



ClaRe: Classification and Regression Tool for Multivariate Time Series

Ricardo Cachucho^{1,2}(✉), Stylianos Paraschiakos², Kaihua Liu¹, Benjamin van der Burgh¹, and Arno Knobbe¹

¹ Leiden Institute of Advanced Computer Science, Leiden, The Netherlands
{r.cachucho,b.van.der.burgh,a.j.knobbe}@liacs.leidenuniv.nl,
lkaihua@gmail.com

² Leiden University Medical Center, Leiden, The Netherlands
s.paraschiakos@lumc.nl

Abstract. As sensing and monitoring technology becomes more and more common, multiple scientific domains have to deal with big multivariate time series data. Whether one is in the field of finance, life science and health, engineering, sports or child psychology, being able to analyze and model multivariate time series has become of high importance. As a result, there is an increased interest in multivariate time series data methodologies, to which the data mining and machine learning communities respond with a vast literature on new time series methods.

However, there is a major challenge that is commonly overlooked; most of the broad audience of end users lack the knowledge on how to implement and use such methods. To bridge the gap between users and multivariate time series methods, we introduce the ClaRe dashboard. This open source web-based tool, provides to a broad audience a new intuitive data mining methodology for regression and classification tasks over time series. Code related to this paper is available at: <https://github.com/parastelios/Accordion-Dashboard>.

1 Introduction

Over the past few years, there is an increased interest in the analysis of multivariate time series data. A great deal of this interest is motivated by advances in sensor technology. In many application areas, deploying sensors for continuous monitoring has become a common strategy. Over the last 10 years, sensors are becoming more accurate, with better data communication protocols, smaller and last but not least, cheaper.

From the data science perspective, sensor systems will produce time series data. In the case of sensor networks, multiple variables are collected simultaneously, producing multivariate time series. Adding to that, when collected continuously, these datasets lead to big data challenges. This raised challenges to the data mining community, on how to deal with large multivariate time series. These challenges have attracted the attention of many researcher and lead to a vast literature on time series mining. With the exception of a few good examples

[1, 2], there is still a gap between most of these methods and the potential end users, who may lack a technical background to implement them.

Most of the sciences based on empirical observations have the potential to benefit from technological advances in sensor systems: (1) Children can be monitored continuously to study their social competence; (2) Environmental sciences can benefit from continuous sensing; Civil engineering can develop predictive maintenance of infrastructures using sensor networks; Life sciences and health are already heavily supported by machinery that uses sensors to measure all sort of phenomena. A common link between all the examples mentioned above is that they rely on sensor monitoring systems for their continuous sampling methodologies. The continuous nature of the measurements, lead to large multivariate time series datasets. As a consequence, the traditional data analysis tools based on classical statistics are commonly not applicable to this kind of data. New tools are an opportunity to bridge between data science and empirical sciences.

One could argue that the data mining community is already encouraging the publication of source code and data associated with publications. However, without a deep knowledge on the published method and the language used to implement the code, such released source code targets only a limited audience. Another very significant effort to make machine learning methods more accessible is the release of packages with collections of algorithms, such as Scikit-learn [3] for Python or Caret [4] for R. The downside of such packages is the need to be proficient both in the programming language that implements the package of methods and the need to know how to build a data science methodology around the chosen method. At last, there are tools for a broad audience such as Weka [1], MOA [2], Knime [5], JMulTi [6] and SPSS [7], which are intuitive and provide graphical user interfaces. These tools lack on the flexibility to implement new methods and most of them are not designed to analyze multivariate time series.

Our proposal to bridge the gap between new methods and a broad audience, is to build easily accessible web-based tools, with a user interface. we propose *ClaRe*, a *Classification and Regression* tool to model supervised multivariate time series. This Software as a Service (SaaS) tool adopts the *Accordion* algorithm from the previous chapter, to learn informative features and allows users to learn regression and classification models from multivariate time series with mixed sampling rates. Its intuitive web-based interface provides options of importing, pre-processing, modeling and evaluating multivariate time series data. In every step, plotting and saving data or results are allowed. Furthermore, source code, experimental data¹ and video tutorial² are available.

2 Tool Overview

ClaRe is a web-based tool that incorporates all the necessary steps for modeling time series with mixed sampling rates. Such time series are often collected from a network of sensors that measures complex phenomena. The output of such

¹ <https://github.com/parastelios/Accordion-Dashboard>.

² <https://www.youtube.com/watch?v=Vomhr9mBSBU>.

sensors are often multiple files that have variables measured at different rates and thus have special needs: (1) Pre-processing needs to include synchronization and merging; (2) Plotting needs to be done using sampling techniques due to the size of such time series; (3) Learning strategies that take into account the temporal nature of the data; (4) Adequate model evaluation strategies that test multiple systems (e.g. people) to reflect the true accuracy of the models.

From a technical perspective, ClaRe also presents benefits in terms of development and deployment. Both front end and server are developed with R, using the *R Shiny* package. This package provides a framework to interact between client and server side through R-scripts. As a result, the tool was easy to implement since only one programming language is used to manage both server and front end. From the deployment perspective, ClaRe's main advantage is its compatibility with all modern web browsers. With ClaRe, one can import and pre-process time series data, build regression or classification models, evaluate them, and export the results. The user can follow the proposed methodology intuitively, using web components that adjust to the user choices and guides the user throughout the data mining methodology. Each panel will be enumerated and explained below, following the CRISP-DM methodology [9].

Import and Pre-processing: When the user accesses the tool online, they are welcomed to the tool by the *Import* panel. To start, the user can upload predictors and target in a single or separate files. In this panel, the user can get a preview of the data and descriptive statistics for all the variables. Having imported the data, the user will be intuitively guided to the *Pre-processing* panel. Here, the user can choose from multiple pre-processing tasks, both generic for all sorts of datasets and specific to sensor-based multivariate time series. The pre-processing tasks include: (1) Selecting the variable the user wants to consider as a target; (2) Normalizing datasets; (3) Removing outliers; (4) Merge multiple files into one dataset, (5) Synchronize time series data with mixed sampling rates; (6) Manage missing values; (7) Plotting inspection as presented in Fig. 1.

Model: After choosing a numeric or nominal target, this panel changes into a regression or classification setup, respectively. The available regression models are a linear regression model and a lag regression model. As for the classification task, the available model is a decision tree. Both classification and regression models construct and select aggregate features using Accordion algorithm [8]. Accordion can be tuned with multiple parameters, which are available in the *Regression* or *Classification* panels. For both classification and regression, one can tune the target's sampling rate, the maximum window size and the number of samples used to perform a greedy search for aggregate features. Additionally, in regression there are multiple options for regression methods (linear and lag).

Evaluation: Having the models learned or loaded, the *Evaluation* panel allows the users to obtain multiple evaluations of the constructed model. For models testing multiple systems, one can use the Leave One Participant Out (LOPO) evaluation. With LOPO, the model is built multiple times, leaving each time one system out of the learning process to validate. This evaluation method is

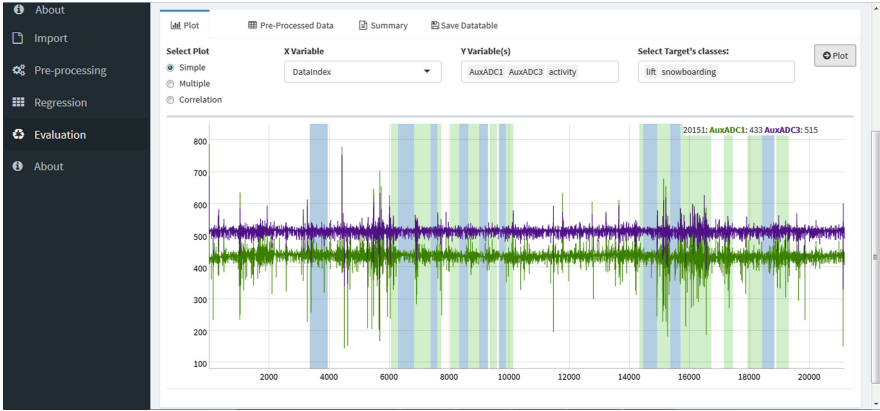


Fig. 1. *ClaRe* dashboard user interface: plotting tab.

especially important to assess the real accuracy of models, once the dataset instances are not independent and identically distributed. Other evaluation functionalities include: (1) statistical summary of the models accuracies/errors; (2) Visualization panel to plot models and predictions; (3) Export models and associated features; (4) Evaluate the model using new datasets; (5) Compare multiple models in new datasets; (6) Flexibility to re-visit these results over multiple user sessions.

3 Conclusion

This paper presents an easily accessible web-tool designated as *ClaRe*. *ClaRe* is a Software as a service (SaaS), which provides any user interested in mining multivariate time series, a methodology for supervised learning. More specifically, it allows users to deal with cases when the multivariate time series data have mixed sampling rates. Making use of intuitive menus, one can easily load one of multiple files, pre-process properly sensor systems data, learn time series models and evaluate the results. At any stage of the mining process, interactive plotting and saving options (for models and data) are available.

References

1. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. In: SIGKDD Explorations, vol. 11 (2009)
2. Bifet, A., Holmes, G., Kirkby, R., Pfahringer, B.: MOA: massive online analysis. *JMLR* **11**, 1601–1604 (2010)
3. Pedregosa, F., et al.: Scikit-learn: machine learning in python. *JMLR* **12**, 2825–2830 (2011)
4. Kuhn, M.: Caret package. *J. Stat. Softw.* **28**, 1–26 (2008)

5. Berthold, M., et al.: KNIME - the Konstanz information miner: version 2.0 and beyond. *ACM SIGKDD Explor. Newslett.* **11**, 26–31 (2009)
6. Krätzig, M.: The Software JMulTi, *Applied Time Series Econometrics* (2004)
7. BM Corp. Released 2017, IBM SPSS Statistics, Version 24.0, IBM Corp, New York
8. Cachucho, R., Meeng, M., Vespier, U., Nijssen, S., Knobbe, A.: Mining multivariate time series with mixed sampling rates. In: *Proceedings of ACM UbiComp*, pp. 413–423 (2014)
9. Wirth, R., Hipp, J.: CRISP-DM: Towards a standard process model for data mining. In: *Proceedings of PADD*, pp. 29–39 (2000)