




## Article

# Industrial Data-Driven Processing Framework Combining Process Knowledge for Improved Decision Making—Part 1: Framework Development

Émilie Thibault <sup>1</sup>, Jeffrey Dean Kelly <sup>2</sup>, Francis Lebreux Desilets <sup>1</sup>, Moncef Chioua <sup>1</sup>, Bruno Poulin <sup>3</sup> and Paul Stuart <sup>1,\*</sup>

- <sup>1</sup> Chemical Engineering, Polytechnique Montreal, 2500 Chemin de Polytechnique, Montreal, QC H3T 1J4, Canada; emilie.thibault@polymtl.ca (É.T.); francis.lebreux-desilets@polymtl.ca (F.L.D.); moncef.chioua@polymtl.ca (M.C.)
- <sup>2</sup> Industrial Algorithms Ltd., 15 St. Andrews Road, Toronto, ON M1P 4C3, Canada; jdkelly@industrialalgorithms.ca
- <sup>3</sup> CanmetENERGY, 1615 Bd Lionel-Boulet, Varennes, QC J3X 1P7, Canada; bruno.poulin@nrcan-rncan.gc.ca
- \* Correspondence: paul.stuart@polymtl.ca

**Abstract:** Data management systems are increasingly used in industrial processes. However, data collected as part of industrial process operations, such as sensor or measurement instruments data, contain various sources of errors that can hamper process analysis and decision making. The authors propose an operating-regime-based data processing framework for industrial process decision making. The framework was designed to increase the quality and take advantage of available process data use to make informed offline strategic business operation decisions, i.e., environmental, cost and energy analysis, optimization, fault detection, debottlenecking, etc. The approach was synthesized from best practices derived from the available framework and improved upon its predecessor by putting forward the combination of process expertise and data-driven approaches. This systematic and structured approach includes the following stages: (1) scope of the analysis, (2) signal processing, (3) steady-state operating periods detection, (4) data reconciliation and (5) operating regime detection and identification. The proposed framework is applied to the brownstock washing department of a dissolving pulp mill. Over a 5-month period, the process was found to be in steady-state 32% of the time. Twenty (20) distinct operating regimes were identified. Further processing with the help of data reconciliation techniques, principal component analysis and k-means clustering showed that the main drivers explaining the operating regimes are the pulp level in tanks, its density, and the shower wash water flow rate. Additionally, it was concluded that the top four persistently problematic sensors across the steady-state spans that would need to be verified are three flow meters (06FIC137, 06FIC152, and 06FIC433), and one consistency sensor (06NIC423). This information was relayed to process experts contacts at the plant for further investigation.

**Keywords:** industrial data; data processing; steady-state detection; data reconciliation; operating regime; framework



**Citation:** Thibault, É.; Kelly, J.D.; Lebreux Desilets, F.; Chioua, M.; Poulin, B.; Stuart, P. Industrial Data-Driven Processing Framework Combining Process Knowledge for Improved Decision Making—Part 1: Framework Development. *Processes* **2023**, *11*, 2376. <https://doi.org/10.3390/pr11082376>

Academic Editor: Wei Sun

Received: 16 June 2023

Revised: 26 July 2023

Accepted: 2 August 2023

Published: 7 August 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Data available in today's industrial processes are abundant and contain the knowledge that can help plants with their decision-making processes. Examples of decisions include identifying which products or recipes are not profitable (cost analysis), which generate the most emissions (environmental analysis), which requires the most resources (energy analysis), when should maintenance be performed (predictive maintenance), and how to best schedule production (optimization analysis). However, big data does not always include good data only; raw process data alone have no refined value and are of limited use until they are converted or transformed into information. Measured process data

from sensors and instruments are the source of process data, but are however inherently corrupted and distorted by different types of errors and therefore data cleaning techniques must be applied before incorporating these data in any analysis [1]. Plants need to pre-process their big data to obtain useful good data of reproducible and reliable quality.

Measurements taken in industrial processes are generally impacted by errors that can be grouped into three main categories: random and gross errors as well as abnormalities. The presence of these errors can be evidenced by noise in process data and the inconsistency between the measured values and the material, energy and momentum balances related to the process. This makes the direct use of raw data inefficient. In order to overcome these issues, data filtering and reconciliation have become established industrial practices [2]. Data processing is widely used for time series data coming from sensors, enabling improved decision making [3–6]. Data manipulation approaches allow us to take full advantage of the information collected and provide many benefits such as better plant knowledge, reduced downtime (hence, throughput increase), decrease in maintenance costs, reduce over-specified feed quality, increase in product quality, minimized energy consumption and improved energy efficiency [7].

An adequate elimination of the errors present in the measurements enables us to increase the effectiveness and efficiency of decisions related to the operation of chemical processes. The latter include the configuration of the control systems, detection and diagnosis operating problems, planning, scheduling, and coordinating, identifying equipment maintenance requirements, and real-time optimization of the process operations and environmental, cost and energy analysis.

The operation of a process can exhibit different regimes (operating strategies) that may be characterized by feedstock changes (nature or quality of feedstock), diverse operating or processing conditions, different products runs and product switching periods, seasons, production rates, maintenance periods, equipment modes, shutdown and ramping-up periods, and recipes.

Segmenting industrial process sensor data (time-series data) into different operating regimes opens the possibility of assessing energy (or resources) consumption, detecting ineffectiveness and inefficiency, monitoring performance, cost, yield, and product quality for each regime. This analysis should help dictate and direct the best way to operate a plant; the identification of the most economic, effective, and efficient operating regimes also determines the least interesting ones [1]. The detection and identification of these regimes may also help debottleneck the process and impact the production coordination, scheduling and planning of the plant.

Process decision making is directly dependent on an adequate interpretation of the process variables trends. We have established that industrial process data must be processed prior to taking any decisions. However, the complexity and specificity of chemical processes require the consideration of a priori process knowledge when analyzing industrial data for decision making in order to extract and analyze these trends and behaviors. The decision-making process should not be disconnected from process knowledge, but must exploit data-driven approaches. There is a need for a systematic methodology that would balance both dimensions and prepare data for future decision-making applications; such methodology is presently unavailable—frameworks presented in Section 2 have limited scope or major limitations.

Therefore, this paper proposes an operating-regime-based data management framework (data quality improvement strategy) that combines the interpretation of the process operation (understanding of the process) with data-driven mathematical approaches for design, production, process and/or operation related decision making. This systematic and structured way to analyze historical process signals for making offline strategic decisions puts a specific emphasis on operating regime detection and identification (ORDI). This practical framework will be applied to the data of a brownstock washing department of a dissolving pulp mill.

To pull together this architecture, Section 2 presents the existing data processing frameworks that are established for industrial decision making and reviews existing methods for industrial process operating regime detection proposed in the literature. Then, Section 3 describes the steps employed to obtain the proposed operating-regime-based data processing framework and Section 4 presents the framework based on the combination of process knowledge and data-driven approach. This section demonstrates the applicability of the framework as well on data from a dissolving pulp mill. Section 5 gives a conclusion of the present work.

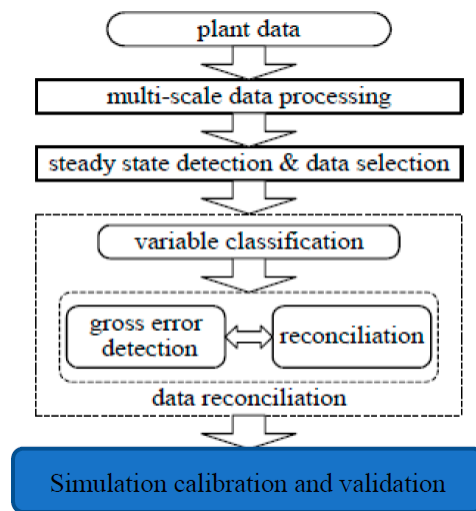
## 2. Review of Data Processing Frameworks for Industrial Decision Making

Industrial processes have a data management strategy (how data are collected, stored, and accessed) as well as a data processing strategy (improving their quality), together they lead to decision making. Software offering a range of data processing steps for various decision-making nature are presented in [8]. The data manipulation and treatment steps required vary from one application (decision) to another [8]. Data processing frameworks employed for industrial decision making, focusing on offline process diagnostics, troubleshooting, optimization, and cost analysis, are summarized in Table 1 to compare the steps undertaken in each of these with what is proposed in this publication. The objective is to assess the pros and cons of existing frameworks, and point out what is missing. Each step could be performed with various techniques. The focus of this publication is not to compare these techniques explicitly, but rather to develop a framework that contains all necessary step, and that these steps are performed with techniques that consider process knowledge as much as the data-driven aspects.

**Table 1.** Comparison of frameworks.

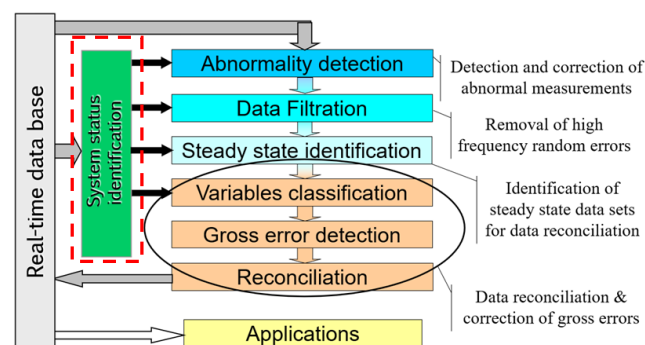
|  | [9]                                   | [10]                           | [1]           | [11]         | [12]         | This Publication             |
|--|---------------------------------------|--------------------------------|---------------|--------------|--------------|------------------------------|
| Scope or final application of processed data | Simulation calibration and validation | Improve accuracy and precision | Cost analysis | Optimization | Optimization | Business operation decisions |
| Data synchronisation and imputation          | No                                    | No                             | No            | No           | No           | Yes                          |
| Inter- and Intra-unit lag correction         | No                                    | No                             | No            | No           | No           | Yes                          |
| Detection and correction of outliers         | Yes                                   | Yes                            | Yes           | No           | Yes          | Yes                          |
| Noise reduction                              | Yes                                   | Yes                            | Yes           | Yes          | Yes          | Yes                          |
| Steady-state detection                       | Yes                                   | Yes                            | Yes           | Yes          | Yes          | Yes                          |
| Data reconciliation                          | Yes                                   | Yes                            | Yes           | Yes          | Yes          | Yes                          |
| Operating regime detection                   | No                                    | Yes                            | Yes           | No           | No           | Yes                          |

As data must be processed for a targeted specific application, it is critical to highlight the scope of the analysis and take into consideration the period that is under study here. Ref. [9] proposed a systematic strategy (Figure 1) to improve and validate a steady-state simulation model of a recaustizing plant in a pulp mill. The processed and reconciled data showed that the simulation is reasonable for practical application such as process sensitivity analysis, investigation of pulp production increase and providing insight related to capital effective options for plant upgrade alternatives.



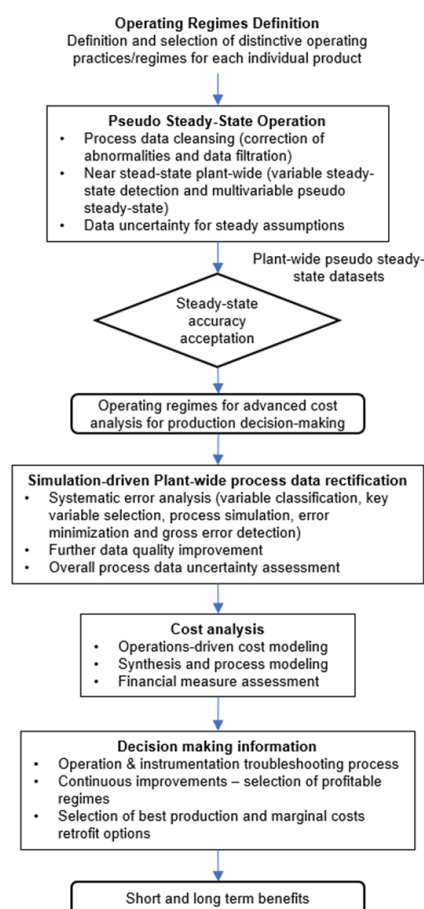
**Figure 1.** Strategy for improving data quality for simulation calibration and validation, adapted from [9].

Based on Jiang's work, Ref. [10] suggested a sequence of steps to increase value by improving the quality of real-time process measurements (Figure 2). Bellec's data processing framework aims to improve the overall accuracy and precision of process data. The selected data treatment techniques are suitable for online applications. Ref. [10] claimed that by compiling steady-state reconciled data in different operating regimes, plants can build a database of accurate process data. The database could lead to benefits related to process operation and troubleshooting such as identification of process leaks and instruments failures for improved process yield [10]. The understanding of the process is improved from knowing the different process operating conditions and plant-wide, or multi-unit-wide optimization could be done by coupling operational data and operating cost data for specific products in particular operating regimes [10].



**Figure 2.** Methodology to improve data quality proposed by Bellec et al., adapted from [10].

In 2011, Korbel proposed a framework (Figure 3) that uses plant-wide rectified process data in cost analysis for identifying short- and long-term benefits, i.e., profitable and unprofitable regimes [1]. The reconciled data obtained following the last step of his framework is used for advanced cost analysis for production decision making; this operations-driven cost analysis is performed through an activity-based cost modelling approach for a given operating regime.



**Figure 3.** Data processing framework for cost analysis, adapted from [1].

Ref. [11] detailed the necessary steps to preprocess data before utilizing them in real-time optimization for implementation. Moreover, Ref. [12] proposed a sequence of data processing steps that lead to process optimization. For instance, they put forward that the essential steps that need to be followed include: (1) data acquisition, (2) data treatment, (3) steady-state detection, (4) data reconciliation, (5) optimization, and (6) solution validation. During the data acquisition stage, the user specifies whether the algorithm will operate offline, independent of the plant's actual state, or online, where values are directly retrieved from the process database. The proposed framework can be applied to all decisions related to business operation, i.e., environmental, cost and energy analysis, optimization, fault detection, debottlenecking, etc.

None of the aforementioned frameworks acknowledge the data synchronization and imputation step. The importance of this step is explained in Section 4.2.

Regarding the inter- and intra-unit lag correction steps, Ref. [10] ensure that the lags in the process are considered by retaining only steady-state periods longer than a threshold value, i.e., the system delay plus a safety factor. Therefore, they targeted steady-state periods long enough for them to compensate for the fact that the various departments in a process are not synchronized.

Both Jiang's and Bellec's frameworks achieved the detection and correction of abnormal measurements by employing a method based on wavelet transforms [13,14]. In the second step of Delou's framework, each input variable may undergo a linear transformation to scale or change units, while also eliminating any unwanted or out-of-limit values. This step is repeated iteratively until all potential errors have been eliminated.

Moving on to the noise removal step. For [9], the data processing step reduced errors and extracted key process trends. To do so, the authors performed a multi-scale data

processing using wavelet transform. They discarded random noise, and extracted process trends by using an approach proposed by [13]. For Bellec, a data filtration technique also based on wavelet transform [14] is used to correct random errors. For Korbel, noise and abnormalities are extracted simultaneously from the process measurement trends by applying optimal wavelet transform parameters which are adjusted iteratively until optimal data pre-processing and accurate steady-state detection are perceived to be achieved [1]. In Reyes's framework, signal denoising was performed using both short-time Fourier transform, and wavelet transform, with a preference for the latter. The authors emphasized the significance of selecting an appropriate mother wavelet and concluded that the Daubechies wavelet family provided the most accurate results in their case.

In Jiang's framework, the steady-state detection is performed using a method based on wavelet transform presented by [13]. In Bellec's work, steady-state operating periods are carefully identified based on the approach mentioned in [14]. For Korbel, the identification of steady-state operation is performed within each of the operating regime under analysis. The denoised signals, series or trends are analyzed for steady-state occurrences using a methodology based on wavelet transform presented in [15]. The detection of steady-state periods is done in two steps, (1) univariately and (2) in a multivariate way by comparing the whole set of key variable states over time. In Reyes's framework, steady-state identification is accomplished using the wavelet method proposed by [14]. Delou et al. mentioned how steady-state detection becomes crucial before proceeding to the reconciliation stage, as it ensures the reliable estimation of the optimization model parameters. Ref. [12] employed two statistical tests to detect steady-state conditions.

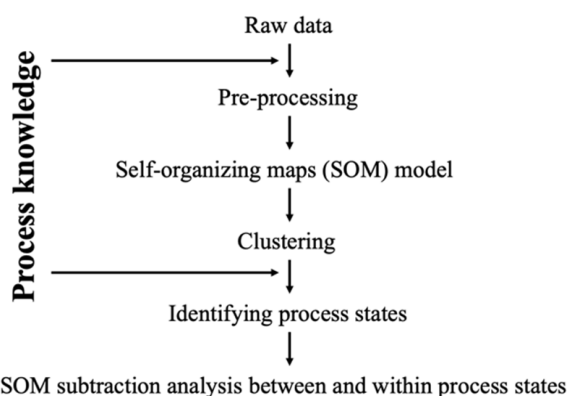
The data reconciliation step is performed using the Sigmafine V. 3 software (Sigmafine, 2022) in Jiang's framework. For Bellec, gross errors are corrected by using a weighted least squares method. In Korbel's framework, plant-wide data reconciliation assesses the data validity by comparing them to the values of the underlying process model, i.e., a simulation that accounts for the material and energy balances and/or other laws of conservation. The details behind the steps performed for data rectification are presented in [1]. To detect gross errors, Reyes et al. employed Hampel's redescending M-estimator and minimized the generalized maximum likelihood objective function. This process was carried out for 24 steady-state intervals, each containing 8000 data points. Regarding Delou's framework, data reconciliation is formulated as a nonlinear programming problem. Once the reconciliation is performed, the optimization problem can be solved, and the resulting solution can be validated.

Operating regimes should be considered in order to account for the mill or plant variability. In [10] framework, the authors included a system status identification where the main purpose is to identify the conditions of operation for the system under study and select the data processing parameters associated with the identified process status (operating regime). The selection is based on an offline investigation of all steady-state operating conditions and that the parameters are set to achieve, as claimed by [10], optimal results. Ref. [10] are vague regarding the optimality of the results, but process knowledge seems to be involved. Additionally, in the validation step of the framework presented by Bellec et al. on historical data from a paper machine, the authors included the notion of operating regimes also known as modes or grades. They underlined that a process may have many different steady-state operating regimes. Those are identified for the system under study and then the characteristics associated with the different operating regimes are compiled and stored in a database. The probability of occurrence, the duration of each regime and the delay occurring between those are assessed. Additionally, the mean and variance of process measurements within each operating regime are also compiled and stocked in a database. These characteristics, associated with each process measurement within an operating regime, are useful to properly identify the operating and processing conditions of the studied process [10]. In Korbel's framework, the initial step consists of defining operating regimes, i.e., characterizing the operating differences that occur while producing each specific product. According to [1], operating regimes are established when

changes occur in process design, when different equipment are used, when changes occur in production target or in setpoint control strategy as well as feedstock diversity.

Changes in processing conditions, such as long-term evolution and transition from one regime to another, have often been ignored when analyzing industrial process data [16]. Modern automated data acquisition systems, improved data storage capacity and retrieval, combined with the proliferation of inexpensive sensors and instrumentation massively increased availability of data. This massive amount of data is not easy to interpret by conventional means. To exploit and take advantage of this data, chemical engineers must use data science tools, such as those discussed in this paper. Industrial (chemical) plants commonly operate in a finite set of operating regimes, which may be classified as modes where process variables vary within a relatively narrow band, i.e., are at steady-state [17]. On the other hand, transition or sequencing periods are characterized by large changes in one or more variables [17]. The latter represent the state of the process between two modes, e.g., a change in setpoint, the opening of a valve, and/or a change in equipment configuration. Research has shown that the behavior of a process may differ greatly between modes [16].

Archived process data can be used to cluster or characterize process states [18]. Heikkinen et al. grouped variables by the *k*-means algorithm into clusters that represent different process states [18]. The state of the process is determined through data analysis. The clusters are analyzed to indicate differences in process states by examining their properties and by using process knowledge. A cluster representing periods of plant shutdown is easily identified based on very low volumetric flow. Heikkinen et al. findings showed that seasonality is one of the clearest distinguishing factors between clusters. One cluster is prevalent in the summertime, another in the wintertime, a third represented periods when the whole process is either out of control or shut down, and a fourth represented periods when the process is unstable or unsteady. Various identified process states gave information about its process behavior. This information is used to determine optimum setpoints for key process variables within a process state. Process knowledge is needed for variable selection and for associating and allocating process states to clusters (Figure 4). There is always measurement noise that affects accuracy; therefore, the variables are filtered by a moving average filter (Figure 4).



**Figure 4.** Data processing chain diagram for self-organizing maps (SOM), adapted from [18].

The idea of using *k*-means clustering to create subsets representing process states is taken up by Liukkonen et al. as well [16,19]. One of the most commonly used unsupervised clustering methods is the *k*-means algorithm [20]. Perhaps the biggest issue when trying to implement this technique is the requirement to set the number of clusters in advance; this is rarely known for complex processes operating under unknown conditions or in a changing environment [21]. Therefore, Ref. [21] used another unsupervised clustering technique to detect operating regimes in condition monitoring data, e.g., vibration signals, called the variational Bayesian Gaussian mixture.

Traditional clustering techniques are computationally expensive and generally perform poorly on time-series data because they are intended for more time-independent observations, Ref. [22] proposed a two-step clustering method based on principal component analysis (PCA). Chemical plants normally operate in a number of operating states (e.g., plant start-up, grade change, shutdown), including steady state, and frequently switch between them. Therefore, in the first step, process states are classified into modes, i.e., the process unit operates in steady-state and transitions, using a novel multivariate algorithm to segment historical data. A PCA algorithm is then used in the second step to compare the different modes and transitions to cluster them based on their similarities. Process control, simulation, fault detection, and alarm management are examples of applications whose parameters must be adjusted to fit the current process state [22].

In practice, translating the obtained clusters to operating regimes periods requires input from operating personnel, i.e., adequate knowledge of process history. When a clear change appears in the operating regime, it might be something new, unknown, or something very common and already identified, i.e., a specific operating mode, and the plant have criteria to identify that regime. It takes well-defined criteria or targeted measurements to know in which operating regime the process is operating under. Distinct operating regimes should be named and characterized (is this regime repeatable, is it relevant to the process, should engineering personnel know about it and act on it, etc.). Ref. [23] used cluster analysis [24] as a framework to simultaneously identify process states, detect transition periods between states, and label times of occurrence using historical time-series data. As part of the cluster interpretation, the most desired operating regimes are identified on the basis of knowledge of the process history.

Even if the notion of process knowledge is considered [18] or mentioned [23], studies, approaches or frameworks that identify operating regimes using a combination of process knowledge and data-driven approaches are lacking in the literature. Therefore, this paper attempts to address this by setting-up an operating-regime-based data processing framework in which regimes are considered and detected using both process knowledge and clustering analysis (principal component analysis and *k*-means).

Prior to operating regimes detection, process data are reconciled. The idea of integrating data reconciliation and principal component analysis have been presented before for energy monitoring [25], fault detection [26,27], smart energy distribution network [28], analytical chemistry [29], data processing [30] and improved data reconciliation [31,32].

Clustering in general, i.e., not only for industrial process operating regime detection, can be performed with a plethora of techniques and algorithms, some being very recent. More details can be found in the following publications regarding expectation-maximization algorithm [33–36], *c*-means [37–40], principal component analysis with *k*-means algorithm [41–44], discriminant analysis [45,46], Kohonen neural network [47,48], genetic algorithm [49–52] and graph clustering algorithm [53–56].

Therefore, compared to [9] framework, Ref. [10] considered operating regimes. To the best of our knowledge, the operating regimes are manually identified by experts in the process, no data-driven techniques were referred to. Additionally, the lags in the process are considered but not systematically and rigorously accounted for; to eliminate them, Ref. [10] retained only steady-state period that lasted a certain amount of time. Additionally, there is no mention of an identification of steady-state periods for a whole system, i.e., a unit-wide or plant-wide steady-state detection. With Korbél's framework [1], the operating regimes are defined based on process knowledge and sound engineering judgement. Therefore, no data-driven techniques were mentioned for operating regime detection in Korbél's framework. Furthermore, there is no mention of data synchronization, imputation, inter- and intra-unit lag correction. Hence, it is found in the literature that most of the data management frameworks for offline decision making considered data processing steps (noise reduction, outlier detection and data reconciliation), but their scope is not always explicit, and operating regime detection is not systematically considered. Neither Reyes et al. nor Delou et al. approach consider operating regimes detection. This paper proposes



a framework that combine process knowledge and data-driven techniques at each step and addresses these limitations. Using process operation knowledge in all steps of data analytics is also addressed in [57]. They built an inferential sensor model to predict the impurity in the product stream for the Dow data challenge problem.

### 3. Methodology: Operating-Regime-Based Data Processing Framework

This paper presents a systematic and structured approach inspired by the frameworks presented in Section 2 to process industrial signals to assist in the making of strategic management decisions. These decisions would be made offline. This framework, based on the combination of process knowledge and a data-driven approach, considers a scope definition, data processing, steady-state detection, data reconciliation and operating regime detection (a specific focus is put on this last step). The proposed framework employed various tools such as EXPLORE version 2.2.0.814 [58], MATLAB V. R2021a, Excel V. 2306 and IMPL-DATA V. Release 1.7.

As part of the framework proposed here, the data reconciliation is performed using IMPL-DATA [59]. The latter is a data reconciliation modeling and solving platform, implemented in Microsoft Excel, to perform any type of unit-wide and plant-wide nonlinear data reconciliation application in off-line and on-line environments. IMPL-DATA is specifically streamlined and suited to the interactive workflow of DR in terms of its gross error detection, identification, and elimination capabilities. Additionally, it contains a smart model building capability to configure small to large nonlinear data reconciliation problems. The straightforward formulation of these models is based on units such as mixers, splitters, and processes interconnected via streams where quantity and quality meters (flows, holdups, densities, etc.) may be assigned or mapped.

In summary, the suggested methodology exploits the advantages of data-driven approaches and uses process expertise as complementarity information sources. This is a generic approach that can be applied for a wide range of industrial processes.

To obtain this architecture, a literature review of existing data processing frameworks is accomplished. Their strengths and weaknesses are highlighted. That leads to the formulation of a series of steps (framework) that address the identified potential improvements to existing frameworks; it is a synthesis of practical data management techniques. Then, the proposed framework is tested in a case study. Finally, the limitations of the approach suggested are presented in the conclusion for them to be addressed in future work.

### 4. Results and Discussion

Following the framework development, an application is demonstrated on data from the brownstock washing department of a dissolving pulp mill in order to put this architecture together and interpret data for decision making.

The following sections present a framework for which all steps consider the combination of process knowledge and data-driven approaches as well as its application to the brownstock washing department on a dissolving pulp mill. The synthesized framework have five major steps: scope definition, data processing, steady-state detection, data reconciliation and operating regime detection. It is based on the limitations presented in Section 2. The objective of the framework is to provide plant personnel with rectified and reconciled data in each operating regimes to be used in various analysis (environmental, cost, energy, debottlenecking, fault detection, troubleshooting, etc.) for improved decision making.

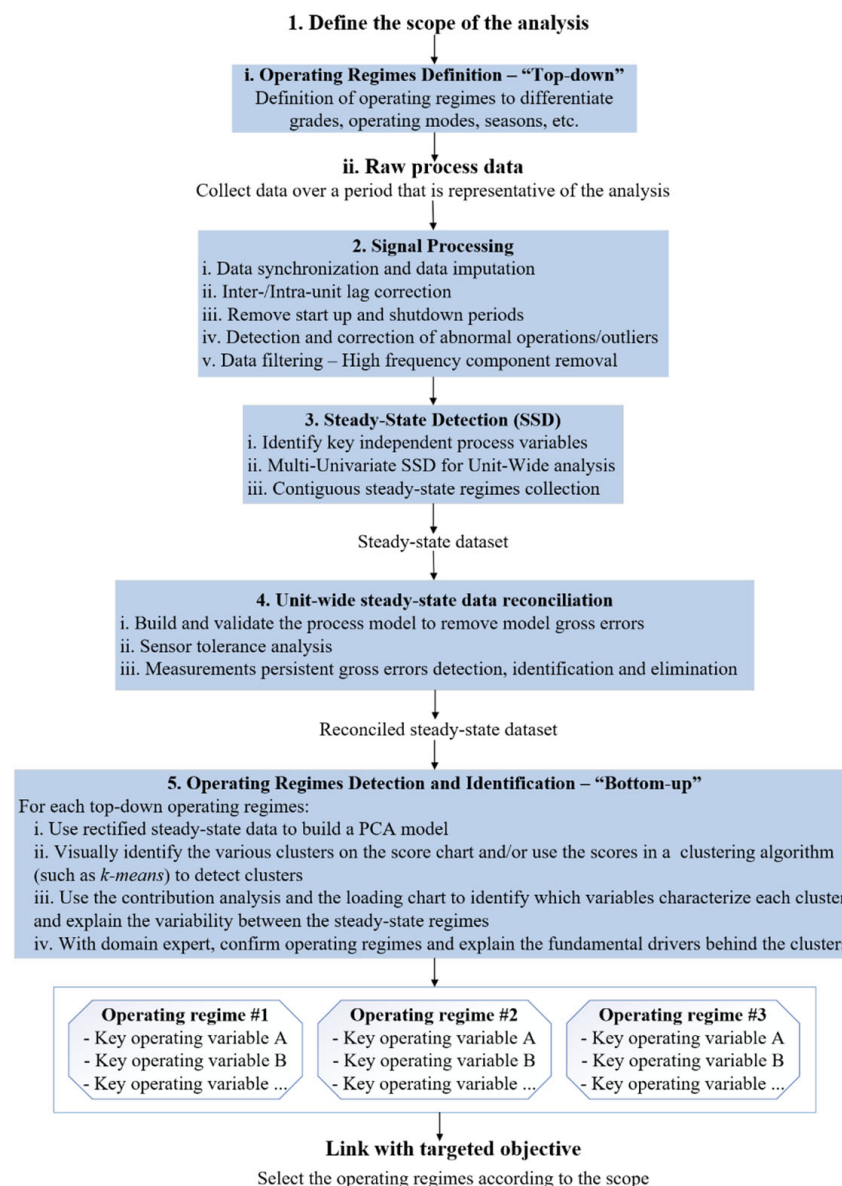
Measurements taken in industry contains several types of errors that can be grouped under three main categories: abnormalities, random errors, and gross errors. These errors are corrected in a series of five steps. First, given the fact that data are always processed according to a specific and targeted goal, the objective of the analysis (the scope) is defined, and data are collected (step 1). Then, data are processed, i.e., data undergo a synchronization step, a lag correction, the abnormal operations are removed, and data are filtered to remove unwanted high frequency components (step 2). The third step is to detect the steady-state data series that will allow the data reconciliation step (step 4). In step #5, the

various operating regimes are detected and identified. This framework is limited to offline operating data (as opposed to real-time online process data) as they are an important source of process insight.

The proposed framework focuses on data that are representative of normal and desirable operating conditions. Abnormal operations, fault detection and diagnosis are critical, but are not the subject under advisement in this paper. Therefore, shutdown and startup periods as well as outliers are removed, and steady-state periods are detected.

Lastly, all steps have their own internal feedback loop. For instance, in the data reconciliation step, the moment one gross error is eliminated, another may be found; a practical way to perform data reconciliation is to remove gross errors one at the time.

Figure 5 depicts the proposed methodological framework.



**Figure 5.** Framework combining process knowledge and data-driven approaches.

#### 4.1. Step 1: Scope Definition

Any data processing analysis starts by defining the objectives of the analysis, e.g., optimizing the production sequence, reducing the operating cost in a plant's department, identifying unprofitable operating regimes. It is critical to put forward why this analysis is performed, what are the data used for and what is being studied. Therefore, the oper-

ating regimes known and recognized by the process experts, i.e., the different recipes as expressed in standard operating procedure that are related to the analysis objective, should be acknowledged. Operating regimes are used to answer questions, solve problematics, and make decisions in industrial processes. Depending on the question, the problematic or the process decision to be made, specific operating regimes will be required.

“Top-down” operating regimes are defined using process knowledge, they are obvious and easily detectable; process experts are aware of those. These operating regimes are driven by changes in the control setpoint strategy caused by different operating conditions (seasons, product, feedstock changes, production rate, maintenance, the usage of different production lines, etc.). The fact to the matter is that those setpoints do not account for everything happening in industrial processes. In fact, even if operating regimes are defined by standard operating procedures (SOP), setpoints are changed by operators since a plant is not 100% automated. The process can be operated in automatic or manual mode; some variables are left in automatic control all the time while others are not. Furthermore, it could be possible to observe differences in some variables when an equipment needs maintenance, hence before and after maintenance. Moreover, equipment performance changes all the time because of fouling, deterioration, therefore the recipe should be adjusted accordingly. Hence, operators are acting when something changes in the process; there could be a problem in the process, variation in raw material, hence the process needs adjustments. Additionally, two different operators will not necessarily take the same action. These considerations are made later on with the “bottom-up” operating regime detection through clustering analysis and the drivers of these operating regimes are assessed (process knowledge is necessary to understand why variables act as they do). In the scope definition, only the “top-down” operating regimes are considered; the idea is to go as far as possible using the recipes (SOP) considering what is required related to the scope, then those regimes will be sub-categorized by using clusters.

The top-down analysis may be assimilated into a white-box approach since it puts forward how a plant is supposed to operate, how it is designed, what grades are produced, what are the setpoints for those grades, what are the standard operating procedure (SOP) says, etc. On the other hand, the bottom-up can be perceived as the black-box approach; process data, through clustering analysis, may uncover unknown elements about the process.

Then, raw process data are collected over a period that is representative for the targeted analysis. Every process generates a wide range of data types that are collected, gathered, and stored in the process database/historian systems. These include process sensor data (pressure, flow, temperature, level), lab results, manufacturing schedule, production mode, product grade, and many others. Therefore, pertinent data must be gathered considering the objective of the analysis.

#### 4.2. Step 2: Signal Processing

Data must be synchronized to the same time step since they are often stored on the change in value, which is not a regular frequency, and variables used in an analysis will not necessarily be sampled at the same rate (for instance hourly for some lab data versus by the second for some sensors). Compression is used to save data storage space, therefore sensor data are often only logged when the values differ significantly from the previously stored value (change-of-value basis), whereas lab data might be available once every hour or every day. Therefore, to perform any analysis, all data should be synchronized, and preferably, to the frequency of interest. Thus, if the sampling frequency of a variable is lower than the frequency of interest for an analysis, data imputation must be performed (linear interpolation of data, repeat the last valid value) [60]. On the other hand, if the sampling frequency is higher than the frequency of interest, then the mean, the closest value, the minimum value, or the maximum value over a sampling interval equal to the desired time step must be considered, depending on the analysis.

For most analyses, it is not mandatory to have all the sampled data points values. For instance, to determine a trend, a few missing values in a dataset are not a problem [61]. Larger segments without data are, however, unusable in most analysis. Therefore, missing half of the values is a problem in the case of contiguously missing data that results in large empty stretches, but, depending on the analytics algorithm, it might be less of an issue if every other value is missing [61].

Once the synchronization process to set data on a regular time interval is done, the lag is accounted for. For an analysis that is taken place inside one department (system or unit) only, intra-unit lag correction is performed. On the other hand, a plant-wide analysis considers inter-unit lag as well—that is, the lag from one unit to another. For instance, when the production stops in the digester units of a pulp and paper mill, it takes a few hours for the repercussions to be visible at the paper machine. The lag accounts for the time required for the operations, the retention time, the transport time, etc. Lags vary according to the production rate. By ignoring the lags, some correlations may be lost. Furthermore, most of the subsequent steps involved in this framework employed reference variables (independent variables that describe the process) to perform the analysis; therefore, all variables must be aligned in time beforehand. The lag correction analysis is achieved based on Fourier and cross-correlation analysis (software such as EXPLORE v. 2.2.0.814 [58] offers this functionality) [62–65]. This approach is robust to noisy signals.

If the data processing framework is applied on batch processes (as opposed to continuous ones), dynamic time warping should be considered for a standardized reference timeline in addition to the lag analysis as the different batches might have different batch time. More information on dynamic time warping can be found in [66].

As part of the data processing phase, the shutdown and start up periods should be removed. To do so, variables of reference are selected to detect those periods. Even when the process is stopped, some pieces of equipment are still functioning, hence some sensors are still collecting data and therefore could not be used as reference. Using reference variables, specific periods are detected visually and deleted from the dataset by numerical validation, i.e., all values under a certain limit that are not representative of the normal and desirable operating conditions. The threshold values are specified by a process expert. The removal of process shutdown and start up events from the reference variables are mirrored onto the other variables used as part of the analysis (values at certain time stamps will be eliminated for all variables). Lastly, additional data points beyond the specified period which are impacted by a process shutdown (a time delay/interval or buffer around these periods) are removed to account for the time required to shut the system and to reach normal operations again. In other words, periods of downtime are extended to include transient periods. The time intervals removed around downtime is chosen based on operating practice and process expert judgement.

Alongside shutdown removal, an important step in signal processing is to detect and remove outliers [67]. Therefore, once downtime periods are eliminated from the dataset, outliers, i.e., observations that deviates from the expected behavior of a process variable—values that are out of range, are removed. These values are often of short duration and present abrupt changes in the signal. The fact that top-down operating regimes are defined earlier on helps in the identification of outliers as segmenting process data reduces variability (and the standard deviation), and thus makes it easier to detect process deviations and avoid unnecessary rejects. Similarly as what is done for the shutdown and start up periods, variables that represent the system under analysis (key process variables) are used to detect outliers. The quartiles analysis [68] is used to identify and remove outliers univariately in the key process variables [69]. This removal is reflected in all the other variables. The quartiles analysis is based on the median and is therefore independent from the distribution and more robust (and efficient) towards outliers detection.

Additionally, a spatial correlation-based outlier detection analysis is carried out using principal component analysis (PCA) to validate the results of the quartile analysis. The PCA identify the residual values by extracting the principal components of the dataset, and

these are evaluated through detection mechanism such as Hotelling  $T^2$  score and squared prediction error to find important discordant data points—observations falling outside the (95% or 99%) confidence level of the model [70]. The Hotelling's 95 or 99-percentile appears as an ellipse on the score chart, showing how each observation fits within the model relative to all the others. Data points around the outliers might also be removed to ensure that the period of disruption, i.e., the period of abrupt change in the signal, is covered. Once identified and removed, abnormal data points may be corrected using imputation. This value can be predicted using previous data, interpolation or by using the mean value. Lastly, abnormal data can also be labeled as missing data or non-naturally occurring number (−99.999). References that considered PCA for outlier removal include [57,71].

Afterwards, in order to reduce noise that affect a signal by making it less representative of the true process state (random errors) and, therefore, distinguish the process trend, signals are filtered. Various sources of noise add a random component to the signal fluctuations. To achieve that, the wavelet transform are used. Some approaches reported in the literature are compared in [72]. It is highlighted that low-pass filters do not have the ability to reduce noise present at different frequencies and that lead to the development of multi-level analysis. In this category, the use of wavelet transform is preferred to that of short time Fourier transform because of the weaknesses towards temporal localization of the latter. The wavelet transform technique offers simultaneous localization in time and frequency domain, it preserves important signal features while removing unwanted components (noise); it is able to separate the finest details in signals. Finally, to ensure that the filtering does not induce unwanted lag, a second lag analysis might be performed.

Similarly as for the outlier detection, the consideration of the inherent variability of the data from the top-down operating regimes definition before noise reduction can benefit this data processing step as the process noise amplitude can fluctuate with operating conditions, and therefore the filter parameters must be adapted [73].

#### 4.3. Step 3: Steady-State Detection

Industrial process signals describing a process cover both transient and steady-state operations. The former happen between steady-state operating periods and occur because of environmental changes, upsets, changes in setpoints, etc. Plants are tracking and reporting what is happening during these transient periods; the analysis of these periods includes how long they lasted, how to minimize the transition time and the asset reliability (equipment failures happening during transient states). However, this framework focuses on steady-state periods to provide a better understanding of how the process operates as well as significant information for offline management decision making to improve the process analytics. Therefore, transient periods are not considered.

In order to detect when a process is operating under steady-state conditions, the first step consists of the identification of critical variables that will be used as the basis for the steady-state detection (SSD). This identification is done using process knowledge. The variables should be those that best characterize the stability of the system and should additionally be independent, i.e., controlled by different setpoints or variables, and not correlated, i.e., do not vary together at the same time. When all the key process variables are simultaneously in steady-state, the system is said to have reached steady-state conditions. An alternate solution would be to claim steady-state if some percentage of the critical variables are at steady-state. This percentage value would vary depending on the complexity of the process.

The SSD algorithm employed as part of this framework is inspired by the one presented by Kelly and Hedengren in 2013 [74]. This window-based algorithm utilizes the Student- $t$  test to determine if the difference between the process signal value and its mean is above or below the standard deviation times its statistical critical value—below would mean that the time instant is steady whereas above would indicate that it is unsteady.

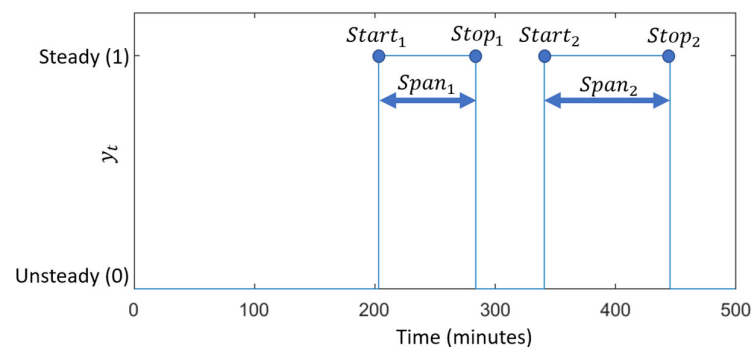
This approach offers a practical, efficient, and robust way to detect steady-state conditions, the implementation is computationally simple (easy to apply/implement or easily

implemented), the criteria are generalizable, the results are reproducible, easily interpretable, and reliable, and lastly, process knowledge can and must be combined with the approach—there are adjustable parameters.

Furthermore, this SSD algorithm can accommodate noisy data, multivariable analysis and is scale independent. The computational load imposed by the approach is minimal as it only involves calculation of a mean, standard deviation, and slope. Lastly, this effective method gained industrial acceptance as it is simple both in concept and implementation. It has been extensively applied to commercial/industrial-scale multivariate processes/processing units. The details of this algorithm are presented in Appendix A.

The algorithm proposed by Kelly and Hedengren was modified. First, in order to increase the user input and process knowledge, the estimated standard deviation of the noise can be provided to the algorithm if it is known by the user. When the standard deviation is endogenously computed, the result is rather liberal (looser); it will most likely be bigger than an externally supplied standard deviation that could be more conservative (tighter/smaller).

Additionally, the modified steady-state detection algorithm detects when a steady-state period starts and stops, and then calculates the span range for each period (Figure 6). The span is a collection over multiple signals (all key process variables). The algorithm is able to keep track of the date where steady-state periods are detected using a time index vector. The start range returns the starting index when  $y_t$  vector first becomes steady for all found spans, then the stop range returns the time index when the last contiguous  $y_t = 1$  is obtained, i.e., when the  $y_t$  vector switches from 1 to 0, and the span range returns the number of data point or time index in the interval (the length of these spans in terms of the number of indexes when it is contiguously steady). The span range considers additionally the number of data points in the time-window. Finally, the SSD algorithm returns some summary metrics based on the number of indexes per span such as the maximum and minimum number of data points in a span, the span mean, mode, median and variance. All data windows or span deemed acceptable (how reached a minimal period of time) are then reconciled. This idea is consistent with the fact that in process operations, there are sporadic and persistent steady-state periods. When looking for something that is persistently steady, a minimal period for steady-state operation should be considered.



**Figure 6.** Representation of start, stop, and span detection for all contiguous steady-state periods.

Hence, following the detection of steady-state periods, a threshold (criteria) for the minimal period of time for the steady-state operation to be deemed representative is considered. The minimal steady-state period is established based on process knowledge; this criteria is often set as the average time required to reach steady-state in the process. This is a generally accepted justification for the threshold as it is generalizable, reproducible, and robust.

#### 4.4. Step 4: Unit-Wide Steady-State Data Reconciliation

Following SSD, the remaining time-series data are reconciled. Data reconciliation (DR) consist of identifying measurement systematic (gross) error—inconsistencies with respect

to known conservation laws—and more specifically persistent gross error. This is done by fitting the data to the material and energy balances. This analysis does indicate the presence of systematic error, but does not specify which instruments (sensors) are wrong—and whether or not there are more than one sensor in fault.

The primary purpose of applying DR is to improve industrial plant data considering the inherent uncertainty and complexity in the process measurement systems. This is done by making sure they satisfy all material, energy, and momentum equality constraints or balances and any other inequality constraints or bounds that may be justifiably included. Therefore, DR leads to the identification/diagnosis of defective, faulty and/or inconsistent measuring instrument. Furthermore, DR provide estimates of unmeasured quantities and qualities to be used, for instance, in daily, weekly, or monthly process, production and/or yield accounting reports. When substituting the raw measured values by the reconciled and estimated (regressed) unmeasured variables, all balances equal zero.

The process of data reconciliation aims to align measured process data with the principles of conservation of matter. By solely relying on these conservation laws, gross errors can be identified. However, if these laws do not reveal any errors, it does not mean that the data are necessarily free of gross errors. More complex models, incorporating transport phenomena, fluid mechanics, and reaction kinetics details, may uncover additional errors. Material, energy, and momentum balances are typically just a subset of the available constraints, and adding more constraints may uncover errors that were not detected using only these three [75].

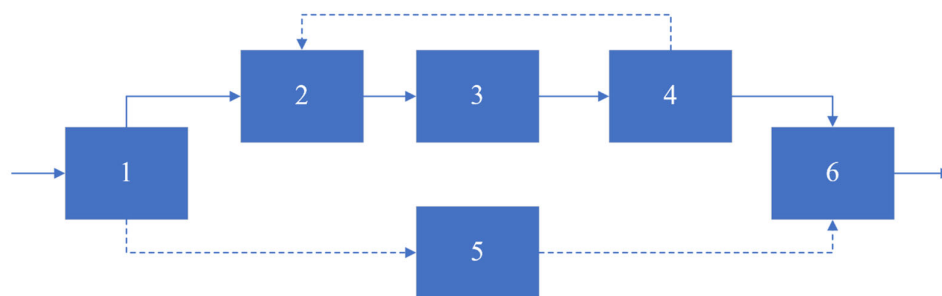
Prior to running the DR algorithm, the process variables are categorized as measured, unmeasured, or fixed (constant in the process). After running the solver, measured variables can be indicated as redundant or not whereas unmeasured variables could be marked as observable or not. This classification highlights which variables can be reconciled (redundant variables), which ones can be estimated (observables) and, finally, those whose reconciliation is not possible and whose accuracy remains unchanged (non-redundant). A measurement is redundant when, if it is removed (marked as unmeasured), that value/variable is still observable. On the other hand, an observable variable is one that is uniquely calculated from the model and measurements. Even if data reconciliation can only take place if there is redundancy (the data reconciliation algorithm still works even if the DOF is null; however, all the sensors become non-redundant) in the system, i.e., there exists more constraints or model equations than unmeasured variables, observability is more important. Among unmeasured variables, defining which may be calculated (which one are solvable) is required. Indeed, unobservable variables are non-unique, not reliable and their values are arbitrary in the sense that they only satisfy the model constraints numerically.

If there are unobservable variables in the system, there are three approaches to remove the unobservabilities: (1) add new sensors which requires capital investments, (2) add more equations if possible and (3) reconfigure and simplify the model to remove the unobservable variables, i.e., the model perhaps can be too granular.

Employing the framework may lead to the recommendation of implementing additional sensors to increase redundancy. The sensor network can be redesigned using the data reconciliation concept. That could be achieved either by using an optimization algorithm that would maximize observability, redundancy, and precision or manually with a simulation, adding sensors one at the time, then running data reconciliation, and assess if observability, redundancy, and precision were improved. That would maximize the reliability of decision making based on data interpretation.

Following variables classification, a data reconciliation model based on physical models (thermodynamic, material and energy balance models) is built to identify inconsistencies. This model represents the flowsheet of the plant or sub-plant, it is a subset of the equations used in a process simulation. This step requires a profound comprehension of the process. DR model relates to the fundamental laws of material, energy, and momentum conservation, and it is also dependent of the processing operations (operating modes, grades, equipment set-ups, start-ups, shutdowns, and switchovers as well as recycle/recirculation

loops via bypassing, repetitive cleaning and purging, etc.). From a DR perspective, these variations in the processing operations may be mathematically expressed using the notion of temporary stream variables (versus permanent) with zero/one, open/closed, on/off or active/inactive stream switches. Temporary streams can be switched on or off depending on the state of the process. Usually, industrial plant operations know a priori when mode, grade and/or any other type of processing logic changes will occur via their planning, scheduling, and coordinating department. As such, before, during or after the changeover occurs, manual and/or automatic indicators such as valve actuator positions, key process conditions, and pump/compressor starts/stops will be available to aid in the determination of when and which stream switches changed. The DR model represents all possible streams, both permanent and temporary. Most of the streams are permanent in industrial processes—these are represented as solid lines in Figure 7. Nevertheless, when the process conditions change (going into another operating mode), there are streams switches or temporary streams (represented in dotted lines in Figure 7). Thus, when a process switches to a different operating mode, some flows are no longer required whereas others may be opened just for the time being. Therefore, handling those stream-switch configurations can reliably indicate a certain mode, grade or changeover that occurred and thus reducing the model gross errors possibility, plausibility, and probability of occurrence.



**Figure 7.** Example of a block flow diagram to present permanent and temporary streams.

Next, the model is validated, i.e., making sure that there are no model errors. Moreover, it should be validated for each “top-down” operating regime. Indeed, in order to properly validate the model, the operating regimes should be acknowledged since for every operating mode, the operations change, there might be different streams (temporary streams). Therefore, if the operating mode changes, then the model could be wrong if it is not changed properly. Hence, the temporary streams due to operating-mode or grade switches are properly accounted for in the model structure.

Before validating the model, both model and measurement gross error are to be expected in the process data sets. However, once it is done, data are only left with measurement gross error; there should not be any more modelling errors. Moreover, when the process is unsteady, there will be model gross error because the mass balance equation will not equal zero. Therefore, in order to only identify measurements gross error, unsteady periods are excluded from the analysis, i.e., this is one of the reason SSD is required.

As part of the model validation concept, a preliminary validation stage could be performed if a simulation of the plant process (or some process units) is available. The simulated values would be regressed against the DR model. If no gross errors are present, then a proper (or the actual) validation is performed based on real process data. Afterwards, gross error detection (GED) might take place.

All three concepts (pre-validation, validation, and GED) can use initial values or starting guesses from the variables from a simulation base case. These may be used every time the DR is run, as they “prime” the DR solver with default results. The initial values are used for a warm start, they can influence the solution since it is a non-convex problem (i.e., multiple local optima exists), convergence to the global optimum cannot be guaranteed.



Hence, the starting guesses may find a different local solution. Lastly, these values are only used as default results and are updated during the reconciliation solving.

Once the process model is validated, it is used to analyze typically averaged or aggregated steady-state industrial process data. All the time-windows that are found to be statistically steady are reconciled. The steady-state data reconciliation (SSDR) is executed for the number of spans detected; it is computed based on the average value of each variable over the multiple time steps declared to be steady in the span. The SSDR algorithm is detailed in Appendix B.

#### 4.5. Step 5: Operating Regimes Detection and Identification

The fifth step of the proposed framework is to translate processed steady-state data into operating regimes. This gives a snapshot of the plant and, therefore, the idea is to perform a multivariate data analysis to detect what is the operating regime when the plant is operating that specific way. In order to classify the snapshots, i.e., to identify the operating regime, independent key process variables (variables of interests that are important in a process unit) that distinguish the operating regimes are identified by process experts.

Starting from a “top-down” perspective, the steady-state processed data are used to build a principal component analysis model. PCA exploits historical process data and discovers hidden phenomena that may be useful for detecting unexpected/novel or unknown operating conditions. The principal component analysis allows to represent process variability and identify and understand significant correlations (between variables) that are inherent to the plant operation. It groups together variables with similar characteristics into clusters unmindful of the link between consecutive time steps. Standard PCA ignores dynamicity (it is not a time-series technique), it is treating all data points (time increments) equivalently, regardless of how far apart they are timewise [76].

Then, the principal components scores are used as input variables in a  $k$ -means clustering algorithm [41–44]. Only the components that explain an important proportion (90% for example) of the variation among the operation variables are selected. Based on the result of the principal component analysis—starting with the number of clusters visible on the score or loading chart—the number of clusters detected by the  $k$ -means algorithm is changed incrementally. It is widely recognized that trial-and-error or multiple random number of clusters to retain are tested [42]. The  $k$ -means algorithm is highly performant, it can be used with large datasets. This unsupervised clustering algorithm offers many insights, is simple to implement and the clusters it returns are effortlessly interpretable and visualizable. Lastly, a priori subject matter knowledge can be used to set the number of clusters [77].

These clustered data are assessed by process experts to highlight what makes a cluster different from another and assess whether they all represent distinct operating regimes. The cluster analysis gives process insight, i.e., existing regimes, actions took by operators, consequence of actions, root cause analysis, cause–effect relationship, etc. This insight is useful to make the process more effective, economical, efficient, helps understanding the process, phenomenon, deviation, and improves decision making. This clustering analysis is a precursor of decision making; subsequent analysis (cost analysis, environmental analysis, energy analysis, etc.) is required in to act on the process.

Additionally, the contribution analysis and the loading chart [78] are employed to identify which variables characterize each cluster and explain the variability between steady-state operating regimes; hence, which variables are important for a particular regime, i.e., which variables are affected by the different clusters. Lastly, the experts confirm the operating regimes and explain the fundamental drivers behind them, i.e., the variable of interest helps identifying the clusters.

These “bottom-up” operating regimes based on a data-driven approach as well as process knowledge offer the opportunity to interpret and understand the process in more depth. Those regimes could be linked with the way the plant is run by operators since they might operate the plant differently when producing the same product.

The identified operating regimes should be mutually exclusive. Hence, some variables specific values must uniquely be attributable to one operating regime. In other words, there is a unique set of parameters values to distinguish each regime. All operating regimes are unique, and they cover all the possibilities for characterization variables (attributes).

Operating regime detection and identification allows improving process knowledge as analyzing each cluster may lead to discovering unknown operation mode. On the other hand, process knowledge is required to interpret the patterns extracted from the principal component analysis and to achieve an accurate representation of the operating regimes in complex processes. The PCA results interpretation must be performed by process experts, and they should be cautious when doing so as the PCA blindly finds correlations; however, the physical reality of the process plays no part in generating the statistical outputs [76]. Therefore, PCA results should be interpreted based on an understanding of the process fundamentals. This is a task for process experts; the “bottom-up” operating regimes are detected and identified by merging data-driven approach and process knowledge.

Performing all the previously mentioned data management steps on the raw process data—including abnormal operations removal and noise reduction—maximizes the usefulness and truthfulness of multivariate analysis as a statistical analysis is only as good as the raw data and pre-treated data represent the process more accurately [76]. Multivariate analysis (such as PCA) are entirely data-driven techniques, and thus highly susceptible to the issue of “garbage in, garbage out”. They are sensitive to outliers and instrument drift; the latter can appear as a long-term trend to which multivariate algorithm could blindly ascribe statistical significance [76].

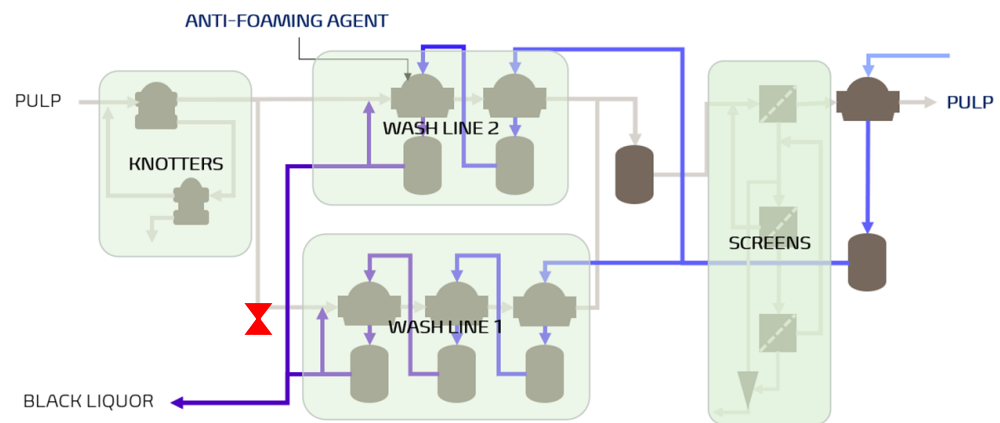
Noise in process data represents a different problem. As each data point (one-hour period for instance) is treated as a separate observation which bears little resemblance to each other before bringing them all to the frequency analysis, some kind of smoothing is therefore required; this filtering improves the multivariate analysis ability to find correlations between variables [76]. Therefore, a direct use of the raw data would yield meaningless multivariate analysis results, since the algorithm could, for instance, blindly attribute most of the correlation to the start/stop phenomenon and not to actual changes in the process [76].

To conclude the framework development, operating regimes resulting from the bottom-up analysis are identified through a clustering analysis that is based on the combination of principal component analysis, *k*-means algorithm, and process knowledge. Some of these regimes are selected according to the scope to solve a management problem (related to process design or operations).

#### *4.6. Application: Brownstock Washing Department in a Dissolving Pulp Mill*

An application is demonstrated on data from the brownstock washing of a dissolving pulp mill (Figure 8) in order to illustrate the benefits of the proposed framework. This unit is isolated by buffer tanks. Those tanks absorb process fluctuations, so events in one system will not impact downstream systems. Therefore, the unit can be marked as independent from the rest of the process. This system was chosen as it is the one showing the highest redundancy (allows data reconciliation analysis); the pulp and paper mills are famously known for their lack of redundancy. The results from a simulation of this department are used to increase redundancy.

The implications of each step of the framework are shown here. This section is divided in accordance with the main parts of the framework; scope definition, signal processing, steady-state detection, unit-wide data reconciliation and operating regime detection and identification. Each step plays a critical role in the whole framework, and together they allow the use of rectified and reconciled (clean) segmented steady-state data for design decision making.



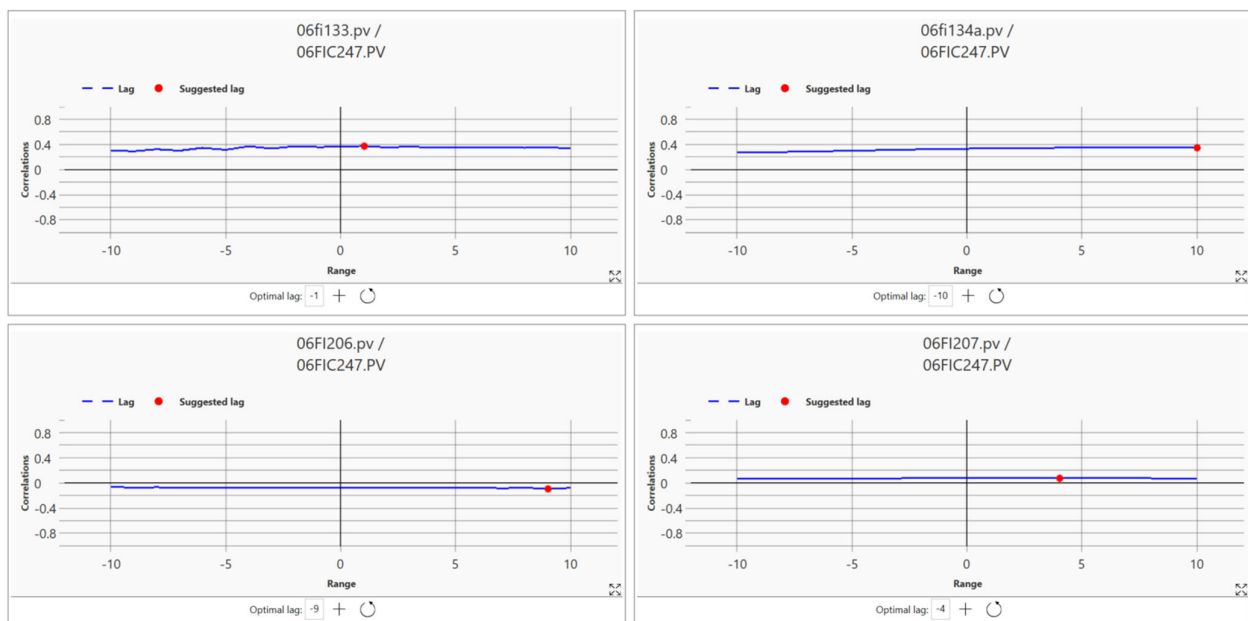
**Figure 8.** Brownstock washing department in a dissolving pulp mill.

### Scope definition

The intended use of the data is to assess how many bottom-up regimes can be detected and identified in the brownstock washing (BSW) through the use of the framework when the process is running smoothly (in steady-state). The top-down regime considered is the summertime when a specific grade is being produced. Brownstock washing data that fit this context is collected.

### Signal processing

In this case, data synchronization and imputation are not required as all data points are sampled at the same rate, i.e., a sampling interval or time step of 10 min (process experts confirmed that a 10-min sampling interval is adequate for this SSD). A sampling interval that reduces autocorrelation in the time-series process data is used. Additionally, as the pulp crosses the whole unit in a few minutes (the overall residence time of the system is around 10 min), there is no need for intra-unit lag correction. This is confirmed by using the lag correction analysis offered in EXPLORE (version 2.2.0.814); a straight line means that there is no lag detected (Figure 9). If a lag was present, the graphs would have shown some waves and a cycle.

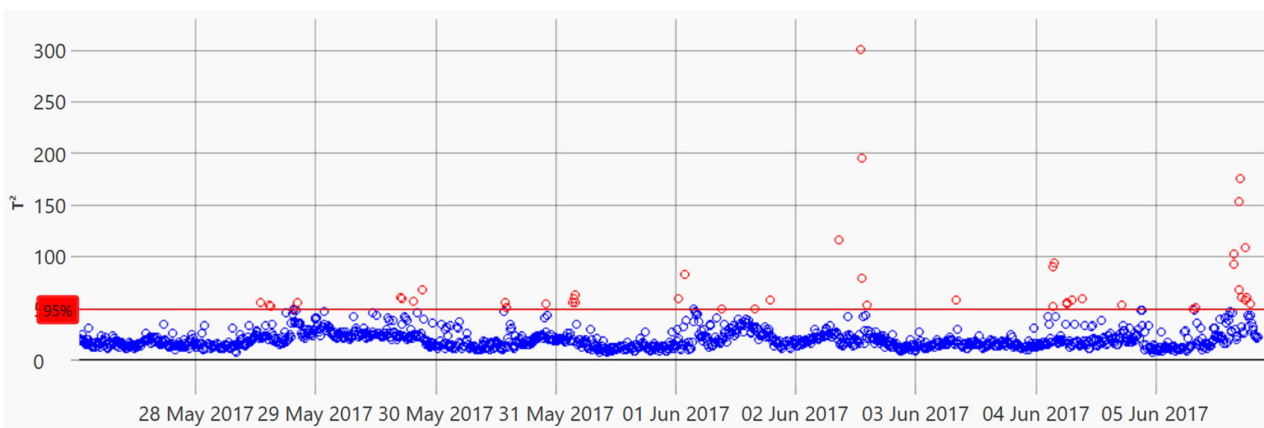


**Figure 9.** Lag analysis where X is the number of lags and Y is the correlation.

From there, start up and shutdown periods are removed. To do so, three independent flow variables that describe the process are used as reference to detect those periods. The

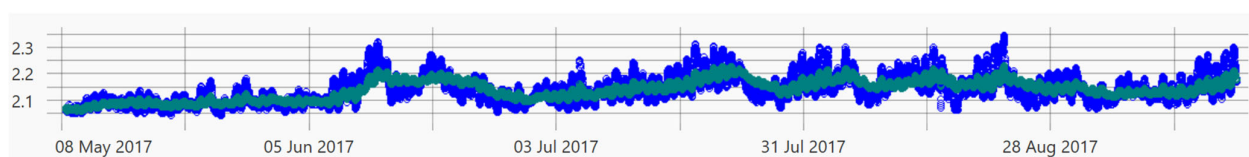
latter are removed by using threshold values specified by a process expert. A two- and four-hour time interval are removed, respectively, before and after each upset periods to account for the time required to go into shutdown and restart the system. These time intervals are based on the time required for the thickener consistency measurement to restabilize itself after an upset. Lastly, the removal of process shutdown and start up from reference variables is mirrored onto the other variables used as part of the analysis.

Then, outliers are removed using quartile analysis as well as the principal component analysis (Figure 10). Similarly as what is done for the shutdown and start up periods, key process flow variables are used to detect outliers. This removal is reflected in all the other variables. Having almost 20,000 data points, outliers are not corrected, but rather attributed a NNON value.



**Figure 10.** Outlier removal using principal component analysis (the outliers are in red).

Last of all, the signals are filtered using wavelet transform (Figure 11). Given that the quality of wavelet transform denoising relies on the optimal configuration of its control parameters, i.e., signal cut-off level (scale), wavelet function and threshold parameters, the selection is done through the use of process knowledge in order to retain only the trend associated with the operation of the process.



**Figure 11.** Consistency signal filtered using wavelet transform with db15, scale 8 (in blue is the raw signal and in green is the filtered signal).

### Steady-state detection

This section describes an industrial implementation of the proposed SSD algorithm. The latter requires a data-vector of time-series data discretized into a time-ordered sequence of uniformly distributed timestep of equal durations with the oldest data point referenced as timestep one (1). The SSD algorithm also necessitates the tuning of the time-window, threshold, standard deviation of the noise (if desired) and cut-off probability; these tuning parameters are part of the configuration of the algorithm. This implies that some prior knowledge and understanding of the process is necessary.

It is not uncommon for experts at the mill (or in plants in general) to be overoptimistic on how well the process unit performs and mentioned how the process runs for many consecutive days in steady-state, possibly overestimating the stability of the plant. There are mainly two factors that may explain this. It might come from the fact that their steady-state analysis is most of the time qualitative. This is what they aim, expect, and wish for,

although without performing a quantitative data-driven analysis, there is no way to confirm that the process is indeed at steady-state. On the other hand, detection of steady-state periods might also be a visually subjective process expert decision. However, this visual steady-state recognition approach requires continual human attention, and it is subject to human error. More specifically, noisy measurements, slow process changes, multiple dynamic trends, and change-of-shift timing are features that may compromise the visual interpretation of data. The statistically-based SSD standardizes the procedure.

In the context of this application, the monitoring horizon is five months in the summer-time whereas the time-window is four hours. The latter should have a number of samples equivalent to three (3) to five (5) times the time constant of the overall process divided by the sampling interval; it should be long enough for the variable to reach steady-state or equilibrium. Here, experts stated that it corresponds to the average time required to reach steady-state in the process. Since the time to reach steady-state varies with operating conditions [73], an average provided by a process expert is used. Too short of a time-window will not give sufficient time to the process to reach some level of stability, thus the steady-state probability will always be low. On the other hand, a too long time-window may lead to the false conclusion that the signal is steady when in fact it is not. Additionally, long windows are not well suited for detecting unsteady behavior with short duration; it is harder to detect. Therefore, the time-window must be long enough to account for the system dynamics, but short enough to detect undesirable changes in the process value that has short duration.

Additionally, since different variables present different dynamics, a slow process should consider a longer time-window, while if the dynamics are fast, the time-window duration should be shorter. Therefore, the time-window is the parameter that can account for both the sampling interval as well as the process dynamics. Furthermore, it is possible for every KPV to have its own time-window size [79].

The threshold is set at 95% confidence interval (Student-t score). This parameter is related to the importance of a period to be labeled properly; for a critical application, a higher  $\alpha$  should be used. Therefore, if the steady-state detection is not extremely critical for a particular application, a 5% probability of incorrectly rejecting steady-state is accepted.

Then, even if the standard deviation of the noise could be assessed by Equation (A5), the latter is provided for all key process variables (KPV) targeted to detect steady-state periods. As the standard deviation of the noise value increases, the maximum steady-state span increases as well—more data are deemed at steady-state, the spans get longer.

The KPV selected by a process expert represents well the process and the operations undertaken by the mill, they have a significant impact on the latter and they should have the least autocorrelation as possible. Experts decide what is the minimum number of KPV required to detect whether the process is at steady-state, and then set up the procedure on that minimal set. For this application, a process expert at the mill mentioned that the best indicator to know if the process is stable is when three distinct pulp flows meet the steady-state conditions at the same time.

The SSD algorithm returns a probability that the data-vector is steady with a probability near one (1) and is unsteady with a probability near zero (0). The determination of the probability limit which indicates a steady or unsteady signal is the responsibility of the user. A suitable cut-off value of whether the process is deemed to be at steady-state depends on the application. In this case, the cut-off probability is set to 95% (probability of the time-window being steady) for each KPV. Once the tuning parameters are established, the SSD function is called for every time-window until the end of the monitoring horizon.

This approach, which uses a combination of good judgment, knowledge and interpretation of process operations, and statistics, has two different modes. The first one runs every time step (sampling interval) continuously (every 10 min in this case)—it incrementally moves over the smallest time step. Therefore, every 10 min, the algorithm is looking back over the time-window duration, and it declares if the process is steady or unsteady for the past 4 h. The second mode is the batch one where the algorithm runs every 4 h,

still looking back, declares the whole time-window steady or unsteady, and then moves forward to another time-window to conclude on its steadiness. However, in both cases, the time-window duration is the minimal threshold of time for which the process can be deemed steady or not, and for which the steady-state duration is judged representative; the algorithm is not going to find anything less than the time-window. In other words, every time the process is declared to be in SS, it lasted for the past time-window. As part of this proposed framework, the first mode is preferred.

In order to label a period as steady, all the key process variables must be steady at the same time, i.e., all KPV have a probability higher than 95%. It would also be possible to label a period as being in steady-state if specified fractions of the KPV are in steady-state at the same time. In this case, since there are only three of those, all three must meet the probability threshold.

As a result, the algorithm evaluated that out of the five (5) months monitoring horizon, the process is in steady-state 32% of the time. However, a major challenge in this case is the constant starting and stopping of production lines and pieces of equipment. Consequently, knowing that there are sporadic and persistent steady-state periods, a run length of a minimum of 12 h is targeted to deem the period as persistently steady; when the process runs reasonably steady for a while, these operating data might be used as a basis for decision making. The run length is the amount of time or samples required to be confident in declaring that the process is persistently steady. In the algorithm, a second routine identifies the contiguous sets of steady data. This routine is a data function that determines the statistics of the contiguous span of when the process is steady; a start, stop and span range are assessed as well as the maximum and minimum number of data points in a span, the span mean, mode, median and variance. The following table (Table 2) presents information about the contiguous steady-state regimes collection.

**Table 2.** Information on the contiguous steady-state regimes collection.

|          | Value |
|----------|-------|
| Minimum  | 24    |
| Maximum  | 182   |
| Mean     | 31    |
| Median   | 26    |
| Mode     | 24    |
| Variance | 214   |

However, considering only when the process unit is contiguously steady for longer than the run length, a total of 20 spans are found. Once all steady-state periods are detected, they all individually and separately go through the process of data reconciliation to determine if there are measurement gross errors—each one of the contiguous steady-state set respecting the run length becomes a dataset for data reconciliation.

#### **Unit-wide data reconciliation for various datasets**

As part of this framework, the data reconciliation problem is modeled from a mathematical programming perspective in opposition to a matrix algebra perspective. From this standpoint, every stream variable has a reconciled value and an adjustment (revision) value. In order to perform data reconciliation, a process model is built and validated. This model consists of a set of equations (See Supplementary Material for the model). There are sensor constraints equations, i.e., adjustment + reconciled = measurement, and model constraints, which are all the laws of conservation of material, energy, and momentum, i.e., mixer, splitter, process, and density equations. The process recycling loops are included in the DR model. These generates unbound variables which lead to an infinite number of solutions. Hence, the simulation results are used to evaluate the split fraction and create a hard constraint.

The equations are a mixture of linear and non-linear constraints. This makes the DR problem more difficult to solve, but it is important to represent the process in its most fundamental way to reduce model gross errors. Having non-linear equations makes the problem non-convex; the problem is subject to local solutions, there are local optimum. Thus, the solver is run a few times and each time, it randomizes the initial values (starting value generation). This also helps with convergence issues. IMPL-DATA can solve non-linear data reconciliation problems.

Generally, adding constraints to any kind of mathematical minimization optimization problem will either increase the value of the objective function or it will stay the same if the information is redundant. The only way adding a constraint could reduce the objective function value is if the solution is a local optimal (it converges to a different local optimum).

The model is validated when it is consistent with the simulation data (given some tolerance). Hence, the process simulation is used for the validation of the data reconciliation model. When the DR model is validated, it is possible to say in confidence that from now on, only measurement GE will be identified.

In industry, not every stream has a flow and/or consistency measurement. This application is no exception; measurements are very sparse. There are many unmeasured and unobservable variables and not enough measurements to identify gross errors. Thus, there is not enough redundancy in the mill; there is redundancy overall, but not around some equipment. Therefore, redundancy is created using the simulation results (this is not a common practice in DR). Hence, a sensor is assigned to all streams; the simulation results are considered as measurements—the simulation approximates the real process. By doing so, observability is increased. There is no level of acceptable observability, but IMPL-DATA have a pre-solve observability functionality that can be applied to better determine the observability, to improve the numerical robustness of the observability detection [80]. The pre-solve algorithm goes through the sparse unmeasured variables incidence matrix and excludes variables that are strongly or guaranteed to be observable in the singleton constraints [80]. Declaring these as observable shrinks the matrix. A large matrix has a large condition number, therefore making the matrix smaller inertly means a smaller condition number, which makes it more numerically reliable. Therefore, the pre-solve provides a smaller matrix that has less constraints and variables to go ahead with the observability detection analysis.

Additionally, to build a DR model, weights are assigned to measurements. These weights reflect the reliability, precision, and accuracy of measurements. For instance, temperature measurements tend to be more accurate than flow measurements, and more specifically, steam flows are more problematic as they are harder to calibrate and need to be corrected for temperature and pressure to get the flows right. On the other hand, liquid flows are generally fairly good. Then, temperature measurements from thermocouple are generally pretty decent, they do not drift much. Next, level measurements from tanks are, most of the time, not even reconciled as plants probably have those values right since the operators would not let tanks overflow or underflow. Lastly, pressure signals are reasonably good and consistency data have a certain range where the sensor seems to work reasonably well.

In this application, only flow and consistency measurements are being reconciled. Measurements are given a tolerance or precision interval which is translated into a raw variance where its inverse is used as an objective function weight in Equation (A8). As consistency measurements present less variation than flow measurements, and are well controlled, process experts decided to assign a tolerance value of 0.5% to consistency measurements and a tolerance value of 2% to flow measurements. These tolerances are for hard sensors (direct measurements); a tighter tolerance is put when the stream has a physical sensor. However, for values assumed from the simulation (soft sensors), a larger tolerance is considered as there is more uncertainty. Therefore, a suitable tolerance for indirect (soft sensor) consistency measurements is 4% whereas 5% is assumed for flows. Lastly, fixed variables are given a tolerance of 0%.

Following the tolerance analysis, for every variable on all streams, there is something indicating whether it is measured (1) or unmeasured (0). This is the sensor switch. When there are a lot of zeros in the sensor switch, the results start showing unobservability, and some flows present negative values (this is a sign of GE). Therefore, to maximize observability, there is a tradeoff that must be found in labeling variables as unmeasured and getting negative flows. The sensor switch helps in turning off the wrong sensors, either hard or soft; to keep the sensor, a value of one (1) is used. Therefore, even if a sensor is assigned to all streams, a value is assumed only for those having their sensor switch value to one (1).

In addition to the sensor switches, which can indicate whether or not a sensor is good, whether it may be trusted or not, stream switches manage permanent and temporary streams. The latter must be considered to make sure that the reconciliation results make sense. Since processes have different operating regimes, the DR model changes and must therefore allow streams to be turned on and off, i.e., it is parameterized in the sense that streams may be active or inactive. Logic is required to manage whether the streams are active or not. For instance, when a stream has a flow sensor, the latter could be used as a stream switch by setting the sensor value to zero when required.

When plants perform data reconciliation, it is mostly based on mass balance. However, most of the flows in plants are volumetric. Therefore, to convert those to mass, density measurements are required. The latter can be assumed, or calculated given an ad hoc formula, density measurements can come from a simulator, or they could be taken every week or every day at the lab. However, densities are not as good of a measurement as flow or consistency because they are not continuous, they are sampled intermittently. Ultimately, the reliability of the density data is poor primarily because there is little, if any, statistically-driven measurement quality feedback being transferred or relayed back to the engineering, operations, instrumentation, and maintenance departments. Additionally, they could be wrong because they are based on assumptions about the process and operating modes that do not necessarily apply. Given the inherent sparsity in the density measuring system, these densities may be biased (wrong, not accurate), especially if these densities are operating-condition or -mode dependent. Hence, many problems could come from inferring the densities. Performing DR can help remove that bias provided that bad densities are detected, identified, and removed from the data reconciliation problem.

Once all variables are accounted for and the model equations are set, a degree-of-freedom analysis is assessed. A negative value of the degree-of-freedom (number of variables—number of equations or constraints—number of measurements or fixed variables) is expected as the simulation values were used as measurements substitution. This non-linear data reconciliation problem reconciles both volume flow and density simultaneously, involving volume, density, consistency, and mass balances. In the present application, there are 53 sub-units, 107 streams, and 19 flow sensors as well as 4 consistency sensors. The sub-units in the brownstock washing are mixers (18), splitters (9) and processes (26). For mixers, both volume (28) and mass (18) balances or equations are required, splitters count 46 equations, and lastly, process equations have a total of 38 mass balances and 30 volumetric equations. In addition to these, the BSW process unit accounts for 107 density equations. Then, as each stream has a flow, consistency and density value, there are 321 variables. Therefore, the number of variables, equations, and measurements (including those assumed from the simulation) yield a DOF of  $321 \text{ variables} - 267 \text{ equations} - 90 \text{ measurements} = -36$ .

In this analysis, the qualities/intensive (compositions—fractions, consistency, properties—density, molecular weight, and conditions—temperature, pressure, velocity) and quantities/extensive (flow, mass inventory, moles, volume, energy, and momentum) are kept distinct to have different constraints on the individual variables: volumetric flow, consistency, and volumetric flow\*density\*consistency (to obtain the mass flow). Both variable categories are reconciled simultaneously.



This DR problem only considers material balance (flow in – flow out = 0). However, if there is a hold-up unit in the model (tank, drum, vessel), the hold-up balance (flow in – flow out + opening – closing = 0) could be performed considering measurements of the level (hold-up, inventory). To do so, the opening hold-up is considered as a fixed variable, meaning that it has 0 uncertainty, or its weight is infinity, and then only the closing value is reconciled. Another way to do this would be to reconcile the difference between the opening and the closing.

In summary, when performing data reconciliation, the first step is to find unobservable unmeasured variables. DR would still run with these, but it would calculate numbers that are completely meaningless since they are non-unique. Hence, from a mathematical perspective, there is no issue; however, from an engineering perspective, unobservable unmeasured variables may be problematic. If required, all unmeasured variables can be made observable either by adding assumed values or shrinking the data reconciliation model. Then, after running the algorithm, one must make sure that there is no negative flows greater than constraint and convergence tolerances, because they mean that there are measurement gross errors. In fact, negative flows could mean that their model directions should be reversed. Once all the negative flows have been resolved, then gross error detection may begin.

For this application, data reconciliation is run offline on averaged steady-state data. However, data reconciliation is generally run online on hourly average steady-state data in order to detect the most persistent or sustained sensor with GE. The execution frequency of DR should be based on how many gross error could happen within that time frame. The goal is to run it when there is zero or no more than one. As a matter of fact, as there are sporadic and persistent GE, a monitoring report is available when reaching the end of the reporting horizon (moment when conclusions about DR is made), such as a shift, a week, or a month, to inform process experts about the most problematic sensors. Hence, only persistent GE are recorded or added to the ongoing list; they may as well be ranked. The ratio of the number of  $H_0$  acceptances over the duration of the reporting horizon gives a probability of occurrence and indicates what percentage of the time a sensor is persistently faulty. Therefore, deploying online unit-wide data reconciliation may continuously improve the reliability of process data, assure that the sensor networks is functioning with consistency and integrity, and provide the level of assurance required for descriptive, predictive, and prescriptive analytics.

Nevertheless, in this case, DR is run on the averaged steady-state spans of arbitrary duration, but always longer than the run length. Table 3 presents the objective function value (Equation (A8)) for all spans. According to the total number of DOF and the 95% confidence interval, the Chi-squared statistic is 85.965. The objective function values are all greater than the chi-squared statistic, hence, out of the 20 steady-state span found from SSD, all of them contain gross errors. Column 3 of Table 3 gives the worst (biggest) maximum power measurement test values (Equation (A9)) across all variables in each span. Lastly, the next column provides the values of Equation (A10). Since all values are higher than the Chi-squared critical value (with one less DOF), 84.821, there are more than one gross error in each span. Therefore, it is difficult to isolate the most-likely bad sensors or to reliably identifier of the sensors with the gross error. Considering all the measurements in each span with a MPMT value close to the worst one, it was concluded that the top four persistently problematic sensors across the steady-state spans that would need to be verified are three flow meters (06FIC137, 06FIC152, and 06FIC433), and one consistency sensor (06NIC423). This information is transmitted to process experts for them to take a look at the problematic sensors. Knowing which sensors are faulty is important because experts base their analysis (such as optimization) on these data, and until these sensors are fixed, gross errors are present in the datasets.

**Table 3.** Objective function value for all spans.

| Spans | Objective Function | Worst MPMT Value | Obj-MPMT |
|-------|--------------------|------------------|----------|
| 1     | 2069.364           | 30.383           | 1146.250 |
| 2     | 1960.541           | 29.919           | 1065.403 |
| 3     | 1899.865           | 28.611           | 1081.287 |
| 4     | 1944.359           | 28.350           | 1140.662 |
| 5     | 1870.016           | 27.920           | 1090.486 |
| 6     | 1910.728           | 28.635           | 1090.786 |
| 7     | 1914.603           | 29.442           | 1047.766 |
| 8     | 2144.591           | 30.767           | 1198.001 |
| 9     | 2077.743           | 31.400           | 1091.786 |
| 10    | 2128.247           | 30.996           | 1167.487 |
| 11    | 2106.752           | 30.351           | 1185.586 |
| 12    | 2041.512           | 29.570           | 1167.146 |
| 13    | 1922.061           | 29.515           | 1050.945 |
| 14    | 2033.123           | 29.566           | 1158.979 |
| 15    | 1861.946           | 29.297           | 1003.613 |
| 16    | 1928.288           | 29.459           | 1060.441 |
| 17    | 1685.618           | 27.045           | 954.180  |
| 18    | 1652.962           | 26.277           | 962.464  |
| 19    | 1706.588           | 25.994           | 1030.917 |
| 20    | 1832.242           | 26.900           | 1108.613 |

### **Operating regimes detection and identification**

In general, in the pulp and paper industry, there is little to no acknowledgement of the fact that a process has many different steady-state operating periods. In other words, operating regimes are not explicitly considered for decision making [8]. However, the value and potential of operating regime detection and identification is well recognized. In the present study, a model based on principal component analysis as well as *k*-means algorithm is used to identify the operating regimes of the brownstock washing department of a dissolving pulp mill. In this application, five months of data are used to detect and identify the process operating regimes.

The clusters apparent on the score chart of the PCA (Figure 12) were confirmed with a *k*-means clustering analysis detailed in Section 4.5. The first and second components separated the observations that are different and gathered the identical observations. The variables that influence these clusters can be observed through a contribution analysis (Figure 13). Analyzing the results with process experts, it is found that the main drivers for the “bottom-up” operating regimes are the pulp level in tanks, its density, and the shower wash water flow rate. The clusters represent changes in the operating conditions.

Clustered data can be difficult to interpret, and as they are interpreted by process experts, errors can happen. Interpretation errors are part of a continuous improvement process. Experts gain insight through the data processing framework.

Lastly, the loading chart of the first and second components is shown in Figure 14. This chart identifies which variables characterized each cluster and explains the variability between the different regimes. Variables close to the center of the chart do not have a lot of importance for component 1 and 2 (in this case) whereas variables away from the center and close to either component will be of great importance—further from the center, more influence they have. Lastly, those located diagonally are influenced by both components.

Figure 14 shows for instance that the pulp consistency explains a lot of variability in the first component while the pulp density explains most the variability of the second component.

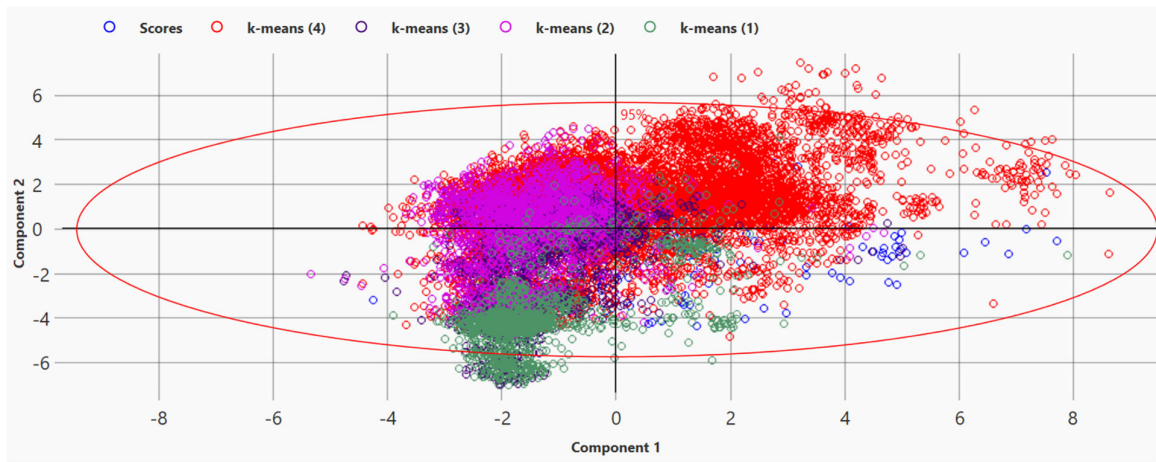


Figure 12. Score chart showing the clusters detected by *k*-means clustering algorithm.

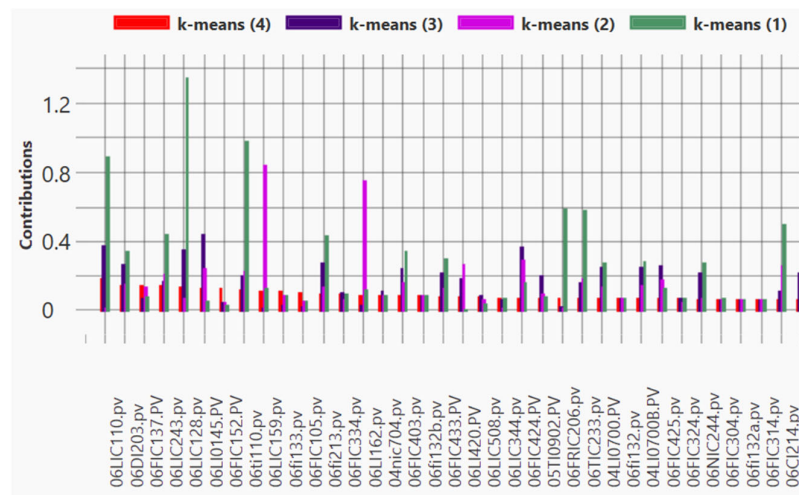


Figure 13. Contribution analysis of four detected clusters through PCA and *k*-means.

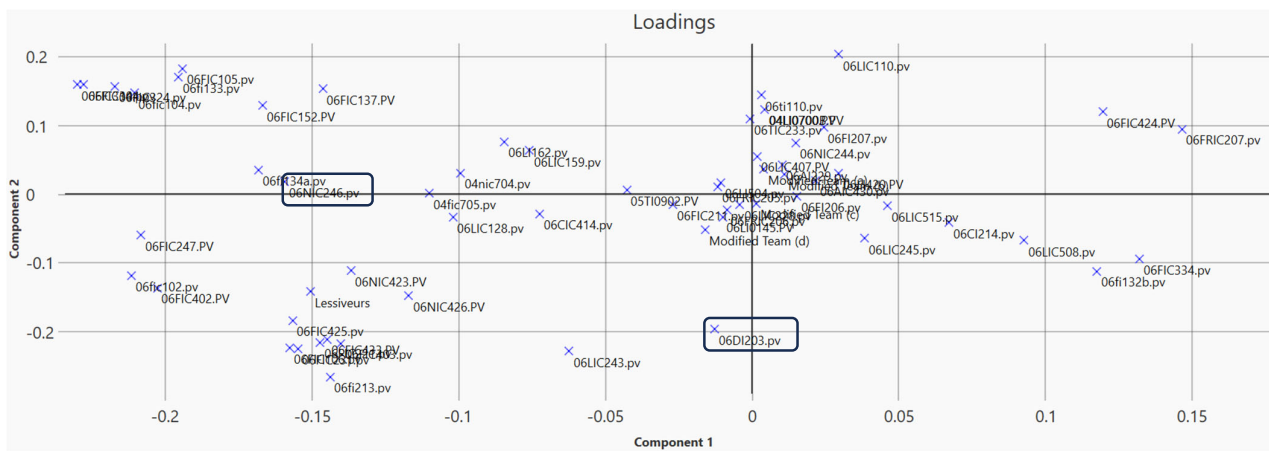


Figure 14. Loading chart of the first and second components to identify variables of significant influence.

## 5. Conclusions

With the quantity of process data being collected increasing year over year, actions should be undertaken to put them to good use, and to do that, one must make sure that the data is of good quality. Several authors proposed methodologies to manage, treat and analyze process sensor data and to use them for decision making [1,9–12]. However, these data processing frameworks missed few critical steps and did not explicitly consider operating regimes nor mention the concept of unit- or plant-wide analysis, even though industrial applications of operating regime detection are extensive. This paper proposes an operating-regime-based data-driven industrial framework for improved process decision making. In this first part, the latter is established. Its value will be further explored in future work, where the processed data from the framework will be used for improved decision making. Examples of decisions that may be taken based on the offline analysis of processed historical data include identifying which products or recipes are not profitable (cost analysis), which generate the most emissions (environmental analysis), which require the most resources (energy analysis), when should maintenance be performed (predictive maintenance), and how to best schedule production (optimization analysis).

The proposed framework employed to make strategic decisions based on historical data exploits advantages from both process knowledge and data-driven approaches. The former is a critical complementary information source. On the other hand, data-driven approaches (including PCA) help to understand the process further and can extract and unlock hidden knowledge that is not accessible from process knowledge. It allows enrichment and updating of existing process knowledge to take accurate management decisions for complex processes.

This combination of process expertise and data-driven approaches starts with the scope definition which is the objective of the analysis. Then, given that the means and variances of the data signals change with operating conditions, top-down operating regimes must be considered early on in the framework.

Once the data are collected, data pre-processing is performed and steady-state periods, intervals or windows are detected. The steady-state detection algorithm provides a reliable and effective way to identify when each key sensor reading is stationary over a suitably chosen time-window. Given that most management decisions, as well as design and diagnostics, are based on the steady-state representation, it is easier to understand and is more practical. Then, unit-wide steady-state data reconciliation is performed. The averaged variable values over the duration of each steady-state span are used for all individual data reconciliation runs. This step highlights persistently faulty sensors in the process and allows ranking them according to their probability of occurrence. Additionally, in opposition to common belief, if there are gross errors in the measurements, then the reconciled values are in fact less reliable than the original raw data as these outliers can unpredictably distort and bias all related or co-incident process variables involved in the model.

The last step of the proposed framework is the “bottom-up” detection and identification of operating regime. In industrial processes, there are often various component material recipes used, different products or grades made, the operating conditions may change, the feedstock may vary (slightly or not), the weather may change, different operators may operate the process differently, etc. All of these aspects may lead to different operation regimes in a plant. Some of the aforementioned changes are known; however, some may be hidden in process data. Therefore, operating regime detection can yield knowledge otherwise not available, provide insights for management decision making and thus potentially create opportunities for better process operations. Putting forward the synergy between process experts and mathematical techniques help detect operating regimes and thus better understand how processes, production, and plant work, and can lead to substantially improved process analytics. Incorporating operating regime detection as described in this framework is an enabler to reach the full potential of process data for decision making.

An application of the developed framework in a brownstock washing unit was demonstrated. We showed that this data treatment tool can be implemented in the process industries. In light of the case study results presented in Section 4.6, we believe that the framework could produce interesting results if deployed in mills, plants, and other types of processing facilities. Furthermore, such applications could help improve the framework and strengthen the link between both process knowledge and data-driven dimensions of the analysis.

The insights generated by the framework will be demonstrated for the optimization of the brownstock washing department operations in future work; the identified steady-state operating regimes will be used for enhanced process decision making. This will enable the analysis of cost-efficiency for various operating regimes. More specifically, future work planned by the authors will include an activity-based costing analysis subsequently of the data processing framework application. Production costs are inevitably related to the operating regime and plants need to know where they stand to assess these costs. This future work will also analyze further what explain the changes in operation between the different steady-state operating regimes, and how does they affect the operating cost.

Industrial data must be processed (cleaned) and their context should be acknowledged (operating regimes) by coupling mathematical methods with process expertise for proper and improved decisions, actions to be taken on the process and/or modification made on the process. These strategic decisions might represent important changes for plants and should therefore be devoted a proportional amount of time. Nevertheless, the data processing techniques used in each step of the framework could employ artificial intelligence and machine learning algorithms in the future. It is however critical to leverage process knowledge and make sure that this aspect is not lost. Therefore, an area of improvement might be to look into automate the data processing steps to make them less time-consuming, without losing the required user input. Even if the data treatment steps are automatic, the data interpretation part must be manual. Understanding complex processes and interpreting data automatically may be one day doable with artificial intelligence and machine learning algorithms. Lastly, future work could address the application of a data processing framework for real-time online decision making. i.e., how to perform each of the framework's steps online in real time.

**Supplementary Materials:** The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/pr11082376/s1>, Model S1: Model of the process, i.e., all the mass and energy balance equations.

**Author Contributions:** Conceptualization, É.T., J.D.K., F.L.D., M.C., B.P. and P.S.; Methodology, É.T., M.C., B.P. and P.S.; Software, É.T., J.D.K. and F.L.D.; Validation, J.D.K.; Formal analysis, É.T. and F.L.D.; Data curation, É.T.; Writing—original draft, É.T.; Writing—review & editing, É.T., J.D.K., F.L.D., M.C., B.P. and P.S.; Visualization, J.D.K.; Supervision, P.S.; Project administration, P.S.; Funding acquisition, P.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** The authors gratefully acknowledge the financial support of The Natural Sciences and Engineering Research Council.

**Data Availability Statement:** The process data from the mill is confidential unfortunately.

**Acknowledgments:** The authors gratefully thank CanmetENERGY for the technical support using the EXPLORE software. The authors gratefully acknowledge the work of Caroline Brucel who provided the simulation of the brownstock washing department of a dissolving pulp mill, as well as her help in building the process model.

**Conflicts of Interest:** Jeffrey Dean Kelly is the President of the company that developed IMPL-DATA. Bruno Poulin is employed by CanmetENERGY, he developed the EXPLORE software.

## Appendix A

It is assumed that any process signal may be represented as

$$x_t = mt + \mu + a_t \quad (\text{A1})$$

where  $m$  is a non-zero slope,  $t$  the relative time within a window (the cycle count),  $mt$  is the deterministic drift component,  $\mu$  is the sample mean or arithmetic average over the time-window with zero slope and  $a_t$  is the independent and identically distributed random error series or white-noise sequence with zero mean and standard deviation  $\sigma_a$ . The  $t$  index indicates the cycle at which the sample is collected.

By taking the discrete difference of  $x_t$  and its immediate past time-shifted value  $x_{t-1}$ , the following equation is obtained

$$x_t - x_{t-1} = m + a_t - a_{t-1} \quad (\text{A2})$$

where  $a_t - a_{t-1}$  has an expected value of zero with a standard deviation of  $2\sigma_a$  by definition. Therefore, the drift slope  $m$  can be estimated as

$$m = \frac{1}{n-1} \sum_{t=2}^n x_t - x_{t-1} \quad (\text{A3})$$

with  $n$  sampled values in the window which are equally spaced in time.

The intercept or mean  $\mu$  is obtained through Equation (A4)

$$\mu = \frac{1}{n} \left( \sum_{t=1}^n x_t - m \sum_{t=1}^n t \right) \quad (\text{A4})$$

Next, the standard deviation of the noise is estimated as

$$\sigma_a = \sqrt{\frac{1}{n-2} \sum_{t=1}^n (x_t - mt - \mu)^2} \quad (\text{A5})$$

The latter could also be provided to the algorithm if it is known by the user, and thus Equation (A5) is ignored. The standard deviation obtained from this equation is rather liberal (looser); it will most likely be bigger than an externally supplied standard deviation that could be more conservative (tighter/smaller).

Both the mean and standard-deviation are corrected for the drift component.

Lastly, considering a specified Student-t critical or threshold value at a particular significance level  $\alpha$  and degrees-of-freedom  $n$  (the DOF is the number of sample in the time-window), the null hypothesis that the process signal is steady may be tested. Therefore, if

$$|x_t - \mu| \leq t_{critical} \sigma_a \text{ then } y_t = 1 \text{ else } y_t = 0 \quad (\text{A6})$$

The null hypothesis is rejected (the window is flagged as unsteady) when  $y_t = 0$  whereas a value of one means that the null hypothesis is accepted, all of the points in the time-window are deemed to be at steady-state. Therefore, if the measurement minus the estimated mean divided by the standard deviation is greater than the  $t_{critical}$ , then one can be  $(1 - \alpha)\%$  confident that it is unsteady. The null hypothesis presumes that there is no autocorrelation in the time-series process data. If there is autocorrelation, the variance will be inflated. Therefore, to ensure that the approach will be insensitive and immune to any autocorrelation in data fluctuations, providing the standard deviation is recommended.

After performing the Student-t test individually on all data points of a variable (tag), the algorithm returns the frequency of  $y_t = 1$ . The sum of  $y_t$  divided by  $n$  (the non-missing data amount number in the time-window) represents the probability of the null hypothesis to be true; it is the percentage of time within the window that the process variable is deemed

to be at steady-state. For instance, a value of 95% would indicate that 5% of the points are not at steady-state.

The algorithm calls this routine (Equations (A1)–(A6)) for each time-window with an increment of one time-interval. In other words, all equations are computed over the time-window continuously (for all samples in the monitoring horizon) given a sliding window. At every time points, the algorithm is looking back at the time-window minus one data point as it includes itself. In the end, every data point has a steady-state flag.

In this algorithm, the samples with time index lower than the time-window (1 to window-1) always have a returned probability of 0 to be in steady-state as there is not enough past data, they are always declared to be unsteady; the algorithm is using a sliding-window that is looking back at the window size. Therefore, after the sample (window), there is enough points so that the algorithm is able to evaluate the probability.

This methodology considers domain-knowledge; it is a combination of process expertise and statistics. The algorithm is subjective as there are a lot of parameters that may impact whether or not a data point accept or reject the null hypothesis, i.e., the process is at steady-state: the number of samples in the time-window, the alpha value (for the threshold, the Student-t critical value) and the standard deviation. In the algorithm [74], the cut-off probability (the sum of  $y_t$  over  $n$ ) is also tunable, it could be a setting of the algorithm; however, in this paper, the control limit for the probability is assumed to be  $1 - \alpha$ . Therefore, for a given time-window, a probability of being steady is assigned by counting the number of time-points that exceed 90%, 95% or 99% confidence-interval defined by the Student-t statistic. If the probability is greater than some upper limit such as 90%, 95% or 99%, then the time-samples are declared steady, else unsteady.

The significance level  $\alpha$  represents the probability of a type I error, or the probability of rejecting the null hypothesis when it is true (false positive/false alarm). Therefore, a 95% confidence interval means that when the null hypothesis is actually true, it will be falsely identified as being unsteady (rejected) 5% of the time. Therefore, alpha is related to the importance of a process value being truly at steady-state. There is also the type II error (significance level  $\beta$ ), or the probability of accepting the null hypothesis when it is false (false negative/missing alarm). Theoretically, if steady-state is critical for a specific application, the SSD must ensure a low probability of a type II error; however, this algorithm does not consider it.

As part of this algorithm, the incoming process signals must contain contiguous data points. Therefore, missing data points are given a non-naturally occurring number (NNON) of  $-99.999$ . These NNON are always considered as unsteady. Furthermore, all the equations presented earlier are handling missing data as they are assessed considering only non-missing data points; there must be at least the number of sample in the time-window of non-missing data for the SSD algorithm to proceed, else it returns a probability of 0 for unsteady.

The SSD algorithm is performed on all the reference variables, univariately, but always considers the presence of the others. To manage multiple process signal that collectively determine whether a system is steady, i.e., multi-univariate SSD, the individual significance level  $\alpha_i$  is evaluated based on Equation (A7). The validity of this equation is predicated on independence of all key process variables. The sole difference between strict univariate and multi-univariate SSD (where there are two or more key independent process variables considered) is the threshold value (more specifically the alpha value). The latter is reduced from the overall process alpha value following the Sidak significance level adjustment equation [74]

$$\alpha'_i = 1 - \sqrt[k]{1 - \alpha_{process}} \quad (A7)$$

where  $k$  is the number of key independent variables,  $\alpha_{process}$  is the desired overall level of significance,  $\alpha_i$  is the required level of significance for each individual variable. Therefore, using  $\alpha'_i$  in the Student-t statistic instead of  $\alpha_{process}$  results in a larger critical value for the same DOF for an individual signal. In a nutshell, the exact same calculation are used for

each variable; however, when looking at more than one,  $\alpha_i$  is used instead in the Student-t statistic. In a multivariable analysis, we are more cautious about rejecting steady-state conditions and more accepting towards the acceptance of steady-state conditions.

Cross correlation between variables is handled by including key process variables that are not correlated, therefore strategically selecting the variables. Additionally, since cross correlation affects the statistical level of significance, the Sidak adjustment is considered in the SSD algorithm. Otherwise, to properly handle cross-correlation across multiple tags, a multivariate statistic such as the Hotelling should be used.

## Appendix B

Data reconciliation consider the impact of process measurement instrumentation by using a tolerance (weight) metric, i.e., the raw sensor or measurement variances are supplied. This metric involve sensor precision, accuracy, and reliability. It considers the type and brand of instrumentation and the measurement data quality. The measurement error tolerance is two times the standard-deviation. Hence, the variance is equal to the square of half the tolerance.

Data reconciliation is based on statistics. A well-established method to detect one or several gross errors is the objective function or global test [81]. This analysis is a multivariate test. The null hypothesis for DR is there is no gross error. This hypothesis is rejected when the objective function is greater than the Chi-squared statistic ( $\chi^2$ ). On the other hand, if its value is less than or equal to its Chi-squared upper control limit, there is no statistically detectable gross error,  $H_0$  is accepted—that means that statistically, all of the measurements are consistent with the model. For the Chi-squared statistic, the degree-of-freedom  $\lambda$  equals to the number of balance equations minus the number independent unmeasured variables. It can be assumed that all unmeasured variables are independent, which means that they are all observable. Coupled with a specified level-of-significance  $\alpha$ , yields its critical value  $\chi^2$  and provides an upper bound or threshold limit for the hypothesis testing.

When DR is performed, there are an infinite number of feasible solutions that satisfies the material, energy, and momentum balance. Every feasible solution is consistent, meaning that it satisfies all of the balances and constraints. The algorithm seeks the one that minimizes an objective function that reads a weighted sum of squares of residuals:

$$\text{Objective function} = \sum \frac{(\text{raw measurement} - \text{reconciled value})^2}{\text{raw variance}} \leq \chi_{\alpha, \lambda}^2 \quad (\text{A8})$$

In Equation (A8), the adjustments of measured variables are weighted, squared, and minimized. However, unmeasured variables do not contribute to the objective function since their weight is zero (0.0). On the other hand, one of the most important pieces of information that comes out of the data reconciliation solving process is the reconciled variances (calculated via the propagation of errors) of all measured variables. The latter provide the gross error detection statistics presented below.

If the null hypothesis  $H_0$  is rejected, the first hypothesis alternative,  $H_1$ , is that there is only one gross error whereas the second alternative,  $H_2$ , is stating that two or more gross errors exist. Therefore, employing the maximum power measurement test statistic (MPMT) gives the opportunity to assess where the GE are in the data set [81]. This is achieved by comparing the MPMT values to a critical value (the Student-t distribution) for statistical significance for each raw variable. The MPMT is calculated as follows:

$$\text{MPMT} = \sqrt{\frac{(\text{raw measurement} - \text{reconciled value})^2}{\text{reconciled variance}}} \leq \text{Student}_{\alpha', \nu} \quad (\text{A9})$$

where  $\alpha'$  is the Sidak significance level adjustment and  $\nu$  is the number of measurements (sensors).



Hence, there is the overall or global hypothesis that there are no gross errors, then there are the individual hypothesis tests for each sensor; the hypothesis test is whether or not an individual sensor is in gross error. Following the calculation of MPMT for every measurement (sensor), the square of the MPMT maximum absolute value is subtracted from the DR objective function value (weighted sum of squares of residuals).

$$\text{Objective function} - \max(|MPMT^2|) \leq \chi_{\alpha, \lambda-1}^2 \quad (\text{A10})$$

If that difference is less than or equal the Chi-squared critical value (with one less DOF),  $H_1$  is accepted, and if it is greater, then  $H_2$  is accepted. When  $H_1$  is accepted, the statistical power of the MPMT can isolate the most-likely bad sensor, it is a reliable identifier of the sensor with the gross error. If  $H_2$  is accepted, then a combinatorial search, also known as a subset-selection enumeration search, is necessary to identify multiple gross errors although serial elimination (remove or delete one measurement at the time) may or may not be effective [80]. In other words, there is no easy way to find multiple gross errors, it becomes a combinatorial problem, and it is even more difficult as the number of expected gross errors needs to be known a priori. Nevertheless, this statistical test is useful when there is a single gross error in the system. In this case, the MPMT statistic can reliably indicate the problematic sensor or instrument—it will be flagged as being in gross error. However, if multiple gross errors exist, then maximum power is not guaranteed, and a more sophisticated analysis is necessary.

Serial elimination can be used to identify two or more gross errors [82]; the first gross error is identified using the maximum absolute MPMT statistics, its corresponding measurement is eliminated, and the DR is repeated.

Consequently, limiting the DR to a scope where one or even two gross errors are expected is the most tractable approach to accurately and precisely detecting, identifying, and eliminating true gross errors in industrial plant data. In that regard, the DR routine is intended to be executed on a regular basis (say on an hourly average frequency) on-line in real-time at a unit-wide level for volume-only and/or volume-density-mass reconciliation. It is also possible to collect past measurement data, and run the DR algorithm off-line (in the past) on hourly averages. A daily plant-wide DR would highlight a lot of gross errors and closing the balance would be nearly impossible. Therefore, by running the GED more often, i.e., on a shorter time interval (every hour instead of every day), and on a smaller scope/scale (unit-wide instead of plant-wide), the ability to diagnose gross errors easily and accurately increases.

As part of the DR algorithm, a distinction is made between sporadic (transient) and sustained (persistent) gross error [83] by pairing the temporal context to the DR problem. First, when a gross error is detected whether it is sporadic or sustained is unknown. In order to distinguish between the two, the number of times a sensor shows up as being inconsistent with the material and energy balances over a monitoring horizon (the probability) is computed; the occurrence is logged and recorded. This metric identifies the most sustained gross errors by ranking the probability of occurrence for the sensors. IMPL-DATA is detecting which of the sensors are persistently in gross error versus sporadically. Generally speaking, the plant personnel will only act on the sustained ones. In other words, data reconciliation focuses on diagnosing persistently or sustained defective, faulty, or bad measurements. Data reconciliation is used to screen the data for persistent gross error and for plant personnel to fix them over time by recalibrating the sensor for instance. Additionally, as mentioned, unless the gross errors are eliminated, the only recourse is to use the raw values with regard to further analysis and reporting since the reconciled values are unreliable.

The data reconciliation stage will avoid removing sensors and instruments. Instead, the sensors declared as persistently in gross error (over a long period) following the DR are flagged and are only ignored temporarily in the DR problem. Specifically, gross error measurements are set to unmeasured variable in the reconciliation, which is identical to

eliminating the sensor value. In order to achieve that, the measurement raw variance is set to be very large, hence its weight becomes null in the objective function. In addition to the probability of occurrence, there is also the notion of severity, i.e., the degree to which the sensors exceed their statistical threshold tolerances. Therefore, based on this quantitative analysis that consider both metrics, justified and appropriate action may be taken to eliminate the most likely gross errors by sensor recalibration, reconfiguration, repair, replacement or modification of its tolerance or uncertainty.

Gross error detection is applied to improve the accuracy in measured data and to identify instrumentation problems that require special maintenance and correction. Moreover, detection of persistent gross errors can reduce maintenance costs and provide smoother plant operation. Eliminating gross error allows to respect material balances to within statistical error every hour, shift, day, etc.

Data reconciliation is primarily focused on the detection, identification, and elimination of gross errors. There seems to be a wide-spread misconception in industry that after performing data reconciliation on process data, the reconciled values are better than their raw measurements, and that reconciled data should be used for subsequent analysis. In the absence of model gross errors, the reconciled values are by definition consistent with respect to its sensor certainty and model constraints and may be considered as conforming, to some degree, to their true values. Yet, if the dataset contain gross error, all the reconciled numbers may in fact be worse than the raw readings given that all (weighted) least squares methods systematically smear or spread the effects of gross errors throughout the model and ultimately corrupt and distort the accuracy of the reconciled data. Even if data reconciliation provides reconciled values, they are rarely employed; the end game of data reconciliation is never to get reconciled values. This is explained by the fact that if there is one gross error in the system (most of the time there are more than one), all the reconciled values are wrong—if one measurement is bad, they are all bad, the whole recipe is wrong. When there is gross errors in a dataset, the latter will propagate into the hole system. Additionally, when there are more than one gross error in the system, it is practically impossible to determine which one are bad. It may sound counter intuitive, but it is better to use the raw, unreconciled measurements for all subsequent analysis such as optimization.

However, if it turns out that after DR, there are no statistically detectable gross errors (all the sensors are consistent and all of their errors are random), then the raw values and the reconciled values can both be used. The only difference between the raw and the reconciled numbers, is that the reconciled values completely satisfy statistically all of the constraints, they close all of the material, energy, and momentum balances perfectly, whereas the raw values, technically do not.

Nevertheless, the objective function (global test) could either be the sum of squares of residuals (L2- or Euclidean-norm), i.e., raw measurements minus reconciled, or it may be the sum of weighted absolute deviations (L1- or Manhattan-norm), i.e., the absolute value of the residual divided by the standard deviation [84]. When performing DR with the 1-norm, the reconciled values are less impacted and sensitive to gross errors (biases or outliers). Hence, if one really wishes to use the reconciled values, while keeping in mind that it is not possible to be a 100% gross error-free, the 1-norm is an interesting alternative and may be considered instead of the 2-norm whereby the reconciled values may exhibit less corruption and distortion.

## References

1. Korbelt, M. On-Line Steady-State Data Reconciliation for Advanced Cost Analysis in the Pulp and Paper Industry. Ph.D. Thesis, Polytechnique Montreal, Montreal, QC, Canada, 2011.
2. Bagajewicz, M. A brief review of recent developments in data reconciliation and gross error detection/estimation. *Lat. Am. Appl. Res.* **2000**, *30*, 335–342.
3. Fang, W.; Shao, Y.; Love, P.E.D.; Hartmann, T.; Liu, W. Detecting anomalies and de-noising monitoring data from sensors: A smart data approach. *Adv. Eng. Inform.* **2023**, *55*, 101870. [[CrossRef](#)]

4. Farsang, B.; Balogh, I.; Nemeth, S.; Szekvolgyi, Z.; Abonyi, J. PCA based data reconciliation in soft sensor development—Application for melt flow index estimation. *Chem. Eng. Trans.* **2015**, *43*, 1555–1560. [[CrossRef](#)]
5. Koren, A.; Jurevi, M.; Prasad, R. Comparison of Data-Driven Models for Cleaning eHealth Sensor Data: Use Case on ECG Signal. *Wirel. Pers. Commun.* **2020**, *114*, 1501–1517. [[CrossRef](#)]
6. Lee, S.; Rao, S.; Kim, M.J.; Esfahani, I.J.; Yoo, C.K. Assessment of environmental data quality and its effect on modelling error of full-scale plants with a closed-loop mass balancing. *Environ. Technol.* **2015**, *36*, 3253–3261. [[CrossRef](#)] [[PubMed](#)]
7. Desbiens, A.; Nunez, E.; Del Villar, R.; Hodouin, D.; Poulin, E. Using process control to increase the energy efficiency of mineral and metal processing plants. *Int. J. Power Energy Syst.* **2008**, *28*, 146–151. [[CrossRef](#)]
8. Thibault, É.; Chioua, M.; McKay, M.; Korbel, M.; Patience, G.S.; Stuart, P.R. Experimental methods in chemical engineering: Data processing and data usage in decision-making. *Can. J. Chem. Eng.* **2023**. [[CrossRef](#)]
9. Jiang, T.; Chen, B.; Jasim, K.; Stuart, P.R. Strategy for improving data quality for a kraft pulp mill recausticizing plant. In Proceedings of the Computer-Aided Process Operations (FOCAPO) Conference, Coral Springs, FL, USA, 12–15 January 2003.
10. Bellec, S.; Jiang, T.; Kerr, B.; Diamond, M.; Stuart, P. On-line processing and steady-state reconciliation of pulp and paper mill process data. *Pulp Pap. Can.-Ont.-* **2007**, *108*, 36–40.
11. Reyes, J.D.; Rodríguez, A.L.; Riascos, C.A.M. Data Analysis and Modelling of a Fluid Catalytic Cracking Unit (FCCU) for an Implementation of Real Time Optimization. In *Computer Aided Chemical Engineering*; Gernaey, K.V., Huusom, J.K., Gani, R., Eds.; Elsevier: Amsterdam, The Netherlands, 2015; Volume 37, pp. 611–616.
12. Delou, P.A.; Ribeiro, L.D.; Paiva, C.R.; Niederberger, J.; Gomes, M.V.C.; Secchi, A.R. A Real-Time Optimization Strategy for Small-Scale Facilities and Implementation in a Gas Processing Unit. *Processes* **2021**, *9*, 1179. [[CrossRef](#)]
13. Jiang, T.; Chen, B.; He, X. Industrial application of Wavelet Transform to the on-line prediction of side draw qualities of crude unit. *Comput. Chem. Eng.* **2000**, *24*, 507–512. [[CrossRef](#)]
14. Jiang, T.; Chen, B.; He, X.; Stuart, P. Application of steady-state detection method based on wavelet transform. *Comput. Chem. Eng.* **2003**, *27*, 569–578. [[CrossRef](#)]
15. Korbel, M.; Bellec, S.; Jiang, T.; Stuart, P. Steady state identification for on-line data reconciliation based on wavelet transform and filtering. *Comput. Chem. Eng.* **2014**, *63*, 206–218. [[CrossRef](#)]
16. Liukkonen, M.; Hiltunen, T.; Hälikkää, E.; Hiltunen, Y. Modeling of the fluidized bed combustion process and NO<sub>x</sub> emissions using self-organizing maps: An application to the diagnosis of process states. *Environ. Model. Softw.* **2011**, *26*, 605–614. [[CrossRef](#)]
17. Srinivasan, R.; Wang, C.; Ho, W.K.; Lim, K.W. Context-based recognition of process states using neural networks. *Chem. Eng. Sci.* **2005**, *60*, 935–949. [[CrossRef](#)]
18. Heikkinen, M.; Poutiainen, H.; Liukkonen, M.; Heikkinen, T.; Hiltunen, Y. Subtraction analysis based on self-organizing maps for an industrial wastewater treatment process. *Math. Comput. Simul.* **2011**, *82*, 450–459. [[CrossRef](#)]
19. Liukkonen, M.; Heikkinen, M.; Hiltunen, T.; Hälikkää, E.; Kuivalainen, R.; Hiltunen, Y. Artificial neural networks for analysis of process states in fluidized bed combustion. *Energy* **2011**, *36*, 339–347. [[CrossRef](#)]
20. Jain, A.K. Data clustering: 50 years beyond K-means. *Pattern Recognit. Lett.* **2010**, *31*, 651–666. [[CrossRef](#)]
21. Saari, J.; Odelius, J. Detecting operation regimes using unsupervised clustering with infected group labelling to improve machine diagnostics and prognostics. *Oper. Res. Perspect.* **2018**, *5*, 232–244. [[CrossRef](#)]
22. Srinivasan, R.; Wang, C.; Ho, W.K.; Lim, K.W. Dynamic Principal Component Analysis Based Methodology for Clustering Process States in Agile Chemical Plants. *Ind. Eng. Chem. Res.* **2004**, *43*, 2123–2139. [[CrossRef](#)]
23. Beaver, S.; Palazoglu, A.; Romagnoli, J.A. Cluster Analysis for Autocorrelated and Cyclic Chemical Process Data. *Ind. Eng. Chem. Res.* **2007**, *46*, 3610–3622. [[CrossRef](#)]
24. Everitt, B. *Cluster Analysis*; Arnold, E., Ed.; Halsted Press: Ultimo, Australia, 1993.
25. Farsang, B.; Nemeth, S.; Abonyi, J. Synergy between data reconciliation and principal component analysis in energy monitoring. *Chem. Eng. Trans.* **2014**, *39*, 721–726. [[CrossRef](#)]
26. Amand, T.; Heyen, G.; Kalitventzeff, B. Plant monitoring and fault detection synergy between data reconciliation and principal component analysis. *Comput. Chem. Eng.* **2001**, *25*, 501–507. [[CrossRef](#)]
27. Yellapu, V.S.; Zhang, W.; Vajpayee, V.; Xu, X. A multiscale data reconciliation approach for sensor fault detection. *Prog. Nucl. Energy* **2021**, *135*, 103707. [[CrossRef](#)]
28. Ramasamy, J.; Devanathan, S.; Jayaraman, D. Comparative analysis of select techniques and metrics for data reconciliation in smart energy distribution network. *Water Sci. Technol. Water Supply* **2021**, *21*, 2109–2121. [[CrossRef](#)]
29. Medeiros, K.A.R.; Matos, A.C.H.d.; Oliveira, E.C.d. Shedding Light on Data Reconciliation Techniques Applied to Analytical Chemistry. *Crit. Rev. Anal. Chem.* **2023**, *53*, 975–985. [[CrossRef](#)]
30. Narasimhan, S.; Bhatt, N. Deconstructing principal component analysis using a data reconciliation perspective. *Comput. Chem. Eng.* **2015**, *77*, 74–84. [[CrossRef](#)]
31. Jeyanthi, R.; Sahithi, M.; Sireesha, N.V.L.; Srinivasan, M.S.; Devanathan, S. Data reconciliation using MA-PCA and EWMA-PCA for large dimensional data. *J. Intell. Fuzzy Syst.* **2021**, *41*, 5731–5736. [[CrossRef](#)]
32. Varshith, C.R.; Rishika, J.R.; Ganesh, S.; Jeyanthi, R. Principal component analysis based data reconciliation for a steam metering circuit. In Proceedings of the International Conference on Soft Computing and Signal Processing, ICSCSP 2018, Hyderabad, India, 22–23 June 2018; Volume 898, pp. 619–626.

33. Liu, Z.; Song, Y.-Q.; Xie, C.-H.; Zhu, F.; Bao, X. Clustering gene expression data analysis using an improved em algorithm based on multivariate elliptical contoured mixture models. *Optik* **2014**, *125*, 6388–6394. [[CrossRef](#)]
34. Mumtaz, A.; Coviello, E.; Lanckriet, G.R.G.; Chan, A.B. Clustering dynamic textures with the hierarchical em algorithm for modeling video. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1606–1621. [[CrossRef](#)]
35. Soor, A.; Mittal, V. An improved method for robust and efficient clustering using EM algorithm with Gaussian Kernel. *Int. J. Database Theory Appl.* **2014**, *7*, 191–200. [[CrossRef](#)]
36. Umatani, R.; Imai, T.; Kawamoto, K.; Kunimasa, S. Time series clustering with an EM algorithm for mixtures of linear Gaussian state space models. *Pattern Recognit.* **2023**, *138*, 109375. [[CrossRef](#)]
37. Arora, J.; Tushir, M.; Dadhwal, S.K. A New Suppression-based Possibilistic Fuzzy c-means Clustering Algorithm. *EAI Endorsed Trans. Scalable Inf. Syst.* **2023**, *10*, e3. [[CrossRef](#)]
38. Bashir, M.A.; Muhiuddin, G.; Rashid, T.; Sardar, M.S. Multicriteria Ordered the Profile Clustering Algorithm Based on PROMETHEE and Fuzzy c-Means. *Math. Probl. Eng.* **2023**, *2023*, 5268340. [[CrossRef](#)]
39. Hashemi, S.E.; Gholian-Jouybari, F.; Hajiaghahi-Keshteli, M. A fuzzy C-means algorithm for optimizing data clustering. *Expert Syst. Appl.* **2023**, *227*, 120377. [[CrossRef](#)]
40. Zhang, Y.; Chen, T.; Jiang, Y.; Wang, J. Possibilistic c-means clustering based on the nearest-neighbour isolation similarity. *J. Intell. Fuzzy Syst.* **2023**, *44*, 1781–1792. [[CrossRef](#)]
41. De la Haba Ruiz, M.A.; Ruiz Perez-Cacho, P.; Dios Palomares, R.; Galan-Soldevilla, H. Classification of artisanal Andalusian cheeses on physicochemical parameters applying multivariate statistical techniques. *Dairy Sci. Technol.* **2016**, *96*, 95–106. [[CrossRef](#)]
42. Goncalves, J.N.C.; Cortez, P.; Carvalho, M.S. K-means clustering combined with principal component analysis for material profiling in automotive supply chains. *Eur. J. Ind. Eng.* **2021**, *15*, 273–294. [[CrossRef](#)]
43. Knadel, M.; Viscarra Rossel, R.A.; Deng, F.; Thomsen, A.; Greve, M.H. Visible-near infrared spectra as a proxy for topsoil texture and glacial boundaries. *Soil Sci. Soc. Am. J.* **2013**, *77*, 568–579. [[CrossRef](#)]
44. Xie, S.; Lawniczak, A.T.; Gan, C. Optimal number of clusters in explainable data analysis of agent-based simulation experiments. *J. Comput. Sci.* **2022**, *62*, 101685. [[CrossRef](#)]
45. Hirose, K.; Miura, K.; Koie, A. Hierarchical clustered multiclass discriminant analysis via cross-validation. *Comput. Stat. Data Anal.* **2023**, *178*, 107613. [[CrossRef](#)]
46. Wang, Q.; Wang, F.; Ren, F.; Li, Z.; Nie, F. An Effective Clustering Optimization Method for Unsupervised Linear Discriminant Analysis. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 3444–3457. [[CrossRef](#)]
47. Mao, Y. Application of Kohonen Neural Network in Sports Cluster. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 2266702. [[CrossRef](#)]
48. Ye, H.; Zhang, L.; Liu, X. Network intrusion clustering based on Fuzzy C-Means and modified Kohonen neural network. *Comput. Model. New Technol.* **2014**, *18*, 154–158.
49. Agarwal, N.; Sikka, G.; Awasthi, L.K. WGSDDMM+GA: A genetic algorithm-based service clustering methodology assimilating dirichlet multinomial mixture model with word embedding. *Future Gener. Comput. Syst.* **2023**, *145*, 254–266. [[CrossRef](#)]
50. Feng, J.; Zhang, J.; Zhu, X.; Wang, J.-H. Gene selection and clustering of single-cell data based on Fisher score and genetic algorithm. *J. Supercomput.* **2023**, *79*, 7067–7093. [[CrossRef](#)]
51. Gunjan; Sharma, A.K.; Verma, K. GA-UCR: Genetic Algorithm Based Unequal Clustering and Routing Protocol for Wireless Sensor Networks. *Wirel. Pers. Commun.* **2023**, *128*, 537–558. [[CrossRef](#)]
52. Uma, K.; Perumal, K. A novel Swarm Optimized Clustering based genetic algorithm for medical decision support system. *Meas. Sens.* **2023**, *28*, 100821. [[CrossRef](#)]
53. Fu, N.; Ni, W.; Zhang, S.; Hou, L.; Zhang, D. GC-NLDP: A graph clustering algorithm with local differential privacy. *Comput. Secur.* **2023**, *124*, 102967. [[CrossRef](#)]
54. Mei, G.; Tu, J.; Xiao, L.; Piccialli, F. An efficient graph clustering algorithm by exploiting k-core decomposition and motifs. *Comput. Electr. Eng.* **2021**, *96*, 107564. [[CrossRef](#)]
55. Moradi, P.; Ahmadian, S.; Akhlaghian, F. An effective trust-based recommendation method using a novel graph clustering algorithm. *Phys. A: Stat. Mech. Its Appl.* **2015**, *436*, 462–481. [[CrossRef](#)]
56. Nascimento, M.C.V.; Carvalho, A.C.P.L.F. A graph clustering algorithm based on a clustering coefficient for weighted graphs. *J. Braz. Comput. Soc.* **2011**, *17*, 19–29. [[CrossRef](#)]
57. Joe Qin, S.; Guo, S.; Li, Z.; Chiang, L.H.; Castillo, I.; Braun, B.; Wang, Z. Integration of process knowledge and statistical learning for the Dow data challenge problem. *Comput. Chem. Eng.* **2021**, *153*, 107451. [[CrossRef](#)]
58. Canada, G.o. EXPLORE. Available online: <https://www.rncan.gc.ca/cartes-outils-et-publications/outils/outils-modelisation/explore/24825> (accessed on 1 May 2023).
59. Kelly, J.D. *Industrial Modeling & Programming Language (IMPL) Manual*; 2023.
60. Huang, G. Missing data filling method based on linear interpolation and lightgbm. *J. Phys. Conf. Ser.* **2021**, *1754*, 012187. [[CrossRef](#)]
61. Gitzel, R. *Data Quality in Time Series Data: An Experience Report*; CBI: New Delhi, India, 2016.
62. Epitropakis, A.; Papadakis, I. Statistical properties of Fourier-based time-lag estimates. *Astron. Astrophys.* **2016**, *591*, A113. [[CrossRef](#)]

63. Setiawan, I.; Morgan, L.K.; Doscher, C. Saltwater intrusion from an estuarine river: A field investigation. *J. Hydrol.* **2023**, *617*, 128955. [CrossRef]
64. Tetarenko, A.J.; Casella, P.; Miller-Jones, J.C.A.; Sivakoff, G.R.; Paice, J.A.; Vincentelli, F.M.; Maccarone, T.J.; Gandhi, P.; Dhillon, V.S.; Marsh, T.R.; et al. Measuring fundamental jet properties with multiwavelength fast timing of the black hole X-ray binary MAXI J1820+070. *Mon. Not. R. Astron. Soc.* **2021**, *504*, 3862–3883. [CrossRef]
65. Ventosa, S.; Schimmel, M.; Stutzmann, E. Towards the processing of large data volumes with phase cross-correlation. *Seismol. Res. Lett.* **2019**, *90*, 1663–1669. [CrossRef]
66. Müller, M. Dynamic Time Warping. In *Information Retrieval for Music and Motion*; Müller, M., Ed.; Springer: Berlin/Heidelberg, Germany, 2007; pp. 69–84.
67. Liu, H.; Shah, S.; Jiang, W. On-line outlier detection and data cleaning. *Comput. Chem. Eng.* **2004**, *28*, 1635–1647. [CrossRef]
68. Dunn, K. Box plots. Dans *Process Improvement Using Data*. 2022. Available online: <https://learnche.org/pid/data-visualization/box-plots> (accessed on 1 May 2023).
69. Kartashov, O.O.; Chernov, A.V.; Polyanichenko, D.S.; Butakova, M.A. XAS data preprocessing of nanocatalysts for machine learning applications. *Materials* **2021**, *14*, 7884. [CrossRef]
70. Krishnamurthi, R.; Kumar, A.; Gopinathan, D.; Nayyar, A.; Qureshi, B. An Overview of IoT Sensor Data Processing, Fusion, and Analysis Techniques. *Sensors* **2020**, *20*, 6076. [CrossRef]
71. Barton, M.; Lennox, B. Model stacking to improve prediction and variable importance robustness for soft sensor development. *Digit. Chem. Eng.* **2022**, *3*, 100034. [CrossRef]
72. Thibault, É.; Désilets, F.L.; Poulin, B.; Chioua, M.; Stuart, P. Comparison of signal processing methods considering their optimal parameters using synthetic signals in a heat exchanger network simulation. *Comput. Chem. Eng.* **2023**, 108380. [CrossRef]
73. Rhinehart, R.R. Automated steady and transient state identification in noisy processes. In *Proceedings of the 2013 American Control Conference*, Washington, DC, USA, 17–19 June 2013; pp. 4477–4493.
74. Kelly, J.; Hedengren, J. A steady-state detection (SSD) algorithm to detect non-stationary drifts in processes. *J. Process Control* **2013**, *23*, 326–331. [CrossRef]
75. Kelly, J.D. Techniques for solving industrial nonlinear data reconciliation problems. *Comput. Chem. Eng.* **2004**, *28*, 2837–2843. [CrossRef]
76. Harrison, R.P.; Stuart, P.R. Data pre-processing techniques for multivariate analysis to treat industrial operating data for retrofit design. In *Proceedings of the Canadian Engineering Education Association (CEEA)*, Kaninaskis, NB, Canada, 18–20 July 2005.
77. Kalloo, G.; Wellenius, G.A.; McCandless, L.; Calafat, A.M.; Sjodin, A.; Karagas, M.; Chen, A.; Yolton, K.; Lanphear, B.P.; Braun, J.M. Profiles and Predictors of Environmental Chemical Mixture Exposure among Pregnant Women: The Health Outcomes and Measures of the Environment Study. *Environ. Sci. Technol.* **2018**, *52*, 10104–10113. [CrossRef]
78. Dunn, K. Principal Component Analysis (PCA). *Process Improvement Using Data*. 2022. Available online: <https://learnche.org/pid/latent-variable-modelling/principal-component-analysis/index> (accessed on 1 May 2023).
79. Moreno, R.d.P. Steady State Detection, Data Reconciliation, and Gross Error Detection: Development for Industrial Processes. Master's Thesis, University of New Brunswick, Fredericton, NB, Canada, 2010.
80. Kelly, J.D.; Zyngier, D. A new and improved MILP formulation to optimize observability, redundancy and precision for sensor network problems. *AIChE J.* **2008**, *54*, 1282–1291. [CrossRef]
81. Narasimhan, S.; Jordache, C. *Data Reconciliation and Gross Error Detection: An Intelligent Use of Process Data*; Gulf Professional Publishing: Houston, TX, USA, 2000.
82. Bagajewicz, M.; Jiang, Q.; Sanchez, M. Performance evaluation of PCA tests in serial elimination strategies for gross error identification. *Chem. Eng. Commun.* **2010**, *183*, 119–139. [CrossRef]
83. Tong, H.; Crowe, C.M. Detecting persistent gross errors by sequential analysis of principal components. *Comput. Chem. Eng.* **1996**, *20*, S733–S738. [CrossRef]
84. Zhan, H.-r.; Miao, Y.; Wang, W. Extended support vector regression based data reconciliation and its application to plant-wide mass balance. *Int. J. Innov. Comput. Inf. Control* **2012**, *8*, 4111–4122.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.