multivariate time series data. Therefore, we apply PCA transformation in each dimension of the multivariate time series data independently and then combine transformed data back to multivariate time series data. In general, when PCA is applied on some dataset, the following inequality has to be satisfied:

$$0 < numComponents \leq min(numSamples, numFeatures) \quad (1)$$

where $numSamples$ is the number of instances and $numFeatures$ is the number of features/attributes in a dataset. For the (univariate) time series data, $numSamples$ is the number of time series in the dataset and $numFeatures$ is the length of the time series (assuming the fact that all time series are equal length). Since, the used datasets have predefined train and test splits, $numSamples$ in the train and test splits are different. However, the length of the series is the same for train and test splits. Since, we apply PCA for each dimension of the multivariate time series independently, the Eq. (1) is formulated as follows:

$$0 < numComponents \leq min(trainSize, testSize, tsLength) \quad (2)$$

where $trainSize$ is the train set size, $testSize$ is the test set size and $tsLength$ is the length of a time series.

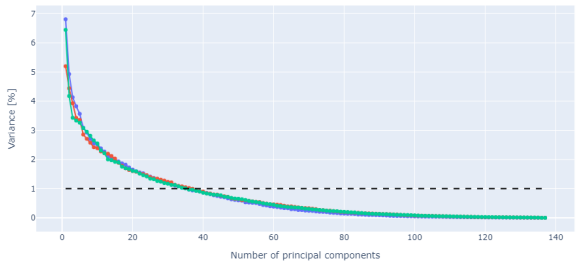

Figure 1: Epilepsy dataset: optimal number of principal components under 1% variance is 35.

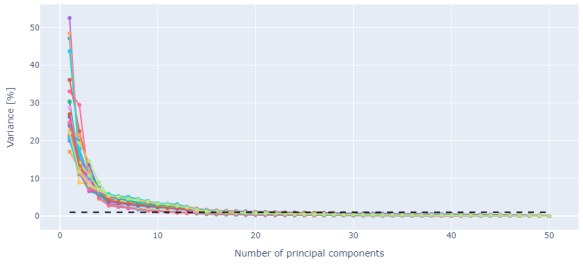

Figure 2: FingerMovements dataset: optimal number of principal components under 1% variance is 20.

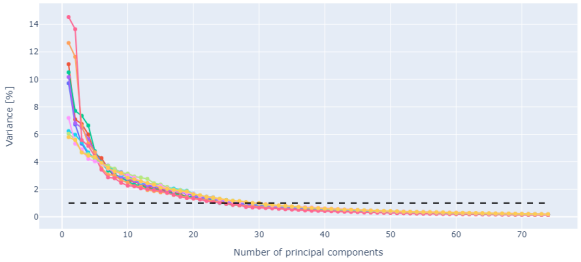

Figure 3: HandMovementDirection dataset: optimal number of principal components under 1% variance is 25.

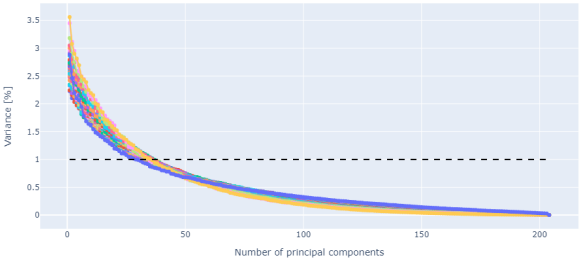

Figure 4: Heartbeat dataset: optimal number of principal components under 1% variance is 35.

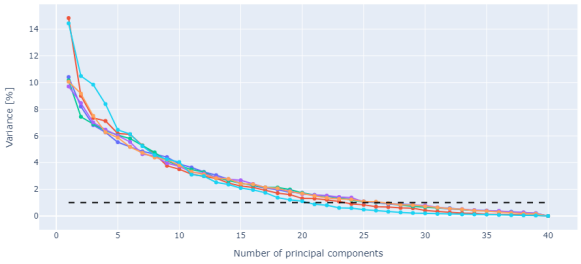

Figure 5: BasicMotions dataset: optimal number of principal components under 1% variance is 25.

PCA performs best with standardized data. We fit standardization transformer **sklearn.preprocessing.StandardScaler** with mean of 0 and variance of 1 on the train set. Then, we transform train and test sets with the fitted transformer. This is a straightforward way, if there are train and test splits in the dataset.

The next step is the selection of fixed number of components for our PCA transformer which transforms each dimension of the multivariate time series into a fixed length series. Then, obtained series

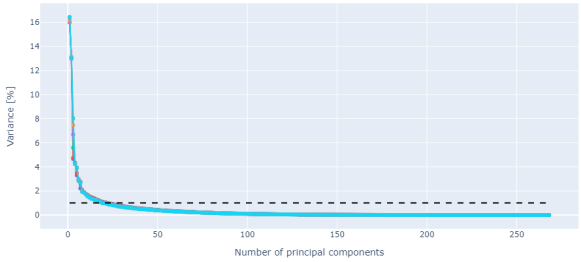

Figure 6: SelfRegulationSCP1 dataset: optimal number of principal components under 1% variance is 20.

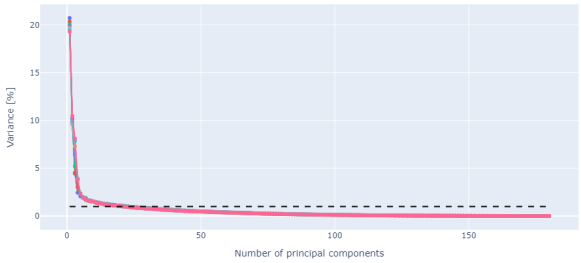

Figure 7: SelfRegulationSCP2 dataset: optimal number of principal components under 1% variance is 25.

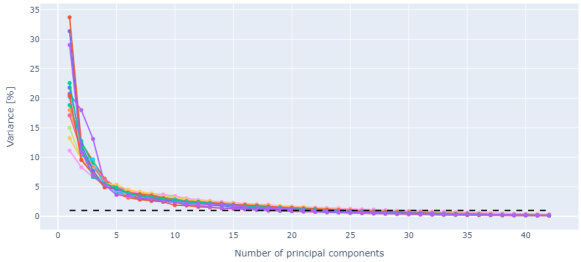

Figure 8: EyesOpenShut dataset: optimal number of principal components under 1% variance is 25.

are combined back to the multivariate time series and classification is performed once again.

To find the fixed number of principal components, we utilize "*explained variance ratio*". The explained variance ratio returns the variance caused by each of the principal components. The number of principal components to keep in a feature set depends on various conditions such as storage capacity, training time, performance and so on. A common rule of thumb is to take number of components that contribute to significant variance and ignore the components with diminishing variance values. A straightforward way for this is to plot the variance against principal components and ignore the principal components with diminishing values. For our datasets, we defined this as a spot where the variance drops below 1%. The number which determines this spot is chosen for all dimensions of multivariate time series data. In this case, the length of time series for every dataset becomes equal to the corresponding number. We provide "*Variance - Number of principal components*" plots in Figure 1 - Figure 8. At the end, we fit a separate PCA transformer for each dimension of the multivariate time series data in the standardized train set. Then, we use fitted PCA transformers to transform the data in the standardized train and test sets. After PCA transformation, Epilepsy, FingerMovements, HandMovementDirection, Heartbeat, BasicMotions, SelfRegulationSCP1, SelfRegulationSCP2 and EyesOpenShut datasets have a length of 35, 20, 25, 35, 25, 20, 25, 25, respectively.

The classification results on the PCA transformed data are given in Table 5 and Table 6. These tables can be directly compared to Table 3 and Table 4 respectively.

For the Epilepsy dataset, the length is dropped from 207 to 35. In terms of performance, we notice a significant drop in classification results with respect to Table 3. However, the dimension independent 1NN classifier with DTW distance still has better scores.

On the FingerMovements dataset, there is no significant drop in the length after PCA transformation. The length of the time series in the dataset is dropped from 50 to 20. We notice that cBoss outperforms all others and has comparable performance to the winner classifier 1NN in Table 3.

For the HandMovementDirection dataset, the length of the dataset is dropped from 400 to 25. However, a winner classifier is not changed; it is still TSF. Its scores are also comparable to the TSF of Table 3. The scores of the other classifiers are comparable to the counterparts in Table 3 as well.

For the Heartbeat dataset, TSF reaches the highest values in three metrics, shows slightly lower value only in Recall metric. The scores of all classifiers are still comparable to Table 3 considering the fact that the length of the dataset is dropped from 405 to 35.

Similar to the Epilepsy dataset, we observe a significant drop in the classification performance for BasicMotions dataset in Table 6. Note that, the length of the BasicMotions dataset is dropped from 100 to 25.

Since, SelfRegulationSCP1 and SelfRegulationsSCP2 were the longest datasets, their lengths are substantially dropped from 896 to 20 and 1152 to 25, respectively. For the former dataset, there is also remarkable drop in the classification performance. However, for the latter dataset, we even observe overall performance increase in all classifiers and specifically for WEASEL in Accuracy, F1 and Recall metrics which is notable with respect to Table 4.

Another dataset where we see a performance increase is EyesOpenShut dataset where the length is dropped from 128 to 25 after PCA transformation. On this dataset, WEASEL outperforms all other classifiers and the winner classifier from Table 4.

Overall, we observed significant performance drops in three datasets (Epilepsy, BasicMotions, SelfRegulationSCP1), comparable or almost comparable performance in three datasets (FingerMovements, HandMovementDirection, Hearbeat) and performance increase in two datasets (SelfRegulationsSCP2, EyesOpenShut) with

Table 5: Performance of classifiers in Epilepsy, FingerMovements, HandMovementDirection and Heartbeat datasets after applying PCA.

	Epilepsy				FingerMovements				HandMovementDirection				Heartbeat			
Class.	Acc.	F1	AUC	Rec.	Acc.	F1	AUC	Rec.	Acc.	F1	AUC	Rec.	Acc.	F1	AUC	Rec.
TSF	0.5652	0.5314	0.8098	0.5587	0.56	0.5598	0.5350	0.5606	**0.3783**	**0.3675**	**0.5895**	**0.3821**	**0.7219**	**0.4192**	**0.5701**	0.5
RISF	0.6231	0.5765	**0.8437**	0.6078	0.49	0.4812	0.4989	0.4927	0.1756	0.1728	0.5116	0.1880	0.7219	0.4192	0.4831	0.5
kNN	**0.6956**	**0.6897**	0.7956	**0.6935**	0.57	0.5696	0.5708	0.5708	0.2972	0.2850	0.5253	0.2880	0.6146	0.5004	0.5011	0.5011
cBOSS	0.4275	0.4134	0.7010	0.4160	**0.58**	**0.5738**	**0.5750**	**0.5778**	0.2027	0.1811	0.4505	0.2178	0.7219	0.4192	0.4139	0.5
W-EL	0.4275	0.4246	0.6934	0.4227	0.45	0.4486	0.4725	0.4511	0.2702	0.2697	0.5669	0.2857	0.7024	0.4804	0.5529	**0.5134**
MrSEQL	0.5724	0.5669	0.7641	0.5662	0.46	0.4591	0.4853	0.4609	0.2567	0.2655	0.5260	0.3059	0.7219	0.4192	0.4632	0.5
MUSE	0.5072	0.4743	0.7284	0.4915	0.43	0.4294	0.4385	0.4307	0.3378	0.3415	0.5855	0.3642	0.7219	0.4192	0.5124	0.5

Table 6: Performance of classifiers in BasicMotions, SelfRegulationSCP1, SelfRegulationSCP2 and EyesOpenShut datasets after applying PCA.

	BasicMotions				SelfRegulationSCP1				SelfRegulationSCP2				EyesOpenShut			
Class.	Acc.	F1	AUC	Rec.	Acc.	F1	AUC	Rec.	Acc.	F1	AUC	Rec.	Acc.	F1	AUC	Rec.
TSF	0.375	0.3188	0.6704	0.375	0.5563	0.5498	0.6028	0.5567	0.5333	0.5333	0.5465	0.5333	0.4285	0.3571	0.4410	0.4285
RISF	0.45	0.4159	0.7354	0.4499	0.4334	0.4278	0.4313	0.4337	0.55	0.5498	0.5174	0.55	0.4523	0.3943	0.4297	0.4523
kNN	**0.7**	**0.6992**	**0.8**	**0.7**	**0.6587**	**0.6575**	0.6589	**0.6589**	0.5388	0.5377	0.5388	0.5388	0.5238	0.5238	0.5238	0.5238
cBOSS	0.275	0.2347	0.52	0.275	0.4539	0.3867	0.4716	0.4550	0.4888	0.4623	**0.5467**	0.4888	0.4285	0.3571	0.3786	0.4285
W-EL	0.4	0.3727	0.6383	0.4	0.4709	0.4707	0.4441	0.4709	**0.55**	**0.5499**	0.5390	**0.55**	**0.6190**	**0.5961**	**0.5804**	**0.6190**
MrSEQL	0.625	0.6166	0.7766	0.6249	0.5494	0.5365	0.5872	0.55	0.5	0.4977	0.5214	0.5	0.5238	0.4166	0.4671	0.5238
MUSE	0.375	0.3794	0.6183	0.375	0.6552	0.6422	**0.8**	0.6559	0.4944	0.4943	0.4707	0.4944	0.4285	0.3	0.297	0.4285

respect to Table 3 and Table 4. Another observation is that only dimension independent classifiers performed better on PCA transformed datasets.

7 DISCUSSION

Our results for original datasets in Table 3 and Table 4 show that even if MUSE performs better on HAR datasets (Epilepsy, BasicMotions), dimension independent approaches are better on EEG based datasets (FingerMovements, HandMovementDirection, SelfRegulationSCP2, EyesOpenShut) despite of the fact they are straightforward adaptations of univariate classifiers for MTSC. One exception here is the AUDIO type Heartbeat dataset and the HAR type BasicMotions dataset where there is no clear winner among classifiers we examined. Another exception is the EEG type SelfRegulationSCP1 dataset, where MUSE was the clear winner.

For PCA transformed datasets in Table 5 and Table 6, only dimension independent approaches were winners. With respect to Table 3 and Table 4, we observed drops in classification performance for 3 datasets (Epilepsy, BasicMotions, SelfRegulationSCP1), comparable results for the other three (FingerMovements, HandMovementDirection, Hearbeat) and performance increase in two datasets (SelfRegulationSCP2, EyesOpenShut). There can be two reasons: i) our method to select the number of principal components (using plots) might be too straightforward; ii) since, a separate PCA transformer picks the principal components for each dimension independently, it might be possible that every transformer picks different principal components (by explained variance ratio) in each dimension discarding the correlation between the dimensions of the multivariate time series.

The paper [22] also proposed ensemble of univariate classifiers for UTSC called HIVE-COTE. Our findings show that HIVE-COTE is not combinable with ColumnEnsembleClassifier to apply it for MTSC. Therefore, we discarded HIVE-COTE in our experiments.

8 CONCLUSION

In this paper, we examined different adapted univariate classifiers for MTSC and compared them to bespoke MTSC algorithms on 8 medical datasets from the UEA archive.

For the original datasets, the dimension independent techniques outperformed MTSC algorithms on four of the examined datasets, the state-of-the-art bespoke MTSC algorithm (MUSE) performed better on two datasets, where on the remaining two datasets, there was no clear winner. For the PCA transformed datasets, only dimension independent techniques were winners.

As our future work, it will be interesting to determine in more detail how different dimensions (variables) are correlated (e.g. by fine-tuning the Principal Component Analysis) and whether this correlation has an effect on the classification performance of dimension independent techniques.

A further important aspect of future work will be making trained models interpretable or explainable. In this manner, it will be possible to detect which variables (dimensions), time intervals or slices in a series are important for a classifier, or which time slices contribute to one class or the other. For example, SHapley Additive exPlanations (SHAP) [18] can be one of the methods for explainability.

8.1 Code Availability

For the sake of reproducing the results obtained in this work, our source code is published in *ipynb* files in a public repository[2][3].

ACKNOWLEDGMENTS

This work was supported by the Fraunhofer Internal Programs under Grant No. Attract 042-601000.

REFERENCES

[1] Physionet A. Goldberger. 2016. UEA archive: Heartbeat Data Set. http://www.timeseriesclassification.com/description.php?Dataset=Heartbeat.
[2] Amaia Abanda, Usue Mori, and Jose A Lozano. 2019. A review on distance based time series classification. *Data Mining and Knowledge Discovery* 33, 2 (2019), 378–412.
[3] Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. 2018. The UEA multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075* (2018).
[4] Anthony Bagnall, Jason Lines, Aaron Bostrom, James Large, and Eamonn Keogh. 2017. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery* 31, 3 (2017), 606–660.
[5] Anthony J Bagnall, Michael Flynn, James Large, Jason Lines, and Matthew Middlehurst. 2020. A tale of two toolkits, report the third: on the usage and performance of HIVE-COTE v1. 0. *CoRR* (2020).
[6] Tony Bagnall. 2020. TSML: Java time series machine learning tools in a Weka compatible toolkit. https://github.com/uea-machine-learning/tsml.
[7] BCI2 Benjamin Blankertz. 2002. UEA archive: FingerMovements Data Set. http://www.timeseriesclassification.com/description.php?Dataset=FingerMovements.
[8] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. 2018. Recurrent neural networks for multivariate time series with missing values. *Scientific reports* 8, 1 (2018), 1–12.
[9] Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Annh Ratanamahatana, and Eamonn Keogh. 2019. The UCR time series archive. *IEEE/CAA Journal of Automatica Sinica* 6, 6 (2019), 1293–1305.
[10] Houtao Deng, George Runger, Eugene Tuv, and Martyanov Vladimir. 2013. A time series forest for classification and feature extraction. *Information Sciences* 239 (2013), 142–153.
[11] Toni Giorgino. 2009. Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package. *Journal of Statistical Software* 31, 7 (2009), 1–24. https://doi.org/10.18637/jss.v031.i07
[12] UEA Jack Clements. 2016. UEA archive: BasicMotions Data Set. http://www.timeseriesclassification.com/description.php?Dataset=BasicMotions.
[13] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data* 3 (2016), 160035.
[14] UEA Jose Ramon Villar. 2016. UEA archive: Epilepsy Data Set. http://www.timeseriesclassification.com/description.php?Dataset=Epilepsy.
[15] Thach Le Nguyen, Severin Gsponer, Iulia Ilie, Martin O'Reilly, and Georgiana Ifrim. 2019. Interpretable time series classification using linear models and multi-resolution multi-domain symbolic representations. *Data Mining and Knowledge Discovery* 33, 4 (2019), 1183–1222.
[16] Jason Lines, Sarah Taylor, and Anthony Bagnall. 2018. Time series classification with HIVE-COTE: The hierarchical vote collective of transformation-based ensembles. *ACM Transactions on Knowledge Discovery from Data* 12, 5 (2018).
[17] Markus Löning, Anthony Bagnall, Sajaysurya Ganesh, Viktor Kazakov, Jason Lines, and Franz J Király. 2019. sktime: A Unified Interface for Machine Learning with Time Series. In *Workshop on Systems for ML at NeurIPS 2019*.
[18] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*. 4765–4774.
[19] Matthew Middlehurst, William Vickers, and Anthony Bagnall. 2019. Scalable dictionary classifiers for time series classification. In *International Conference on Intelligent Data Engineering and Automated Learning*. Springer, 11–19.
[20] John Paparrizos, Chunwei Liu, Aaron J Elmore, and Michael J Franklin. 2020. Debunking Four Long-Standing Misconceptions of Time-Series Distance Measures. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 1887–1905.
[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.
[22] Alejandro Pasos Ruiz, Michael Flynn, and Anthony Bagnall. 2020. Benchmarking Multivariate Time Series Classification Algorithms. *arXiv preprint arXiv:2007.13156* (2020).
[23] Patrick Schäfer. 2015. The BOSS is concerned with time series classification in the presence of noise. *Data Mining and Knowledge Discovery* 29, 6 (2015), 1505–1530.
[24] Ikaro Silva, George Moody, Daniel J Scott, Leo A Celi, and Roger G Mark. 2012. Predicting in-hospital mortality of icu patients: The physionet/computing in cardiology challenge 2012. In *2012 Computing in Cardiology*. IEEE, 245–248.
[25] BCI4 Stephan Waldert. 2009. UEA archive: HandMovementDirection Data Set. http://www.timeseriesclassification.com/description.php?Dataset=HandMovementDirection.
[26] BCI2 Thilo Hinterberger. 1999, 2007. UEA archive: SelfRegulationSCP1 Data Set. http://www.timeseriesclassification.com/description.php?Dataset=SelfRegulationSCP1.
[27] BCI2 Thilo Hinterberger. 1999, 2007. UEA archive: SelfRegulationSCP2 Data Set. http://www.timeseriesclassification.com/description.php?Dataset=SelfRegulationSCP2.
[28] Michael E Tipping and Christopher M Bishop. 1999. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61, 3 (1999), 611–622.
[29] UEA Vinicius Souza. 2013. UEA archive: EyesOpenShut Data Set. http://www.timeseriesclassification.com/description.php?Dataset=EyesOpenShut.

[2] www.github.com/CavaJ/BenchmarkUEAMedical/blob/main/Individual%20tests.ipynb
[3] www.github.com/CavaJ/BenchmarkUEAMedical/blob/main/PCA.ipynb