Developing a Data Analytics Toolbox to Support CPS-based Services

Massimiliano Zanin*, Ernestina Menasalvas Ruiz* †, Alejandro Rodríguez-González * †,
Christian Wolff‡, Juana Wendt§, Elisa A. Herrmann¶, Pavel Smrz‖
*Centro de Tecnología Biomédica, Universidad Politécnica de Madrid, Madrid, Spain
Email: [massimiliano.zanin, ernestina.menasalvas, alejandro.rg]@upm.es
†Escuela Técnica Superior de Ingenieros Informáticos, Universidad Politécnica de Madrid, Madrid, Spain
‡Institute for Applied Systems Technology Bremen GmbH, Bremen, Germany
Email: wolff@atb-bremen.de
§Volkswagen AG, Wolfsburg, Germany
Email: extern.juana.wendt1@volkswagen.de
¶ARI Atos IT Solutions and Services Iberia, Madrid, Spain
Email: elisa.herrmann@atos.net
‖Brno University of Technology, Brno, Czech Republic
Email: smrz@fit.vutbr.cz

***Abstract*—The fast growth of Cyber-Physical Systems (CPSs) has brought us new opportunities to benefit from ever-increasing quantities of data describing our environment and behaviours. These data have a strong potential to become the basis of novel innovative services and products. However, the nature of CPS data streams makes it challenging to apply known data analytics methods and tools in an efficient way. This contribution discusses these challenges and shows how they could be tackled. Specifically, we present the initial development of a Data Analytics Toolbox designed to deal with some of them, like the streaming nature of the information they provide, and the need for efficient filtering techniques. As a case study, we further describe an application of the toolbox based on a real business case, aimed at improving high resolution weather forecast models.**

I. INTRODUCTION

The current interconnected society can be characterised by the growing popularity and relevance of cyber-physical systems (CPSs) [1], [2]. Everyday objects become unexpected sources of information about our environment and behaviours. New opportunities but also challenges emerge from the large amounts of data coming from heterogeneous sources and contexts. To illustrate the quantities, let us consider an average modern car that processes 4,000 signals at different time resolutions (even with no advanced driver-assistance sensors), translating to more than 500 MB of data per hour [3]. Similarly, modern home automation systems involve hundreds of sensors per building [4]. The potential of such data is enormous: they can foster a completely new market of innovative Cyber Physical Products (CPPs), i.e. products and services that have the features of CPSs at their core.

The promises of new business opportunities yielded by CPSs / CPPs also bring several new challenges. First of all, it is not simple to combine streams of data coming from different systems / products, as they may have different quality, may be a subject of various commercial limitations, and may have been provided by proprietary vendor-specific implementations, thus requiring ad-hoc interfaces. Beyond that, CPS data always need to be associated with confidentiality, privacy, security, IPR and ethical aspects for which the scientific community has yet no general answers.

Beyond the previously described challenges, an additional one has received much less attention from the scientific community: the problem of analysing such large volume of data. As it is well-known in machine learning, having data is not tantamount to gain knowledge, as the former have to be transformed into the latter. While a plethora of different data mining algorithms and models have been developed in the last decades, there are several requirements that are specific to CPSs. First of all, while data may be available in large volume, only a fraction of them may be relevant for a given application, thus efficient filtering strategies are needed to reduce the computational cost of any analysis. Along the same line, the streaming nature of CPS data implies that the evolution of a value through time can be more relevant than the value itself, thus imposing a dynamical view to data. Thirdly, data coming from CPSs are intrinsically unreliable and seldom dependable [5], implying the need of algorithms robust against noise and missing information. And, finally, data from different sources have heterogenous resolutions and references, complicating the process of synchronising the analysis across different CPSs.

In this contribution we describe the initial development of a Data Analytics Toolbox designed to fulfil these requirements.

978-1-7281-6949-1/20/$31.00 ©2020 IEEE

It has been developed within the umbrella of *Cross-CPP*, a European H2020 project devoted to the construction of a framework for supporting the development of services based on CPS data [6]. The Analytics Toolbox is based on a modular structure, in which new analytics services can be added to fulfil new user requirements; and it is aimed at supporting both fast prototyping of new ideas, and efficient implementation of data synthesis and analysis techniques. The remainder of the contribution is organised as follows. Firstly Sec. II discusses the need for, and the advantages associated to an integrated analytics toolbox, as opposed to the use of out-of-the-box solutions. Afterwards, Sec. III describes the global Cross-CPP solution, how the Analytics Toolbox interface with it, and the toolbox internal structure. Sec. IV further presents a use case, based on a real application that is being developed by a CPP user, involving the improvement of weather prediction models of very high spatial resolution. Finally, Sec. V concludes with a discussion of the main learnt lessons and steps ahead.

II. WHY A DEDICATED ANALYTICS TOOLBOX?

As a result of the recent surge of interest in data analysis, numerous analytics solutions can be found both in the market and the literature, from low-level libraries for machine learning (e.g. *scikit-learn* [7] or *TensorFlow* [8]), to out-of-the-box and integrated software solutions (e.g. Amazon's *SageMaker* [9] or Microsoft's *Azure* [10]). One may then ask what is the advantage, or even the need, of providing a custom data analytics toolbox inside a CPS system, as other options are already available to the user. We here argue that the availability of a tailored analytics toolbox has three main advantages.

First of all, it can provide ways for fast prototyping. While users may be expert on specific business aspects, they may not have in-house expertise on data analysis, nor the resources to deploy an analytics solution. The toolbox would then solve the initial problem, by allowing a first feasibility evaluation of a business concept at essentially no cost.

Secondly, a toolbox can provide access to advanced algorithms that are not standard in general purpose solutions, for being specifically tailored to CPS data; and that may not be part of the usual expertise of a data analyst. To illustrate, our Data Analytics Toolbox includes a module for network-based data analysis, as will be described in Sec. III-D. While network theory has attracted much attention in the academic world [11], [12], its adoption in real business applications is still far from being widespread.

Finally, using external data analytics tools requires downloading all relevant data; on the contrary, an integrated toolbox could filter the data prior to analysis, or even provide synthetic views of the same, for instance through averaging, clustering, or event-driven triggers. An example of this will be described in Sec. IV.

III. DATA ANALYTICS INSIDE CROSS-CPP

The objective of the Cross-CPP project is to establish an IT environment for the integration and analytics of data streams coming from high volume products with cyber physical features. The envisioned solution is based on the information flow depicted in Fig. 1 - see also [6] for further details. Data, coming both from cars and buildings, are pre-processed and stored in the CPP Cloud Storage; the CPP Big Data Marketplace is then in charge of keeping an index of the stored data, and of controlling the access. Finally, a set of service providers (i.e. the end users of the system) can use the Marketplace to access the data or to execute analytics over them.

In line with this view, the toolbox has been designed with a modular structure in mind, in which five modules, each one devoted to a specific analysis, communicate with the Cross-CPP Marketplace to get the data and yield back results. The modules, described in detail below, only share the communication interface with the Marketplace, i.e. they all expose a REST API interface; and are accessible though the same web interface in the Marketplace - see Fig. 2 for screenshots. Yet, beside these commonalities, they are completely independent - e.g. they have been developed in different programming languages, and are deployed in different physical servers.

In what follows, the five modules of the Analytics Toolbox are briefly described.

A. Basic statistics

There are many situations in which the service provider may not want to analyse a full set of data, but only have a simple overview of it; for instance, a given service may only be provided in cold weather, thus a first step may require checking the average ambient temperature. This module aims at providing with some very simple statistical functions, calculated over a subset of the stored data, and with the objective of minimising communication overheads. Several metrics are included, like:

- metrics of location and dispersion, i.e. average, median and standard deviation;
- maximum and minimum;
- metrics of shape, including skewness and kurtosis;
- for distributions, including temporal and spatial ones, their Shannon entropy.

B. Time series

In the analysis of many real-world systems, knowing when one specific parameter or characteristic changes can be as important as estimating its exact value. Several are the reasons behind this.

On one hand, the existence of changes is related to the stationarity of the system. If one wants to simulate (or forecast) a system through a model, e.g. constructed through a data mining approach, such model is only valid as long as the main properties of the underlying system do not change. To illustrate, if one builds a model to forecast the time needed by a

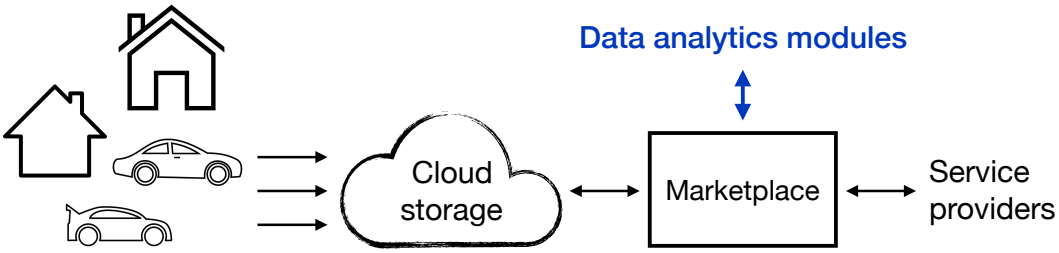

Fig. 1: Basic data flow inside Cross-CPP - see main text for details.

Fig. 2: Screenshots of the Marketplace web interface for accessing the analytics modules. (Top) Selection of the analytics module. (Bottom) Selection of the time series analysis to be executed.

driver to reach a destination in the summer, with ideal weather conditions, such model may be completely wrong when the winter arrives, and with it several centimetres of snow. Any real-world analysis executed in a changing environment must therefore be aware of its non-stationarity, i.e. when and how its main characteristics change.

On the other hand, and beyond such theoretical reason, knowing when a time series suffers a sudden change can also buttress two specific applications. Firstly, time series segmentation, i.e. transforming a unique signal into a set of sub-time series, each one of them of guaranteed stationarity. This can then be used to create simpler models of each time window. For instance, the full time series can be represented as a set of piecewise linear segments, which can support the definition of simpler distance metrics and of time series clustering algorithms [13]. Secondly, when the system is expected by construction to be stationary, sudden changes in the time series it produces can be used to detect malfunctions and abnormal conditions. To illustrate, a sudden change in the temperature of a car engine designed to work at 90°C can indicate a failure in the cooling system. For some examples on the use of time series segmentation in fault analysis, the reader may refer to [14], [15].

This module thus provides a set of tools for detecting when a time series, representing the evolution of a measurement, experiences a sudden change - see Fig. 2 bottom panel for a screenshot of this module's web interface. Note that the definition of such changes, also known as *drifts* [16], strongly depends on the considered application; to illustrate, one may be interested in a change in the average value, in the standard deviation, or even in the presence of correlations between two or more time series. Several methods for detecting drifts are thus implemented, including:

- Models based on statistical properties of the time series, including significant changes in the average or in the whole probability distribution of the time series.
- Models based on dynamical properties of the time series, including changes in entropy [17] or irreversibility [18].
- Models based on assessing the structure of relationships between multiple time series. This can include, for instance, the assessment of the correlation or causality between pairs of time series, and of their evolution through time [19].
- Models based on predictive machine learning algorithms. These models are based on training a forecasting algorithm using historical data, for then comparing the actual observed value with the forecast of the model [20]; the higher the error, the more distant is the system from stationarity.

C. Trajectory analysis

The concept of "trajectory analysis" is a very general one, encompassing many different analyses on data that encode

a spatio-temporal evolution. Most of the CPP elements are expected to move, at some point of their life. Additionally, it has to be noted that analysing a trajectory is not equivalent to analysing a multivariate time series, as the different components (e.g. longitude and latitude) are inherently linked. With these concepts in mind, this library aims at making available a set of basic tools simplifying the handling and manipulation of this mathematical object. The following functions are available to the user:

- simple statistics, like the calculation of velocities and accelerations;
- interpolation of trajectories, i.e. creating a new trajectory with a higher (and constant) time resolution, using the available data to derive the position when not known [21];
- error detection, i.e. the detection of reported positions that do not comply with physical limitations (e.g. maximum velocity) or with the past dynamics of the CPP [22];
- multivariate statistics, including the calculation of the average trajectory from a set of trajectories;
- trajectories clustering, i.e. the identification of sub-groups of trajectories with high similarity [23];
- group interpolation, understood as the interpolation of one trajectory by taking into account the dynamics of other (related) objects; and
- interactions, that is the identification of relationships between pairs of trajectories - e.g. minimum and maximum distance reached by two CPPs.

D. Network analysis

Sensors belonging to cyber-physical systems are organised in complex interaction structures, whose understanding and analysis can be of relevance. To illustrate, let us consider the case of a service provider that is interested in getting information from a set of electrical consumption sensors in a building, to understand how much energy will be available for e.g. charging electric cars. In such scenario, it is to be expected that several sensors will describe rooms of similar characteristics, and thus that the corresponding consumptions will be highly correlated. In a similar fashion, it may be expected that some sensors will "lead" others, i.e. that the information of the former precedes that of the latter - as for instance can be the case of room temperature at sunset, in which east-facing rooms will cool faster and west-facing ones.

In all these situations, the elements composing the system (being them sensors, cars, or any other elements of interest) can be represented as nodes of a complex networks [11], [12], [24], which are pairwise connected when a relevant relationship is detected between them. The resulting structure can then be used as input of a data mining task, as discussed in [25]. While complex networks theory has received a growing attention from the scientific community in the last decade [26], its application to real-world problems has been rather limited, among other reasons because of its mathematical complexity and high computational cost. The network analysis module is thus a user-friendly solution to include network-based analyses in the Cross-CPP system.

Network analysis is usually a computationally intensive task, as the cost of the calculation of many topological metrics (i.e. measures describing a particular aspect of the structure) scales as the square or the cube of the number of nodes. On the other hand, many networks may evolve slowly in time - for instance, the structure between sensors in a building may essentially be static. This module is thus designed to help the analysis of evolving networks, by identifying those elements that have not changed since the last iteration, and by avoiding redundant calculations. Some topological metrics included are: the *efficiency* of the network, defined as the average of the inverse of the geodesic distance between all pairs of nodes [27]; *transitivity*, defined as the average number of triangles in the network, over the total number of possible triangles [28]; *centrality* of nodes, i.e. measures describing how important these are inside the network [29]; and *modularity*, i.e. the best division into disjoint communities, where a community is usually defined as set of nodes strongly connected between them, but loosely connected with the remainder of the network [30].

E. Machine learning

As a last module, we include the possibility of training and evaluating machine learning models, able to predict the class of a future event (classification) or a magnitude of a variable (regression) [31]. It is worth noting that the design of the best machine learning model is strongly dependent on the considered application - i.e. it is almost impossible to find a *one-fits-all* solution. Still, the aim of this module is aligned with the general philosophy of the toolbox: providing a first solution for a fast prototyping, which can then be optimised according to the requirements of the problem. The service provider can then test an initial idea using the provided models; and, if results are promising, in-house resources can be committed to create a tailored solution.

This module supports incremental learning algorithms by means of existing libraries and frameworks that proved to be applicable in high velocity settings, as, for instance, the Yahoo/Microsoft *Vowpal Wabbit* [32]. Selected methods are interconnected with push data subscription and change notification channels, to enable incremental improvement/building of a model and its direct use in the same time. The machine learning module thus learns and predicts at the same time, solving the "cold start" problem [33]. The clear distinction between the phases of model training and of its use, typical for the batch machine learning applications, is not relevant for the case of incremental training from data streams. A streaming model would typically be built for a prediction task in which the correct value would form a part of the stream data with a defined time delay.

The implementation builds on open-source machine learning libraries (scikit-learn [7], PyTorch and Google Tensorflow [8]),

well-established industrial standards that are used by many large companies every day.

IV. USE CASE: MONITORING RAIN IN A CITY

As an example of application, we here consider a real scenario identified by the partners composing the Cross-CPP consortium. Suppose a service provider interested in the market of high resolution meteorological forecasts, e.g. in the development of a model able to predict the presence of rain with a city block precision. Achieving such resolution requires the execution of meteorological models on input data which should have the same, or even higher, precision; thus the classical approach of gathering data from a few meteorological stations per city would not be feasible. The CPP framework would make this possible: cars driving in the city could send weather information, as for instance the presence of rain through their selected wipers level. While the service provider could simply download these data (i.e. all wipers activities) in real time, this entails two problems. On one hand, in the case of large cities with tens of thousands of cars sending information every second, the amount of information to be handled would be substantial, with most of it being irrelevant. On the other hand, wipers information is expected to be noisy, as a driver may for instance activate them just to clean the windshield; it is thus necessary to discriminate such random activations from collective and coherent ones. As will be described below, this scenario is a perfect case in point of the usefulness of an integrated analytics toolbox.

Due to the work-in-progress status of the Cross-CPP infrastructure, real data from cars were not available when writing this contribution; they have instead been simulated, through a set of time series representing the wipers activation in two large groups of cars. All computations, on the other hand, have been performed through the system's libraries. The starting point is that cars usually activate their wipers when it is raining, but they might do it even in the absence of rain; furthermore, they are activated with varying speed (level). We finally suppose that all cars in a group are in the same part of the city, thus their activation will be synchronised; yet, the two groups find rain at different moments. As a naïve solution to the problem, the service provider may download all time series. These are depicted in Fig. 3, in which the two top panels represent the evolution of five time series (i.e. the evolution of wipers velocity for five cars) for each group. More efficient solutions can nevertheless be devised.

First of all, instead of the full set of data, the service provider may request the average of the wipers activation from all cars. This is represented by the red lines in the top two panels of Fig. 3. Note that the averaged data can previously be filtered, for instance to only include cars driving in a specific part of the city.

Secondly, it is worth noting that the presence of two groups of cars, located in different regions of the city, is not known beforehand. The analytics toolbox provides ways of highlighting such structure. To illustrate, the pairwise correlation between the time series coming from cars can be calculated, and the result depicted as a network. Whenever a high correlation between pairs of cars is detected, this will map into a strong link between the two corresponding nodes; the presence of communities can then be evaluated, e.g. through a simple graphical representation. When this is performed on our synthetic data set using the functions of the time series module (specifically, a Spearman's rank correlation), the result is what depicted in the central panel of Fig. 3.

Finally, the service provider can ask the system for a statistical measure alerting about the presence of changes. The blue line in the bottom panel of Fig. 3 depicts the evolution of the Permutation Entropy, a well-known metric able to distinguish between stochastic (i.e. random) and deterministic dynamics [34], [35]. This entropy has been calculated over the average of all time series composing the first group. As it can be appreciated, values of the metric close to 1 indicate a random dynamics, i.e. that wipers are activated and stopped without any clear trend - to simplify the interpretation, the grey band indicate the 68% confidence band observed in random data. On the other hand, the metric presents three clear minima, which correspond to the time in which the rain started for the first group; this is because, as all drivers are activating the wipers at almost the same time, the evolution of the time series is no longer stochastic, but instead follows a clear and deterministic trend. The service provider could then monitor this metric, and only download the underlying data when a change is detected.

V. CONCLUSION

In this contribution we have presented the initial development of an Analytics Toolbox for the analysis of CPS data. It includes both a general architecture for the integration of different modules, and five specific analysis tools, covering the most important needs associated with the processing of car and building information.

From the experience gained in the development of this Analytics Toolbox, one important lesson has to be highlighted: the design of the different modules has to reach a fine balance between generality and specificity. On one hand, the provided functions ought to be general enough, i.e. should be of relevance for the largest possible pool of services and users. On the other hand, too general functions may become useless, as each service has specific needs. This latter case is well represented by classification models, which have to be tailored to the input data they are going to process. Such balance can be achieved in different ways.

First of all, the analytics toolbox should offer different algorithms and techniques to reach the same analytics goal. To illustrate, the time series module provides different metrics for detecting drifts, e.g. based on statistical and information science concepts. It is then the responsibility of the user to select the one most suitable for the problem at hand. On the other hand, this requires the provision of an exhaustive documentation, as

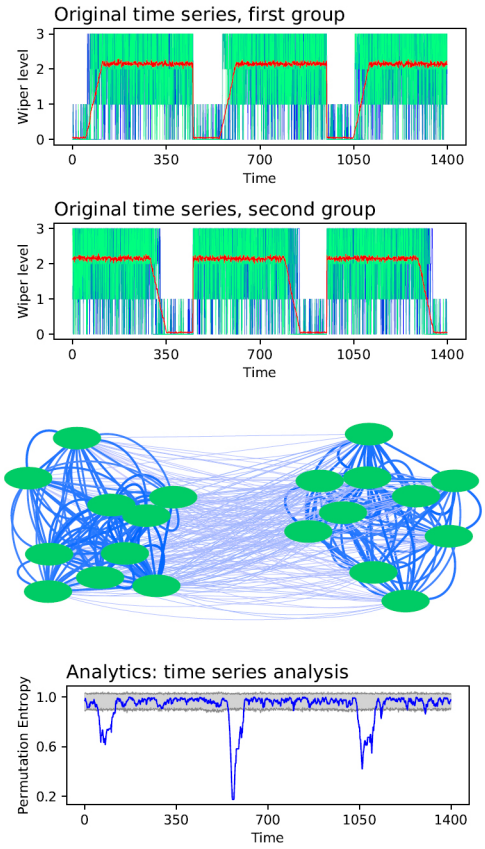

Fig. 3: Example of the analysis of wiper data. The top two graphs depict the temporal evolution of the wiper level of two groups of five cars; the central one the network representation of the correlation between the ten time series; finally, the bottom one the evolution of the Permutation Entropy for the first group. See Sec. IV for details.

users may not be familiar with the idiosyncrasies of each one of these techniques.

Secondly, flexibility can be increased by providing ways of concatenating different modules together, in order to exponentially increase the array of available analyses. In the initial implementation here presented, the average of a set of time series is itself a time series, which can be analysed by the corresponding module.

Finally, it is important to design a flexible and modular environment since the beginning of the development, in which new modules can seamlessly be introduced according to the user needs.

In line with this lesson, future work will be devoted to extend the array of analytics tools offered within the system; and to provide easier and richer ways of merging different modules together.

ACKNOWLEDGMENT

This paper presents work developed in the scope of the project Cross-CPP. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 780167. The content of this paper does not reflect the official opinion of the European Union. Responsibility for the information and views expressed in this paper lies entirely with the authors.

REFERENCES

[1] W. H. Wolf, "Cyber-physical systems." *IEEE Computer*, vol. 42, no. 3, pp. 88–89, 2009.

[2] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: the next computing revolution," in *Design Automation Conference*. IEEE, 2010, pp. 731–736.

[3] D. Work, A. Bayen, and Q. Jacobson, "Automotive cyber physical systems in the context of human mobility," in *National Workshop on high-confidence automotive cyber-physical systems*, 2008, pp. 3–4.

[4] L. Gurgen, O. Gunalp, Y. Benazzouz, and M. Gallissot, "Self-aware cyber-physical systems and applications in smart buildings and cities," in *2013 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2013, pp. 1149–1154.

[5] L. Miclea and T. Sanislav, "About dependability in cyber-physical systems," in *2011 9th East-West Design & Test Symposium (EWDTS)*. IEEE, 2011, pp. 17–21.

[6] A. Correia, C. Wolff, M. Zanin, E. Menasalvas, E. A. Herrmann, V. Corral, M. Kachelmann, R. DeLong, and P. Smrz, "Cross-cpp - an ecosystem for provisioning, consolidating, and analysing big data from cyber-physical products," in *Proceedings of the 1st Workshop on Cyber-Physical Social Systems co-located with the 9th International Conference on the Internet of Things (IoT 2019)*, 2019.

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[8] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.

[9] A. V. Joshi, "Amazon's machine learning toolkit: Sagemaker," in *Machine Learning and Artificial Intelligence*. Springer, 2020, pp. 233–243.

[10] R. Barga, V. Fontama, W. H. Tok, and L. Cabrera-Cordon, *Predictive analytics with Microsoft Azure machine learning*. Springer, 2015.

[11] S. H. Strogatz, "Exploring complex networks," *nature*, vol. 410, no. 6825, p. 268, 2001.

[12] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, "Complex networks: Structure and dynamics," *Physics reports*, vol. 424, no. 4-5, pp. 175–308, 2006.

[13] E. Keogh, S. Chu, D. Hart, and M. Pazzani, "Segmenting time series: A survey and novel approach," in *Data mining in time series databases*. World Scientific, 2004, pp. 1–21.

[14] L. Martí, N. Sanchez-Pi, J. Molina, and A. Garcia, "Anomaly detection based on sensor data in petroleum industry applications," *Sensors*, vol. 15, no. 2, pp. 2774–2797, 2015.

[15] A. Nikolaou, P. A. Gutiérrez, A. Durán, I. Dicaire, F. Fernández-Navarro, and C. Hervás-Martínez, "Detection of early warning signals in paleoclimate data using a genetic time series segmentation algorithm," *Climate Dynamics*, vol. 44, no. 7-8, pp. 1919–1933, 2015.

[16] J. Gama, *Knowledge discovery from data streams*. Chapman and Hall/CRC, 2010.

[17] H. Azami and J. Escudero, "Amplitude-aware permutation entropy: Illustration in spike detection and signal segmentation," *Computer methods and programs in biomedicine*, vol. 128, pp. 40–51, 2016.

[18] M. Zanin, A. Rodríguez-González, E. Menasalvas Ruiz, and D. Papo, "Assessing time series reversibility through permutation patterns," *Entropy*, vol. 20, no. 9, p. 665, 2018.

[19] X. Xuan and K. Murphy, "Modeling changing dependency structure in multivariate time series," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 1055–1062.

[20] J. J. Valdés and G. Bonham-Carter, "Time dependent neural network models for detecting changes of state in complex processes: Applications in earth sciences and astronomy," *Neural Networks*, vol. 19, no. 2, pp. 196–207, 2006.

[21] C. Sun, C. Wu, D. Chu, L. Xie, L. Liu, and H. Li, "Vehicle trajectory restoration based on vondrak filtering and cubic spline interpolation," in *CICTP 2016*, 2016, pp. 235–248.

[22] R. Laxhammar and G. Falkman, "Online learning and sequential anomaly detection in trajectories," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 6, pp. 1158–1173, 2013.

[23] G. Yuan, P. Sun, J. Zhao, D. Li, and C. Wang, "A review of moving object trajectory clustering algorithms," *Artificial Intelligence Review*, vol. 47, no. 1, pp. 123–144, 2017.

[24] S. Boccaletti, G. Bianconi, R. Criado, C. I. Del Genio, J. Gómez-Gardenes, M. Romance, I. Sendina-Nadal, Z. Wang, and M. Zanin, "The structure and dynamics of multilayer networks," *Physics Reports*, vol. 544, no. 1, pp. 1–122, 2014.

[25] M. Zanin, D. Papo, P. A. Sousa, E. Menasalvas, A. Nicchi, E. Kubik, and S. Boccaletti, "Combining complex networks and data mining: why and how," *Physics Reports*, vol. 635, pp. 1–44, 2016.

[26] L. d. F. Costa, O. N. Oliveira Jr, G. Travieso, F. A. Rodrigues, P. R. Villas Boas, L. Antiqueira, M. P. Viana, and L. E. Correa Rocha, "Analyzing and modeling real-world phenomena with complex networks: a survey of applications," *Advances in Physics*, vol. 60, no. 3, pp. 329–412, 2011.

[27] V. Latora and M. Marchiori, "Efficient behavior of small-world networks," *Physical review letters*, vol. 87, no. 19, p. 198701, 2001.

[28] M. Á. Serrano and M. Boguna, "Clustering in complex networks. i. general formalism," *Physical Review E*, vol. 74, no. 5, p. 056114, 2006.

[29] E. Estrada, *The structure of complex networks: theory and applications*. Oxford University Press, 2012.

[30] M. E. Newman, "Modularity and community structure in networks," *Proceedings of the national academy of sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.

[31] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[32] J. Langford, L. Li, and A. Strehl, "Vowpal wabbit online learning project," 2007.

[33] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, "Methods and metrics for cold-start recommendations," in *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2002, pp. 253–260.

[34] C. Bandt and B. Pompe, "Permutation entropy: a natural complexity measure for time series," *Physical review letters*, vol. 88, no. 17, p. 174102, 2002.

[35] M. Zanin, L. Zunino, O. A. Rosso, and D. Papo, "Permutation entropy and its main biomedical and econophysics applications: a review," *Entropy*, vol. 14, no. 8, pp. 1553–1577, 2012.