Meta - Multivariate Time Series Feature Engineering

Team Member: Krisha Mehta, Marc Mascarenhas, Sindhu Madhadi, Tommy Chen
Sponsor: Dawei Li, Research Data Scientist at Meta

MOTIVATION

In this project, we aim to extract features from Multivariate time series datasets to perform classification tasks. Feature engineering, in general, creates a strong relationship between the input and output by either selecting, manipulating or transforming data into features. As we know, multivariate time series data are large and thus, we aim to limit our data size by extracting features using the mindset of reservoir computing and see if we can still get a good accuracy for the classification. The goal of this project is to add functionality to Meta's opensource Kats framework.

APPROACH

We first selected a few multivariate time-series classification datasets of different sizes. We then performed some detailed analysis on these datasets to understand them, by plotting them and even extracting the time-series features. Next, we reduced the dimensionality of the data by performing element wise multiplication and convoluted the 3-d dataset with a randomly generated matrix. We tested the performance of the system by applying machine learning algorithms such as Logistic Regression, Support Vector Machines, Random Forest and Gradient Boosting to test the performance of our feature engineering function.

QUICK Q&A

> What is Reservoir computing?

> What is the Meta Kats library?

Kats is a lightweight, easy-to-use, and generalizable framework for generic time series analysis, including forecasting, anomaly detection, multivariate analysis, and feature extraction/embedding. Kats is the first comprehensive Python library for generic time series analysis, which provides both classical and advanced techniques to model time series data.

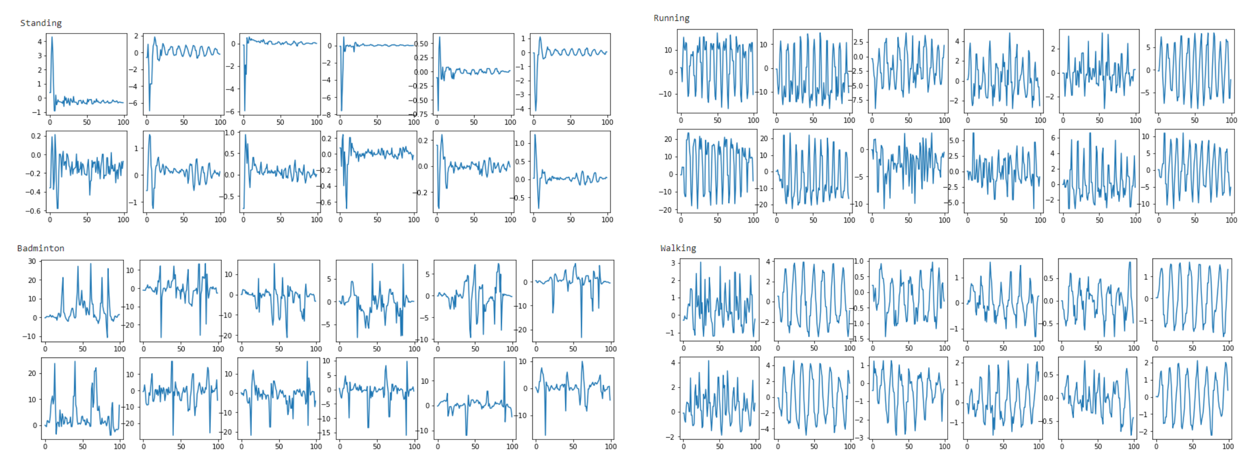
> What are some of the future steps for the project?

We're seeing good performances on some datasets and less than ideal results for others, and thus more exploration and evaluation is needed for these cases. In addition, we're seeing big difference between performances (accuracies and AUC scores) acquired from the train set and the test set, and that is worth looking into. Another future step for us is to make a pull into the Meta Kats library and further validate our method.

DATA ANALYSIS

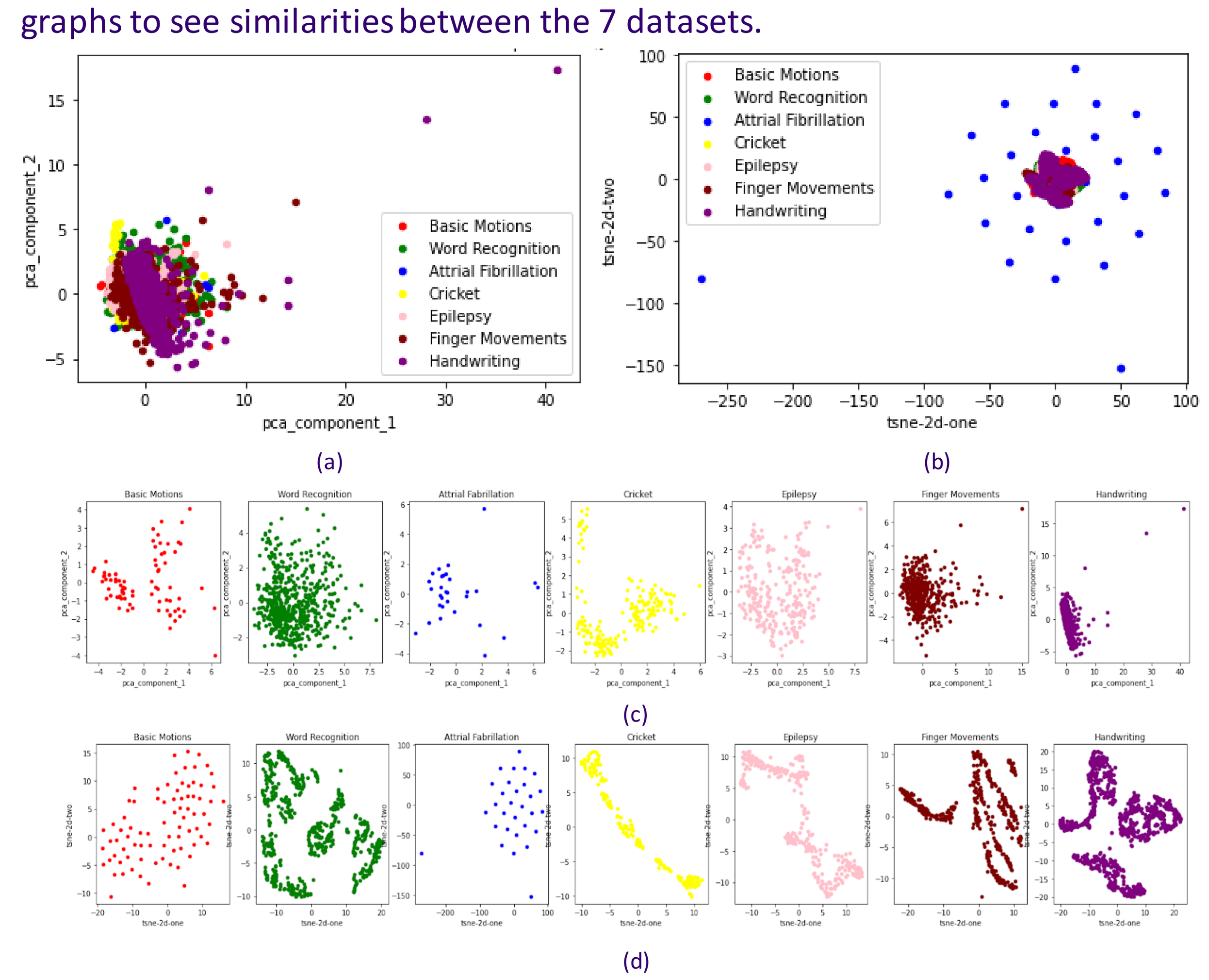
Dataset	Size	Entropy	Variance	Lumpiness	Stability
Basic Motions	(80, 60, 100)	0.744123	0.996673	0.630846	0.098126
Word Recognition	(575, 9, 144)	0.521606	0.999698	0.094382	0.774965
Atrial Fibrillation	(30, 2, 640)	0.732757	0.956361	1.326227	0.286923
Cricket	(180, 6, 1197)	0.516064	0.985533	0.138552	0.856744
Epilepsy	(275, 3, 206)	0.525518	0.834375	0.324354	0.324354
Finger Movements	(416, 28, 50)	0.581074	0.327436	0.023929	0.179505
Handwriting	(1000, 3, 152)	0.687698	0.964678	4.811281	0.245734

We used 7 multivariate time series datasets with different sizes to test our hypothesis. We import them from the pyts library and the sktime library. We studied each of these datasets in detail and also extracted the time series features using TsFeatures from kats, a library by Facebook.



Time series graphs for Basic Motions dataset

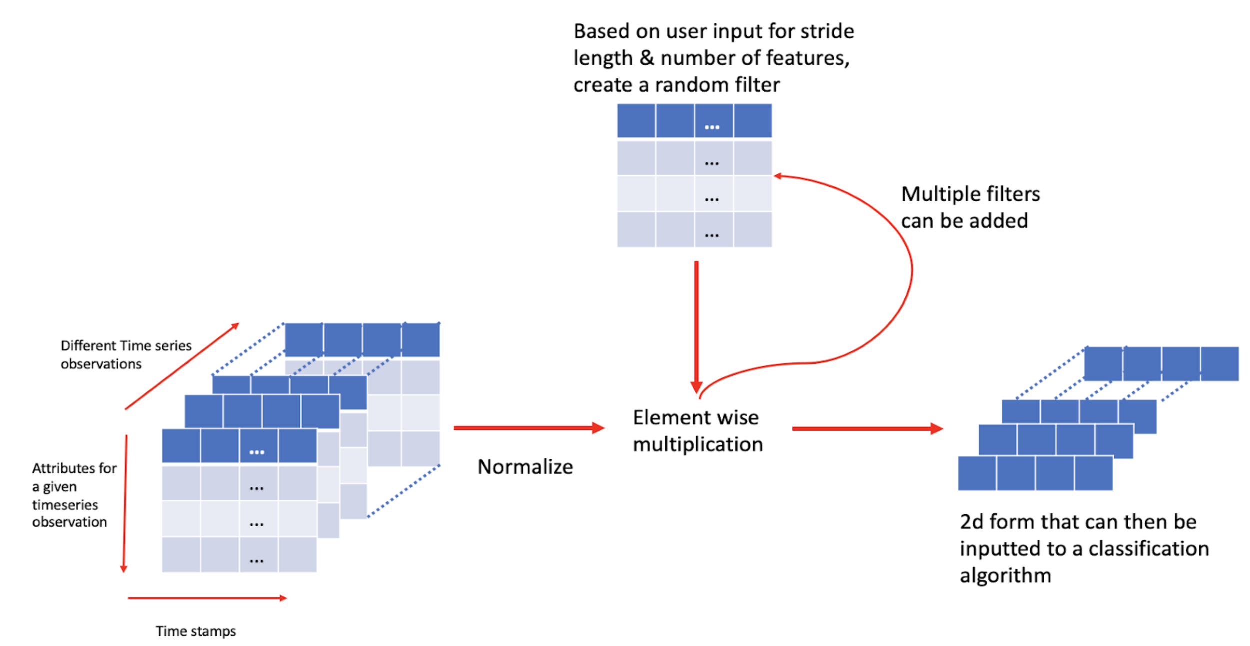
Using the features extracted from TsFeatures, we then plot PCA and TSNE graphs to see similarities between the 7 datasets.



(a) Dimension reduced time series samples using PCA (b) Dimension reduced time series samples using TSNE (c) Individual dataset PCA reduced time series plots (d) Individual dataset TSNE reduced time series plots

METHODOLOGY

To test our hypothesis, we needed to create a testing framework to streamline the evaluation process. To do this, we created a python class that could easily ingest the data, transform it from a 3-dimensional time series data frame to a 2-dimensional form that can be inputted to a machine learning classification algorithm.



Flowchart of testing framework

EVALUATION

> In this project, we used 7 datasets for the evaluation process. (information displayed below). The Basic Motions dataset is from the pyts library while the rest is imported from the sktime library.

> All datasets are train-test-split upon import. We only did the re-split (50/50) for the Finger Movements dataset and the Handwriting dataset.

Dataset	Train Size	Test Size	Data Transformation
Basic Motions	(40, 60, 100)	(40, 60, 100)	6*100 -> 16
Word Recognition	(275, 9, 144)	(300, 9, 144)	9*144 -> 40
Atrial Fibrillation	(15, 2, 640)	(15, 2, 640)	2*640 -> 40
Cricket	(108, 6, 1197)	(72, 6, 1197)	6*1197 -> 200
Epilepsy	(137, 3, 206)	(138, 3, 206)	3*206 -> 16
Finger Movements	(208, 28, 50)	(208, 28, 50)	28*50 -> 16
Handwriting	(500, 3, 152)	(500, 3, 152)	3*152 -> 10

> The next section displays the evaluation results. We used both accuracies and AUC scores in our evaluation.

		Basic Motions	Articular Word Recognition	Atrial Fibrillation	Cricket	Epilepsy	Finger Movements	Hand Writing
Logistic Regression	Accuracy	0.350	0.453	0.266	0.305	0.2681	0.486	0.034
	AUC Score	0.672	0.910	0.487	0.861	0.510	0.570	0.553
Support Vector Machine	Accuracy	0.525	0.690	0.333	0.653	0.377	0.572	0.082
	AUC Score	0.720	0.941	0.647	0.943	0.591	0.594	0.606
Random Forest	Accuracy	0.475	0.923	0.266	0.472	0.3405	0.495	0.08
	AUC Score	0.712	0.998	0.470	0.884	0.597	0.480	0.538
Gradient Boosting	Accuracy	0.400	0.563	0.266	0.361	0.268	0.509	0.125
	AUC Score	0.705	0.953	0.466	0.656	0.529	0.522	0.544

CONCLUSION

> From the testing results generated by the 7 datasets, Support Vector Machine performs much better than Logistic Regression, and Random Forest performs better than Gradient Boosting.

> Performance from best to worst (Dataset-wise): Articular Word Recognition > Cricket > Basic Motions > Epilepsy > Atrial Fibrillation > Finger Movements (binary classification) > Handwriting

> We observe relatively high AUC scores and similar performance on the models trained using the Cricket dataset and the Articular Word Recognition dataset. We also see similar performance on the Atrial Fibrillation dataset and the Epilepsy dataset. However, the performances are relatively worse than the previous two datasets.

> There are two datasets that display much lower accuracies – the Finger Movements dataset and the Handwriting dataset.

> Unlike the other datasets, the Finger Movements dataset has only two labels (Left and Right), thus making it a binary classification task, and we're seeing accuracies around 0.5 for Logistic Regression, Random Forest, and Gradient Boosting. Support Vector Machine performs slightly better results on this dataset.

> For the Handwriting dataset, all four models give low accuracies. However, we observe a gap between the accuracies and the AUC scores. Further research is needed to explain the big difference.