TÉCNICO LISBOA

Autonomous Time Series Data Processing on Historical and Real-Time Settings

Ricardo Filipe Lourenço Nunes de Sousa

Thesis to obtain the Master of Science Degree in

Computer Science and Engineering

Supervisors: Prof. Maria da Conceição Esperança Amado
Prof. Rui Miguel Carrasqueiro Henriques

Examination Committee

Chairperson: Prof. José Carlos Martins Delgado
Supervisor: Prof. Rui Miguel Carrasqueiro Henriques
Member of the Committee: Prof. Susana de Almeida Mendes Vinga Martins

January 2021

Abstract

Heterogeneous sensor networks, including water distribution systems and traffic monitoring systems, produce abundant time series data with an arbitrarily-high multivariate order for monitoring network dynamics and detecting events of interest. Nevertheless, errors and failures in the calibration, data storage or acquisition can occur on some of the sensors installed in those systems, producing missing and/or anomalous values. This thesis proposes a computational system, for the fully autonomous cleaning of multivariate time series data using strict quality criteria assessed against ground truth extracted from the targeted series data, on historical and real time data contexts. The proposed methodology is parameter-free as it relies on robust principles for the assessment, hyperparameterization and selection of methods. This work supports an extensive set state-of-the-art methods for (multivariate) time series imputation and outlier detection-and-treatment, considering both point and segment/serial occurrences. A comprehensive evaluation of system is accomplished using heterogeneous sensors from two water distribution systems with varying sampling rates, water consumption patterns, and inconsistencies. Results confirm the relevance of the proposed autonomous processing approach, and its extensibility towards real-time settings under tigtht optimality guarantees.

Keywords

parameter-free learning, multivariate time series, missing values imputation, outlier detection, heterogeneous sensor networks, real-time data

Resumo

Redes de sensores heterogéneos, incluindo sistemas de distribuição de água e os seus sistemas de monitorização de tráfego, produzem dados de séries temporais abundantes com uma ordem multivariada arbitrariamente alta para monitorizar a dinâmica da rede e detectar eventos de interesse. No entanto, erros e falhas na calibração, armazenamento ou aquisição de dados podem ocorrer em alguns dos sensores instalados nestes sistemas, produzindo valores omissos e/ou anómalos. Esta tese propõe um sistema computacional, para a limpeza totalmente autónoma de dados de séries temporais multivariados usando critérios de qualidade rigorosos avaliados contra os valores reais extraídos dos dados da série alvo, em contextos de dados históricos e em tempo real. A metodologia proposta é livre de parâmetros por se basear em princípios robustos para avaliação, hiperparameterização e seleção de métodos. Este trabalho oferece suporte a um extenso conjunto de métodos do estado da arte para imputação de séries temporais (multivariadas) e detecção e tratamento de valores anómolos, considerando ocorrências pontuais e de segmento/sequência. Uma avaliação abrangente do sistema é realizada usando sensores heterogéneos de dois sistemas de distribuição de água com taxas de amostragem variadas, padrões de consumo de água e inconsistências. Os resultados confirmam a relevância da abordagem proposta de processamento autónomo e a sua extensibilidade para configurações em tempo real sob garantias de otimização.

Palavras Chave

aprendizagem livre de parâmetros, séries temporais multivariadas, imputação de valores omissos, deteção de valores anómalos, redes de sensores heterogéneos, dados em tempo real

Contents

List of Figures

List of Tables

1

Introduction

Contents

The placement of heterogeneous sensors within complex systems – whether physiological, mechanical, digital, geophysical, environmental or urban – offers the possibility to acquire comprehensive views of their behavior along time. Sensorized systems produce abundant time series data, used for monitoring purposes or the detection of events of interest. However, the placed sensors are susceptible to failures and errors associated with sensor calibration and data acquisition-transmission-storage Gill et al. [1], producing time series data with missing and anomalous values. In this context, time series data are generally subjected to initial processing stages to enhance their quality for the subsequent mining stages.

Processing time series data produced by networks of heterogeneous sensors is, nevertheless, a laborious process due to four major reasons. First, the selection and parameterization of the processing methods is highly dependent on the regularities of the target series data and challenged by the wide diversity of approaches currently available. Second, the profile of errors can be diversified, each leading to different processing choices. In this context, the type and amount of anomalies and missing values can largely affect decisions. Third, different types of sensors – such as water flow, pressure and water quality sensors in water distribution systems – may benefit from dissimilar processing methods. In fact, sensors of the same type but with singular calibrations, sampling rates, or positioning within the monitored system can as well benefit from different choices. Fourth and finally, different systems equipped with identical sensors do not necessarily benefit from the same processing options. Consider water distribution network (WDN) systems, water consumption patterns can highly vary between WDNs or along time, impacting decisions. Even more, different WDNs may be susceptible to unique externalities, affecting the profile of observed errors, such as the sampling rate, sensor distribution, and system's susceptibility of sensors can further affect processing choices.

In addition, time series data processing generally give sub-optimal results. First, cross-variable relationships in multivariate time series data are commonly disregarded. For instance, flow and pressure sensors in WDNs are generally correlated, and thus co-located or nearby sensors can guide the treatment of low-quality series data. Second and understandably, optimal decisions are challenged by the wide diversity of available processing approaches, multiplicity of sensors, and profile of errors observed per sensor.

1.1 Research contributions

This thesis proposes a methodology for the fully autonomous cleaning of multivariate time series that is able to address the introduced challenges. It is relevant in the historical data context, where the time series have been totally collected throughout the time and is already stored, and in the real-time data context, where new observations are continuously arriving into the system from the multiple sensors in

the network. The proposed methodology referred as AutoMTS (**Auto**nomous **M**ultivariate **T**ime **S**eries data processing), offers three major contributions. First, provides strict guarantees of optimality as it places robust processing decisions against ground truth extracted from the targeted series data. To this end, data series are automatically explored in order to detect conserved segments and identify the profile of observed errors, which are then planted in the conserved segments for the sound comparison of available processing choices.

Second, AutoMTS provides a comprehensive coverage of available processing options, currently providing over twenty state-of-the-art methods for missing imputation, outlier detection and gross-error removal from time series data. Particular attention was placed to guarantee the presence of state-of-the-art methods able to consider cross-variable dependencies in the presence of multivariate time series data. Also, we further guarantee the presence of methods able to deal with both point and segment/serial missing and outlier values.

Third, AutoMTS is parameter-free as it relies on robust principles to assess, hyperparameterize and select state-of-the-art processing methods.

In the real-time data context, AutoMTS requires to ensure that the processing of the incoming data is achieved in a period of time that is similar to the next new observation that will get into the system, in order to complete the task without delaying future observations processing.

To assess the significance of the proposed contributions, AutoMTS is extensively evaluated in two water distribution network systems with heterogeneous sensors, producing observations at varying sampling rates, and subjected to unique water consumption patterns and error profiles.

The gathered results confirm the relevance of the proposed AutoMTS methodology, highlighting that processing choices are highly specific to each sensor and thus guarantees of optimality can only be provided under comprehensive and robust assessments. Also, results further offer a thorough comparison of state-of-the-art imputation and outlier detection methods, assessing their ability to handle diverse error profiles in real-world series data with varying regularities.

AutoMTS is provided as both a graphical and programmatic tool satisfying strict usability criteria.

1.2 WISDOM project

This thesis is inserted in the WISDOM project, which is a collaboration between Instituto Superior Técnico (IST), Instituto Politécnico de Setúbal (IPS), IST-ID, INESC-ID and the water utilities Infraquinta, Câmara Municipal do Barreiro and Câmara Municipal de Beja.

Each of the water utilities provided time series representative of their telemetry systems. Infraquinta provided 62 time series, each one representing one sensor of their water flow network and another 24 for the water delivery points. All the series represent the year 2017. The time series representing the

network have more than 360.000 observations, with a shifting time granularity. They provided information about 62 sensors divided into: pressure meters, reservoir water level meters, energy analyzers, volumetric meters, water flow meters and chlorine meters. The time series representing the water delivery points have more than 800.000 observations, but no information about the time granularity. Câmara Municipal do Barreiro provided 26 time series about their water flow network, representative of the 2018 year. The time series has 8473 observations, with a regular time granularity of 1 measure per hour. They provided information about 14 sensors, where some are divided into maximum value, minimum value and mean value of the hour, making it the total of 26 time series. At last the Câmara Municipal de Beja provided 2 time series (on the water flow network), representing almost a 2 years period from 16/05/2017 to 24/04/2019. The time series has 207.949 observations, with a shifting time granularity. They provided information about two water pressure and water flow sensors from their network. About the water delivery points system, there are 206 time series representing a period from 1/10/2018 to 23/4/2019 with a regular time granularity of measurement per day.

The Figure 1.1 depicts the water flow and water pressure series from sensors located near the principal tanks in the Barreiro and Beja WDNs, for five illustrative days where it is possible to perceive the distinct sample granularity and specific patterns.

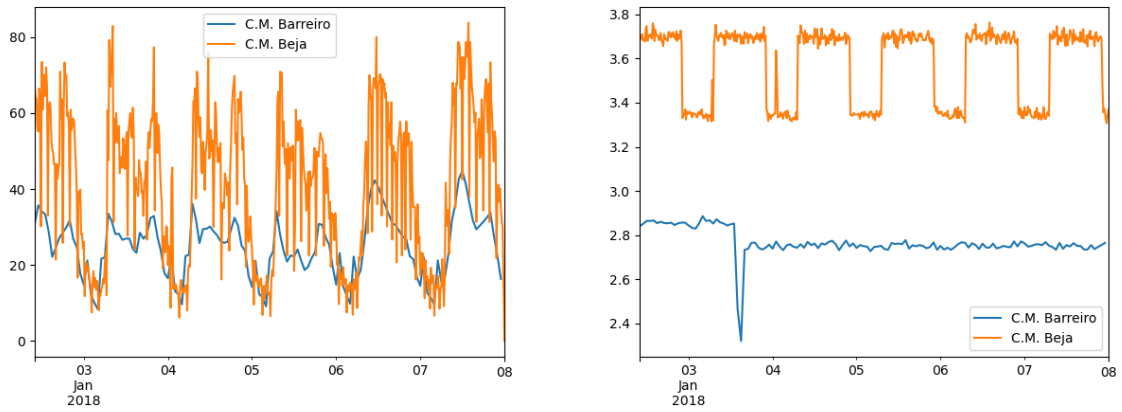

Figure 1.1: Sensor measurements over 5 illustrative days for both Barreiro and Beja WDNs.

This sensors are susceptible to errors and failures like calibration, data store or data acquisition, which produce faulty data.

1.3 Document Outline

This thesis is is organized as follows: Chapter 1 gives a short introduction about the main objectives and the project. Chapter 2 introduces essential concepts that form structure of this thesis. Chapter 3 gives

an overview of some contributions in the pre-processing of the outliers and missing values imputation, and respective state-of-the-art methodologies, for either historical and real-time contexts. Chapter 4 describes the proposed processing approach for the real-time and historical settings and which main decisions were taken. Chapter 5 gives a comprehensive evaluation of our methods using two real world heterogeneous networks and study cases. Finally Chapter 7, shows the concluding remarks and the future work plans.

2

Background

Contents

This chapter describe fundamental concepts referred along the report.

2.1 Water Distribution Networks

A Water Distribution Network (WDN) is a network of pumps, pipelines, storage tanks, sensors and other accessories, that are capable of delivering adequate quantities of water for the everyday needs. Usually each country is divided into multiple sub-systems called District Metered Area (DMA). According to Farley et al. [2], a DMA is a well defined and permanent boundary, where due to the installation of water flow (amount of water flowing per unit of time) and pressure (flow of water from the tap) meters at strategic points throughout the distribution system, each meter recording the flows into a discrete area. There are also other metrics measured in DMAs, e.g., water pH, chloride levels, etc. DMAs are useful to create boundaries which will help in the management of the water pressure and water flow, and in monitoring of each district enabling the presence of unreported bursts and leakages to be identified and locations. As stated before, the creation of most of the DMAs requires the installation of adequate equipment which is usually used to store information using a data logger, a device which records data over time, via external sensors using a telemetry system.

2.2 Time Series

A time series is a sequence of observations of the same random variable, repeated in time order (ordered chronologically), usually spaced at uniform time intervals. The importance of the use of time series is to understand the past and the present and try to predict the future, such as anticipation of events of interest, anomaly detection and motif discovery. Although the behavior of systems in these domains is dynamic in nature (e.g. continuous variation of water flow and pressure along time and space), the monitored data are discrete.

Time series can be either univariate or multivariate. The univariate time series occurs only one attribute is being measured over time and it follows,

$$(x_1, x_2, x_3, ..., x_t, ...) \tag{2.1}$$

where $x_\mathrm{t} \in \mathbb{R}$ and x_t is an observation on the time stamp t. The multivariate time series have multiple attributes that are being measured over time, and its notation is the same as the univariate but $\mathbf{x}_\mathrm{t} \in \mathbb{R}^\mathrm{m}$, where *m* is the multiple attributes being measured in the time stamp t.

Time series typically have an internal structure with, which have domain-specific meanings, e.g., an abrupt increase in the water flow is specific to water distribution systems. The structural elements of a time series are usually abstractions of the original data, and the understanding of the structures

are essential to a better understanding of the time series. Thus, it is necessary to choose a suitable representation which will ease and increase the efficiency of time series analysis. There are great amount of time series representations (numeric and symbolic) according to Lin et al. [3], e.g., Discrete Fourier Transform (DFT), Discrete Wavelet Transform (DWT), Piecewise Aggregate Approximation (PAA) and Symbolic Aggregate Approximation (SAX). These representations offer benefits such as reduction of dimensionality (amount of attributes), variability (how spread is the data) and cardinality (uniqueness of data values contained in one attribute).

Usually time series data can show several patterns and, in this context, may be convenient to decompose it. Five well-studied patterns or components are: Level also referred as **L** (or Horizontal), Trend referred as **T**, Seasonal variation as **S**, Cyclical variation known as **C** and Random as **R**. Level is the fluctuation of data around a constant mean. Trend happens when there is a long-term upward/downward in the data, which it is not always linear (can also be polynomial, exponential, logarithmic, etc). Seasonal variation happens when the time series is affected by a well defined/fixed periodic (yearly, monthly, weekly, daily) fluctuations on the data, e.g., an increase in the cases of the flu during the winter months. Cyclical variations happens when the time series is affected by a non-fixed fluctuations (usually with larger periods than the previous season period) on the data and may or not be periodic, e.g., every decade a country has a decrease on the gross domestic product for a non-fixed period of time. The Random component is usually called the noise of the time series which usually has short duration and it is non-regular, non-repeatable and unpredictable. Some possible causes for this component are usually measurement errors, anomalies or unnatural changes, e.g, error on the measurement on the water flow sensors.

According to Hyndman et al. [4] a time series is a composition of the components explained in the previous paragraph (where the Trend and Cyclical components can be combined into a single Trend-Cycle component). In this context, there are two main models of decomposition: the additive and the multiplicative. Generally an additive model,

$$x_t = T_t + S_t + C_t + R_t, \tag{2.2}$$

is more suitable for environmental data, while the multiplicative model,

$$x_t = T_t \times S_t \times C_t \times R_t, \tag{2.3}$$

is commonly better suited for economical data. The additive model is the most suitable if the variations around the trend, seasonal and cyclical components do not vary with the level of the time series. If the variation of the previous mentioned components appears to be proportional or tends to vary with the time series level, the most suitable model is the multiplicative. It is also possible to first make transfor-

mations on the data, until the level of the time series appears to be stable over time and then use the addictive decomposition as an alternative for the the multiplicative decomposition. The decomposition is an essential operation for the processing of time series, since the treatment is easier after the removal noise patterns (random level).

2.3 Historical and Real Time Data

In the past two decades, along with the improvements in the technology, the amount of data generated is increasing extremely fast. According to Chen et al. [5], the overall created and copied data volume in 2011 was 1.8ZBytes (10^{21}Bytes), which increased nearly nine times within five years. This data is also referred as big data which is mainly used to described large datasets obtained from multiple sources, e.g., sensor systems, social network or financial transactions. Since the size those datasets can reach could be to large, it is not possible to process the data with the traditional tools. This data have an immense value, which can be used in the different fields of the industry as analytical tools. The main challenge nowadays is to process this data and make a good use of it.

According to Mohamed and Al-Jaroodi [6], big data differs from regular data in three characteristics: volume, velocity and variety. The expanding use of the Internet, the extremely usage of sensors in infrastructures and the constant monitoring of everyday mechanisms, increases the volume of this huge data. Furthermore, it is important to manage the speed as the data is created and generated in a way that is possible to extract, process and integrate in the existing datasets. It is also necessary to integrate the data that are generated from different sources and in different formats.

Large scale data can be classified as historical and real time data. These data contexts differ regarding both the analysis and the time that is required to process them. Nonetheless, both the data contexts can be acquired in the same way.

2.3.1 Historical Data

All the data that was already acquired and stored can be called historical data. The data are usually collected over a period of time and then stored in a variety of ways, e.g., files, databases, records. Historical data is usually only processed after it is stored where it is possible to make the specific analysis. Both the processing and the analysis usually do not have high requirements on response time. Although we consider historical data as big data the speed characteristic it is not relevant since it does not matter how fast the data was generated, processed (if processed at all) and stored, we will only analyze it further.

2.3.2 Real Time Data

Data continuously arriving into a system, and that can be processed and analyzed before its storage is here referred as real time data. As the data flows into the system, both the processing and analysis have a schedule to be assigned and must be completed before their deadline. The processing and the analysis deadlines can be divided into tasks, which can be classified, according to Shin and Ramanathan [7], as hard, firm and soft. If a task is not completed in the deadline and it causes a catastrophic consequence in the system it is called hard. A deadline is firm if the results of the task cease to be useful when the deadline expires and the consequences of not meeting the requirements are not very severe. If the deadline is neither one of the above categories it is called soft, and the results produced by a task with a soft deadline decreases over time after the deadline expires. The deadlines are specific of the system and its needs.

The real time data is commonly more resourceful expensive in comparison with historical data, since could be necessary to allocate extra resources in order to complete essential tasks in specific deadlines. Moreover, these tasks have a precedence constraint, where the tasks may be dependable of previous ones and have performance performance constraints, where the tasks may have to meet reliability, availability or performance requirements.

There is also the problem that the data can be acquired from different data sources and it is necessary to consolidate these sources in one of the tasks.

2.4 Data Cleaning

As explained in the introductory section Chapter 1, sensorized systems are susceptible to failures and errors associated with sensor calibration and data acquisition-transmission-storage, therefore producing faulty data. The major sources of problems are briefly defined below.

2.4.1 Duplicates

Duplicates is when occurs a missing observation and the value of the next observation is duplicated, because the system considered that the value should be accumulated from the previous observation, or when there are two observations for the same time point. We assume only the latter definition for duplicates in the thesis.

2.4.2 Outliers

Outliers can be seen as observations which significantly deviate from expectations that may induce error in the system or may themselves denote abnormal behaviours or anomalies in the systems. Outliers can

either exist as a singular or a subsequences of observations. The singular outliers can be global if their value deviates from the entire dataset, or contextual if their value deviates from observations in the same context. Similarly, an outlier subsequence is sequence of observations whose pattern deviates from expectations. In fact, both the singular and sub-sequence outliers share similarities, but can be processed differently. Outliers can be dealt in two different way: by removing the outliers from the dataset, which is not the best option specially dealing with time series, or transforming that value into a missing observation and using imputation methods in order to impute that value with a more suitable one. However, should be noted that some outliers can be an interesting event and should be analyzed itself, like a burst in a water flow time series data. In this case, after detection and before treatment those observations should be saved and reported to the water network manager.

Outliers can further categorized in accordance with the effect they yield on the data. The additive outliers (AO) affects the time series for only one time period. The level shift outliers (LS) affects for all future time. The temporary change outlier (TC) effects the time series with a exponential decaying over time. The innovational outlier (IO) effects the time series over the subsequent observations.

2.4.3 Gross errors

Gross errors are observations similar to the outliers, whose values are out of the possible range for a given sensor. Illustrating, the water flow sensors can not have negative values, in order that, any value below 0 is considered a gross error. Similarly to the outliers, typically this type of errors are turned into a missing observations and are dealt with the same way the missing observations are.

2.4.4 Missing Values

The target time series data is incomplete, i.e., an arbitrarily-high amount of observations in the data are missing. Missing observations, commonly referred as missing values, can be divided into three main categories associated with the underlying stochastic processes that describe their occurrence: missing completely at random (MCAR), missing at random (MAR) and not missing at random (NMAR).

- **MCAR**: the missing observations occur entirely at random, where the probability of the observation being missing is the same for every time point (memoryless). This implies that the missing observations are totally unrelated to the data and there is no systematic mechanism on the way the data is missing., e.g., malfunction on the transmission, storage and acquisition of the data.

- **MAR**: the missing observations occur at random, where the probability of the observation being missing is the same only within the observed data. This implies that the missing observations are independent of the value of the observation itself, but it is dependent on the other non-missing

observations, e.g., malfunction of the sensors occurs mostly when the temperatures are below -30ºC.

- **NMAR**: the missing observations does not occur at random, where the probability of the observation being missing varies for unknown reasons and depends only on the value of the observation itself. e.g., water flow sensors not measuring values bellow 0.

In accordance with the properties of water distribution systems, in this thesis we will assume that our missing values are all MAR. We also need to have into account that there are two types of missing observations: missing sequence observations, which are represented by a long range where there is nothing but missing observations, or discrete missing observations, which are represented by individual missing observations among the data.

There are four typical approaches to deal with missing values:

- Delete the observations from the dataset where the missing values occurs. This will create inconsistencies and gaps on the time series. The subsequent time series analysis becomes dependent on the ability of the applied methods to handle incomplete time series.
- Replace the missing values with an impossible/dedicated/special value. This value will be considered as an outlier and will get the same pre-processing as the outliers do.
- Replace the missing values with data imputation methods.
- Keep the missing observations in the data and apply models which are capable of dealing work with missing data.

2.5 Hyperparameterization

Most of the available pre-processing methods are parametric. In this context, their suitability for a given data context depends on the correct selection of parameters, the process referred as hyperparameterization. This task can be manual, where the user explores specific parameterizations in order to achieve the best possible results. This is a laborious task given the huge number of available combinations. To address this, hyperparameterization can alternatively be an autonomous task, where the user select the range for each parameter and it autonomously select whose values are better for each parameter.

3

Related work

Contents

This section surveys state-of-the-art methods for outliers detection and missing value imputation fields will be discussed. Within each field, both approaches for historical and real time data will be considered.

3.1 Outliers

Gupta et al. [8], survey approaches to detect outliers on temporal data, mainly on time series data, data streams, distributed data, spatio-temporal data and network data. For the purpose of the thesis, only the time series data and data streams (which can be seen as real time data) will be discussed, and specifically approaches for single time series data analysis.

3.1.1 Historical Time Series Data

Gupta identifies two major types of outliers in the context of a single time series: 1) find singular observations (one time point) within the time series which are outliers or 2) find a sub-sequence of observations (sequence of time points) which represents a bulk of outliers. On the time point perspective, it is possible to detect outliers by using: 1) prediction models, 2) profile based models, 3) information theoretic techniques, 4) classification and clustering approaches.

Prediction models, assigned a score to each observation as a deviation from the predicted value computed on a generic prediction model. This predicted value at time t can be obtained using the following prediction models to compute the predicted value (on univariate time series): using it as a median of a $2k$-size window from t-k to t+k. Using the nearest cluster predictor by computing the average of all observations in the cluster that the observation at time t was assigned to (this cluster was obtained as the b most similar observations in the time series, based on euclidean distance). Using single-layer linear networks which predicts an observation at time t by using a linear combination of the q previous observation and a set of weights w which define the relationship between a moving window and the expected value from the observation t. Using a simple regression models and support vector regression method, where a more detailed presentation on the latter can be found in Smola and Schölkopf [9]. The usage of multi-layer perceptron which is well documented in Bishop [10], using a feed-forward network with sigmoid activation function in the hidden layers and a linear activation in the output layer. On non-Gaussian time series it is used a Mixture Transition Distribution models where it is possible to obtain a full predictive distribution that takes into account the possibility of future outliers and exploit the presence of different prediction intervals. An ARIMA model it is also proposed, which can identify different types of outliers, as referred in 2.4.2. Also it is possible to identify non-consecutive singular outliers by using re-weighted maximum likelihood estimation which is a generalization of the M step of EM algorithm as will be explained in 4.1.4.A or a new approach using Gibbs sampling as specified in Justel et al. [11]. There

are many more of variants of these techniques such as Chen and Liu [12]. At last there are approaches for prediction on multivariate time series such as selecting on a projection direction instead of testing on the time series directly.

Profile-based models trace the normal profile of the series behavior. Then, any new observation is compared with this normal profile and, like the prediction models, an outlier score is assigned.

In *information theoretic techniques*, there is an exploration of the space-deviation trade off, by fixing the deviation instead of fixing the space as in conventional models. If the removal of a observation in a sequence results in a more succinctly representation, e.g. histograms, than the one with the removed observation, than that observation is considered an outlier and therefore, a deviant, as stated in Jagadish et al. [13]. Other proposals that use dynamic programming are presented to solve the problem that is difficult to find observations which the removal results in a representation with a lower error bound than the original.

Classification and clustering methods can also be used to detect outliers. In addition to surveyed categories of approaches for point outliers, there are also approaches to detect subsequences of outliers in single time series. Citing Keogh et al. [14], a subsequence is an outlier if "*given a time series T, the subsequence D of length n beginning at position l is said to be the outlier of T if D has the largest distance to its nearest non-self match*". A solution is called brute force where it is considered all subsequences of length n in T and then compute the distance for each with each non-overlapping subsequence. Keogh et al. [14] outline alternative solutions more efficient than the previous referred, e.g., heuristic reordering of candidate subsequences, locality sensitive hashing, Haar wavelet, augmented tries and SAX with augmented tries. These are used for an improved ordering of subsequences, for an effective selection of the outliers. The most used distance metric between subsequences in these methods is the Euclidean. Others discussions on the the topic lay on the fact that the time series may have not unequal intervals, i.e., value sampled at equal time intervals. Chen and Zhan [15] defined two methods based on this premise where a pattern (subsequence of two consecutive points) is defined and the patterns that are considered outliers are those which have very few patterns with the same slope and length.

3.1.2 Real Time Data

In contrast with historical data, streaming (real time) data yields a significant difference: the non fixed length of the data and the requirement that the processing task must be efficient enough to deal with the sampling rate of the new coming observations. Therefore, the referred methods for outliers detection can substantially differ. According to Gupta, the methods can be divided into: 1) evolving prediction models, 2) distance based outliers for sliding windows and 3) outliers in high-dimensional data streams.

Evolving prediction models are updated as new data arrives into the system, so that is possible to capture trend in the data. These models can be divided into online sequential discounting, dynamic

cluster maintenance and dynamic Bayesian networks. On the online sequential discounting, the following papers Yamanishi and ichi Takeuchi [16] and Yamanishi et al. [17], present a method which learns, as the new data arrives, a probabilistic mixture model with a decay factor. For categorical attributes they present the sequentially discounting Laplace estimation (SDLE) method, which by partitioning the values space of all categorical attributes allows the creation of cells and when a new observation arrives the cells count is adjusted with a temporal discounting and a Laplace smoothing is applied. For numerical attributes three models are proposed. A Gaussian mixture model (GMM) in a parametric case where an incremental EM with discounting on the effect of the past examples, a kernel mixture model (KMM) in a non-parametric case where it iteratively learns the means of the kernels, and a time series model which learns the parameters of an auto regressive model with time discounting. In other methods it is given more weight to recent data by using exponential greater importance. On the dynamic cluster, maintenance profiles are created and when new subsequences of data arrive to the system there is a comparison to those profiles and then a rating it is given to the subsequence as classifies if it has outliers. Dynamic Bayesian networks are used when updating the model is not enough. There are two major methods: 1) Bayesian credible interval (BCI), which uses a hidden Markov model (HMM) to infer posterior distributions of hidden and observed states to build itself and then any observation that falls out of the interval is considered an outlier, and 2) Maximum a posterior measurement status (MAP-ms) method which uses a two layered deep belief network which estimates the status (e.g. normal or outliers) to each new observation.

Distance based outliers for sliding windows describe that a observation is considered an outlier o if there are less than k points at distance d from o, on a specific window of the data. In this approach outliers can be found globally, in the dataset, or locally, in the current sliding window (neighbours). Some methods are proposed to deal with the global outliers. Using a data structure called indexed stream buffer (ISB) which supports a range query on the entire window, and the exact answer of the data stream outlier query can be computed. It is also proposed that only a fraction of the safe inliers (an observation that will have always at least k succeeding observations during the data streaming) is retained in the ISB and that is enough to store only the fraction of safe inliers. Also a subsequent window can be predicted by checking the observations that will participate in each of those windows as explained in Yang et al. [18]. For local outliers it is mostly used the local outlier factor (LOF) method which computes the local deviation of an observation with all its neighbours. The LOF method has multiple derivations.

Outliers in high-dimensional data streams can be detected using a system called stream projected outlier deTector (SPOT), Zhang et al. [19], which is a technique that: 1) creates a window-based time model to capture statistics from the data stream, 2) a set of sub-spaces is obtained using unsupervised and/or supervised processes to detect projected outliers effectively and 3) carry out online self-evolution to compete with the dynamics of the data streams.

3.2 Missing values imputation

The imputation of missing values is a broadly studied area. There are single imputation approaches and multiple imputation approaches. In single imputation approaches, only one version of the dataset is imputed. On the other hand, multiple imputation is a missing data imputation approach where multiple versions of the dataset are imputed with different values, in order to use appropriately all the information present in the dataset with missingness. This approach can reduce the bias and increase efficiency in comparison with other methods. A multiple imputation framework creates multiple datasets with different imputed versions of the data, which are used to create multiple models and in the end the mean over all the models is computed and sum the within and between-imputation variance. It is also important to refer that one condition to use multiple imputation is that the missing data distribution must be MAR.

3.2.1 Historical Time Series Data

The imputation of missing values for historical data is an area where a lot of research has been done throughout the years. A considerably high number of techniques have been developed for any type of data and others specifically for time series, either for univariate and multivariate. In this section we will present multiple researches on the imputation of missing values.

Moritz et al. [20] did an extensive evaluation of multiple imputation methods on univariate time series available on R. Other multivariate imputation methods are also referred but not used on the experiments. They tested the imputations methods on four different datasets, each one having a different combinations of trend and seasonal components of time series, i.e., no trend and no seasonality, trend and no seasonality, no trend and seasonality, and trend and seasonality. The methods used in their experiment were the following: Aggregated values, from the R package *zoo* Zeileis and Grothendieck [21], which imputes the missing observations by overall mean, monthly mean, etc. In the experiments they only impute by overall mean. LOCF, from the R package *zoo*, which uses the value of the previous observation to impute. Seasonal Kalman filter, from the R package *zoo*, where the missing observations are interpolated using a seasonal Kalman filter. Interpolation, from the R package *forecast* Hyndman et al. [22], and from the R package *zoo*. The former has a different step from the later, which is the removal of the season component of the time series at first, and after the imputation the component it is added again. Model-based imputation, from the R package *VIM* Kowarik and Templ [23], which induces lags for the time series to create a multivariate dataset. With that they predict the missing observations of one variable by using the others as regressors.

In order to evaluate the performance of each imputation method on each dataset the following procedure was done. First, they artificially deleted observations in a completed dataset using a missing

data simulation function which is based on the assumption that the data is following a MCAR distribution. Second, they used the imputation methods referred above to fill the missing observations. Third, they compared the results of each imputation method with the original completed dataset using the performance Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE). They also evaluated the run-time of each method. The experiments were also performed using different percentages of missing observations (0.1, 0.3, 0.5, 0.7) and the missing data creation function was used deleted the observations using 25 different seeds, in order to replicate the removal of a bigger range of specific observations. With that their experiment was performed 600 times for each dataset (25 different seeds, 4 different missing rates, 6 different methods).

Linear interpolation and seasonal Kalman filter were the methods with best overall performance in imputing the missing values correctly. On the other hand, the best run-time performance methods were the aggregated values and the LOCF, which depending on the datasets they lead into very misleading imputation results, so it was advised to use it with caution. The model-based imputation lead into mediocre imputation results and it was the slowest of the methods.

Osman et al. [24] did an extensive research about general missing data imputation methods specific to WDN, and also developed an approach for a imputation method selection. This approach is based on three steps which are: the categories associated with the underlying stochastic process that describe the missing values as referred previously in section 2.4.4, the percentage of missingness in the data and in a categorization of missing data methods. First it is necessary to understand in which category the missingness of the data is associated. If the data is NMAR it is due to selection bias, and it is possible to use the Heckman correction to correct the bias by using a simple regression method to estimate behavioral functions free of selection bias, as described in Heckman [25]. If the data is either MAR or MCAR, we go into the next step. If the percentage of missing data is lower than 5% it is used traditional imputation techniques. On the other hand if the percentage of missing data is greater than 5% it is used modern imputation methods.

A – Traditional Techniques: On the traditional imputation techniques, the first one presented is the deletion of the missing observations. This deletion can be divided into two techniques: listwise and pairwise. Listwise deletion results in the removal of all missing observations, which can induce the bias on the data, mainly when the percentage of missing observations is to high and the dataset is too small. On the other hand the pairwise deletion is a more controlled technique, where the elimination occurs only in the cases with high percentage of missing observations or with an overall small dataset, which tries to minimize the data loss. The deletion method in overall is not the best option but has the advantage of being easily implemented. The next referred technique by Osman et al. [24] is the single imputation, which contains the following methods: mean imputation where the values are imputed with average of

the dataset. It has the advantage of being easily implemented but the imputation results may not be the ideal since it depends rigorously in the data distribution. Hot-deck and cold-deck imputations methods are the following in the single imputation. Hot-deck imputation uses values from similar observations as imputation. These method can be divided into distance function approach where the value used as imputation is the closest non-missing observation (like the LOCF), and pattern matching approach where the dataset is divided into multiple similar groups and the imputed values is selected randomly among the values in the same group. Similarly, the cold-deck imputation, select the sample, but from a different dataset. The last method from the traditional techniques is the regression method where the value imputed is estimated by using a function (linear, quadratic, cubic, etc). This method predicts the most likely value for the missing observations but does not takes into account the error and the uncertainty related to the imputed value. Stochastic regression attempts to rectify this problem by generating and adding an error term to induce variance to the missing observation.

B – Modern Techniques: On the modern techniques, the first one presented by Osman et al. [24] is the multiple imputation method which was already explained in section 3.2. It is a complex method but induces variability in order to find a range of possible imputation values. The next technique referred is model-base which embrace the EM method, which will be explained in section 4.1.4.A and estimates excellent parameters but does not provide standard errors as part of the process. The last modern techniques are called machine learning techniques and are approaches derived from the pattern recognition and learning theory, which creates a predictive model that estimates the missing observations based on the information in the dataset. This technique embraces the following methods: Gaussian process regression (GPR), which is a non parametric Bayesian approach to regression, where the key assumption of the Gaussian process is that the data can be represented as a sample from a multivariate Gaussian distribution and the regression function used is dependent of the data and not of specific model. Another modern technique is based on the principal component analysis (PCA). Although this technique it is most used on data mining analysis, Ilin and Raiko [26], did an extensive work on the reconstruction of missing data using PCA. The next modern technique presented is the kNN method, where the average to the k nearest neighbours is computed. This method can also be seen has a hot-deck method for the reasons explained in the traditional techniques. The next presented model from the modern techniques is the decision trees. Mostly used as a classification method, it has a wide range of methods (ID3, C4.5, CN2, etc), where each one of them uses a different approach to deal with missing observations, e.g., the CN2 imputes missing observations by filling the missing observations with the attribute most common value. The next model is the neural networks (NN) outlined by Bishop [10]. NN are organized in layers: the input layer, hidden layer (which can have as many as possible) and the output layer. Each layer is composed by neurons which are connected to all neurons of the next layer in order to propagate

information. NN have multiple variations including: feedfoward NN which are knowledge-based systems where the knowledge of the data passes trough each layer and in the end it is used to impute the missing observation, feedback NN which are error-based systems and the missing observations imputation values are updated, during multiple iterations, using feedback connections (propagating the error to the previous layers) and auto-associative NN where it first learns from complete cases and only after it replicates the process for cases with missing observations. The last technique from the modern techniques is tensor based methods. Tensors can be seen as multi-dimensional arrays. From their decomposition and by learning inherent collaborative relationships from non-missing data it is possible to impute the missing observations as described in Garg et al. [27].

Osman et al. [24] did a brief comparison of these methods. He concluded that the deletion method is the simplest technique in comparison with multiple imputation, GPR and expectation maximization which are more complex and challenging. Moreover, deletion, mean imputation and hot-deck, are mathematically easier to understand but the results might be misleading leading into a reduction of population variance and bias. EM, GPR and multiple imputations methods are more complex mathematically because they require specification of likelihood and posterior distribution from which inferences and uncertainty measures from missing data can be obtained.

Besides all the previous methods for missing value imputation, Osman also presented imputation methods used specifically in WDN. The first one mentioned is the two-phase model presented by Quevedo et al. [28], which have the advantage of being able to deal with large amounts of data. However, on the other hand, it can only use its own data to impute the missing values and the large amounts of data is required because it is needed to produce consumer demand patterns. The second approach for missing values imputation on WDN is the combined model approach, Barrela et al. [29], which is a combination of both forecast and backcast missing observations values generated by TBATS and ARIMA models, accommodating multiple seasonality. Moreover, this method can be used in both online and historical data. The least squares-Kalman filter, Bennis et al. [30] it is the following presented approach. Is an improvement on the least squares method (an approach of the regression method) by combining it with the Kalman filter, in order to increase the accuracy. Other methods are also mentioned but are mainly focused on real time data, which will be detailed in the next section 3.2.2.

3.2.2 Real Time Data

Research on imputation approaches for real time data is not as common as the class of approaches outlined in the previous section.

Osman et al. [24] present approaches for missing values imputation in real time data from a WDN. The first system uses a real-time dynamic hydraulic model which is used for portable water loss reduc-

tion. Real-time data is constantly entering the model which is capable of evaluate the network condition and sends automatically control to the various networks components, adjusting the WDN and make it more efficient. This system has a built-in missing data imputation feature which uses a least squares-Kalman filter method like the on mentioned in the previous section. The other referred system in the paper it is not processed at real time, but uses data obtained in a semi-short period of time, using the virtual sensors concept. This concept focus on the installation of permanent and temporary sensors in the WDN. The permanent sensors are the already built-in sensors of the system, but the temporary are installed in strategically chosen optimal locations in the WDN for a period of seven to ten days. Real-time data will constantly feed the system using both types of sensors. After the defined period, the temporary sensors are removed, and accumulated data from that period of both sensors compared. A Gaussian process regression imputation method is suggested to correlate the temporary sensors data into the permanent sensors data.

Fan et al. [31] proposed a model called On-line Missing Value Imputation (OLMVI), which can analyze tuples (observations) of information and impute the missing observations before they are added into a database. Each position of the tuple represents a value in a feature/attribute. Because of that it is necessary to have a correlation matrix always updated as the data enters the system, which is good for scalability. Using this correlation matrix and the attributes, imputation candidates are computed by assigning an imputation score to each of them. The candidate with the highest score is the one used as imputed value. The process of analyzing if an observation has a missing value and the process of imputation are divided. Because of that, observations with missing values will not prevent the analysis of non-missing observations. Also, the correlation matrix is always being updated, which makes this method an incremental learning type method.

4

Solution

Contents

Despite the relevance of the surveyed contributions, existing time series pre-processing methods are generally oriented towards specific data regularities and types of errors. Thorough comparisons are thus necessary to place proper decisions, a generally laborious and difficult process due to the difficulty of performing objective assessments in the absence of ground truth.

In this context, we propose an approach for the fully autonomous processing of multivariate time series data, in historical and real time data contexts.

The major idea behind this approach is to generate precise ground truth for the sound and quality-driven evaluation of available processing options. To this end, our method relies on two major principles: i) detection of conserved segments within the imputed series data, and ii) modeling the type and amount of observed errors. Under these principles, the evaluation can be performed by purposefully planting errors along the conserved segments.

The approach provides a good coverage of available processing options, providing over twenty state-of-the-art methods for missing imputation and outlier detection from time series data. With the objective to deal with errors of varying profile, our approach incorporates processing methods able to deal with both point and serial missing and outlier values. In addition, is able explore the aided processing guidance provided by correlated variables within multivariate time series data. To this end, state-of-the-art processing methods able to capture cross-variable dependencies are further supported.

The proposed method uses a sequential approach that will be depicted for the historical data contexts in section 4.1 and the real time data contexts in section 4.2.

4.1 Historical data

Section 4.1.1 introduces the pipeline of the proposed method. Section 4.1.2 depicts how to handle duplicates and remove the gross errors. Section 4.1.3 enumerates the available methods for outlier detection, the hyperparameterization functionality and the possibilities for treating outliers offered to the user. Section 4.1.4 enumerates the available methods for missing values imputation, the hyperparameterization functionality and how to impute them.

4.1.1 Pipeline

As stated before, AutoMTS uses a sequential approach for pre-processing time series. The four major steps are depicted in the figure 4.1. Given a time series (univariate or multivariate), the *first step* is to treat duplicates and remove them. After the time series is cleansed of duplicates, the *second step* is the detection of gross-errors and their removal. The *third step* is the detection of outliers and their removal or retention. The *fourth step*, and the last, is the imputation all the missing observations that exist throughout the time series.

In order to evaluate and select the most promising outliers detection and missing imputation methods, the proposed method selects the largest segment of observations without errors, i.e., missing values, that exist in the original time series. The evaluation of the most promising methods makes use of this segment and not of the original series. When the most promising methods are selected, they are applied (in the same order of the pipeline) to the original series to obtain the pre-processed time series.

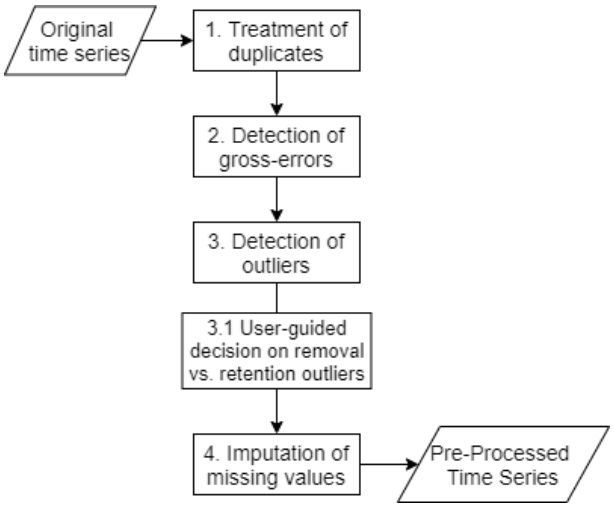

Figure 4.1: Pipeline of the historical data pre-processing methodology.

4.1.2 Early processing steps

4.1.2.A Autonomous duplicates treatment *(step 1)*

The first step detects and treats the duplicate observations that subsist in the time series. In case the detected duplicates have the same values, either one of the observations is removed. In the other hand, if the detected duplicates have different values the following approach is performed: i) remove the duplicate observations and insert a new observation as a missing value (with the same time stamp as the duplicates); ii) impute the missing values using a default imputation method available in the approach; iii) the duplicate observation which has the smaller difference with the imputed missing value is selected and the other is removed from the time series.

4.1.2.B Gross-errors treatment *(step 2)*

The second step detects the gross errors that exist in the time series against background knowledge. This means the user can specify, in the given file, the ranges that each of the series must have. For instance, in the context of water flow and pressure sensors, lower bounds are generally zero and upper bounds fixed in accordance with the pipe specifications. Any value out of the ranges are turned into a

missing observation and will be dealt latter in the process by the step 4. If the user does not specify the ranges, we assume that the series does not have any value limitations.

4.1.3 Autonomous outliers detection *(step 3)*

The third step autonomously detects the outliers observations presented in the data. It is possible to classify outliers as *point-wise*, which are outliers that appear randomly and individually from the others, and *sequential*, which are outliers that occur in clusters/sequences of outliers.

4.1.3.A Available methods

The outliers detection is a field where multiple approaches have been proposed in the available literature as referred in section 3.1.1. Our methodology provides the following outliers detection methods:

- **Standard Deviation:** If the data follows a normal distribution, then we can say that approximately 68% of the data are in one standard deviation of the mean. In this approach we assume that observations whose values that are within three standard deviations, or more, are considered outliers.
- **Interquartile range (IQR):** Since not all data follows a normal distribution, a good way to summarize a non-normal distribution sample of the data is the interquartile range. The IQR is computed as the difference between the 75th and the 25th percentiles of the data. The IQR it is used to identify outliers by defining limits on the values that are a factor k of the IQR below the 25th percentile or above the 75th percentile. Any observation whose value is out of the limits, is considered an outlier.
- **Isolation forest:** This method proposed by Liu et al. [32], isolates the outliers instead of focusing on creating profiles and assigning outliers scores to each observation. It takes into advantage that the outliers are the minority of the data and are very different from the normal observations. The method isolates random observations by selecting an attribute and then randomly select a value between the maximum and minimum value of that attribute. The number of splittings required to isolate an observation is equivalent to the path length from the root node of the tree to the terminating node. Random partitioning produces shorter paths for anomalies. For each observation is given an anomaly score.
- **Local outlier factor (LOF):** This method proposed by Breunig et al. [33], finds outliers by computing a score based on the local density deviation of an observation with respect to its neighbours (average of the local density deviation). A normal observation is expected to have a score similar to its neighbours, while an outlier is expected to have a score much smaller than its neighbours.

This method also takes in consideration both local and global specifications of the dataset into consideration.

- **The Density-Based Spatial Clustering of Application with Noise (DBScan):** The DBSCAN method Ester et al. [34] is a cluster based method where, it regard clusters as areas of high density separated by areas of low density. There are a set of core samples which are in areas of high density and non-core samples that are close to a core samples, but are not core samples themselves. Any non-core sample that is a far distance from any core sample is considered an outlier.

- **HOT SAX:** This method Lin et al. [35], converts the observations into symbols, where similar values have the same symbols. Symbols that are out of normal range are considered outliers. This methods works mostly for sequence of outliers, since those are all assembled into one specific symbol. Even more, this method is useful because it allows reduction of dimensionality which is one of the main problems of symbolic methods, since the dimensionality is the same as the original data.

From the listed methods the standard deviation, IQR, isolation forests, LOF, DBScan work either for point or sequence outliers. On the other hand, HOT SAX only works consistently for sequential outliers.

4.1.3.B Hyperparameterization

Some of the methods stated in the previous section have a set of parameters that could change how task of detecting outliers performs. As stated in section 2.5, the user can select manually the values for each parameter, but it is a laborious task. Moreover, the parameters selection is task dependent and the best parameters for a specific time series could not achieve the best results for another. So it is important to detect the best parameters autonomously for each of the time series available. According to Bergstra and Bengio [36], a combination of grid search and manual search is possibility to this end. It is only required to choose a set of values for each of the parameters that we want to test on a method and it automatically select the best parameters combinations. The problem here is that the grid search can take a lot of time, since it scales with the number of parameters and the values we want to test. Although in the historical data context the time is not one of the main constraints (as opposing to the real time data context), we want the parameter selection to be as efficient as possible since it will be in either way a time consuming task.

In order to be as efficient as possible we will use the Bayesian optimization Snoek et al. [37], a probability model which builds and objective function and use it to select the most promising parameters. This method evaluates only the most promising areas of the search space, and it is possible to choose

how many maximum iterations we want the method to run. In our case, the objective function is to maximize the F1-Score, a evaluation metric that will be explained further in document, in section 5.2.

Nevertheless, to maximize the F1-Score, we need to know what are the ground truth (GT) outliers (what we consider that are true outliers), to be compared with the outliers detected by our methods. Therefore we purposefully plants artificial outliers in the conserved segments explained in section 4.1.1. We can either plant point and/or sequence of GT outliers. All the methods are evaluated against the GT outliers, and the hyperparameterization will return the best method for a given time series, and if the method is parametric, the best set of parameters.

4.1.3.C Outliers removal

The outlier detection method selected by the autonomous hyperparameterization step is then used to detect outliers in the original time series. These detected outliers can be seen by the user as events of interest or simply errors in the data. The detected outliers, along with their anomaly values, will be given to the user and he may opt to either discard the outliers or mark some of the detected outliers to be retained in the time series. The discarded outliers are turned into missing values and will be dealt latter in the process in section 4.1.4.

4.1.4 Missing values imputation *(step 4)*

The fourth and last step autonomously detects and cleans the missing values presented. In like manner as the outliers, it is possible to classify missing observations as *point-wise* and *sequential*.

4.1.4.A Available methods

- **Mean and Median:** Both these methods consist in replacing the missing values of an attribute with the mean/median of all historical non-missing values of time series.
- **Random sample:** The missing observations are imputed using a random non-missing observation of the time series.
- **Interpolation:** The imputation of each missing value consists in the average of the last observation and next observation of that missing value. Linear interpolation can capture time dependencies using the previous and next observations combined.
- **Last Observation Carried Forward(LOCF) and Next Observation Carried Back-ward(NOCB):** Both these methods consist in imputing the missing value with the value of the last observed value for the LOCF or with the value of next observed value for the NOCB. Both these methods assume that there is no change since the last or next observation. Although, both these methods can be

used in any type of tabular data, they are used on time series data since they can impute the missing observations based on the previous and the next temporal observation and capture at least one time stamp dependency.

- **Moving average:** This method consists in imputing the missing observation by taking the average of several sequential values, which are represented by a window of *m* size, where *m* is the number of observations used on the average computation. Therefore, these sequential values are centered on the missing observation. For a two-sided moving average,

$$x_t = \frac{1}{m} \sum_{j=-k}^{k} x_{t+j}, t = k+1, k+2, \ldots, n-k, \tag{4.1}$$

where $m = 2k + 1$. Thus, we estimate the missing observation at time *t* using *k* periods. We can also call this a m-Moving Average, meaning a moving average of order *m*. When the sequential values are all missing observations, the window size will expand further until two non-missing values occur. It is possible to say that Linear Interpolation is specification of a 2-Moving Average and because of that it can also capture some time dependencies.

- **Random forests:** The missing observations are imputed using an iterative method based on a random forest. A random forest is a machine learning method which construct multiple decision trees and outputs the mode or mean from all of them. According to Stekhoven and Bühlmann [38] this method pretends to input any type of data and makes as few as possible assumptions about the data structure. The random forests method Breiman [39] is able to deal with mixed-type data and as a non-parametric method it allows for interactive and regression (non-linear) effects. It also has a integrated routine for handling missing observations, by weighting the frequency of the observed values in a variable, which requires a complete response variable for training the forest. On the other hand, the MissForests method predicts the missing values using a random forest trained on the observed parts of the dataset (which is represented by a matrix). The iterative imputation scheme is addressed by:

  1. Making an initial guess for the missing observations using a classic imputation method.
  2. Sorting the variables by frequency of missing observations, starting for the ones with lowest amount.
  3. For each variable *Y*, imputing the missing observations, at entries *i(s)*, by fitting a random forest, using as response the observed values of the variable *Y* and as predictors the all the other variables. Then predict the missing observations by applying the trained Random Forest to the variables other than *Y*, with observations *i(s)*.

4. Checking stopping criterion.

This procedure is repeated until the stopping criterion is met, which is, the difference between the newly imputed matrix and the previous one increases for the first time. As the time property is not required, it can be applied in tabular data.

- **Expectation maximization:** The Expectation Maximization algorithm (EM) is an iterative method for computation of maximum likelihood estimates of parameters when the observations have missing values. This method assumes the existence of two observations spaces, i.e., two variables, and a mapping between both. We will call this two observation spaces X and Y which have the joint distribution (X, Y). According to Dempster et al. [40] the observed observations y are a realization from Y and x belongs to X, but is only observed indirectly through y. Thus, it is possible to assume that there is a mapping x $\rightarrow y(x)$ and x only lies in the subset $X(y)$ of X determined by the equation $y = y(x)$, where y, as stated before, is the observed data. Both X and Y are distributed according to some probability P_θ for some parameter θ in a set of parameters Θ. Thus, we have a complete data specification $f(x|\Theta)$ and the incomplete data specification $g(y|\Theta)$. Thus, the EM algorithm wants to find the value Θ which maximize $g(y|\Theta)$ making use of $f(x|\Theta)$.

  Initially the algorithm chooses one random θ from set Θ. At each iteration t, it goes from Expectation step (E-step) to Maximization step (M-step) until it converges or until it runs for a maximum number of iterations.

  The E-step will compute the expected value of θ_t from the complete data specification $f(x|\Theta)$ given the observation y,

$$Q(\Theta, \Theta_t) = E(\log f(x|\Theta)|y, \Theta). \tag{4.2}$$

  Since there are many possible complete data specification $f(x|\Theta)$ that will generate $g(y|\Theta)$, the M-step estimates the parameters Θ $t+1$ that maximizes $Q(\Theta, \Theta_t)$,

$$\Theta_{t+1} = \arg\max_{\theta} Q(\Theta, \Theta_t). \tag{4.3}$$

  It is also used on any type of tabular data.

- **k-Nearest Neighbors (kNN):** This method consists in imputing the missing observations using other similar observations. Thus, this similar observations are obtained by getting the k closest neighbors using a distance. According to Troyanskaya et al. [41] the Euclidean Distance was a sufficiently accurate enough distance norm. After that we compute the weighted or unweighted

average, and impute the value on the missing observation. It is prepared for tabular data. Moreover, it is also possible to use the kNN not focusing exclusively on the similar observations, but also, we could also segment the time series and apply the kNN on similar segments.

- **Multivariate Imputation by Chained Equations (MICE):** MICE, proposed by van Buuren and Groothuis-Oudshoorn [42] is a method that imputes missing observations on multivariate datasets using a multiple imputation framework, as will be explained in section 3.2, associated with chained equations technique, which are based in the multiple imputation concept and can be seen as concatenation of multiple univariate procedures to impute missing value according to the following steps:

  1. Impute all the missing observations of the dataset, using a simple imputation method. These imputed values can be seen as place holder for the following steps.
  2. The place holder of one of the variables are set back again to missing value.
  3. Use the variable with the new missing observations as the dependent variable in a regression model and all the other variables are the independent variable in the regression model.
  4. The missing observations are then imputed by the predictions of the regression model.
  5. Repeat the process for all the other variables and repeat all the cycle again until the distribution of the of the parameters of the imputations converge.

  This method is originally prepared for tabular data and might be extensible towards time series data.

- **Amelia:** Multiple imputation approach (available as a R package) as introduced in section 3.2, but the imputation method that is specifically used in each of the dataset versions is the Expectation Maximization method with bootstrapping (EMB). It also neglects the time dimension.

From the listed methods the mean, median, random sample, interpolation, LOCF, NOCB and moving average are specific for univariate time series, while the random forests, expectation maximization, kNN, MICE and Amelia are specific for multivariate time series.

4.1.4.B Hyperparameterization

The hyperparameterization for the missing imputation method selection have the same goal as the hyperparameterization in section 4.1.3.B. In matter of fact, the Bayesian optimization is also performed but instead of maximizing the F1-score metric, we want to minimize the root mean squared error (RMSE) (section 5.2) metric that will be explained further in the document. Again, to compute the RMSE, we need to purposefully plant artificial missing values in the conserved segment explained in section 4.1.1. Since

we know the original value, our GT, of the newly planted missing values, we use our methods to impute them, and the imputed value is compared with the original value. All the methods are evaluated against the GT and the hyperparameterization will return the best method for the given time series, and if the method is parametric, the best set of parameters.

4.1.4.C Imputation

The missing imputation method selected in the hyperparameterization will be used to impute all the missing values in the original series, plus, the new missing observations created by the steps 2 and 3. After the imputation is concluded we got the final pre-processed time series available for the user.

4.2 Real time data

The following sections are organized as follows: section 4.2.1 shows the pipeline of the AutoMTS. Section 4.2.2 shows how our methodology deals with duplicates and remove the gross errors. Section 4.2.3 enumerate the available methods for outlier detection, the hyperparameterization functionality and how to remove them. Section 4.2.4 enumerate the available methods for missing values imputation, the hyperparameterization functionality and how to impute them. At last, section 4.2.5 illustrates of the script available for the real time data.

Even more, we have to consider that in the real-time setting we have a time limitation, i.e., we have to complete the four previous steps, before a new observation comes into the system. Because of that we need to take into account two new principles: historical window and buffer. For our real-time approach to work we require a historical window, i.e., previous stored observations. This window will help our methods to behave better since they have the window size of observations to work with. The buffer is what we call a window of saved observations before they enter the system. With this we can use methods that require observations next to the observation we are working with. The window and buffer sizes are dependable of the periodicity of the new coming observations, e.g., if an observation takes 1 hour to get into the system, we can use a bigger window and buffer size than if the periodicity of the series is 1 minute.

4.2.1 Pipeline

As stated before, AutoMTS uses a sequential approach for pre-processing time series, although in the real-time setting, the approach is also conditional. Instead of having access to the totality of the time series, we receive a new observations with a specific period, so we have to deal observations-wise instead of dataset-wise. The four major steps are depicted in the figure 4.2. Given a new observation,

the *first step* is to check if it is a duplicate. If it is then we remove the observation, if it is not we go to the next step. The *second step* is to detect if the observation is a gross-error. If it is we remove the observation and impute as a missing value, if it is not we go to the next step. The *third step* is to detect the observation as an outlier. If it is we remove the observation and impute as missing value, if it is not we got to the next step. The *fourth step*, we check if the observation is a missing value. If it is we impute it and if it is not we can store the observation because we know it is cleaned.

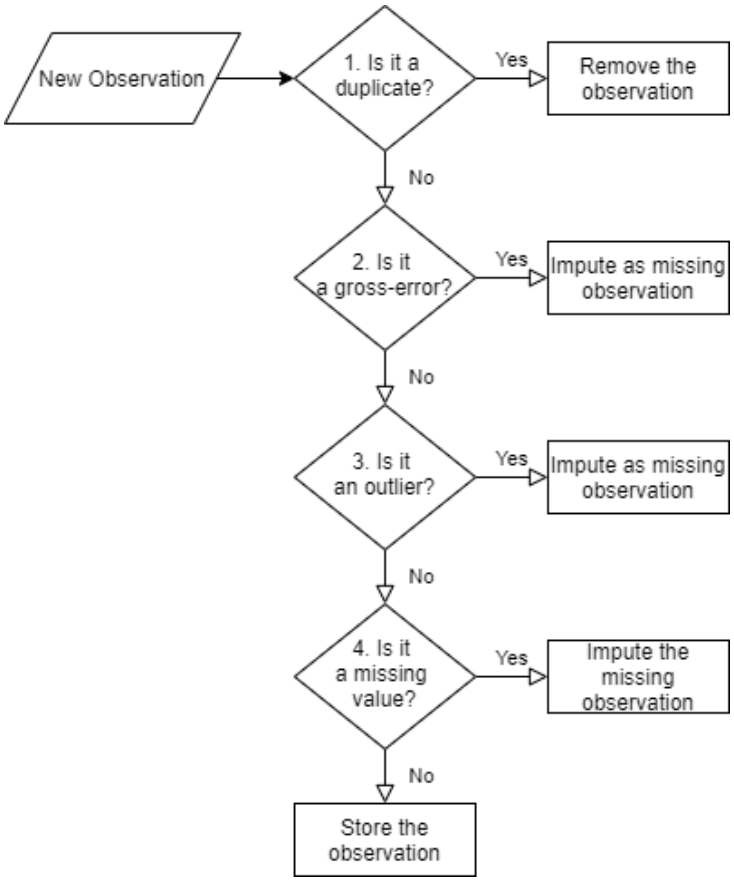

Figure 4.2: Pipeline of the real-time data pre-processing methodology.

4.2.2 Early processing steps

4.2.2.A Autonomous duplicates treatment *(step 1)*

The first step detects if the new observation is a duplicate. We check if the new observation as the same time point as the last observation stored. If it has we perform similarly as in section 4.1.2: i) remove the duplicate observations and insert a new observation as a missing value (with the same time stamp as the duplicates); ii) impute the missing values using a default imputation method available in AutoMTS; iii) the duplicate observation which has the smaller difference with the imputed missing value is selected

and the other is removed from the time series.

4.2.2.B Gross-errors treatment *step 2*

The second step detects if the new observation is a gross-error, by using the same range constraints in section 4.1.2.

4.2.3 Autonomous outliers detection *step 3*

The third step autonomously detects if the new observation is an outlier. The available methods are the same as section 4.1.3.A, but instead of doing the hyperparameterization explained in section 4.1.3.B and run the hyperparameterization for every new coming observation, we run the hyperparameterization with a pre-defined time period, e.g., once a day. Because of that we have a pre-defined outlier detection method and every time the hyperparameterization occurs, we select the best new method with all the observations stored in the system, including the new observations since the previous hyperparameterization.

For the regular detection the window and buffer are used and given to the method.

4.2.4 Missing values imputation *(step 4)*

The fourth step autonomously detects if the new observation is a missing value. The available methods are the same as section 4.1.4.A and the hyperparameterization is performed in the same way as in section 4.2.3.

For the regular imputation the window and buffer are used and given to the method, more specifically for methods that require further observations to the current one, e.g., moving average and NOCB. Although, the addiction of the buffer is mainly used for this methods, they might not be able to perform, because of the buffer size, or simply because the buffer are composed with missing observations.

4.2.5 Script

Although we have a graphical user interface in the historical setting, in the real-time data we created a script that mimics how a real-time system behaves. We have two different processes, one is returning a new observation with a constant time period (which mimics the behaviour of the sensors writing the values in the system) and the other is receiving that observation and is pre-processing it (which mimics the system). Each one of the previous steps is represented as a condition and are disposed sequentially, in order to detect the faults in the observation. When it would be necessary to implement the script in a real system, only the second process is required.

4.3 Computational complexity

Considering the presence of k_1 pre-processing methods, each with $O(T_i)$ complexity, then the complexity of executing them is $\sum_i^{k_1} O(T_i) = O(k_1 T_{\max})$. Assuming that the conducted Bayesian optimization per method converges in a bounded number of k_2 iterations for each method, then $O(k_1 k_2 T_{\max})$. Finally, considering the presence of k_3 testing settings in accordance with the detected error profiles in the original series (e.g. k_3=2 for missing and outlier segments with well-defined rate and length distributions), then our methodology has $O(k_1 k_2 k_3 T_{\max})$ complexity. k_1 and k_3 are constants. Given a window of bounded size w, the majority of pre-processing methods are linear on the window size, yielding $O(k_1 k_2 k_3 w)$.

5

Evaluation

Contents

This chapter is organized in four major steps. First, we describe the networks of heterogeneous sensors that will be used as study cases, exploring some of the produced time series. Second, we explain the metrics used for assessing outliers detection and missing imputation approaches. Third, we provide a thorough comparison of state-of-the-art methods to detect outliers and impute missings, showing that their adequacy is highly dependent on the time series regularities and error profiles. Finally, we evaluate our approach, quantifying its performance gains, in either historical and real-time settings. At the end, a description of the graphical interface is presented.

5.1 Study cases

As specified in section 1.2, the water utilities C.M. Barreiro, C.M. Beja and Infraquinta made available a temporal sample of the measurements made by their sensors to be used and studied. To use as study case a set of time series were randomly selected from pressure and flow sensors for the Barreiro and Beja WDNs. Figure 1.1 (left) depicts the water flow series from sensors located near the principal tanks in the Barreiro and Beja WDNs, while Figure 1.1 (right) depicts the time series produced by the approximately co-located water pressure sensors. In addition, sensors of the same type show considerably different regularities for different water distribution systems. These observations motivate the need to perform processing decisions separately for each sensor from the monitored systems. Moreover, since Beja has a irregular and low time granularity, re-sampling of the series was accomplished in order to the evaluation process be more efficient. For the water flow sensor, the re-sampling corresponds to the integral of the series and for the water pressure sensor we re-sample using the mean.

5.2 Evaluation metrics

Now that we have our time series ready to be evaluated we need to understand which metrics will be used. Outliers detection is a task where we need to know if the system is detecting correctly an observation. Taking this in consideration, let *TP* (true positives) be the correctly detected outliers, *TN* (true negatives) be observations correctly identified as non-outliers, *FP* (false positives) be the incorrectly detected outliers, and *FN* (false negatives) be the non-detected outliers wrongly. To evaluate the behavior of outlier detection methods, we suggest as essential performance views the analysis of recall,

$$\text{recall} = \frac{TP}{TP + FN},$$

to understand the percentage of correctly identified outliers, as well as precision,

$$\text{precision} = \frac{TP}{TP + FP},$$

to understand whether the retrieved outliers were identified at the cost of retrieving non-outlier observations (false positives). To objectively guide the hyperparameterization and selection steps, these complementary views can be combined within scores, such as the F1-score,

$$\text{F1-score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}},$$

which is not free of criticisms Davis and Goadrich [43] due to the inherent characteristics of the harmonic mean.

On the other hand the missing imputation task is not about knowing how many observations we are imputing correctly, but how distant our methods guess is from the real value. As such, residue-based scores are considered, including the mean absolute error (MAE),

$$\text{MAE} = \Sigma_{i=1}^{n} |\hat{\mathbf{x}}_{t_i} - \mathbf{x}_{t_i}|,$$

where $\mathbf{x}$ and $\hat{\mathbf{x}}$ are the observed and imputed time series respectively, and n is the number of missings; the root mean squared error (RMSE),

$$\text{RMSE} = \sqrt{\sum_{i=1}^{n} \frac{(\hat{\mathbf{x}}_{t_i} - \mathbf{x}_{t_i})^2}{n}},$$

the symmetric mean absolute percentage error (SMAPE); and the percentage of missing values imputed since not all imputation methods may not encounter necessary conditions for imputing certain missing observations.

Also the percentage of correctly imputed missing is used in order to evaluate the behaviour of our methodology, which is represented by % in the coming tables.

5.3 Experimental setting

To assess the impact of placing appropriate choices along the processing stages in accordance with the characteristics and inconsistencies observed along time series, we consider the water flow and pressure time series from Barreiro and Beja WDNs and applied the proposed methodology to generate ground truth. To facilitate the interpretability of results, we further varied the profile of the planted inconsistencies for some of the conducted- analyzes. The major parameters controlling the experimental setting are:

- available methods for point outlier detection (e.g. isolation forests) and sequential outlier detection (e.g. SAX), and the corresponding parameters;

- planted outlier profiles, including: i) frequency of outliers (2% and 10%); ii) type of outliers (point versus sequential); and iii) length of sequential outliers;

- available methods for missing imputation from univariate series (e.g. moving average) or multivariate series (e.g. Random forests), and corresponding parameters;
- planted missing profiles, including: i) frequency of missing values (from 2% and 10%); ii) type of missings (point versus sequential); and iii) length of sequential missing observations.

The presented results provide the average performance collected from 30 simulations. A stochastic process to generate inconsistencies in accordance with the introduced parameters is used to produce each simulation. Random seeds are considered to guarantee fair comparisons between methods.

5.4 AutoMTS performance in historical setting

5.4.1 Outliers detection

Tables A.1 and A.3 provide a comprehensive analysis of the performance of the multiple outlier detection methods on time series data produced from one water flow and one water pressure sensor, installed within the Barreiro and Beja WDNs.

Considering F1-score as a reference metric for the outliers detection task, we can say that the inter quartile range and isolation forests are the best performing methods in Barreiro WDN. Even more, the isolation forests is dominant in the water pressure sensor, whereas, the inter quartile range is better for the water flow sensors. This is also true for either 2% or 10% of planted anomalies. Following the isolation forests for the water pressure sensors, local outlier factor also produce good results for the 2% of artificial anomalies, but are heavily surpassed by the inter quartile range for 10% of errors. Following the inter quartile range for the water flow sensors is the standard deviation for 2% and the isolation forests for the 10% of induced errors. Also important to refer is that the methods perform worst with a lower quantity of artificial outliers, i.e., some of the best performing methods can not get better results than 33%, while for a higher quantity of artificial outliers the best methods have a great performance with the lower value for the best performing method being 85.4%. Even more, the DBScan method obtains only favorable results for water flow sensors, while for water pressure sensors is the worst performing method with 0% of F1-score.

For Beja WDN, considering 10% of planted outliers, results are similar too the Barreiro's, where the inter quartile range is the best performing method for water flow sensors and the isolation forests for water pressure sensors. For 2% of artificial errors, the results quite different. For water pressure point-wise, the HOT SAX method have a nice performance, as opposed to what we expected. Standard deviation is dominant in the water flow sensors. The DBScan and local outlier factor have limited performance for all water pressure sensors. For the water flow sensors the local outlier factor have results also close to 0%.

Overall, from all the available methods the standard deviation, inter quartile range, isolation forests and HOT SAX are the four best performing methods for the outliers detection task. Complementary, Figure 5.1 offers a graphical description of previous results for the Barreiro WDN, further showing how the performance of different outlier detection methods vary with the amount of planted outliers. HOT SAX is not competitive when considering a low amount of outliers, yet performance improves with a medium-to-high amount of outliers. The analysis of these figures further highlights that there are significant changes in performance associated with changes on the amount of outlier values. These variations can affect processing decisions (e.g. isolation forests versus inter quartile range in pressure sensors), further supporting the relevance of the proposed methodology.

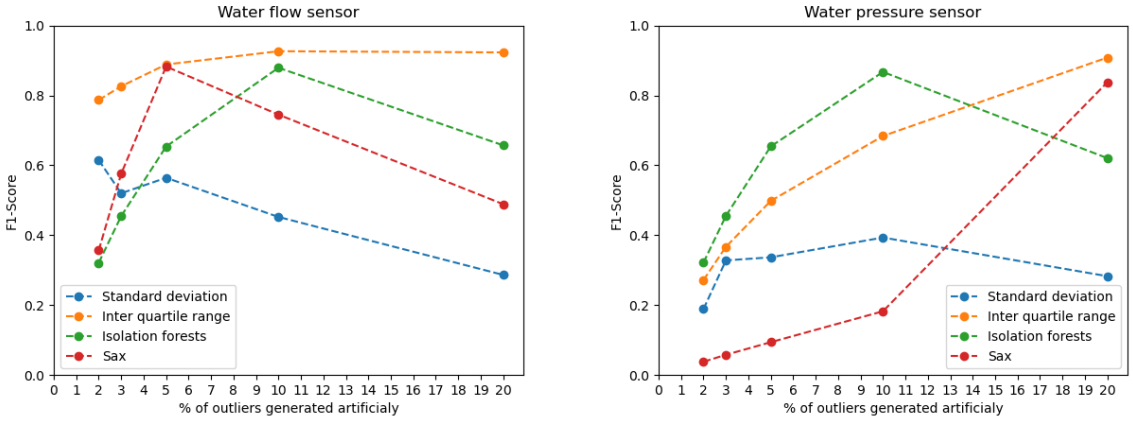

Figure 5.1: Performance of outlier detection methods with varying percentage of planted point outliers in time series produced from heterogeneous sensors in Barreiro WDN.

5.4.2 Missing values imputation

Tables A.2 and A.4 provide a comprehensive analysis of the performance of the multiple missing values imputation methods on time series data produced from one water flow and one water pressure sensor, installed within the Barreiro and Beja WDNs.

Considering RMSE as a reference metric for the missing values imputation task for the Barreiro WDN we can say that for 2% of artificial generated missings univariate imputation methods such as interpolation, LOCF, moving average, etc, have better performance than the multivariate methods. Moreover, moving average have one of the best performances for water pressure sensors, can not secure the total imputation for sequential missing values. Interpolation is the best performing method for water flow sensors. To our astonishment the median, which is usually a non-competitive method, is the best performing method for the water pressure sensor with sequential missings, with a close call from the mean method. With 10% of planted missings moving average is the undisputed method for water pressure sensors

with point-wise missings, with an almost perfect percentage of imputed missings and for the sequential once again the median and mean methods are best performing methods. For the water flow sensors, interpolation is the best performing method point-wise followed by the LOCF and NOCB, and moving average is the method with best RMSE for sequential, but with a poorly percentage of imputed values. As a result, we can see that mean and median are again the best performing method with a close call from multivariate methods such random forests and kNN.

For the Beja WDN, interpolation is by far the best method for point-wise for either 2% or 10% of artificial missings. At first glance, the moving-average is the best method for sequential planted missings, but with a poor imputation percentage, giving the lead to other methods such as interpolation. For water pressure sensor with sequential missings, the mean and median are again best performing methods.

Overall we can say that interpolation, LOCF and NOCB are usually the best methods or close to the best one, but we can observe a decreased performance of single-value methods such as LOCF and NOCB for imputing missings segments and in increased performance of multi-points such as moving average (nonetheless with lower imputed percentage). To further complementary evaluation figure 5.2 offer a complementary graphical description of previous results for the Beja WDN. Generally, the higher the amount of missing observations, the higher the imputation difficulty. These figures highlight the presence of significant performance differences related with the amount of missing observations, further suggesting the relevance of understanding the missing profiles when placing pre-processing decisions. For instance, while random forests is generally a non-competitive method for a small amount of missings, it is the suggested option to impute high amounts of missing observations in water pressure series. This last remark further pinpoints the relevance of considering cross-variable dependencies.

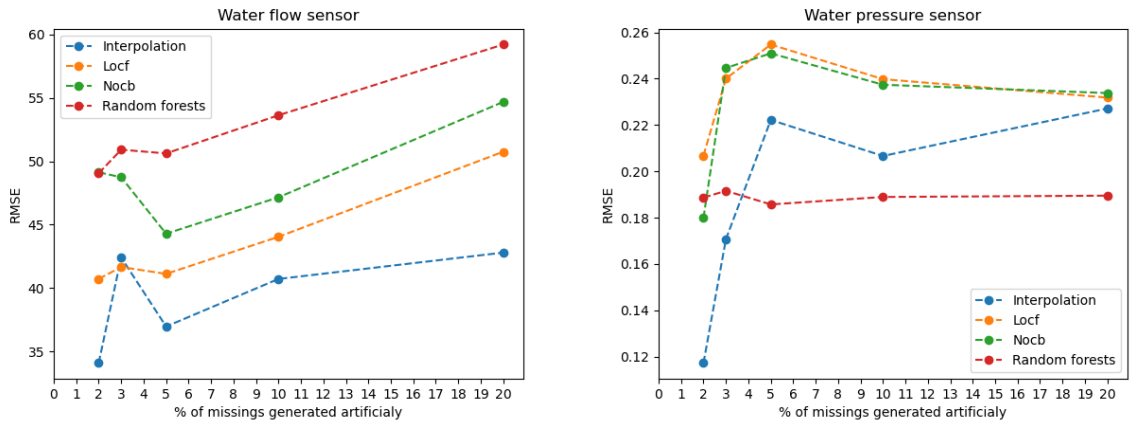

Figure 5.2: Performance of missing imputation methods with varying percentage of sequential missings planted in time series from heterogeneous sensors in Beja WDN.

5.5 AutoMTS performance in real-time setting

In this setting the behaviour of the methods may vary compared to the historical setting, because we have access to a more restrict window of the observations. This window size for our evaluation methodology had a size of 200 and a buffer size of 6.

5.5.1 Outliers detection

Tables B.1 and B.3 provide a comprehensive analysis of the performance of the multiple outlier detection methods on time series data produced from one water flow and one water pressure sensor, installed within the Barreiro and Beja WDNs, for the real time setting. The ΔT column in the tables represent the average time in seconds that each method takes to detect a single observation.

For Barreiro WDNs, inter quartile range is constantly the best method for either the 2% and 10% of planted missings, exceptionally for the water flow sensors with segments of outliers and for water pressure sensor with 2% of segments. With a performance close to the inter quartile range is the standard deviation method, mostly for the 2% of outlier planted and for the 10% of outliers usually the isolation forests is the second best performing method. DBScan is inconsistent, with a considerably limited behaviour for water flow sensors and, but with some positive notes on the water pressure. The HOT SAX method acts poorly under low percentage of outliers, yet its performance significantly increases for a greater number of outliers.

For Beja WDN, isolation forests is every single time the best performing method. For the 2% of planted outliers the local outlier factor is right behind, with an overall similar performance, but usually 10x faster than the previous. The inter quartile range has his worst performance so far with similar results as the DBScan. For the 10% setting the HOT SAX is the second best performing method, again with a faster processing time than the isolation forests. Here the inter quartile range got better results similar to the local outlier factor. Standard deviation behaves poorly overall for the Beja WDN.

Overall we can say that isolation forests, inter quartile range, HOT SAX and standard deviation are the most promising methods, although the local outlier factor is close to some of this methods. All methods have an outlier detection under 1 second, but the isolation forests is the one that takes the most time to detect if the given observation is an outlier with an approximate time of 0.27 seconds. To further complementary evaluation Figure 5.3 offers a complementary graphical description of previous results for the Barreiro WDN. Once again we can see that HOT SAX is not competitive when considering a low amount of outliers, yet the performance improves with medium-to-high amount of outliers. We can also highlight that the increasing of outliers also improve the performance of the tested methods and once again the specificity's of a time series affect which one is the better method.

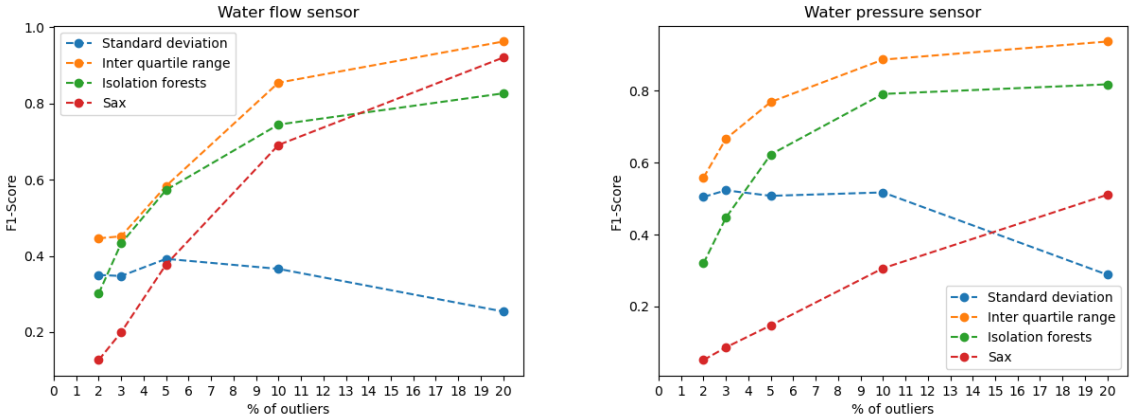

Figure 5.3: Performance of outlier detection methods with varying percentage of point outliers planted in time series from heterogeneous sensors in Barreiro WDN.

5.5.2 Missing values imputation

Tables B.2 and B.4 provide a comprehensive analysis of the performance of the multiple missing value imputation methods on time series data produced from one water flow and one water pressure sensor, installed within the Barreiro and Beja WDNs, for the real time setting. Again the Δ T column in the tables represents the average time in seconds that each method takes to detect a single observation.

For the Barreiro WDN, the water pressure sensors are dominated by the mean and median methods. Although most of the time these methods are discarded, as we can see they are the best performance methods here. These two methods are usually closed followed by the random forests and moving average methods. In the other hand, the water flow sensors are dominated by the interpolation method which has a huge lead compared with the second best performing method (e.g. LOCF or NOCB). The moving average method usually gets good results in terms of RMSE, just like the NOCB, but the latter when used for 10% of sequential planted missings have a low percentage of imputed observations. The random forests and the Knn are the best performing methods for the multivariate, with similar results to the second candidate method.

For the Beja WDN, the interpolation is the best candidate getting the best results for the water pressure sensors with 2% of planted errors and for sequential point-wise with 10% of missings. Random forests and Knn are again very competitive with the former being the best method for the water flow sensor with 2% of sequential missings. Again the NOCB method have a great performance, mainly for the 10% of sequential planted observations, but again for this specific setting it get lower percentages of imputed values.

Overall we can say that the interpolation, random forests, NOCB and mean methods are the best performing methods, with less than 1 second to impute one observation, besides the random forests which have an average time of 2 seconds. Figure 5.4 offers a complementary graphical description

of previous results for the Beja WDN. As expected, the performance of the methods decrease when there is a higher percentage of missing values in the data, although the mean method maintains an approximately constant behaviour. It is also important to see that the point-wise methods like LOCF and interpolation tend to get worse with an increasing amount of missings while the multivariate tend to stabilize.

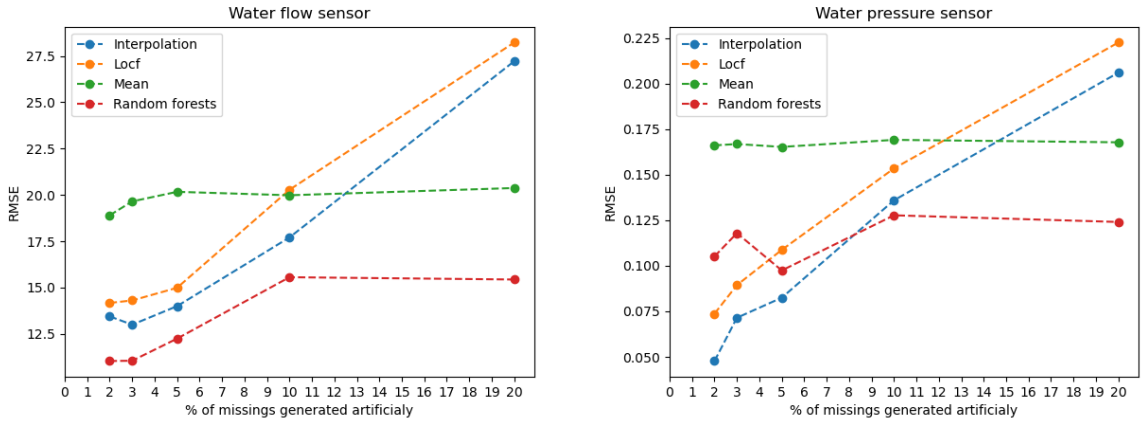

Figure 5.4: Performance of missing imputation methods with varying percentage of point missings planted in time series from heterogeneous sensors in Beja WDN.

6

Software tool

Contents

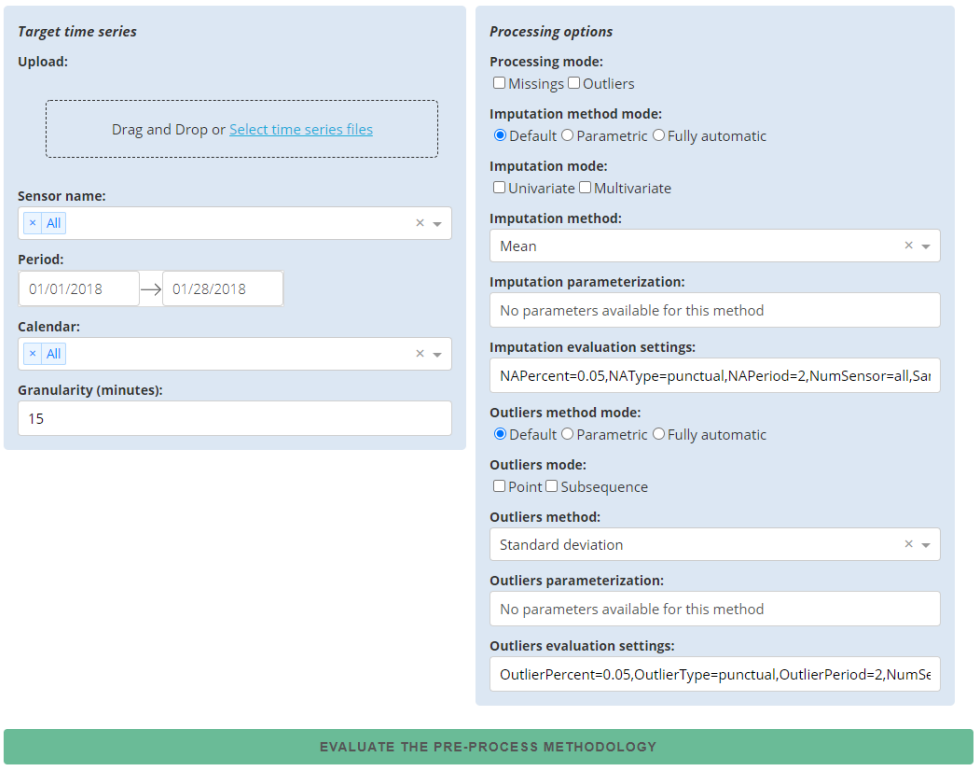

Figure 6.1: Graphical user interface.

The introduced methodology for the historical data contexts processing of time series data was programmed using Python 3.7.3 and is made available within a software tool provided via GitHub in `https://github.com/RicardoFLNSousa/AutoMTS/tree/master`. The tool supports a graphical user interface, where the user can change multiple settings. Figure 6.1 provides a snapshot of the available tool, where it can be divided into two main panels: on the left the *Target time series* panel and on the right the *Processing options* panel.

6.1 Target time series

The *Target time series* panel is used to upload the file desired to be pre-processed and to apply constraints on it. The following bullets are the available functionalities in this panel:

- **Upload:** It is possible to upload the file which contains the time series dataset. Different file formats are supported, including .xlsx and .csv. The data representation must be the same as figure 6.2, where the first row can be used as file name, the second row indicates the name of each sensor (which is a time series), the third row specify the range constraints for the gross errors (if not designated, the sensor has no constraints), and the rest of the rows are the observations, with the individual time stamps and their values for each sensor.

	Nível C1 Máx.	Nível C1 Méd.	Nível C1 Mín.	Nível C2 Máx.	Nível C2 Méd.	Nível C2 Mín.
Limits	0-100	0-?				
01/01/2018	3,1	2,96	2,8	3,2	2,91	2,8
01/01/2018 01:00:00	3,3	3,23	3,1	3,3	3,09	3,0
01/01/2018 02:00:00	3,6	3,49	3,4	3,7	3,38	3,2
01/01/2018 03:00:00	3,9	3,77	3,6	3,9	3,61	3,5
01/01/2018 04:00:00	4,2	4,06	3,9	4,1	3,90	3,8
01/01/2018 05:00:00	4,5	4,34	4,2	4,3	4,21	4,1
01/01/2018 06:00:00	4,6	4,57	4,5	4,6	4,47	4,3
01/01/2018 07:00:00	4,6	4,56	4,5	4,5	4,52	4,5
01/01/2018 08:00:00	4,5	4,51	4,5	4,5	4,46	4,4
01/01/2018 09:00:00	4,5	4,44	4,4	4,5	4,40	4,3
01/01/2018 10:00:00	4,4	4,35	4,3	4,3	4,29	4,2

Figure 6.2: Representation of the desired file structure.

Even so, during the upload the step 1 and 2 of the pipeline in section 4.1.1 are performed. First, duplicated values are detected and removed from the series. Second, the using the second row of the file, we execute the removal of observations out of the limits constraints. This observations will be considered from now on, as missing values, to be deal later in the process. To guarantee that ground truth is assessed over the provided series data, each sensor needs to have at least one period of four weeks without missing observations, or the file is not considered valid, and therefore, can not be used.

- **Sensor name:** Once the uploaded dataset proceed through the initial validation process, it is possible to filter the dataset by selecting the time series (sensors) that we want to process. This can be done using *sensor name* field, where a list with all the sensors name is given to the user and he can select which he wants to pre-process.

- **Calendar:** It is possible to further filter the observations by time period on the *period* field (e.g.weekdays, holidays, Saturdays). This could be useful if the user require a specific constraint over the week, e.g., want only a final time series with the weekends.

- **Granularity:** Finally it is possible to filter by time granularity for the target time series in the *granularity* field. The user have to take into account that only under-sampling is performed, i.e., we can only decrease the granularity of the time series, e.g., from 1 hour time period to 15 minutes.

6.2 Processing options

On the *Processing options* panel is possible to select the options we want to achieve the step 3 and 4 of the pipeline in section 4.1.1, i.e., the outlier detection and missing values imputation. Along this process, we use as dataset the largest segment without missing values, as explained in section 4.1.1. The following bullets are the available functionalities in this panel:

- **Processing mode:** Here it is possible to select if we want to conduct missing imputation and/or outlier detection. If none is selected, both the options are performed.
- **Imputation/Outliers method mode:** For the option selected in the previous bullet, it is possible to select one of three distinct modes: i) the *default* mode which provides a pre-defined method (the usually best performing method in the chapter 5); ii) the *parametric* mode which allows the user to select a desirable method method and its parameters; and, at last, iii) the *fully automatic* mode which performs the hyperparameterization to identify the best method for each one of the sensors selected in the *Target time series* panel.
- **Imputation/Outliers mode:** Here the user can filter the available methods. For example, if the user select univariate for the imputation mode, only univariate methods will be available.
- **Imputation/Outliers evaluation settings:** The user can optionally specify the profile of the artificially planted missing values and outlier values to be considered along the evaluation stage, as well as to provide statistics whenever the user opts to select default and parametric modes. Here the user can select the type, percentage and duration of artificial missings and outliers. It is also possible to select the number of sensors on where we want to plant the artificial inconsistencies. Finally, the user can also specify whether the inconsistencies must occur at the same time for the imputed set of sensors or planted for each sensor individually, thus mimicking different real-world problems in heterogeneous networks.

When the user has all the options defined, he can click in the button *"Evaluate the pre-process methodology"* to run the tool. Immediately upon it finish's to run, all the processed series will be plotted for the *processing mode* defined in the previous steps, as example in figure 6.3, where the user can use interactive zooming and filtering facilities on the displayed series, and access a generated report with the results of the evaluation for each selected sensor with a similar format as the ones presented.

Specifically for the outliers detection method the selection methodology of figure 6.4 is presented to the user where he can select which detected outliers will be removed. If the option "all" is selected, all detected outliers are removed. After that the user press the button *"Pre-process the given file"*, where the outliers are removed and the data is fully cleaned. At last, the link to download the new cleaned file is generated.

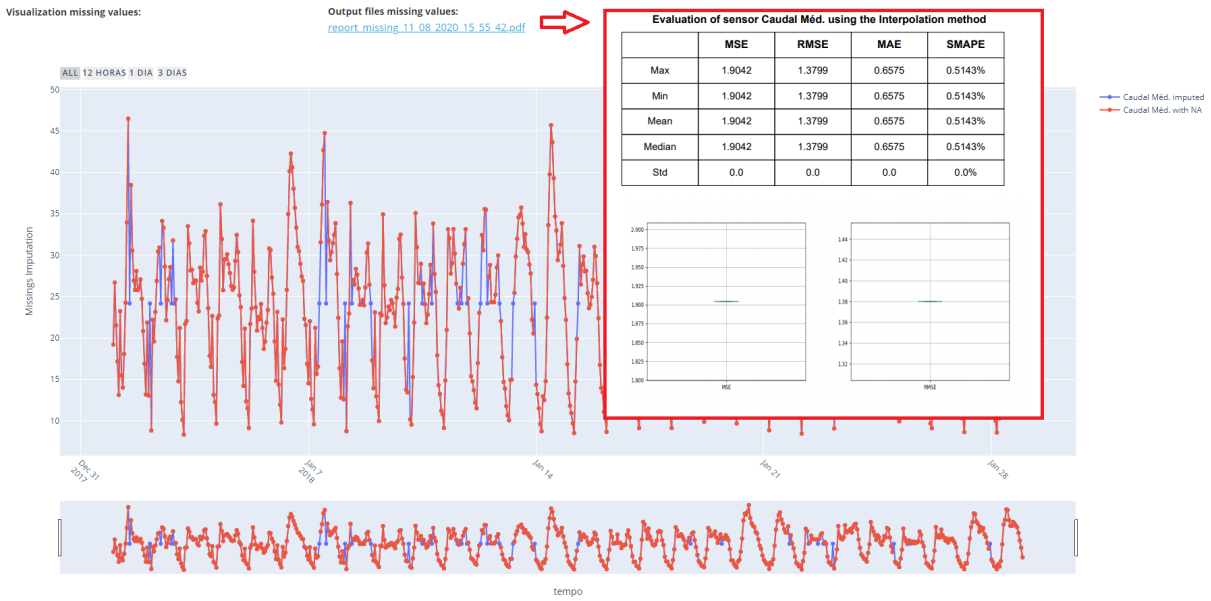

Figure 6.3: Output overview.

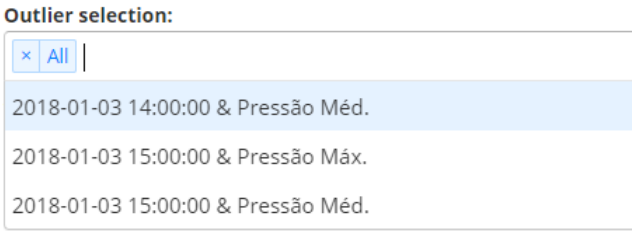

Figure 6.4: Selection of the outliers to remove.

7

Conclusion

Contents

In this chapter the main conclusions are drawn, along some notes for the future work.

7.1 Summary

In this thesis we were able to complete our objectives and produce two approaches for the fully-autonomous and quality-driven processing of time series data produced by networks of heterogeneous sensors. Each one is parameter-free and offers guarantees of optimality. To optimize pre-processing choices, ground truth is created from conserved time series segments for possible error profiles. In addition, AutoMTS provide a coverage of state-of-the-art methods of outliers detection and missing values imputation. AutoMTS implements processing methods able to work with point-wise or sequence observations, and with cross-variable dependencies in the presence of multivariate time series data. Also, our methodology can work with varying types and amount of missing and outliers values, including both point and sequential occurrences of different duration and recurrence.

The experimental evaluation in two real world study cases of water distribution network systems with different sampling rates, water consumption patterns and error profiles confirm the significance of AutoMTS contributions and highlight that pre-processing choices are highly specific to each sensor. Thus guarantees of optimality can only be provided under a robust evaluation. Also the results further offer a comparison of the available methods, showing the strengths and limitations when handling multiple error profiles in real-world time series.

Also the time that takes to pre-process in the real-time setting and its results show that our approach is applicable for the treatment of time series in data streams.

Our approach also enables and facilitate the incorporation and/or implementation of other methods, with almost no effort.

At last, this work provides a functional tool for the WISDOM project, that would help the detection of events of interest and acquire a comprehensive view of the behaviour produced by their time series data.

7.2 Future work

Even though the thesis was completed with success, some complementary extensions could be implemented in order to improve our approach. For the imputation methods, some specific time series methods could be added, e.g., the 10-Min flow model from Quevedo, as long some variations of the moving average method which could deal better with sequential missings.

Moreover, the implementation of our real-time script in a real system could provide a better insight on the behaviour of our methodology, practical validation with the water entities would be performed and

evaluation of the pre-processing in the detection leakage and on consumption patterns. Finally, further evaluation on irregular time series sampling can be supported.

7.3 Scientific communication

During the development of this thesis the article Sousa et al. [44] was published and presented in the EAI Qshine 2020 - 16th EAI International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness. This article only depicted the historical data setting of the AutoMTS tool. The paper that extend the contributions for the real time setting is under submission.

Bibliography

[1] Phillipa Gill, Navendu Jain, and Nachiappan Nagappan. Understanding network failures in data centers: Measurement, analysis, and implications. In *Proceedings of the ACM SIGCOMM 2011 Conference*, SIGCOMM '11, page 350–361, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450307970. URL https://doi.org/10.1145/2018436.2018477.

[2] Malcolm Farley, Sanitation World Health Organization. Water, Health Team, Water Supply, and Sanitation Collaborative Council. Leakage management and control : a best practice training manual / malcolm farley, 2001.

[3] J. Lin, E. Keogh, S. Lonardi, and B. Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, DMKD '03, pages 2–11, New York, NY, USA, 2003. ACM. URL http://doi.acm.org/10.1145/882082.882086.

[4] Rob J. Hyndman, George Athanasopoulos, and OTexts.com. *Forecasting : principles and practice / Rob J Hyndman and George Athanasopoulos*. OTexts.com [Heathmont?, Victoria], print edition. edition, 2014. ISBN 9780987507105.

[5] Min Chen, Shiwen Mao, and Yunhao Liu. Big data: A survey. *Mob. Netw. Appl.*, 19(2):171–209, April 2014. ISSN 1383-469X. URL http://dx.doi.org/10.1007/s11036-013-0489-0.

[6] Nader Mohamed and Jameela Al-Jaroodi. Real-time big data analytics: Applications and challenges. 07 2014.

[7] K. G. Shin and P. Ramanathan. Real-time computing: a new discipline of computer science and engineering. *Proceedings of the IEEE*, 82(1):6–24, Jan 1994. ISSN 1558-2256.

[8] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han. Outlier detection for temporal data: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 26(9):2250–2267, Sep. 2014. ISSN 2326-3865.

[9] Alex J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and*

Computing, 14(3):199–222, August 2004. ISSN 0960-3174. URL https://doi.org/10.1023/B:STCO.0000035301.49549.88.

[10] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., New York, NY, USA, 1995. ISBN 0198538642.

[11] Ana Justel, Daniel Peña, and Ruey Tsay. Detection of outlier patches in autoregressive time series. *Statistica Sinica*, 11:651–673, 07 2001.

[12] Chung Chen and Lon-Mu Liu. Joint estimation of model parameters and outlier effects in time series. *Journal of the American Statistical Association*, 88(421):284–297, 1993. URL https://doi.org/10.1080/01621459.1993.10594321.

[13] H. V. Jagadish, Nick Koudas, and S. Muthukrishnan. Mining deviants in a time series database. In *Proceedings of the 25th International Conference on Very Large Data Bases*, VLDB '99, pages 102–113, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. ISBN 1-55860-615-7. URL http://dl.acm.org/citation.cfm?id=645925.758373.

[14] Eamonn Keogh, Jessica Lin, Sang-Hee Lee, and Helga Van Herle. Finding the most unusual time series subsequence: algorithms and applications. *Knowledge and Information Systems*, 11(1): 1–27, Jan 2007. ISSN 0219-3116. URL https://doi.org/10.1007/s10115-006-0034-6.

[15] Xiao-Yun Chen and Yan-Yan Zhan. Multi-scale anomaly detection algorithm based on infrequent pattern of time series. *Journal of Computational and Applied Mathematics*, 214(1):227 – 237, 2008. ISSN 0377-0427. URL http://www.sciencedirect.com/science/article/pii/S0377042707001100.

[16] Kenji Yamanishi and Jun ichi Takeuchi. A unifying framework for detecting outliers and change points from non-stationary time series data. In *Proc. of the Eighth ACM SIGKDD, ACM*, pages 676–681. Press, 2002.

[17] Kenji Yamanishi, Jun-ichi Takeuchi, Graham Williams, and Peter Milne. On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. *Data Mining and Knowledge Discovery*, 8(3):275–300, May 2004. ISSN 1573-756X. URL https://doi.org/10.1023/B:DAMI.0000023676.72185.7c.

[18] Di Yang, Elke A. Rundensteiner, and Matthew O. Ward. Neighbor-based pattern detection for windows over streaming data. In *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, EDBT '09, pages 529–540, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-422-5. URL http://doi.acm.org/10.1145/1516360.1516422.

[19] J. Zhang, Q. Gao, and H. Wang. Spot: A system for detecting projected outliers from high-dimensional data streams. In *2008 IEEE 24th International Conference on Data Engineering*, pages 1628–1631, April 2008.

[20] Steffen Moritz, Alexis Sardá-Espinosa, Thomas Bartz-Beielstein, Martin Zaefferer, and Jörg Stork. Comparison of different methods for univariate time series imputation in R. 10 2015.

[21] Achim Zeileis and Gabor Grothendieck. zoo: S3 infrastructure for regular and irregular time series. *Journal of Statistical Software, Articles*, 14(6):1–27, 2005. ISSN 1548-7660. URL https://www.jstatsoft.org/v014/i06.

[22] Rob J. Hyndman, George Athanasopoulos, Christoph Bergmeir, Gabriel Caceres, Leanne Chhay, Mitchell O'Hara-Wild, Fotios Petropoulos, Slava Razbash, Earo Wang, and Farah Yasmeen. forecast: Forecasting functions for time series and linear models, 4 2018.

[23] Alexander Kowarik and Matthias Templ. Imputation with the r package vim. *Journal of Statistical Software, Articles*, 74(7):1–16, 2016. ISSN 1548-7660. URL https://www.jstatsoft.org/v074/i07.

[24] M. S. Osman, A. M. Abu-Mahfouz, and P. R. Page. A survey on data imputation techniques: Water distribution system as a use case. *IEEE Access*, 6:63279–63291, 2018. ISSN 2169-3536.

[25] James Heckman. Sample selection bias as a specification error. *Applied Econometrics*, 31(3): 129–137, 2013. URL https://ideas.repec.org/a/ris/apltrx/0220.html.

[26] Alexander Ilin and Tapani Raiko. Practical approaches to principal component analysis in the presence of missing values. *J. Mach. Learn. Res.*, 11:1957–2000, August 2010. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1756006.1859917.

[27] L. Garg, J. Dauwels, A. Earnest, and K. P. Leong. Tensor-based methods for handling missing data in quality-of-life questionnaires. *IEEE Journal of Biomedical and Health Informatics*, 18(5): 1571–1580, Sep. 2014. ISSN 2168-2208.

[28] J. Quevedo, V. Puig, G. Cembrano, J. Blanch, J. Aguilar, D. Saporta, G. Benito, M. Hedo, and A. Molina. Validation and reconstruction of flow meter data in the barcelona water distribution network. *Control Engineering Practice*, 18(6):640 – 651, 2010. ISSN 0967-0661. URL http://www.sciencedirect.com/science/article/pii/S0967066110000791.

[29] Rui Barrela, Conceição Amado, Dália Loureiro, and Aisha Mamade. Data reconstruction of flow time series in water distribution systems – a new method that accommodates multiple seasonality. *Journal of Hydroinformatics*, 19(2):238–250, 12 2016. ISSN 1464-7141. URL https://doi.org/10.2166/hydro.2016.192.

[30] S. Bennis, F. Berrada, and N. Kang. Improving single-variable and multivariable techniques for estimating missing hydrological data. *Journal of Hydrology*, 191(1):87 – 105, 1997. ISSN 0022-1694. URL http://www.sciencedirect.com/science/article/pii/S0022169496030764.

[31] F. Fan, Z. Li, and Y. Wang. On-line imputation for missing values. In *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pages 1–5, Oct 2017.

[32] Fei Tony Liu, Kai Ming Ting, and Zhi hua Zhou. Isolation forest. In *In ICDM '08: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining. IEEE Computer Society*, pages 413–422, 2008.

[33] Markus Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. Lof: Identifying density-based local outliers. In *PROCEEDINGS OF THE 2000 ACM SIGMOD INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA*, pages 93–104. ACM, 2000.

[34] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press, 1996.

[35] Jessica Lin, Eamonn Keogh, Li Wei, and Stefano Lonardi. Experiencing sax: A novel symbolic representation of time series. *Data Min. Knowl. Discov.*, 15:107–144, 08 2007.

[36] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13(null):281–305, February 2012. ISSN 1532-4435.

[37] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2951–2959. Curran Associates, Inc., 2012. URL http://papers.nips.cc/paper/4522-practical-bayesian-optimization-of-machine-learning-algorithms.pdf.

[38] Daniel J. Stekhoven and Peter Bühlmann. MissForest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118, 10 2011. ISSN 1367-4803. URL https://doi.org/10.1093/bioinformatics/btr597.

[39] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001. ISSN 1573-0565. URL https://doi.org/10.1023/A:1010933404324.

[40] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society, series B*, 39(1):1–38, 1977.

[41] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman. Missing value estimation methods for DNA microarrays . *Bioinformatics*, 17(6):520–525, 06 2001. ISSN 1367-4803.

[42] Stef van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in r. *Journal of Statistical Software, Articles*, 45(3):1–67, 2011. ISSN 1548-7660. URL https://www.jstatsoft.org/v045/i03.

[43] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240, 2006.

[44] Ricardo Sousa, Conceição Amado, and Rui Miguel Carrasqueiro Henriques. Automts: fully autonomous processing of multivariate time series data from heterogeneous sensor networks. In *16th EAI Int. Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine)*, October 2020.

A

Historical data results

		Barreiro WDN				Beja WDN			
		F1-score	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall
pressure: point	Standard deviation	0.189±0.11	0.976±0.00	0.149±0.09	**0.262±0.13**	0.132±0.06	0.982±0.00	0.072±0.03	**1.0±0.00**
	Inter quartile range	0.272±0.00	0.896±0.00	**1.0±0.00**	0.157±0.00	0.058±0.04	0.981±0.00	0.031±0.02	0.767±0.42
	Isolation forests	**0.322±0.00**	0.918±0.00	**1.0±0.00**	0.192±0.00	0.337±0.01	0.922±0.00	**1.0±0.00**	0.202±0.00
	Local outlier factor	**0.322±0.00**	0.918±0.00	**1.0±0.00**	0.192±0.00	0.0±0.00	0.881±0.00	0.0±0.00	0.0±0.00
	Dbscan	0.0±0.00	0.979±0.00	0.0±0.00	0.0±0.00	0.0±0.00	0.98±0.00	0.0±0.00	0.0±0.00
	SAX	0.038±0.00	0.023±0.00	**1.0±0.00**	0.019±0.00	**0.684±0.29**	0.941±0.11	**1.0±0.00**	0.582±0.29
	Tool	0.322±0.00	0.918±0.00	1.0±0.00	0.192±0.00	0.684±0.29	0.941±0.11	1.0±0.00	0.582±0.29
flow: point	Standard deviation	0.615±0.11	0.989±0.00	0.454±0.12	**1.0±0.00**	**0.779±0.03**	0.991±0.00	0.772±0.04	**0.788±0.02**
	Inter quartile range	**0.787±0.10**	0.993±0.00	0.659±0.13	**1.0±0.00**	0.445±0.01	0.95±0.00	**1.0±0.00**	0.287±0.01
	Isolation forests	0.321±0.00	0.918±0.00	**1.0±0.00**	0.191±0.00	0.332±0.00	0.92±0.00	**1.0±0.00**	0.199±0.00
	Local outlier factor	0.309±0.02	0.917±0.00	0.964±0.07	0.184±0.01	0.089±0.02	0.891±0.00	0.269±0.07	0.054±0.01
	Dbscan	0.602±0.06	0.976±0.00	0.959±0.12	0.439±0.04	0.415±0.02	0.949±0.00	0.912±0.06	0.269±0.01
	SAX	0.357±0.12	0.916±0.05	**1.0±0.00**	0.224±0.09	0.504±0.08	0.959±0.01	**1.0±0.00**	0.341±0.07
	Tool	0.787±0.10	0.993±0.00	0.659±0.13	1.0±0.00	0.779±0.03	0.991±0.00	0.772±0.04	0.788±0.02
pressure: segment	Standard deviation	0.141±0.12	0.977±0.00	0.114±0.10	**0.189±0.15**	0.136±0.06	**0.981±0.00**	0.074±0.03	**0.967±0.18**
	Inter quartile range	0.253±0.00	0.895±0.00	**1.0±0.00**	0.145±0.00	0.054±0.05	0.98±0.00	0.028±0.03	0.733±0.44
	Isolation forests	**0.301±0.00**	0.917±0.00	**1.0±0.00**	0.177±0.00	**0.341±0.01**	0.922±0.00	**1.0±0.00**	0.205±0.00
	Local outlier factor	0.3±0.00	0.917±0.00	**1.0±0.00**	0.177±0.00	0.0±0.00	0.881±0.00	0.0±0.00	0.0±0.00
	Dbscan	0.0±0.00	0.981±0.00	0.0±0.00	0.0±0.00	0.0±0.00	0.98±0.00	0.0±0.00	0.0±0.00
	SAX	0.035±0.00	0.021±0.00	**1.0±0.00**	0.018±0.00	0.64±0.31	0.925±0.13	0.976±0.13	0.561±0.31
	Tool	0.301±0.00	0.917±0.00	1.0±0.00	0.177±0.00	0.341±0.01	0.922±0.00	1.0±0.00	0.205±0.00
flow: segment	Standard deviation	0.653±0.11	0.991±0.00	0.494±0.12	**1.0±0.00**	**0.787±0.03**	0.992±0.00	0.778±0.05	**0.799±0.02**
	Inter quartile range	**0.827±0.07**	0.995±0.00	0.711±0.10	**1.0±0.00**	0.441±0.03	0.948±0.01	**1.0±0.00**	0.283±0.02
	Isolation forests	0.3±0.00	0.917±0.00	**1.0±0.00**	0.177±0.00	0.336±0.00	0.92±0.00	**1.0±0.00**	0.202±0.00
	Local outlier factor	0.294±0.01	0.916±0.00	0.981±0.04	0.173±0.01	0.093±0.03	0.891±0.00	0.277±0.10	0.056±0.02
	Dbscan	0.597±0.06	0.976±0.00	0.983±0.09	0.429±0.04	0.416±0.03	0.948±0.00	0.91±0.07	0.27±0.01
	SAX	0.366±0.12	0.928±0.04	**1.0±0.00**	0.231±0.09	0.508±0.09	0.958±0.01	**1.0±0.00**	0.346±0.08
	Tool	0.827±0.07	0.995±0.00	0.711±0.10	1.0±0.00	0.787±0.03	0.992±0.00	0.778±0.05	0.799±0.02

Table A.1: Performance of outlier detection methods for water pressure and flow sensors from Barreiro and Beja WDNs with planted point and sequential outliers on 2% of observations, for the historical setting.

		Barreiro WDN				Beja WDN			
		RMSE	MAE	SMAPE	%	RMSE	MAE	SMAPE	%
pressure: point	Mean	0.051±0.07	0.025±0.02	0.941±0.89	100	0.166±0.01	0.154±0.01	4.325±0.21	100
	Median	0.051±0.07	0.024±0.02	0.927±0.89	100	0.19±0.02	0.124±0.02	3.495±0.52	100
	Random sample	0.058±0.07	0.034±0.04	1.268±1.43	100	0.23±0.05	0.169±0.06	4.759±1.64	100
	Interpolation	0.052±0.08	0.023±0.02	0.897±1.04	100	**0.048±0.01**	**0.03±0.01**	0.842±0.17	100
	Locf	**0.039±0.07**	**0.019±0.02**	0.747±0.93	100	0.067±0.02	0.036±0.01	1.017±0.22	100
	Nocb	0.073±0.11	0.03±0.03	1.247±1.53	100	0.057±0.02	0.033±0.01	0.923±0.22	100
	Moving average	0.047±0.07	0.021±0.02	0.822±0.88	100	0.08±0.02	0.042±0.01	1.168±0.27	100
	Random forests	0.074±0.07	0.037±0.02	1.425±1.00	100	0.173±0.01	0.132±0.01	3.709±0.39	100
	EM	0.092±0.06	0.065±0.02	2.422±0.87	100	0.231±0.02	0.19±0.02	5.343±0.50	100
	Knn	0.059±0.05	0.032±0.02	1.216±0.82	100	0.164±0.01	0.132±0.01	3.707±0.41	100
	Mice	0.083±0.06	0.046±0.02	1.722±0.95	100	0.22±0.02	0.152±0.02	4.304±0.50	100
	Amelia	0.095±0.06	0.069±0.02	2.547±0.95	98	0.229±0.01	0.187±0.01	5.247±0.37	97
	Tool	0.039±0.07	0.019±0.02	0.747±0.93	100	0.048±0.01	0.03±0.01	0.842±0.17	100
flow: point	Mean	8.89±1.25	7.461±1.31	34.015±6.62	100	47.609±5.38	36.619±3.97	48.716±4.41	100
	Median	9.061±1.27	7.47±1.35	33.91±7.02	100	49.51±6.59	34.352±4.72	45.846±4.99	100
	Random sample	11.894±4.25	9.956±4.17	41.997±12.05	100	58.912±19.86	46.49±20.79	60.353±29.31	100
	Interpolation	**2.801±0.86**	**2.0±0.66**	10.056±4.14	100	**16.871±2.52**	**11.96±1.36**	9.655±2.54	100
	Locf	4.221±1.22	3.264±0.90	15.714±4.51	100	20.677±3.41	14.189±1.94	23.204±3.36	100
	Nocb	4.52±1.17	3.484±0.91	16.934±4.57	99	20.526±2.83	14.425±1.73	23.604±3.12	100
	Moving average	6.68±2.04	5.314±1.63	25.299±6.89	100	20.534±3.08	14.683±1.64	23.764±3.29	100
	Random forests	10.115±1.86	8.315±1.73	37.123±8.07	100	46.03±5.63	34.514±3.76	46.435±3.94	100
	EM	12.125±2.88	9.982±2.50	47.819±11.22	100	64.264±6.33	49.785±4.78	77.618±6.87	100
	Knn	9.771±1.60	8.051±1.46	36.148±7.26	100	48.345±5.78	36.213±3.97	48.019±4.41	100
	Mice	12.69±2.57	10.433±2.42	48.719±11.95	100	72.407±6.63	54.85±5.43	67.096±5.80	100
	Amelia	12.548±2.03	10.346±1.76	46.246±8.02	98	67.114±5.31	53.254±5.21	75.529±6.72	97
	Tool	2.801±0.86	2.0±0.66	10.056±4.14	100	16.871±2.52	11.96±1.36	19.655±2.54	100
pressure: sequential	Mean	0.025±0.06	0.014±0.02	0.535±0.80	100	0.166±0.03	0.156±0.02	4.399±0.66	100
	Median	**0.024±0.06**	**0.013±0.02**	0.518±0.80	100	0.185±0.07	0.129±0.06	3.653±1.69	100
	Random sample	0.035±0.06	0.024±0.03	0.908±1.20	100	0.227±0.07	0.174±0.08	4.924±2.23	100
	Interpolation	0.03±0.06	0.021±0.04	0.834±1.82	100	0.117±0.03	0.09±0.03	2.538±0.92	100
	Locf	0.039±0.10	0.029±0.07	1.176±3.10	100	0.207±0.08	0.15±0.08	4.228±2.31	100
	Nocb	0.029±0.06	0.018±0.02	0.67±0.87	100	0.18±0.07	0.123±0.08	3.489±2.17	100
	Moving average	0.03±0.07	0.022±0.05	0.875±2.12	67	**0.061±0.06**	**0.044±0.05**	1.224±1.42	13
	Random forests	0.074±0.09	0.034±0.03	1.35±1.30	1.00	0.189±0.03	0.149±0.03	4.187±0.77	100
	EM	0.086±0.05	0.064±0.02	2.371±0.75	100	0.227±0.03	0.188±0.03	5.286±0.71	100
	Knn	0.049±0.06	0.026±0.02	1.013±0.83	100	0.177±0.03	0.145±0.03	4.073±0.79	100
	Mice	0.047±0.06	0.027±0.02	0.994±0.81	100	0.229±0.03	0.164±0.03	4.633±0.84	100
	Amelia	0.082±0.05	0.063±0.02	2.302±0.88	100	0.236±0.03	0.194±0.03	5.441±0.74	100
	Tool	0.024±0.06	0.013±0.02	0.518±0.80	100	0.117±0.03	0.09±0.03	2.538±0.92	100
flow: sequential	Mean	8.922±3.13	7.691±3.31	33.079±12.44	100	49.262±15.83	38.472±11.40	44.656±7.20	100
	Median	8.956±3.00	7.574±3.26	32.503±12.70	100	52.412±20.38	38.268±14.57	44.13±10.56	100
	Random sample	10.477±4.33	8.774±3.75	37.703±16.27	100	61.061±22.51	49.609±20.98	59.152±29.97	100
	Interpolation	**5.889±2.26**	**4.855±2.15**	21.426±8.18	100	34.086±11.53	25.531±7.72	31.003±8.26	100
	Locf	9.482±3.27	7.23±2.73	32.741±11.95	100	40.719±13.46	31.17±9.71	37.961±13.23	100
	Nocb	8.971±3.37	7.244±3.08	31.694±12.54	100	49.136±19.13	37.041±13.07	43.945±12.47	100
	Moving average	9.006±3.95	7.262±3.38	33.103±13.17	67	**24.745±12.15**	**19.515±9.65**	23.337±12.06	13
	Random forests	10.211±2.87	8.412±3.00	35.549±11.01	100	49.06±14.73	37.128±10.54	43.422±6.87	100
	EM	12.111±3.38	10.346±3.01	47.673±11.92	100	69.439±16.02	54.84±12.28	78.177±9.67	100
	Knn	10.035±2.98	8.375±3.01	35.499±11.47	100	51.228±14.47	39.216±10.56	46.073±6.67	100
	Mice	12.611±4.13	10.549±3.47	47.185±14.90	100	72.33±13.95	56.269±11.41	64.379±8.45	100
	Amelia	12.232±3.02	10.263±2.76	45.0±11.82	100	68.419±13.26	55.239±10.25	72.444±7.02	100
	Tool	5.889±2.26	4.855±2.15	21.426±8.18	100	34.086±11.53	25.531±7.72	31.003±8.26	100

Table A.2: Performance of missing imputation methods for water pressure and flow sensors from Barreiro and Beja WDNs with planted point and sequential missing values on 2% of observations, for the historical setting.

		Barreiro WDN				Beja WDN			
		F1-score	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall
pressure: point	Standard deviation	0.393±0.06	0.921±0.01	0.259±0.05	0.832±0.05	0.414±0.02	0.926±0.00	0.261±0.02	**1.0±0.00**
	Inter quartile range	0.684±0.01	0.908±0.00	1.0±0.00	0.52±0.01	0.686±0.02	0.952±0.00	0.522±0.03	**1.0±0.00**
	Isolation forests	**0.867±0.03**	0.973±0.01	**0.873±0.03**	**0.861±0.03**	**0.75±0.03**	0.95±0.01	**0.751±0.03**	0.75±0.03
	Local outlier factor	0.179±0.07	0.835±0.01	0.18±0.07	0.178±0.07	0.0±0.00	0.801±0.00	0.0±0.00	0.0±0.00
	Dbscan	0.0±0.00	0.899±0.00	0.0±0.00	0.0±0.00	0.0±0.00	0.9±0.00	0.0±0.00	0.0±0.00
	SAX	0.182±0.00	0.106±0.00	1.0±0.00	0.1±0.00	0.61±0.11	0.945±0.01	0.448±0.11	**1.0±0.00**
	Tool	0.867±0.03	0.973±0.01	0.873±0.03	0.861±0.03	0.75±0.03	0.95±0.01	0.751±0.03	0.75±0.03
flow: point	Standard deviation	0.452±0.04	0.93±0.00	0.293±0.03	**1.0±0.00**	0.477±0.02	0.932±0.00	0.313±0.02	**1.0±0.00**
	Inter quartile range	**0.926±0.02**	0.986±0.00	0.864±0.04	**1.0±0.00**	**0.964±0.01**	0.993±0.00	**0.981±0.01**	0.949±0.00
	Isolation forests	0.879±0.05	0.976±0.01	**0.886±0.05**	0.873±0.05	0.855±0.02	0.971±0.00	0.856±0.02	0.854±0.02
	Local outlier factor	0.15±0.05	0.829±0.01	0.151±0.05	0.149±0.05	0.084±0.03	0.817±0.01	0.084±0.03	0.083±0.03
	Dbscan	0.837±0.06	0.966±0.01	0.881±0.09	0.801±0.03	0.774±0.01	0.944±0.00	0.954±0.02	0.651±0.01
	SAX	0.745±0.10	0.96±0.01	0.603±0.12	**1.0±0.00**	0.598±0.11	0.944±0.01	0.435±0.10	**1.0±0.00**
	Tool	0.926±0.02	0.986±0.00	0.864±0.04	1.0±0.00	0.964±0.01	0.993±0.00	0.981±0.01	0.949±0.00
pressure: segment	Standard deviation	0.373±0.05	0.918±0.00	0.243±0.04	0.807±0.04	0.415±0.03	0.926±0.00	0.262±0.02	**1.0±0.00**
	Inter quartile range	0.665±0.00	0.898±0.00	**1.0±0.00**	0.498±0.00	0.692±0.02	0.953±0.00	0.529±0.03	**1.0±0.00**
	Isolation forests	**0.854±0.03**	0.97±0.01	0.853±0.03	**0.855±0.03**	**0.75±0.04**	0.95±0.01	**0.751±0.04**	0.749±0.04
	Local outlier factor	0.151±0.07	0.828±0.01	0.15±0.07	0.151±0.07	0.0±0.00	0.801±0.00	0.0±0.00	0.0±0.00
	Dbscan	0.0±0.00	0.897±0.00	0.0±0.00	0.0±0.00	0.0±0.00	0.9±0.00	0.0±0.00	0.0±0.00
	SAX	0.185±0.00	0.108±0.00	**1.0±0.00**	0.102±0.00	0.617±0.11	0.946±0.01	0.455±0.11	**1.0±0.00**
	Tool	0.854±0.03	0.97±0.01	0.853±0.03	0.855±0.03	0.75±0.04	0.95±0.01	0.751±0.04	0.749±0.04
flow: segment	Standard deviation	0.45±0.05	0.928±0.00	0.292±0.04	**1.0±0.00**	0.486±0.02	0.932±0.00	0.322±0.01	**1.0±0.00**
	Inter quartile range	**0.929±0.03**	0.987±0.01	0.869±0.05	**1.0±0.00**	**0.956±0.01**	0.991±0.00	**0.982±0.03**	0.934±0.03
	Isolation forests	0.889±0.05	0.978±0.01	**0.889±0.05**	0.889±0.05	0.855±0.02	0.971±0.00	0.856±0.02	0.854±0.02
	Local outlier factor	0.157±0.05	0.829±0.01	0.157±0.05	0.157±0.05	0.091±0.04	0.819±0.01	0.091±0.04	0.091±0.04
	Dbscan	0.816±0.10	0.962±0.02	0.864±0.14	0.779±0.07	0.773±0.03	0.944±0.01	0.953±0.05	0.651±0.02
	SAX	0.795±0.08	0.966±0.01	0.667±0.11	**1.0±0.00**	0.61±0.11	0.945±0.01	0.448±0.11	**1.0±0.00**
	Tool	0.929±0.03	0.987±0.01	0.869±0.05	1.0±0.00	0.956±0.01	0.991±0.00	0.982±0.03	0.934±0.03

Table A.3: Performance of outlier detection methods for water pressure and flow sensors from Barreiro and Beja WDNs with planted point and sequential outliers on up to 10% of observations, for the historical setting.

		Barreiro WDN				Beja WDN			
		RMSE	MAE	SMAPE	%	RMSE	MAE	SMAPE	%
pressure: point	Mean	0.064±0.05	0.025±0.01	0.933±0.43	100	0.166±0.00	0.154±0.00	4.335±0.08	100
	Median	0.065±0.05	0.024±0.01	0.921±0.42	100	0.192±0.01	0.125±0.01	3.517±0.18	100
	Random sample	0.07±0.05	0.033±0.03	1.247±0.91	100	0.23±0.04	0.168±0.05	4.737±1.45	100
	Interpolation	0.058±0.05	0.019±0.01	0.731±0.37	99	**0.05±0.00**	**0.031±0.00**	0.857±0.06	100
	Locf	0.062±0.05	0.021±0.01	0.796±0.36	99	0.068±0.01	0.036±0.00	0.999±0.10	100
	Nocb	0.062±0.06	0.021±0.01	0.808±0.43	99	0.065±0.01	0.035±0.00	0.973±0.09	100
	Moving average	**0.055±0.04**	**0.018±0.01**	0.712±0.33	99	0.078±0.01	0.04±0.00	1.113±0.10	100
	Random forests	0.089±0.04	0.037±0.01	1.414±0.41	100	0.175±0.01	0.133±0.01	3.752±0.18	100
	EM	0.098±0.03	0.065±0.01	2.404±0.28	100	0.228±0.01	0.188±0.01	5.275±0.23	100
	Knn	0.069±0.04	0.032±0.01	1.19±0.38	100	0.166±0.01	0.135±0.01	3.806±0.15	100
	Mice	0.084±0.04	0.043±0.01	1.574±0.49	100	0.221±0.01	0.154±0.01	4.337±0.24	100
	Amelia	0.104±0.04	0.068±0.01	2.508±0.54	90	0.234±0.01	0.19±0.01	5.342±0.21	90
	Tool	0.055±0.04	0.018±0.01	0.712±0.33	99	0.05±0.00	0.031±0.00	0.857±0.06	100
flow: point	Mean	8.878±0.58	7.331±0.57	32.584±2.57	100	47.573±2.71	36.25±1.68	48.104±1.96	100
	Median	8.976±0.57	7.312±0.58	32.392±2.70	100	49.678±3.18	34.009±2.14	45.249±2.21	100
	Random sample	12.186±4.31	10.31±4.14	42.831±12.28	100	59.523±17.85	46.715±19.59	60.721±28.42	100
	Interpolation	**2.927±0.32**	**2.055±0.25**	9.821±1.23	99	**17.246±1.34**	**12.137±0.79**	19.849±1.08	100
	Locf	4.602±0.47	3.49±0.34	16.146±1.65	99	21.231±1.60	14.692±0.85	23.877±1.62	100
	Nocb	4.878±0.55	3.649±0.35	16.924±1.43	99	21.133±1.57	14.565±1.01	24.047±1.59	100
	Moving average	7.021±0.79	5.48±0.59	25.245±2.28	99	21.595±1.46	15.242±0.83	24.077±1.57	100
	Random forests	10.086±0.83	8.085±0.78	35.154±3.40	100	47.097±3.20	34.74±2.29	46.244±2.00	100
	EM	12.08±0.85	9.806±0.72	46.242±3.74	100	66.136±3.22	50.566±2.36	79.039±3.73	100
	Knn	9.639±0.73	7.857±0.66	34.557±2.72	100	47.803±2.11	35.747±1.66	47.494±2.00	100
	Mice	13.291±1.05	10.893±1.03	49.257±5.83	100	72.356±5.04	54.984±4.83	66.829±5.40	100
	Amelia	12.11±0.94	9.78±0.82	42.972±4.07	90	65.755±2.65	51.807±2.21	71.872±3.35	90
	Tool	2.927±0.32	2.055±0.25	9.821±1.23	99	17.246±1.34	12.137±0.79	19.849±1.08	100
pressure: sequential	Mean	0.021±0.02	0.013±0.00	0.464±0.13	100	**0.169±0.01**	0.157±0.00	4.416±0.13	100
	Median	**0.02±0.02**	**0.012±0.00**	0.427±0.13	100	0.197±0.01	**0.13±0.01**	3.684±0.34	100
	Random sample	0.03±0.03	0.022±0.03	0.816±0.95	100	0.23±0.04	0.168±0.05	4.729±1.31	100
	Interpolation	0.03±0.06	0.021±0.04	0.817±1.58	100	0.207±0.01	0.164±0.01	4.618±0.39	100
	Locf	0.043±0.11	0.032±0.08	1.265±3.34	100	0.24±0.03	0.177±0.04	4.998±1.15	100
	Nocb	0.022±0.02	0.014±0.01	0.533±0.20	100	0.237±0.03	0.175±0.03	4.955±0.91	100
	Moving average	0.031±0.07	0.022±0.05	0.887±2.12	12	0.061±0.07	0.044±0.05	1.225±1.42	3
	Random forests	0.064±0.04	0.027±0.01	1.013±0.33	100	0.189±0.01	0.147±0.01	4.155±0.24	100
	EM	0.078±0.01	0.061±0.01	2.21±0.19	100	0.23±0.01	0.189±0.01	5.326±0.22	100
	Knn	0.045±0.02	0.024±0.00	0.871±0.16	100	0.178±0.01	0.146±0.01	4.112±0.19	100
	Mice	0.056±0.02	0.032±0.01	1.167±0.31	100	0.229±0.01	0.162±0.01	4.587±0.30	100
	Amelia	0.073±0.02	0.056±0.01	2.026±0.41	89	0.238±0.01	0.194±0.01	5.433±0.33	89
	Tool	0.02±0.02	0.012±0.00	0.427±0.13	100	0.169±0.01	0.157±0.00	4.416±0.13	100
flow: sequential	Mean	8.839±1.09	7.415±1.25	32.556±4.45	100	53.752±13.36	40.987±9.06	46.883±3.85	100
	Median	8.884±0.98	7.321±1.20	32.073±4.38	100	56.946±18.43	41.172±12.99	46.95±8.74	100
	Random sample	11.589±3.59	9.644±3.29	41.054±11.91	100	65.0±21.24	51.602±19.29	60.692±29.12	100
	Interpolation	10.273±1.21	8.493±1.13	37.44±4.24	100	40.72±10.25	31.124±7.20	37.789±6.21	100
	Locf	12.074±2.19	9.705±1.90	43.785±8.47	100	44.045±11.33	33.286±8.19	40.576±11.90	100
	Nocb	11.092±2.30	9.055±1.90	39.8±8.14	100	47.163±15.20	37.099±12.08	44.922±12.72	100
	Moving average	**8.803±3.96**	**7.065±3.39**	31.923±13.69	12	**24.396±12.08**	**19.166±9.47**	23.783±12.51	3
	Random forests	9.913±1.07	7.961±1.05	34.648±3.79	100	53.638±13.67	39.78±9.41	45.504±4.33	100
	EM	12.193±1.44	9.957±1.25	46.583±5.41	100	73.736±13.64	57.226±10.29	79.993±5.61	100
	Knn	9.431±1.13	7.696±1.17	33.592±4.34	100	54.104±12.18	40.735±8.19	47.224±3.77	100
	Mice	12.148±1.79	9.931±1.55	44.663±7.75	100	75.86±6.55	58.838±5.76	67.721±4.48	100
	Amelia	12.125±1.38	9.771±1.29	43.056±5.90	89	70.062±8.17	55.106±6.29	70.747±3.56	89
	Tool	8.839±1.09	7.415±1.25	32.556±4.45	100	40.72±10.25	31.124±7.20	37.789±6.21	100

Table A.4: Performance of imputation methods for water pressure and flow sensors from Barreiro and Beja WDNs with planted point and sequential missing values on 10% of observations, for the historical setting.

B

Real-time data results

		Barreiro WDN				Beja WDN			
		F1-score	Precision	Recall	ΔT	F1-score	Precision	Recall	ΔT
pressure: point	Standard deviation	0.505 ± 0.2	0.437 ± 0.19	**0.611 ± 0.2**	0.017	0.02 ± 0.06	0.011 ± 0.03	0.1 ± 0.3	0.003
	Inter quartile range	**0.56 ± 0.02**	**0.978 ± 0.05**	0.393 ± 0.01	0.018	0.062 ± 0.05	0.181 ± 0.14	0.038 ± 0.03	0.019
	Isolation forests	0.315 ± 0.02	0.97 ± 0.06	0.188 ± 0.01	0.257	**0.267 ± 0.02**	0.985 ± 0.05	**0.155 ± 0.01**	0.366
	Local outlier factor	0.309 ± 0.02	0.97 ± 0.06	0.184 ± 0.01	0.019	0.266 ± 0.02	**0.989 ± 0.04**	0.154 ± 0.01	0.026
	Dbscan	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.019	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0
	Sax	0.052 ± 0.01	0.97 ± 0.08	0.027 ± 0.00	0.013	0.061 ± 0.01	0.989 ± 0.04	0.031 ± 0.01	0.014
	Tool	0.56 ± 0.02	0.978 ± 0.05	0.393 ± 0.01	0.018	0.267 ± 0.02	0.985 ± 0.05	0.155 ± 0.01	0.366
flow: point	Standard deviation	0.365 ± 0.17	0.24 ± 0.14	0.967 ± 0.18	0.015	0.256 ± 0.13	0.17 ± 0.09	**0.544 ± 0.2**	0.017
	Inter quartile range	**0.435 ± 0.17**	0.296 ± 0.17	**1.0 ± 0.0**	0.018	0.096 ± 0.06	0.241 ± 0.15	0.06 ± 0.03	0.019
	Isolation forests	0.301 ± 0.01	0.978 ± 0.04	0.178 ± 0.00	0.256	**0.274 ± 0.01**	**0.993 ± 0.02**	0.159 ± 0.00	0.367
	Local outlier factor	0.331 ± 0.01	0.985 ± 0.03	0.199 ± 0.00	0.019	0.222 ± 0.02	0.848 ± 0.11	0.128 ± 0.01	0.026
	Dbscan	0.324 ± 0.01	0.985 ± 0.03	0.194 ± 0.0	0.019	0.079 ± 0.00	**0.993 ± 0.02**	0.041 ± 0.00	0.021
	Sax	0.124 ± 0.02	**0.989 ± 0.03**	0.066 ± 0.01	0.013	0.105 ± 0.01	**0.993 ± 0.02**	0.056 ± 0.00	0.014
	Tool	0.435 ± 0.17	0.296 ± 0.17	1.0 ± 0.0	0.018	0.274 ± 0.01	0.993 ± 0.02	0.159 ± 0.00	0.367
pressure: segment	Standard deviation	**0.567 ± 0.17**	0.521 ± 0.21	**0.647 ± 0.10**	0.02	0.021 ± 0.08	0.012 ± 0.04	0.067 ± 0.24	0.001
	Inter quartile range	0.541 ± 0.00	**1.0 ± 0.0**	0.371 ± 0.00	0.022	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.02
	Isolation forests	0.276 ± 0.00	**1.0 ± 0.0**	0.16 ± 0.00	0.298	**0.245 ± 0.00**	**1.0 ± 0.0**	0.14 ± 0.00	0.264
	Local outlier factor	0.264 ± 0.00	**1.0 ± 0.0**	0.152 ± 0.00	0.025	0.239 ± 0.00	**1.0 ± 0.0**	0.136 ± 0.00	0.019
	Dbscan	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.027	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0
	Sax	0.044 ± 0.00	0.971 ± 0.08	0.022 ± 0.00	0.022	0.054 ± 0.01	**1.0 ± 0.0**	0.028 ± 0.00	0.014
	Tool	0.567 ± 0.17	0.521 ± 0.21	0.647 ± 0.10	0.02	0.245 ± 0.00	1.0 ± 0.0	0.14 ± 0.00	0.264
flow: segment	Standard deviation	0.251 ± 0.19	0.158 ± 0.13	**0.733 ± 0.44**	0.013	0.198 ± 0.15	0.133 ± 0.11	0.406 ± 0.27	0.017
	Inter quartile range	0.262 ± 0.22	0.171 ± 0.16	0.7 ± 0.45	0.015	0.041 ± 0.04	0.112 ± 0.12	0.025 ± 0.02	0.02
	Isolation forests	0.266 ± 0.01	0.954 ± 0.06	0.155 ± 0.01	0.295	**0.243 ± 0.01**	0.996 ± 0.02	0.138 ± 0.00	0.265
	Local outlier factor	0.275 ± 0.03	0.938 ± 0.11	0.161 ± 0.01	0.025	0.233 ± 0.02	0.971 ± 0.07	0.132 ± 0.01	0.019
	Dbscan	**0.298 ± 0.02**	0.979 ± 0.06	0.175 ± 0.01	0.027	0.072 ± 0.00	**1.0 ± 0.0**	0.037 ± 0.0	0.021
	Sax	0.094 ± 0.01	**1.0 ± 0.0**	0.049 ± 0.00	0.021	0.081 ± 0.00	**1.0 ± 0.0**	0.042 ± 0.00	0.014
	Tool	0.298 ± 0.02	0.979 ± 0.06	0.175 ± 0.01	0.027	0.243 ± 0.01	0.996 ± 0.02	0.138 ± 0.00	0.265

Table B.1: Performance of outlier detection methods for water pressure and flow sensors from Barreiro and Beja WDNs with planted point and sequential outliers on 2% of observations, for the real time setting.

		Barreiro WDN				Beja WDN			
		RMSE	MAE	SMAPE	ΔT	RMSE	MAE	SMAPE	ΔT
pressure: point	Mean	0.014 ± 0.00	0.011 ± 0.00	0.392 ± 0.09	0.003	0.161 ± 0.02	0.151 ± 0.02	4.265 ± 0.65	0.002
	Median	**0.011 ± 0.00**	**0.009 ± 0.00**	0.345 ± 0.09	0.003	0.178 ± 0.06	0.122 ± 0.04	3.446 ± 1.36	0.003
	Random sample	0.037 ± 0.07	0.021 ± 0.02	0.814 ± 1.11	0.003	0.271 ± 0.04	0.225 ± 0.06	6.379 ± 1.73	0.003
	Interpolation	0.031 ± 0.04	0.017 ± 0.01	0.648 ± 0.63	0.003	**0.039 ± 0.02**	**0.027 ± 0.01**	0.747 ± 0.39	0.003
	Locf	0.032 ± 0.06	0.018 ± 0.02	0.712 ± 1.10	0.003	0.047 ± 0.04	0.03 ± 0.01	0.84 ± 0.53	0.002
	Nocb	0.035 ± 0.07	0.02 ± 0.02	0.759 ± 1.03	0.003	0.042 ± 0.03	0.028 ± 0.01	0.794 ± 0.42	0.002
	Moving average	0.027 ± 0.04	0.017 ± 0.02	0.622 ± 0.79	0.003	0.06 ± 0.04	0.036 ± 0.02	1.004 ± 0.71	0.003
	Random forests	0.03 ± 0.05	0.017 ± 0.02	0.659 ± 0.87	1.741	0.076 ± 0.04	0.045 ± 0.02	1.264 ± 0.62	1.688
	Expectation maximization	0.068 ± 0.02	0.049 ± 0.01	1.791 ± 0.68	0.006	0.215 ± 0.04	0.18 ± 0.04	5.082 ± 1.15	0.004
	Knn	0.03 ± 0.045	0.019 ± 0.01	0.727 ± 0.66	0.006	0.081 ± 0.03	0.051 ± 0.02	1.428 ± 0.56	0.004
	Tool	0.011 ± 0.00	0.009 ± 0.00	0.345 ± 0.09	0.003	0.039 ± 0.02	0.027 ± 0.01	0.747 ± 0.39	0.003
flow: point	Mean	8.435 ± 1.48	7.034 ± 1.29	30.253 ± 7.1	0.003	19.772 ± 3.18	17.084 ± 3.34	47.291 ± 10.82	0.003
	Median	8.37 ± 1.56	6.865 ± 1.42	29.487 ± 8.10	0.003	20.111 ± 3.21	17.014 ± 3.34	46.755 ± 11.32	0.003
	Random sample	11.031 ± 2.46	9.231 ± 2.14	39.891 ± 8.16	0.003	29.432 ± 5.19	25.472 ± 5.15	69.133 ± 15.43	0.003
	Interpolation	**2.324 ± 0.83**	**1.798 ± 0.60**	7.983 ± 2.65	0.005	11.768 ± 3.84	8.521 ± 2.69	22.814 ± 6.91	0.003
	Locf	4.08 ± 1.15	3.155 ± 0.86	13.615 ± 3.88	0.003	13.497 ± 4.44	9.555 ± 2.99	26.409 ± 8.10	0.002
	Nocb	4.324 ± 1.23	3.407 ± 0.76	14.976 ± 3.67	0.003	12.887 ± 5.27	9.225 ± 3.80	25.484 ± 8.69	0.002
	Moving average	7.055 ± 2.27	5.508 ± 1.75	24.393 ± 6.74	0.003	**11.716 ± 3.59**	**8.513 ± 2.73**	23.321 ± 7.86	0.003
	Random forests	9.775 ± 1.91	8.071 ± 1.77	34.99 ± 7.72	1.706	12.779 ± 3.53	10.185 ± 2.88	28.314 ± 8.15	1.752
	Expectation maximization	11.559 ± 2.56	9.593 ± 2.34	44.161 ± 9.94	0.006	27.58 ± 6.01	22.823 ± 5.18	67.884 ± 15.60	0.004
	Knn	8.748 ± 1.69	7.255 ± 1.45	31.334 ± 6.04	0.008	13.155 ± 3.32	10.466 ± 2.75	28.769 ± 7.53	0.004
	Tool	2.324 ± 0.83	1.798 ± 0.60	7.983 ± 2.65	0.005	11.716 ± 3.59	8.577 ± 2.73	23.321 ± 7.86	0.003
pressure: sequential	Mean	**0.014 ± 0.00**	**0.011 ± 0.00**	0.394 ± 0.18	0.004	0.166 ± 0.02	0.161 ± 0.01	4.558 ± 0.59	0.002
	Median	0.014 ± 0.00	0.011 ± 0.00	0.401 ± 0.16	0.003	0.198 ± 0.08	0.154 ± 0.07	4.362 ± 2.00	0.003
	Random sample	0.035 ± 0.07	0.021 ± 0.02	0.82 ± 1.25	0.003	0.309 ± 0.04	0.28 ± 0.07	7.955 ± 2.04	0.003
	Interpolation	0.017 ± 0.00	0.012 ± 0.00	0.444 ± 0.17	0.003	**0.045 ± 0.04**	**0.036 ± 0.03**	1.025 ± 1.11	0.003
	Locf	0.018 ± 0.00	0.014 ± 0.00	0.507 ± 0.19	0.002	0.064 ± 0.08	0.05 ± 0.07	1.416 ± 2.0	0.002
	Nocb	0.018 ± 0.00	0.014 ± 0.00	0.515 ± 0.20	0.002	0.055 ± 0.07	0.042 ± 0.05	1.175 ± 1.58	0.002
	Moving average	0.015 ± 0.00	0.012 ± 0.00	0.42 ± 0.16	0.003	0.062 ± 0.08	0.048 ± 0.06	1.347 ± 1.96	0.003
	Random forests	0.016 ± 0.00	0.012 ± 0.00	0.453 ± 0.14	1.588	0.095 ± 0.05	0.061 ± 0.04	1.712 ± 1.17	1.595
	Expectation maximization	0.038 ± 0.02	0.028 ± 0.01	1.026 ± 0.64	0.005	0.237 ± 0.05	0.194 ± 0.04	5.494 ± 1.28	0.005
	Knn	0.015 ± 0.00	0.012 ± 0.00	0.439 ± 0.18	0.004	0.089 ± 0.05	0.061 ± 0.04	1.735 ± 1.24	0.004
	Tool	0.014 ± 0.00	0.011 ± 0.00	0.394 ± 0.18	0.004	0.045 ± 0.04	0.036 ± 0.03	1.025 ± 1.11	0.003
flow: sequential	Mean	8.284 ± 2.30	7.214 ± 2.51	32.43 ± 12.04	0.004	19.729 ± 4.25	17.912 ± 4.37	49.144 ± 14.98	0.002
	Median	8.358 ± 2.46	7.134 ± 2.57	31.99 ± 12.42	0.003	19.994 ± 4.12	18.058 ± 4.45	49.23 ± 16.02	0.003
	Random sample	10.658 ± 2.80	8.872 ± 2.65	39.969 ± 10.16	0.003	31.205 ± 6.61	28.679 ± 6.58	79.958 ± 21.45	0.003
	Interpolation	**3.864 ± 1.57**	**3.174 ± 1.31**	14.958 ± 6.22	0.003	13.456 ± 3.77	9.521 ± 3.17	25.268 ± 6.02	0.003
	Locf	7.498 ± 3.40	5.746 ± 2.69	27.063 ± 11.01	0.002	14.936 ± 5.66	11.401 ± 4.98	30.576 ± 11.40	0.002
	Nocb	8.025 ± 4.04	6.515 ± 3.28	28.526 ± 10.71	0.002	14.136 ± 5.94	10.548 ± 5.83	28.409 ± 13.26	0.002
	Moving average	8.692 ± 2.98	6.664 ± 2.55	31.635 ± 9.79	0.003	12.303 ± 4.87	9.718 ± 4.38	26.544 ± 10.61	0.003
	Random forests	9.352 ± 2.88	8.018 ± 2.64	35.493 ± 11.61	1.585	**12.093 ± 3.5**	**10.076 ± 3.20**	29.159 ± 7.83	1.56
	Expectation maximization	11.255 ± 3.14	9.598 ± 2.84	46.204 ± 13.35	0.005	28.161 ± 8.07	24.174 ± 7.28	72.085 ± 20.27	0.005
	Knn	8.563 ± 2.54	7.487 ± 2.63	33.364 ± 11.30	0.004	12.483 ± 3.66	10.548 ± 3.68	30.189 ± 8.04	0.004
	Tool	3.864 ± 1.57	3.174 ± 1.31	14.958 ± 6.22	0.003	12.093 ± 3.5	10.076 ± 3.20	29.159 ± 7.83	1.56

Table B.2: Performance of imputation methods for water pressure and flow sensors from Barreiro and Beja WDNs with planted point and sequential missing values on 2% of observations, for the real time setting.

		Barreiro WDN				Beja WDN			
		F1-score	Precision	Recall	ΔT	F1-score	Precision	Recall	ΔT
pressure: point	Standard deviation	0.365 ± 0.17	0.237 ± 0.13	0.967 ± 0.18	0.015	0.16 ± 0.10	0.092 ± 0.07	**0.858 ± 0.22**	0.016
	Inter quartile range	**0.435 ± 0.17**	0.296 ± 0.16	**1.0 ± 0.0**	0.018	0.311 ± 0.08	0.334 ± 0.10	0.292 ± 0.07	0.019
	Isolation forests	0.301 ± 0.01	0.978 ± 0.04	0.178 ± 0.00	0.256	**0.64 ± 0.02**	0.782 ± 0.04	0.543 ± 0.02	0.286
	Local outlier factor	0.331 ± 0.01	0.985 ± 0.03	0.199 ± 0.00	0.019	0.512 ± 0.10	0.577 ± 0.12	0.46 ± 0.09	0.019
	Dbscan	0.324 ± 0.01	0.985 ± 0.03	0.194 ± 0.01	0.019	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0
	Sax	0.124 ± 0.02	**0.989 ± 0.03**	0.066 ± 0.01	0.013	0.501 ± 0.08	**0.984 ± 0.0**1	0.341 ± 0.07	0.013
	Tool	0.435 ± 0.17	0.296 ± 0.16	1.0 ± 0.0	0.018	0.64 ± 0.02	0.782 ± 0.04	0.543 ± 0.02	0.286
flow: point	Standard deviation	0.382 ± 0.10	0.241 ± 0.07	**1.0 ± 0.0**	0.015	0.357 ± 0.12	0.229 ± 0.09	**0.9 ± 0.04**	0.017
	Inter quartile range	**0.868 ± 0.05**	0.77 ± 0.08	**1.0 ± 0.0**	0.018	0.584 ± 0.06	0.681 ± 0.1	0.513 ± 0.05	0.019
	Isolation forests	0.754 ± 0.03	0.847 ± 0.03	0.681 ± 0.03	0.277	**0.703 ± 0.03**	0.855 ± 0.03	0.598 ± 0.03	0.287
	Local outlier factor	0.416 ± 0.06	0.452 ± 0.06	0.387 ± 0.05	0.026	0.297 ± 0.08	0.353 ± 0.10	0.257 ± 0.07	0.019
	Dbscan	0.688 ± 0.02	0.964 ± 0.03	0.536 ± 0.02	0.024	0.307 ± 0.00	**0.984 ± 0.01**	0.182 ± 0.00	0.021
	Sax	0.704 ± 0.05	**0.987 ± 0.01**	0.551 ± 0.06	0.015	0.649 ± 0.07	**0.984 ± 0.01**	0.489 ± 0.08	0.013
	Tool	0.868 ± 0.05	0.77 ± 0.08	1.0 ± 0.0	0.018	0.703 ± 0.03	0.855 ± 0.03	0.598 ± 0.03	0.287
pressure: segment	Standard deviation	0.453 ± 0.14	0.317 ± 0.13	**0.867 ± 0.04**	0.025	0.108 ± 0.1	0.06 ± 0.06	**0.9 ± 0.3**	0.015
	Inter quartile range	**0.892 ± 0.00**	**1.0 ± 0.0**	0.805 ± 0.01	0.031	0.263 ± 0.13	0.301 ± 0.17	0.236 ± 0.11	0.022
	Isolation forests	0.691 ± 0.02	0.857 ± 0.04	0.579 ± 0.02	0.289	**0.542 ± 0.04**	0.687 ± 0.06	0.448 ± 0.02	0.311
	Local outlier factor	0.418 ± 0.04	0.485 ± 0.06	0.368 ± 0.03	0.019	0.341 ± 0.02	0.404 ± 0.03	0.295 ± 0.02	0.026
	Dbscan	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.019	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0
	Sax	0.295 ± 0.02	0.98 ± 0.03	0.174 ± 0.01	0.013	0.401 ± 0.04	**0.993 ± 0.02**	0.252 ± 0.03	0.018
	Tool	0.892 ± 0.00	1.0 ± 0.0	0.805 ± 0.01	0.031	0.542 ± 0.04	0.687 ± 0.06	0.448 ± 0.02	0.311
flow: segment	Standard deviation	0.145 ± 0.07	0.08 ± 0.04	0.933 ± 0.24	0.024	0.149 ± 0.08	0.085 ± 0.05	**0.741 ± 0.17**	0.017
	Inter quartile range	**0.71 ± 0.1**	0.559 ± 0.11	**1.0 ± 0.0**	0.03	0.373 ± 0.10	0.4 ± 0.13	0.351 ± 0.08	0.022
	Isolation forests	0.66 ± 0.03	0.774 ± 0.05	0.576 ± 0.03	0.287	**0.6 ± 0.03**	0.76 ± 0.05	0.496 ± 0.02	0.311
	Local outlier factor	0.411 ± 0.04	0.465 ± 0.06	0.37 ± 0.03	0.019	0.36 ± 0.05	0.445 ± 0.07	0.304 ± 0.04	0.026
	Dbscan	0.705 ± 0.02	0.965 ± 0.04	0.556 ± 0.02	0.02	0.318 ± 0.00	0.996 ± 0.01	0.189 ± 0.00	0.028
	Sax	0.504 ± 0.03	1.0 ± 0.0	0.337 ± 0.03	0.013	0.47 ± 0.04	**0.999 ± 0.00**	0.308 ± 0.03	0.018
	Tool	0.71 ± 0.1	0.559 ± 0.11	1.0 ± 0.0	0.03	0.6 ± 0.03	0.76 ± 0.05	0.496 ± 0.02	0.311

Table B.3: Performance of outlier detection methods for water pressure and flow sensors from Barreiro and Beja WDNs with planted point and sequential outliers on up to 10% of observations, for the real time setting.

		Barreiro WDN				Beja WDN			
		RMSE	MAE	SMAPE	ΔT	RMSE	MAE	SMAPE	ΔT
pressure: point	Mean	0.072 ± 0.07	0.021 ± 0.01	0.827 ± 0.51	0.004	0.165 ± 0.00	0.152 ± 0.00	4.3 ± 0.237	0.00
	Median	**0.07 ± 0.07**	**0.019 ± 0.01**	0.778 ± 0.53	0.005	0.193 ± 0.01	0.128 ± 0.02	3.627 ± 0.59	0.003
	Random sample	0.095 ± 0.07	0.03 ± 0.01	1.199 ± 0.64	0.003	0.276 ± 0.01	0.227 ± 0.02	6.431 ± 0.65	0.003
	Interpolation	0.086 ± 0.06	0.024 ± 0.01	0.977 ± 0.5	0.004	**0.046 ± 0.01**	**0.027 ± 0.00**	0.769 ± 0.15	0.003
	Locf	0.09 ± 0.07	0.027 ± 0.01	1.074 ± 0.64	0.003	0.058 ± 0.02	0.031 ± 0.00	0.86 ± 0.23	0.002
	Nocb	0.089 ± 0.07	0.026 ± 0.01	1.034 ± 0.58	0.003	0.059 ± 0.02	0.03 ± 0.00	0.846 ± 0.21	0.002
	Moving average	0.077 ± 0.06	0.022 ± 0.01	0.905 ± 0.47	0.003	0.075 ± 0.01	0.037 ± 0.00	1.052 ± 0.25	0.003
	Random forests	0.097 ± 0.05	0.028 ± 0.01	1.152 ± 0.49	2.377	0.103 ± 0.01	0.055 ± 0.00	1.561 ± 0.26	2.29
	Expectation maximization	0.112 ± 0.04	0.058 ± 0.01	2.19 ± 0.44	0.008	0.224 ± 0.01	0.185 ± 0.01	5.209 ± 0.49	0.005
	Knn	0.083 ± 0.05	0.026 ± 0.01	1.043 ± 0.46	0.009	0.101 ± 0.01	0.062 ± 0.00	1.756 ± 0.26	0.005
	Tool	0.07 ± 0.07	0.019 ± 0.01	0.778 ± 0.53	0.005	0.046 ± 0.01	0.027 ± 0.00	0.769 ± 0.15	0.003
flow: point	Mean	8.679 ± 0.65	7.165 ± 0.66	31.273 ± 3.45	0.004	19.437 ± 1.37	16.628 ± 1.46	45.317 ± 4.56	0.003
	Median	8.738 ± 0.68	7.153 ± 0.69	31.162 ± 3.66	0.005	20.083 ± 1.46	16.798 ± 1.51	45.164 ± 5.05	0.003
	Random sample	11.189 ± 1.05	9.113 ± 0.93	39.756 ± 4.36	0.003	29.363 ± 1.61	25.211 ± 1.79	68.021 ± 5.67	0.003
	Interpolation	**2.454 ± 0.35**	**1.773 ± 0.23**	7.816 ± 1.13	0.004	**11.815 ± 1.58**	**8.132 ± 1.17**	22.053 ± 2.42	0.003
	Locf	4.537 ± 0.64	3.355 ± 0.40	14.816 ± 1.82	0.003	13.717 ± 2.18	9.214 ± 1.55	25.721 ± 3.47	0.002
	Nocb	4.431 ± 0.61	3.381 ± 0.40	15.09 ± 1.88	0.003	13.606 ± 1.91	9.142 ± 1.38	25.343 ± 3.03	0.002
	Moving average	7.909 ± 0.85	6.055 ± 0.73	27.195 ± 3.25	0.003	12.243 ± 1.78	8.868 ± 1.15	23.931 ± 2.53	0.003
	Random forests	10.143 ± 0.93	8.246 ± 0.97	35.709 ± 4.70	2.37	13.117 ± 1.23	10.074 ± 1.09	27.597 ± 2.86	2.338
	Expectation maximization	11.999 ± 0.96	9.71 ± 0.96	45.178 ± 5.32	0.008	26.534 ± 2.30	21.895 ± 2.04	64.074 ± 7.21	0.005
	Knn	9.339 ± 0.75	7.695 ± 0.76	33.471 ± 3.87	0.009	13.418 ± 1.69	10.473 ± 1.41	28.388 ± 3.33	0.005
	Tool	2.454 ± 0.35	1.773 ± 0.23	7.816 ± 1.13	0.004	11.815 ± 1.58	8.132 ± 1.17	22.053 ± 2.42	0.003
pressure: sequential	Mean	**0.014 ± 0.00**	**0.011 ± 0.00**	0.392 ± 0.06	0.002	0.167 ± 0.01	0.16 ± 0.01	4.508 ± 0.34	0.002
	Median	**0.014 ± 0.00**	**0.011 ± 0.00**	0.385 ± 0.04	0.003	0.196 ± 0.06	0.139 ± 0.05	3.947 ± 1.60	0.003
	Random sample	0.058 ± 0.08	0.025 ± 0.02	1.005 ± 1.09	0.003	0.269 ± 0.01	0.214 ± 0.02	6.08 ± 0.68	0.003
	Interpolation	0.018 ± 0.00	0.014 ± 0.00	0.522 ± 0.14	0.003	0.145 ± 0.09	0.104 ± 0.08	2.954 ± 2.31	0.003
	Locf	0.019 ± 0.00	0.015 ± 0.00	0.539 ± 0.15	0.002	0.167 ± 0.10	0.122 ± 0.09	3.457 ± 2.62	0.002
	Nocb	0.017 ± 0.00	0.013 ± 0.00	0.488 ± 0.19	0.002	**0.046 ± 0.05**	**0.035 ± 0.04**	0.993 ± 1.13	0.002
	Moving average	0.015 ± 0.00	0.012 ± 0.00	0.424 ± 0.08	0.003	0.169 ± 0.09	0.124 ± 0.08	3.5 ± 2.52	0.003
	Random forests	0.036 ± 0.04	0.016 ± 0.00	0.611 ± 0.39	1.872	0.12 ± 0.03	0.07 ± 0.02	1.979 ± 0.80	2.006
	Expectation maximization	0.055 ± 0.03	0.041 ± 0.02	1.49 ± 0.88	0.008	0.231 ± 0.02	0.192 ± 0.02	5.426 ± 0.65	0.009
	Knn	0.033 ± 0.02	0.015 ± 0.00	0.574 ± 0.20	0.004	0.114 ± 0.03	0.078 ± 0.03	2.21 ± 0.89	0.005
	Tool	0.014 ± 0.00	0.011 ± 0.00	0.392 ± 0.06	0.002	0.046 ± 0.05	0.035 ± 0.04	0.993 ± 1.13	0.002
flow: sequential	Mean	**8.637 ± 1.41**	**7.309 ± 1.37**	32.281 ± 4.93	0.002	20.088 ± 2.43	17.47 ± 1.85	47.349 ± 7.33	0.002
	Median	8.684 ± 1.37	7.236 ± 1.38	31.886 ± 4.95	0.003	19.998 ± 1.49	17.312 ± 1.91	46.634 ± 9.95	0.003
	Random sample	12.265 ± 1.25	10.072 ± 1.21	44.458 ± 4.44	0.003	26.923 ± 3.32	23.181 ± 3.04	63.795 ± 6.56	0.003
	Interpolation	11.751 ± 2.54	9.615 ± 2.14	43.713 ± 9.83	0.003	18.467 ± 5.44	14.711 ± 4.93	40.009 ± 14.68	0.003
	Locf	11.692 ± 2.68	9.601 ± 2.31	43.747 ± 10.97	0.002	20.87 ± 5.72	16.61 ± 5.13	44.595 ± 15.12	0.002
	Nocb	8.705 ± 4.37	7.063 ± 3.58	30.515 ± 12.70	0.002	**13.917 ± 5.12**	**10.883 ± 4.76**	32.353 ± 13.73	0.002
	Moving average	10.795 ± 2.17	8.827 ± 1.91	39.962 ± 9.18	0.003	21.252 ± 5.92	16.954 ± 4.88	46.519 ± 13.91	0.003
	Random forests	9.874 ± 1.37	8.138 ± 1.23	35.761 ± 4.63	1.93	15.008 ± 3.53	11.593 ± 2.91	31.675 ± 6.50	2.065
	Expectation maximization	11.938 ± 1.47	9.754 ± 1.30	46.534 ± 6.23	0.008	27.71 ± 5.16	22.599 ± 4.26	66.938 ± 8.90	0.01
	Knn	8.976 ± 1.51	7.497 ± 1.35	33.164 ± 4.78	0.004	14.95 ± 3.22	11.931 ± 2.90	32.811 ± 6.44	0.005
	Tool	8.637 ± 1.41	7.309 ± 1.37	32.281 ± 4.93	0.002	13.917 ± 5.12	10.883 ± 4.76	32.353 ± 13.73	0.002

Table B.4: Performance of imputation methods for water pressure and flow sensors from Barreiro and Beja WDNs with planted point and sequential missing values on 10% of observations, for the real time setting.