Stream Correlation Monitoring for Uncertainty-Aware Data Processing Systems

Aleka K. Seliniotaki

Thesis submitted in partial fulfillment of the requirements for the

Masters' of Science degree in Computer Science

University of Crete
School of Sciences and Engineering
Computer Science Department

Thesis Advisor: Prof. Vassilis Christophides

This work is partially supported by HYDROBIONETSproject (ICT-2011-7) funded by the European commission in FP7 (GA-2011-287613)

Abstract

In several industrial applications, monitoring large-scale infrastructures in order to provide notifications for abnormal behavior is of high significance. For this purpose, the deployment of large-scale sensor networks is the current trend. However, this results in handling vast amounts of low-level, and often unreliable, data, while an efficient and real-time data manipulation is a strong demand. In this thesis, we propose an uncertainty-aware data management system capable of monitoring interrelations between large and heterogeneous sensor data streams in real-time. To this end, an efficient similarity function is employed instead of the typical correlation coefficient to monitor dynamic phenomena for timely alerting notifications, and to guarantee the validity of detected extreme events. Experimental evaluation with a set of real data recorded by distinct sensors in an industrial water desalination plant reveals a superior performance of our proposed approach in terms of achieving significantly reduced execution times, along with increased accuracy in detecting extreme events and highly correlated pairs of sensor data streams, when compared with state-of-the-art data stream processing techniques.

Περίληψη

Σε πολλές βιομηχανικές εφαρμογές η παρακολούθηση υποδομών μεγάλης κλίμακας είναι υψηλής σημασίας, προκειμένου να παρέχονται οι κατάλληλες ειδοποιήσεις για μη αναμενόμενη συμπεριφορά. Αυτός είναι και ο λόγος που αυξήθηκε η τάση για την ανάπτυξη δικτύων αισθητήρων μεγάλης κλίμακας. Ωστόσο, αυτή η επεξεργασία από τα δίκτυα αισθητήρων καταλήγει στο χειρισμό μεγάλου όγκου δεδομένων πληροφορίας με χαμηλή ποιότητα. Η αναξιοπιστία στα δεδομένα επεξεργασίας, αποτελεί έναν ανασταλτικό παράγοντα στα συστήματα διαχείρισης δεδομένων πραγματικού χρόνου των οποίων η ζήτηση είναι αυξημένη. Σε αυτήν την εργασία προτείνουμε ένα σύστημα επεξεργασίας δεδομένων χωρίς αβεβαιότητα, το οποίο είναι ικανό να παρακολουθεί τη συμπεριφορά και τις αλληλεπιδράσεις μεταξύ μεγάλων και ετερογενών ροών δεδομένων από αισθητήρες, σε πραγματικό χρόνο. Η προσέγγισή μας χρησιμοποιεί μία συνάρτηση ομοιότητας, αντί τον τυπικό συντελεστή συσχέτισης , έτσι ώστε να πραγματοποιείται η παρακολούθηση στην εξέλιξη δυναμικών φαινομένων για την έγκαιρη ανακοίνωση σημαντικών ειδοποιήσεων, καθώς και για την εξασφάλιση της εγκυρότητας για ανίχνευση σημαντικών γεγονότων. Η πειραματική μας αξιολόγηση σε ένα σύνολο πραγματικών δεδομένων τα οποία καταγράφονται από αισθητήρες σε μία βιομηχανική μονάδα αφαλάτωσης νερού, εμφανίζει αυξημένη απόδοση όσον αφορά στην επίτευξη μειωμένων χρόνων απόκρισης του συστήματος. Επιπλέον, παρουσιάζει μεγάλη ακρίβεια στην ανίχνευση υψηλά συσχετιζόμενων ροών δεδομένων, σε αντίθεση με διαφορετικές τεχνικές επεξεργασίας ροών δεδομένων.

Ευχαριστίες

Πρωτίστως θα ήθελα να ευχαριστήσω ενθέρμως τον επόπτη καθηγητή μου κ. Βασίλη Χριστοφίδη, καθ'ότι η παρούσα μεταπτυχιακή εργασία δεν θα μπορούσε να ολοκληρωθεί χωρίς την συμβολή του και την αμέριστη προσοχή του. Τον ευχαριστώ για την αγαστή συνεργασία μας, για την ορθή του καθοδήγηση και για την επιμονή του στην ποιοτική δουλειά. Η οργανωτική του εμπειρία και το γεγονός ότι είναι ένας υπέροχος άνθρωπος, με γενναιόδωρο χαρακτήρα, με βοήθησαν να βελτιώσω τον εαυτό μου και τις ικανότητές μου.

Κατόπιν, ευχαριστώ τον κ. Παναγιώτη Τσακαλίδη για τη συμβολή του στη στατιστική επεξεργασία σήματος και για τη συνεισφορά του σε κάθε φάση αυτής της μελέτης, διακρίνοντάς τον πάντα εξαιρετική ευγένεια και αμεσότητα.

Ευχαριστώ πολύ την κ. Μαρία Παπαδοπούλη για τις σημαντικές συμβουλές της και για την καθοδήγηση που μου έδωσε με μεγάλο ενθουσιασμό, για την ολοκλήρωση της παρούσας διατριβής.

Φυσικά, δεν πρέπει να παραλείψω να ευχαριστήσω τον κ. Γιώργο Τζαγκαράκη, για τις υποδείξεις του σε σημαντικά και ασήμαντα ζητήματα, που όμως ήταν καθοριστικά για την ολοκλήρωση της εργασίας μου. Οι γνώσεις του, η αναλυτικότητά του, ο επαγγελματισμός του και η υπομονή του λειτούργησαν ως πρότυπο και αξίες τόσο για αυτήν την εργασία όσο και για μελλοντικές.

Επίσης, θα ήθελα να ευχαριστήσω τον σύζυγό μου Γιάννη και τον γιο μας Κωνσταντίνο για την κατανόηση που έδειξαν, όταν αφιέρωνα τον κοινό μας χρόνο για την ολοκλήρωση της εργασίας μου. Τους ευγνωμονώ που ήταν διπλά μου με αφοσίωση, με υπομονή και κατανόηση και με στήριξαν σε αυτό το πιεστικό και ταυτοχρόνως σημαντικό βήμα των σπουδών μου.

Τέλος, θα ήθελα να ευχαριστήσω το εργαστήριο Πληροφοριακών Συστημάτων του Ιδρύματος Τεχνολογίας και Έρευνας και το HYDROBIONETS project για την υλικοτεχνική υποστήριξη.

Table of Contents

List of Figures

List of Tables

1. Introduction

Recent advances in information and communication technology (ICT) have led to a significant progress in the design of devices incorporating wireless communication, processing and storage capabilities, as well as diverse sensing and actuation functionalities in a single unit that is compact, economical, autonomous and destined to become ubiquitous. This revolution appears in the form of dense and distributed large-scale self-organized wireless sensor networks (WSN) for carrying out various tasks that are of great societal interest, such as environmental monitoring and surveillance or monitoring and management in large-scale industrial infrastructures. The HYDROBIONETS project[1] is a characteristic example of such an infrastructure for water resource management. Specifically, it targets at developing a real-time micro-biological wireless networked control system for water desalination and treatment plants, providing the fundamental design principles of a wireless BioMEM network (WBN) with distributed multi-sensing and multi-actuation capabilities.

The HYDROBIONETS infrastructure focuses on the monitoring of the complete water cycle in large-scale water treatment and desalination plants. The deployment of a WBN aims at monitoring critical microbiological and electrochemical parameters of water at different stages of the desalination process. The associated distributed, autonomous sensing is further exploited to produce *intelligent reasoning* over the data by supporting advanced operations, such as the detection of high fouling concentration in seawater, the control of biocide and chlorine dosage by measuring bacteria in seawater at different stages of water treatment (pre-filtered, pre-treatment and reverse osmosis phases) at periodic time intervals. These functionalities essentially provide the building blocks of the actuation process for water desalination at different locations in the plant.

At the core of the HYDROBIONETS system, which curries out those operations, is an efficient *data processing* module. This module comprises of distinct collaborating computational nodes, which monitor and control several physical entities and dynamic phenomena. The sensor data and metadata, which are produced in streams by the sensors, can be either processed in real time or stored for further exploitation. Those data can be raw (as produced by the sensors) or aggregated, which are produced based on calculations at the node level. To accommodate the requirements of our industrial paradigm we focus on the design and development of a set of tools to deal with high-level analysis of the collected data. These tools will work on the available data and they report and employ in a coherent manner an appropriate statistical analysis in order to: (i) monitor continuously a dynamic system, (ii) detect extreme events (e.g. presence of highly contaminant substances) and provide specific alerts depending on the level of severity of the event, (iii) guarantee the validity of the detected extreme events, and (iv) account for the underlying uncertainty of the recorded data.

Rather than computing single stream statistics, such as average and standard deviation, our data analysis is focusing on *finding high correlations* among pairs of data streams from distinct sensors. More specifically, a system operator may rely on

[1] http://www.hydrobionets.eu/

pairwise sensor stream correlations to reveal interrelations between seemingly independent physical quantities monitored by distinct sensors. This can be further exploited to guarantee the validity of a detected extreme event and provide the necessary alerting notifications. For instance, temperature and pressure sensors, which monitor an industrial plant, could provide evidence of an increasing bacteria presence. Depending on their physical location in the plant we expect that corresponding data streams will be highly correlated. Moreover, in HYDROBIONETS, the measurements from heterogeneous sensors, distributed over a geographic area, need to be processed efficiently in order to reconstruct the spatio-temporal behavior of desired physical variables or to detect, identify and localize sources and events of interest.

Whereas traditional statistical machine learning provides well-established mathematical tools for data analysis [14][23][24][25], their performance is limited when processing high-dimensional data streams. Specifically, existing techniques for monitoring pairwise stream correlations exhibit several drawbacks. In a recent work [27] the problem of maintaining data stream statistics over sliding windows is studied, with the focus being only on single stream statistics. On the other hand, [28] introduced an extension for monitoring the statistics of multiple data streams, but the computation of correlated aggregates is limited to a small number of monitored streams. In addition, StatStream [20] has been proven to be a successful data stream monitoring system, which enables the computation of single- and multiple-stream statistics. However, the main drawback of this technique is the difficulty to define an appropriate "similarity" function for data streams describing dynamic phenomena with unknown prior distributions, which is normally the case in an industrial environment.

The aforementioned solutions do not apply in the case of monitoring and comparing the behavior of data streams. The challenges of this study include: (i) the dynamic evolution of the phenomena and lack of an *a-priori* knowledge of the characteristics of their values and errors that may accur, (ii) the comparison between independent physical quantities measured in different scales and (iii) the inherent data uncertainty, due to the presence of incomplete, imprecise, and even sometimes misleading data , which hind an accurate and reliable decision making.

In this thesis, we overcome the limitations of the previous approaches by introducing a computationally efficient *"similarity extraction"* module, which enables the monitoring of pairwise correlations between high-dimensional sensor data streams on the fly. We note here that time synchronization is also performed between the acquired data streams, prior to the extraction of highly correlated pairs, based on their corresponding time stamps, which are available as a part of the transmitted packets. In particular, instead of computing all pairwise correlations between the original full-dimensional data streams, we exploit the compressibility property of the discrete Fourier transform (DFT) to concentrate the inherent energy content of a given signal in the first few high-amplitude coefficients, as in [20]. Then, a suitable peak similarity measure is applied on the associated pairs of truncated DFTs as a proxy for the corresponding correlation coefficients. Thus the problem of identifying

highly correlated pairs of data streams is reduced to a problem of identifying pairs of truncated DFTs with high peak similarity values.

It is worth also to stress that usually WSN nodes do not handle any quality aspect of physical device data but rather interface with a high-level representation and reconstruction of the sensed physical world. As a result, the HYDROBIONETS data processing subsystem has to additionally cope with the data *uncertainty*, where stream data may be incomplete, imprecise, and even misleading [40], thus impeding the task of an accurate and reliable decision making. *Uncertainty-aware data management* [4] presents numerous challenges, in terms of collecting, modeling, representing, querying, indexing and mining the data. Since many of these issues are interrelated, they cannot be easily addressed independently. Uncertainty has been recently recognized as an additional source of information that could be valuable during data analysis, and thus, should be preserved. More specifically, a *spreadsheet-based* approach is employed to identify, quantify, and combine the underlying uncertainty from the most dominant potential sources of uncertainty, as presented in [31].

Another major functionality assigned to our *uncertainty-aware data processing system* is to perform high-level operations, and specifically to provide notifications of *extreme events* by employing raw sensor data [25]. Two widely-used methods for notifying a system operator whether the data has unexpected values are: (i) *compliance with operating limits* (COL), and (ii) the method of *peaks over a threshold* (POT) [41]. Since the detection of abnormal behavior is affected by the underlying uncertainty, the above two extreme event detectors are modified accordingly so as to account for the imprecise nature of the raw sensor data.

Our proposed system is completed with the integration of appropriate rules for uncertainty propagation after a query execution has finished. The result of the uncertainty of an aggregation will not be measured directly. For instance, what is the error in Z = A + B, where A and B are two measurements with errors ΔA and ΔB respectively? A first thought might be that the error in Z would be just the sum of the errors in A and B, that is, $(A + \Delta A) + (B + \Delta B) = (A + B) + (\Delta A + \Delta B)$. However, this assumes that, when combined, the errors in A and B have the same sign and maximum magnitude, that is, they always combine in the worst possible way. This could only happen if the errors in the two variables were perfectly correlated. Uncertainty propagation may be also viewed from the perspective of queries sent by an operator, for actions to be taken on the recorded data streams (*e.g.,* join, aggregation). We identify the most appropriate methods to achieve a robust estimation of (i) the raw data uncertainty and (ii) the uncertainty resulting from query processing. This results also in a balanced trade-off between the computational burden and the accurate estimation of the underlying uncertainty.

The performance of our proposed system is evaluated using a set of real-world data provided by ACCIONA Agua, recorded by a set of distinct electromechanical sensors in the La Tordera's desalination plant[2].Specifically, it achieves highly

[2] http://aca-web.gencat.cat/aca/documents/ca/sensibilitzacio/desal_Tordera/dessalinitzacio_en.pdf

reduced execution times in conjunction with accurate estimation of the highly-"similar" pairs of sensor streams, as well as a timely alerting performance, when compared with existing widely used data analysis techniques.

To summarize, the main contribution of this thesis is threefold: (i) a fast and robust method is proposed for uncertainty-aware monitoring of pairwise interrelations ("similarities") between distinct sensors, which outperforms state-of-the-art pairwise correlation extraction methods; (ii) in contrast to common data management, which relies on the raw measurements, we verify that the underlying data uncertainty is a valuable source of information, which should be preserved, towards providing more ubiquitous data descriptions; and (iii) the performance of two widely-used extreme event detection methods is enhanced by incorporating the inherent data uncertainty component.

Our utmost goal is to provide a valuable insight into the design and implementation principles of an efficient and robust data processing system. The integration of the above three functionalities in industrial monitoring and surveillance applications has indicated the role of the underlying data uncertainty as an additional source of information, which should be preserved across all stages of the data processing chain.

2. Problem Description

The problem of quality and quantity of water resources is a global challenge for the upcoming years. Both an adequate amount of water and adequate water quality are essential for public health and hygiene. Waterborne diseases are among the leading causes of morbidity and mortality in low-and middle-income countries, frequently called developing countries.

In recent years, treated wastewater has been used as a source of water for certain applications. This is generally named "Water Reuse". Wastewater reclamation is gaining popularity worldwide as a means of conserving natural resources used for drinking water supply. Recycled water is most commonly used for non-potable purposes, such as agriculture, landscape, public parks, and industrial applications, among others. Both water treatment and desalination plants play a major role in terms of obtaining large quantities of water with good quality.

The above requirements, made the need for the implementation and deployment of a large-scale Self-Organized Wireless BioMEM Network (WBN). The WBN will be responsible for microbiological autonomous monitoring and decentralized control of water quality in industrial environments. So, it gives us the opportunity to improve the quality of life, safety and security of water supply. The HYDROBIONETS project is a characteristic example of such an infrastructure for water resource management. Specifically, it targets at developing a real-time microbiological wireless networked control system for water desalination and treatment plants, providing the fundamental design principles of a wireless BioMEM network with multi-sensing and multi-actuation capabilities.

We propose a data processing subsystem, which aims to support the HYDROBIONETS WSN infrastructure for multi-sensing and multi-actuation in water treatment and desalination plants. In our case, a desalination pilot plant is located in La Tordera, which is equipped with a number of various electrochemical sensors, scattered in distinct locations, for monitoring several physical and mechanical variables in the plant. The major contribution of this thesis is the main component of this data processing subsystem. It is responsible for finding high *correlations* among pairs of data streams. A system operator may use this information to identify interrelations between seemingly unrelated physical quantities monitored by distinct sensors, or to guarantee the validity of a detected extreme event. Thus on-the-fly monitoring of potential correlations in the recorded details is crucial to extract meaningful information and provide the necessary notifications. Moreover, in HYDROBIONETS, measurements from heterogeneous sensors, distributed over a geographic area, need to be processed efficiently in order to reconstruct the spatio-temporal behaviour of desired physical variables or to detect, identify and localize sources and events of interest.

This chapter describes the actuation process in HYDROBIONETS infrastructure. More specifically in section 2.1 we mention the main objectives of the HYDROBIONETS project and explain the phenomenon of biofouling developed due to water treatment. In the section 2.2 we describe the sensing performed at various points in the plant at different times. By describing the complexity of the phenomena presented in such cyber-physical systems, we justify why we don't use simple

models for monitoring the actuation process on them. In fact, only with detailed data analysis processing we can find spatiotemporal correlations, which allow us the timely actuation in water treatment process. A major challenge arises from our data analysis processing· the existence of uncertainty in our data streams. For this reason, in subsection 2.2.1, we analyse the data quality characteristics for the sensors of HYDROBIONETS' project. Also we indicate how the knowledge of these key dimensions of data quality enhances the non-existence of uncertainty in the data to be processed. Finally, in section 2.3 we describe the requirements for high-level data analysis.

2.1. *Industrial Monitoring Setting*

Our developed data processing system is at the core of the HYDROBIONETS project, which focuses on the research and development of Self-Organized Wireless BioMEM Networks and their integration in a global system to monitor the complete water cycle in large-scale water treatment and desalination plants. The WBNs will achieve a distributed monitoring and control of critical microbiological parameters of water in the different stages of process in desalination plants. Water treatment plants as a solution to water scarcity and water treatment for reuse has a number of advantages:

(i) low energy requirement for water production,
(ii) potential for using the water in different manner, and
(iii) environmentally friendly.

Different technologies can be applied for water desalination and specifically seawater desalination. The HYDROBIONETS project focuses on desalination by reverse osmosis and in the waste-water treatment plants by Membrane Bio-Reactor (MBR). The use of MBR technology has been proven to be a feasible and efficient method of producing reclaimed water [1]. The osmotic membrane also is referred to as a semi-permeable membrane because of its capability to allow some constituents to pass through it while holding back others.

There are two major control problems that have been studied in this project:

(i) Control of the aeration process in membrane tanks to avoid *fouling* of the fibre surface and adjusting also the level of aeration to save energy.
(ii) Control of the MBR cleaning procedure by estimating more precisely the need and frequency of membrane cleaning, as well as the dose of chemicals to be used.

The cleaning in MBR systems is performed by chemical shock and backwashing techniques, taking into account any sensor information, thus it is usually performed less aggressively than needed. The duration of the cleanings depend on how severe the biofouling is (which is currently estimated by observing the pressure drop measurements or by membrane autopsy). If cleaning is not frequently needed, the associated costs are relatively low.

Fouling refers to the accumulation of unwanted material on solid surfaces, most often in an aquatic environment. The fouling material can consist of either living organisms (biofouling) or a non-living substance (inorganic or organic). However, in practice, when the fouling takes place, it includes all types of material, that is: organic, inorganic and

bacterial fouling (biofouling). Fouling phenomena are common and diverse, ranging from fouling of ship hulls, natural surfaces in the marine environment (marine fouling), fouling of heat-transfer components through ingredients contained in the cooling water or gases, and very often, in desalination membranes and MBR membranes.

The growth of a fouling layer due to the deposition of undesirable materials on the membrane is a persistent problem in water treatment membrane processes. Particular fouling, that is the deposition of suspended solids, colloids and microbiological cells onto or into the membranes, is an especially delicate issue in the membrane filtration operation. Its complete removal by intensive pretreatment of the feed water is not always feasible. A technique for early warning and fouling monitoring is the desire of all engineers to achieve the long-term and stable operating performance of a membrane process.

To fulfill this requirement, an integrated technique is proposed for the online fouling monitoring of a water treatment membrane filtration process. This online monitoring technique provides dynamic and real-time information about a fouling phenomenon and includes the process-oriented capabilities of

(i) in situ measurement of fouling layer thickness,
(ii) dynamic analysis of fouling layer structure and
(iii) monitoring of membrane fouling potential in membrane filtration processes for water treatment applications.

Membrane fouling has been, and continues to be, a major issue in the MBR systems. Most MBR plants operate at relatively modest constant flux as a strategy to slow down the membrane fouling rate and hence reduce the frequency of membrane chemical cleaning. It is also prevented with aeration, which inhibits particles from attaching to the surface of the ultrafiltration membranes. Biofouling that occurs in MBR systems is associated with other foulants, such as suspended solids, nutrients etc. which are quite difficult to differentiate. This is why when referring to MBR, this phenomena become more general, since it is caused by both inorganic and organic matter, and is simply called *fouling.*

To fulfill these requirements, an appropriately deployed WSN acquires measurements from distinct physical variables recorded by various electrochemical sensors, such as, temperature, turbidity, conductivity, oxygen content, pH, redox potential, nitrate and chlorine. Based on the monitoring process further operations take place including:

(i) the detection of high fouling potential,
(ii) the optimization of chemical cleaning of the ultrafiltration membranes,
(iii) the MBR membranes cleaning, and
(iv) the control of chlorine dosage during the reverse osmosis phase.

2.2. ***Sensing in La Tordera's desalination pilot plant.***

The autonomous control of the MBR fouling and cleaning procedures is achieved with sensing and actuation functionalities. In HYDROBIONETS, there are two main classes of important sensors, *electrochemical* sensors and *bacteriological* sensors (Chlorine and Biofilm). Electrochemical sensors can measure quantities, such as temperature, turbidity,

conductivity, oxygen content, pH, redox potential, nitrate and chlorine concentration, etc., while bacteriological sensors used to measure bacteria, biocides and bio-fouling, detecting and measuring traces of Escherichia coli, Salmonella, Shigella, Pseudomonas, Legionella, etc.

Using the Chlorine and Biofilm (BioMEMs) sensor measurements at different locations is expected to help to obtain a more precise measurement of water biofouling potential, predicting earlier the possible growth of biofouling. The electrochemical sensors are located before and after of the water desalination stages (pre-filtered, pre-treatment and reverse osmosis) and they are characterized as *conventional liquid analysis* sensors. The measurements of these sensors are using to achieve and synchronize, with the best way, the operations of the water treatment. In most cases, to build in complex processes, the sensor measurements need to be collected and jointly processed. The following paragraphs report the main usage of BioMEMs and of selected electrochemical sensors (pH, temperature, pressure).

The *Biofilm* sensor is able to measure fouling potential (not only bacteria) at constant flow. The sensor measurements are using for measuring the fouling potential and for chemical cleaning. More specifically, the information of the Biofilm sensor data helps us for:

(i) the optimization for MBR cleaning procedure,
(ii) the optimization of biocide dosage at different stages of the water treatment and
(iii) the optimization of aeration in MBR systems.

By law, residual chlorine must be below a certain value and the MBR membranes cannot be in contact with chlorine. The *Chlorine* sensor is used as a security system checking that there is no residual chlorine before the MBR membranes. The chlorine sensor could be used to optimize the dosing of chlorine and also to know the current concentration of chlorine at different points of the process. Briefly, *Chlorine* sensor measurements are used to control (i) chlorine entering into the membranes and (ii) chlorine at different stages of the water treatment.

The complex phenomenon of biofouling can be ascertained by several variables, such as pH and temperature of water. Also, the presence of sodium hydroxide and hydrochloric acid in the water is testified by PH values and the water PH should be adjusted before its input in the different stages of water desalination. The PH sensor measures the water pH in frequent intervals and it is placed in different stages at water treatment plant, such as in pre-treatment and reverse osmosis stage.

Pressure sensors give a signal when a certain process situation is achieved, for example: high or low pressure. These sensors are located near in membranes and are capable to give signal, before water pressure has increased significantly. Generally, they control the water pressure lying in the interval from 0 to 3 bars. The water pressure is associated directly with the phenomenon of biofouling, since the biofouling affects the flux of water and, therefore, an increase of differential pressure occurs. The MBR cleaning procedure is also optimized by the monitoring of pressure. By observing the pressure measurements, is estimated how severe the biofouling is.

Another important water property that should be monitored is the *salinity*. The quality of water effluent from MBR membranes is increasing by reducing the salinity of reclaimed water. The salinity property is tracked by conductivity sensor measurements.

The growth of biofouling starts when organic matter begins to accumulate on the surface of the first MBR membrane on reverse osmosis phase. This organic matter consists mainly of bacteria, whose growth is monitored by the biofilm sensor. The complex phenomenon of biofouling can be affected negatively by several variables, such as organic matter, pH and temperature of feed water. All these variables have to be controlled during water treatment, as a modification in the acquired data streams, indicates the existence of highly concentrated bacteria. The correlations between data streams from the above sensors (i) warn us about the development of the biofouling phenomenon and (ii) give us the guarantees for the existence of this.

Biofouling affects the flux of water that is processed through membranes and, therefore, an increase of differential pressure occurs. In order to maintain the flux of product water, high pressure pumps have to increase in frequency, resulting in higher energy consumption. When differential pressure has increased, biofouling has already been formed. The duration of the cleanings depend on how severe the biofouling is, which is currently estimated by observing the pressure measurements or by membrane autopsy.

In the RO desalination process, a pressure to the saline water greater than a distinct value will cause fresh water to flow faster through the MBR membranes, holding back the salts. The higher the applied pressure is, the higher the rate of fresh water transports across the membranes. Measuring the pressure and flow rate (this water property is tracked by electromagnetic flowmeter sensor) of water at specific positions in the plant, the analogue correlation between these measurements, inform us that the flow of fresh water is properly carried out.

Redox measurements are used to control the chlorine dosage at different positions of the process. Currently, the activation, deactivation and regulation (dosage) of the biocide pumps is done based on redox measurements. Redox sensors have a low response time if there is a sudden rise of chlorine concentration. So, if we observe reduced response from redox sensor, then we check the chlorine concentration.

The autonomous sensing described above, proves how complex is the surveillance, monitoring and management of large-scale infrastructures. The monitoring of dynamic phenomena (such as the development of biofouling or the modifications in water temperature/pH/pressure), as described in the last paragraphs, becomes more complex, when the data *uncertainty* is appearing in senor measurements. We consider the *uncertainty* as an additional source of valuable information for data analysis which should be preserved.

2.2.1. Managing Streaming Data and their Quality

In a typical wireless sensor network, measurements from heterogeneous sensors distributed over a geographic area need to be processed in order to reconstruct the spatiotemporal behavior of desired physical variables or to detect, identify, and localize sources and events of interest. The HYDROBIONETS large-scale

infrastructure was designed under constraints on cost, bandwidth and energy resources while optimizing performance metrics such as reconstruction fidelity, detection performance, latency etc. In this setting, data quality is becoming a crucial issue in the design of real sensor systems. It is nowadays widely recognized that a typical characteristic of sensor data is their *uncertain* and *erroneous* nature, due to discharged batteries, network failures, and imprecise readings from low-cost sensors. This poses significant limitations on data utilization, since applications using data with low quality may yield unsound results. To address this issue, it is essential to assess as early as possible the quality of data, and process data while reflecting the data quality. *Consistency, accuracy, reliability,* and *survivability* concerns have to be addressed in sensor data acquisition, storage, fusion and analysis. Some of these are in fact straightforward to compute; others are very difficult to precisely infer.

In the rest of this section we survey state-of-the-art in declarative modeling of the data uncertainty capturing various forms of data imperfections (e.g., impreciseness, unreliability, incompleteness, etc.). Having acquired all the information needed to describe the measurement capability of electrochemical sensors, we estimate the inherent uncertainty preserving in the raw data streams. The estimation is carried out in consecutive steps, namely, *identification* of all the potential sources of uncertainty, followed by their *quantification* and *propagation* (each one of these steps is described in detail in chapter 3).

A sensor network's data stream presents, almost by definition, complex issues related to data quality. Data is often missing, and when not missing is subject to potentially significant noise and calibration effects. For example, temperature and moisture sensors report voltages that must be converted to temperature (Celsius) and moisture (partial pressure and dew point) units. Also, because sensing relies on some form of physical coupling, the potential for faulty data is tremendous. Depending on where a fault occurs in the data reporting, observations might be subject to unacceptable noise levels (for example, due to poor coupling or analog-to-digital conversion) or transmission errors (packet corruption or loss). Applications that draw on this data, or end users hoping to perform an analysis, will need to contend with observations that involve incomplete and/or incorrect [3].

In this context, *data uncertainty* may be the result of the fundamental limitations of the underlying measurement infrastructures, the inherent ambiguity in the domain, or they may be a side-effect of the rich probabilistic modeling typically performed to extract high-level events from sensor row data. Uncertainty is a state of limited knowledge, where we do not know which of two or more alternative statements is true. Traditional approaches more or less consider uncertainty as a problem, as something to be avoided or resolved during data gathering and integration. However, uncertainty has recently been recognized as an additional source of information that could be valuable and should be preserved. *Uncertain data management* [4] presents a variety of challenges in terms of collecting, modeling, representing, querying, indexing and mining the data. It should be stressed that many of these issues are inter-related and cannot easily be addressed independently. Uncertainty can be represented using *quantitative* methods, e.g., specifying the probability that a statement is correct, or *qualitative* methods, e.g., using fuzzy sets and possibility

theory to represent preferences about the correctness of a statement. Quantitative models are the ones most frequently adopted [5], but qualitative approaches have also been explored in the literature — usually with the aim of reducing the complexity of manipulation of uncertainty. While traditional Statistical Machine Learning (SML) has provided well-founded mathematical tools for uncertainty management, such tools are not targeted at the declarative management and processing of large scale data sets. Moving away from statistical approaches, several data management works have focused on how to represent multiple alternative statements that could be true based on our limited knowledge, and this leads to the production of multiple possible integrated tuples, one for each choice.

In contrast to traditional database tuples, each *uncertain tuple* contains a set of possible alternative values representing the various different options about what is true. The reliability of a particular set of data is dependent upon the uses to which it is put. Data which are completely inappropriate in one context may be totally adequate in a different context (or vice versa). Data quality is therefore to some extend a relative concept dependent upon the context. The emphasis has therefore tended to switch away from simply trying to make the data as error free as possible to providing potential users with the information which they require to make an informed decision about the adequacy of the data for a particular purpose. This information is referred to as metadata.

Table 1: Description of data quality metadata.

Metadata	Description of measurement capability
Accuracy	The closeness of agreement between a measured value and the true value.
Precision	The closeness between independent measurements of a quantity under the same conditions.
Measurement Range	The set of values that the sensor can return as the result of an observation under the defined environmental conditions with the defined measurement properties.
Response Time	The time between a change in the value of an observed quality and a sensor 'settling' on an observed value.
Frequency	The smallest possible time between one observation and the next
Latency	The time between a request for an observation and the sensor providing a result.
Resolution	The smallest change that the sensor can detect regarding the quantity it measures.

Metadata is data about data. In this context, each data set - or uncertain tuple (Figure 1) – should be accompanied by metadata explaining not only what it contains but how and when it was collected, and details relating to its quality. The Table 1 indicates the type of information the metadata might include. In the literature there are various models for describing uncertain/incomplete/probabilistic data sets, e.g [6][7][8], but in all of them answering queries and representing their results is based on annotating uncertain tuples with information about their lineage or provenance

[9]. This provenance information captures the relationship among source and derived data along with the query operators that were involved in the derivations, and can be materialized in the repository where integrated data from various sources is stored [10]. Then, provenance can later be used to compute annotations for integrated data [11][12]–such as trust scores or probabilities–"on the fly", based e.g., on the degree of confidence any particular user has about the possible alternative values in the sources and how they were combined through query operators during the integration process.

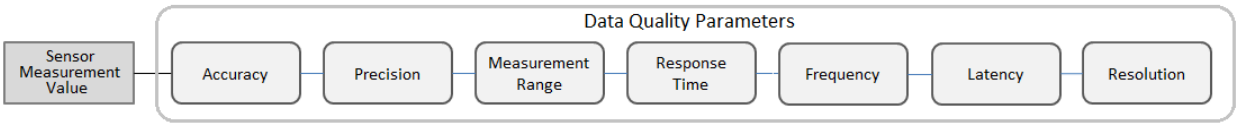


Figure 1: Data quality parameters of uncertain tuples

To conclude, in most cases the *uncertainty* is supposed to be the "umbrella" term for *accuracy* and *precision*. Essentially, it is the component of a reported value that characterizes the range of values within which the true value is asserted to lie. An uncertainty estimate should address error from all possible effects (both systematic and random) and, therefore, usually is the most appropriate means of expressing the accuracy and precision of results. The sensor *accuracy* describes the systematic measurement error resulting from static errors in the measurement process [13], due to miscalibration, retroactions of the measured method, or environmental influences on the measured values. The *precision* is a measure of how well a measurement can be made without reference to a theoretical or true value. Since precision is not based on a true value there is no bias[3] or systematic error in the value, but instead it depends only on the distribution of random errors [13]. Figure 2 depicts the target model, which correlates the precision and accuracy with uncertainty. The notions of error and bias are also shown. Using four different cases of shots at the center of the target helps to distinguish the meaning of precision and accuracy:

- Not accurate, not precise (bottom left corner): The shots are neither accurate (not close to the center) nor precise (not close to each other).
- Precise, not accurate (bottom right corner): The shots are precise (close together), but not accurate (not close to the centre of the target).
- Accurate, not precise (top left corner): The shots are scattered across the target, but the location of each of them is very close to the centre of the target. These shots are accurate, but not precise.
- Precise and accurate (top right corner): The shots are very close to the centre of the target (accurate) and very close together (precise). In this case the uncertainty is fully determined and we notice that increasing accuracy and precision, the uncertainty decreases.

[3] **Bias** is the difference between the average value of the large series of measurements and the accepted true one.

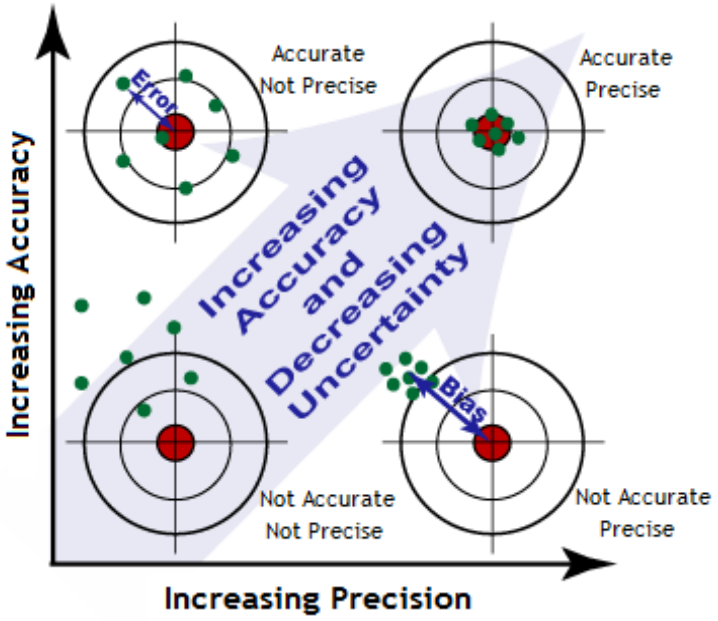


Figure 2: Target model to illustrate accuracy and precision. The centre of the target denotes the (unknown) true value

To sum up, the *accuracy* of a set of observations is the difference between the average of the measured values and the true value of the observed quantity. The *precision* of a set of measurements is a measure of the range of values found, that is the reproducibility of the measurements. The relationship of accuracy and precision is depicted in Figure 2, where the red target denotes the unknown true value. In the bottom left corner we can see that our results are imprecise and not accurate, so the uncertainty for this measurement value has a very large range. In contrast, in the right top corner case where we have high precision and accuracy, the uncertainty value range is decreased. Finally, we can see that good precision doesn't imply good accuracy (in case of right bottom corner).

2.3. ***Requirements for High-Level Data analysis***

The problem of producing data that are *unreliable*, *low-level*, and *rarely usable directly* by applications, still affects the development of sophisticated integrated sensing systems. Usually, applications do not deal with any aspect of physical device data, but rather interface with a high-level representation and reconstruction of the physical world created by a sensor infrastructure. As a result, we often witness uncertain data streams, where data may be incomplete, imprecise, and even misleading. Consequently, the final results presented to end applications are often of unknown quality, thus, impeding the task of an accurate and reliable decision making.

The major task of HYDROBIONETS project is to monitor and control the fouling phenomena developed during the different stages of the water treatment. The growth of a fouling layer due to the deposition of undesirable materials on the membrane is a persistent problem in water treatment and desalination plants. Specific types of fouling, such as, the deposition of suspended solids, colloids, and microbiological

cells, onto or into the membranes, are a severe issue which impedes the normal operation of the membranes. Complete removal of fouling mass by intensive pre-treatment of the feed water is not always feasible. Motivated by this, developing techniques for monitoring the fouling formation and providing early warning notifications for pre-defined alerts is a necessity in order to achieve the long-term and stable operation of the filtration membranes, while reducing the energy consumption and maintenance expenses. Some examples of the alerters framework of HYDROBIONETS' project include (i) the detection of high fouling concentration in seawater, (ii) the control of biocide and chlorine dosage by measuring bacteria in seawater at different stages of water treatment (pre-treatment, pre-filtered and reverse osmosis phases) and (iii) the optimization of chemical cleaning of the ultrafiltration membranes. These functionalities essentially provide the building blocks of the actuation process for water desalination at different locations in the plant. Thought appropriate alerts we enable the monitoring and notification in the HYDROBIONETS' infrastructure, when in a sensing node the values change or they are out of the ordinary.

The main contribution of this thesis is the identification of an appropriate infrastructure for the monitoring of a dynamic system. This infrastructure extracts the interrelations between pre-defined pairs of data streams, driven by their behaviour across the time. The observation of the streams behaviour contributes:

(i) In the overall monitoring of dynamic phenomena that aren't characterized by a specific distribution. This monitoring provides timely and valid actuation process in dynamic systems, as in the case of HYDROBIONETS project
(ii) To guarantee the validity of detected extreme events in uncertain data streams.

The design of this infrastructure becomes more complex when arising the following challenges due to the data analysis:

(i) The appearance of *uncertainty* in our data that if we don't take into consideration will affect the decision making in our system
(ii) The interrelation/comparison between data streams with *different scaling*. For example, in the case of comparison between a temperature and a pressure data stream, the first one is measured in the Celsius scale and the second one is measured in bars scale.
(iii) The monitoring of the *concurrent* behaviour in our data streams across the time.
(iv) The data should be processed *quickly* and at low cost due to the large amount of data we have to manage.

Ours goals in this thesis include (i) the identification of appropriate monitoring tools for the characterization of the system behaviour in real time, and (ii) the provision of the most appropriate data services to manipulate the BioMEM uncertain sensor measurements. By this way we provide timely and valid actuation for our system. Uncertainty awareness of the acquired data streams consists the basis of the proposed tools for monitoring the BioMEM sensor network and alerting in case of abnormal events. These actions along with the observation of the streams behaviour are integrated in an *uncertainty-aware data processing* infrastructure, as described in the following chapter.

3. Hydrobionets Data Processing Services

Taking important decisions is often based on the results of a prior *quantitative analysis*. Whenever decisions are based on analytical results, it is necessary to have some indication of the quality of the results. That is, the extent to which they can be relied on for the purpose of interest. *Confidence* in the obtained data is a prerequisite to meeting this objective, especially when the users of these results work in "sensitive" areas, such as those concerned with public health and hygiene. To this end, we need to monitor continuously and in an online fashion the interrelations between a number of distinct data streams produced by sensors at different stages of water treatment (*e.g.,* pre-filtering, pre-treatment and reverse osmosis), while accounting for their inherent imprecision expressed in terms of uncertainty. Although this uncertainty component may be due to hardware defections or environmental variations, its effects can be only observed and quantified from the recorded sensor measurements.

The appearance of *uncertainty* in our data streams may lead to wrong decisions concerning the source and existence of an extreme event. Timely actuation is crucial, so providing guarantees for a detected extreme event is also of high significance. Besides, the propagation of the uncertainty information through the operator queries may affect the progress of the water treatment, since we have a self-organized sensor network. In order to add an extra control in the quality of the alerters framework we should be able to extract efficiently the correlation information arising from data streams interrelations.

Rather than single stream statistics, such as average and standard deviation, data analysis is focusing on finding high *correlations* among pairs of data streams from distinct sensors. For instance, temperature and pressure sensors which monitor an industrial plant could provide evidence of an increasing bacteria presence. Depending on their physical location in the plant we expect that corresponding data streams to be highly correlated, since this pair of data streams displays analogous behavior in our case.

More generally, a desalination plant operator may rely on such stream correlation engine to reveal interrelations between seemingly independent physical quantities monitored by distinct sensors, or to guarantee the validity of a detected extreme event (e.g. high chlorine concentration in the water) and provide the necessary notifications. Moreover, in HYDROBIONETS, measurements from heterogeneous sensors, distributed over a geographic area, need to be processed efficiently in order to reconstruct the spatio-temporal behavior of desired physical variables or to detect, identify and localize sources and events of interest. Whereas traditional statistical machine learning provides well-established mathematical tools for data analysis [14] their performance is limited when processing high- dimensional data streams.

Another major functionality assigned to our uncertainty-aware data processing infrastructure is to perform high-level operations, as the notification of *extreme*

events from raw sensor data. Since the detection of abnormal behavior is affected by the underlying uncertainty, incorporation of the estimated uncertainty for the extraction of potential correlation between pairs of data streams is expected to yield more meaningful results. This thesis introduces a set of statistical techniques yielding efficient detection of rare events in complicated datasets, to be employed in the final HYDROBIONETS infrastructure.

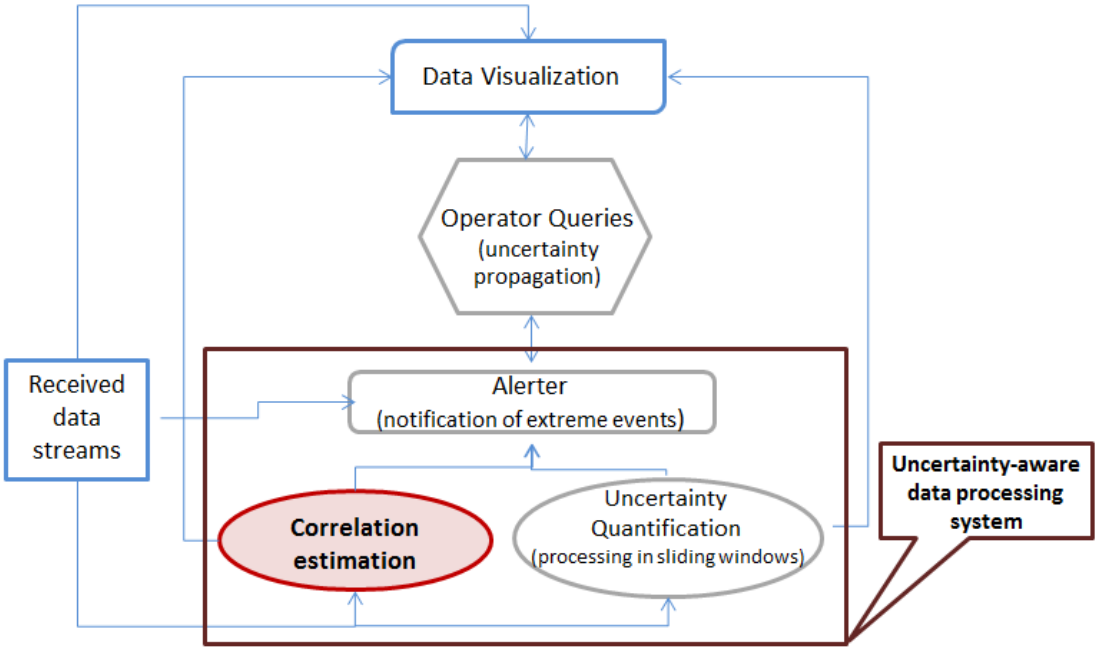


Figure 3: The uncertainty-aware data management infrastructure in HYDROBIONETS project

Figure 3 presents the overall infrastructure of data management in HYDROBIONETS' project, which combines the previous functionalities. Emphasis should be given to the uncertainty-aware data processing system. This generic structure consists of the following three building blocks:

Correlation estimation module: The *fast correlations extraction* between uncertain data streams constitutes the key component for the identification interrelations between seemingly unrelated physical quantities. The HYDROBIONETS infrastructure comprises of collaborating computational nodes, which observe and control distinct physical entities and dynamic phenomena. The existence of *correlation* among several distinct types of sensors arises naturally. Rather than single stream statistics, such as average and standard deviation, data analysis is focusing on finding high *correlations* among pairs of data streams from distinct sensors. More details about this module will be presented in chapter 4, since it constitutes the major contribution of this thesis.

***Uncertainty quantification module*:** Given that uncertainty has been recognized as an additional source of valuable information for data analysis which should be preserved, in contrast to existing data management systems, our approach incorporates an appropriate submodule to handle the inherent data uncertainty. More specifically, a *spreadsheet-based* approach is employed to identify, quantify, and

combine the underlying uncertainty from the most dominant potential sources of uncertainty.

***Alerter module*:** This module combines the received data streams along with their quantified uncertainty, the extracted correlations and the detectors for extreme events to estimate the presence of extreme events and provide the necessary notifications (queries or decision making).

The uncertainty-aware data management infrastructure of HYDROBIONETS' project is completed with the integration of the ***operator queries module.*** This module transmits the recorded information between the user and the system taking into account the uncertainty propagation, based on the rules will be described in section 3.2. The combination of the above modules enables higher-level analysis, which forms the basis for the development of an integrated uncertainty-aware data management system for monitoring dynamic sensor networks and alerting in case of abnormal events.

This thesis is the result of our work to select and implement appropriate analytical techniques for the HYDROBIONETS project, concerning

(i) on the fly *monitoring* and *extraction* of pairwise correlations between high-dimensional sensor data streams
(ii) *modeling*, *management* and *propagation* of uncertainty in the generated raw data streams, and
(iii) designing appropriate alerting tools notifying for *extreme events*.

The scope of this chapter is to describe the services which are selected and used to compose the modules of the uncertainty-aware data management infrastructure of HYDROBIONETS project. To be more specific, in section 3.1 different approaches are presented for modeling the uncertainty in data streams. The one of them has emerged for the HYDROBIONETS project needs as described in [31]. The section 3.2 describes the rules for uncertainty propagation and finally, the section 3.3 is referring in two widely used techniques for extreme events detection by incorporating the underlying estimated data uncertainty.

3.1. *Uncertainty modeling in data streams*

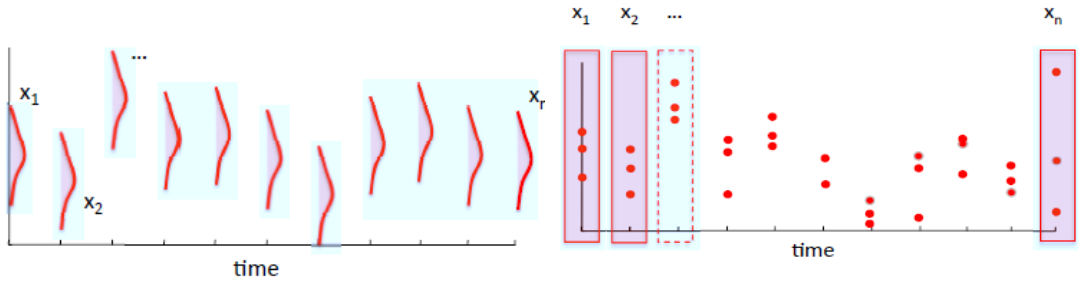
The definition of *measurement uncertainty* is as follows: *"A parameter associated with the result of a measurement that characterizes the dispersion of the values that could reasonably be attributed to the measurand*[4]*"*, where a parameter can be, for instance, a standard deviation, or the width of a confidence interval. In general, measurement uncertainty consists of several distinct components. Some of these components may be evaluated directly from the available information from each sensor of the recorded measurements (see Table 1), while the rest of the components can be evaluated based on an empirical assumption for the probability distributions according to our experience or some other prior information.

[4] *Measurand* refers to a clear and unambiguous statement of what is being measured, along with a quantitative formulation relating the value of the measurand to the parameters on which it depends.

3.1.1. Existing approaches

While the problem of managing and processing uncertain data has been studied in the traditional database literature since the 80's [32], the attention of researchers was only recently focused on the specific case of uncertain time series. Two main approaches have emerged for modeling uncertain time series and both of them are based on this general definition: An uncertain time series X is defined as a sequence of random variables $\langle x_1, x_2, ..., x_n \rangle$ where x_i is the random variable modeling the real valued number at timestamp i.

In the first approach [24], an uncertain time series is modeled by a streaming time series of random variables, where each random variable represents the uncertainty of the value in the corresponding timestamp (Figure 4[a]). The probability density function (pdf) over the uncertain values is estimated by using some a priori knowledge of the general characteristics of the data distribution, namely its means and variance. In [26], uncertainty is modeled by means of repeated observations at each timestamp, as depicted in Figure 4[b](in each orthogonal shape there are the corresponding repeated observations for each observation).



(a) Uncertainty modeling by means of pdf estimation

(b) Uncertainty modeling by means of repeated observations

Figure 4: Examples of modelling the uncertainty in time series *X={x1,...xn}*

These two techniques are based on the assumption that the values of the time series are independent from one another. That is, the value at each timestamp is assumed to be independently drawn from a given distribution. Evidently, this is a simplifying assumption, since neighboring values in time series usually have a strong temporal correlation. The main difference between [26] and [24] is that the first represents the uncertainty of the time series values by recording multiple observations for each timestamp. This can be considered as sampling from the distribution of the value errors. In contrast, [24] consider each value of time series to be a continuous random variable following a certain probability distribution. The amount of preliminary information, i.e. a priori knowledge of the characteristics of the time series values and their errors, varies greatly among the techniques. The approach of [26] does not need to know the distribution of the time series values, or the distribution of the value errors. It simply operates on the observations available at each timestamp. On the other hand, [24] needs to know the distribution of the error at each value of the data stream. In particular, this technique requires knowing the standard deviation of the uncertainty error, and a single observed value for each timestamp. Also, it

assumes that the standard deviation of the uncertainty error remains constant across all timestamps.

These approaches can't be used by HYDROBIONETS project infrastructure, since the knowledge of the distribution is limited due to the nature of our data streams. We assume that uncertainty is an additional source of information that is valuable and should be preserved. So the *quantification* of the inherent uncertainty plays a fundamental role on the certification in high-consequence decisions.

3.1.2. Uncertainty quantification

Having acquired the raw sensor data from the distinct electrochemical sensors distributed across the plant, our proposed infrastructure (Figure 3) estimates their corresponding inherent uncertainty. The estimation is carried out in two consecutive steps, namely, *identification* of all the potential sources of uncertainty, followed by their *quantification* and *propagation*. In the following, each one of the *identification sources* and *quantification* steps is described, as in [31]. The rules of propagation step are described in section 3.2.

Step 1: Identification of uncertainty sources

Identification of uncertainty sources comprises the first step towards the design of our integrated uncertainty-aware data management system. In practice, the underlying uncertainty may arise due to several distinct sources, such as, an incomplete definition of the observed quantities, sampling effects and interferences, varying environmental conditions, and inherent uncertainties of the equipment.

A very convenient way to determine the most dominant uncertainty sources, along with their potential interdependencies, is to exploit the so-called *cause and effect (or Ishikawa) diagram*. This diagram also ensures comprehensive coverage, while helping to avoid double counting of sources. Once the set of most significant uncertainty sources is formed, their effects can be usually represented in terms of a measurement model.

As a typical example, Figure 5 shows a cause and effect diagram for a temperature sensor. The first source of uncertainty is the sensor's functionality by itself. However, its performance is affected by several distinct factors, such as, its sensitivity and precision, the calibration, the operating temperature, and the water flow-rate and pressure. On the other hand, the accuracy of the recorded values depends also on the sensors' deployment density and location, as well as on the sampling process we use. Possible misplacement or a very sparse time-sampling is expected to increase the uncertainty, especially when the monitored variable varies rapidly across time.

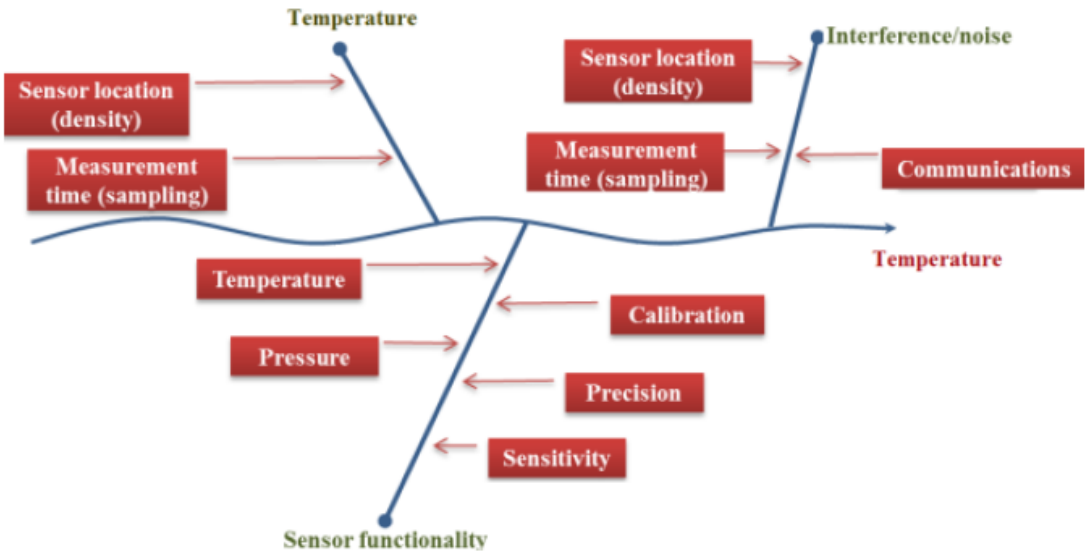


Figure 5: Cause and effect diagram for a temperature sensor

Step 2: Quantification of uncertainty

The identification of uncertainty sources is followed by a quantification process. This is done by estimating the uncertainty of each individual source and then combining them appropriately to obtain a single overall uncertainty. The underlying data uncertainty in a given data stream is distinguished into two separate categories, *type A* (aleatoric, or statistical) and *type B* (epistemic, or systematic) uncertainty:

Uncertainties of *type A* are characterized by the estimated variances σ_i^2 (or the standard deviations σ_i), which are obtained by statistical analysis of the observations in the raw data streams. Following the *sliding window* approach, as it was described in a previous section, the variance σ_i^2 of the *i*-th sensor is estimated from its measurements in the current window. This is equivalent to obtaining a standard uncertainty from a probability density function (pdf) derived from an observed frequency (empirical) distribution. Let $\boldsymbol{y}$ be a data stream with *N* values {*y1, ..., yN*}, which corresponds to a specific observed variable. Then, the standard uncertainty of $\boldsymbol{y}$, which is denoted by $u(\boldsymbol{y})$, is expressed in terms of the corresponding standard deviation σ_y, estimated directly from the observations y_i as follows,

$$u(y) = \frac{\sigma_y}{\sqrt{N}} \tag{12}$$

For uncertainties of type B, the estimated "variance" s_j^2 is obtained from an assumed probability density function based on prior knowledge for the corresponding source of uncertainty, which may include:

(i) data from previous measurements,
(ii) experience or knowledge of the properties of instrumentation and materials used,
(iii) manufacturer's specifications, and
(iv) data calibration.

In general, concerning type B uncertainties, the quantification is performed either by means of an external information source, or from an assumed distribution. Typical assumptions for the prior distributions include the Gaussian (*e.g.,* when an estimate is made from repeated observations of a randomly varying process, or when the uncertainty is given as a standard deviation or a confidence interval), the uniform (*e.g.,* when a manufacturer's specification, or some other certificate, give limits without specifying a confidence level and without any further knowledge of the distribution's shape), and the triangular distribution (*e.g.,* when the measured values are more likely to be close to a value *a* than near the bounds of an interval with mean equal to *a*).

Having estimated the individual uncertainties, expressed as standard uncertainties, the next step is to combine them in the form of a *combined standard uncertainty*. Although in practice there may exist correlations between the individual uncertainty sources, however, it is usually impossible to compute those correlations accurately. For this purpose, it is more convenient to rely on an assumption of independence between the individual uncertainty sources.

In the following, let y denote the observed variable associated with the acquired data stream $\boldsymbol{y}$. Furthermore, let $y = f\{x_1,...,x_L\}$ be an observed variable, which depends on L input variables x_l through a functional relation $f(\cdot)$. Then, the *combined standard uncertainty* of y, for independent input variables x_l, $l = 1, ..., L$, is given by:

$$u_c(y) = \sqrt{\sum_{l=1}^{L}\left(\frac{\partial f}{\partial x_l}\right)^2 u^2(x_l)} \tag{13}$$

where $u(x_l)$ denotes the standard uncertainty of the input variable x_l (either of type A, or of type B), while the partial derivatives $\partial f / \partial x_l$, the so-called *sensitivity coefficients*, quantify how much the output y varies with changes in the values of the input variables x_l $l=1, ... L$. Finally, the combined standard uncertainty, which may be thought of as equivalent to one standard deviation, is transformed into an *overall expanded uncertainty*, *U,* via multiplication with a coverage factor *k,* that is,

$$U(y) = k \cdot u_c(y) \tag{14}$$

where the value of k is determined in terms of the desired confidence level as shown in Table 2.

Table 2: Coverage factor as a function of confidence level for the Gaussian distribution

Coverage factor (k)	Confidence level (%)
k=1	67%
k=1.96	95%
k=2.576	99%
k=3	99.7%

The most convenient way to summarize all this information and compute the overall uncertainty is by means of *spreadsheet tables*. A spreadsheet table lists the dominant

sources of uncertainty and categorizes them according to their type. Based on that, the individual standard uncertainties are stated explicitly, along with the overall combined uncertainty. An example of such a table for a temperature sensor is shown in Table 3.

Table 3: Example of a spreadsheet table for a temperature sensor.

Source of uncertainty			**Value** (±)	**Probability distribution**	**Divisor**	**Standard uncertainty** $u(x)$
Type B	Sensor	Calibration	C_1	Normal	2	$C_1 / 2$
		Precision (Resolution)	C_2	Rectangular	$\sqrt{3}$	$C_2 / \sqrt{3}$
		Sensitivity	C_3	Rectangular	$\sqrt{3}$	$C_3 / \sqrt{3}$
	Sensor density		C_4	Rectangular	$\sqrt{3}$	$C_4 / \sqrt{3}$
	Sampling		C_5	Rectangular	$\sqrt{3}$	$C_5 / \sqrt{3}$
Type A	Temperature		C_T	-		σ_T
	Pressure		C_P	-		σ_P
Combined standard uncertainty $u_{c,b}(y)$						
Coverage factor k_b						
Expanded uncertainty U_b						

The final output of the above spreadsheet-based approach is the assignment of the combined and expanded uncertainty values to the current windows of all the sensors. This completes the first building block of our uncertainty-aware data processing system as presented in [31]. In the following section we describe the uncertainty propagation building block, namely, the rules for further query processing by accounting the estimated uncertainties.

3.2. *Uncertainty propagation in derived data streams*

In the previous section, we referred to the quantification of uncertainty of individual components for the HYDROBIONETS sensors, as described in [31]. The next step is to apply appropriate rules for propagating the estimated uncertainties upon a specific query operation, since the recorded sensor data streams are exploited to support and optimize production automation processes, as well as complex application decisions.

After asserting the uncertainty of raw data streams obtained from HYDROBIONETS sensors, they go through various operators to produce final results. Since sensors allow for the automatic collection of a huge volume of data, the additional propagation of data uncertainty results in an overhead for data transfer and management, which may shape up as very expensive. Furthermore, if data uncertainty information is lost, the executed data processing steps have to be mirrored in a data quality processing framework.

To extract the complex knowledge that we need, sensor data is merged, transformed, and aggregated by applying traditional data stream queries, complex signal analysis, or elementary numerical operators. During the data stream processing, the initial sensor-inherent errors are amplified. Additionally, new errors may be introduced. Finally, if the sensor data are incorrect or misleading, derived decisions are likely flawed. Hence, it is also important to capture uncertainty of such processing results.

So, the quantification of uncertainty in our data has to be processed with the right way from the query operators to avoid invalid decision making. With the rules that introduced in the following, we extend the existing operators, selection-aggregation-join, to take into account the uncertainty of data. Mathematical functions are introduced to compute the effects of different operators on the uncertainty components *accuracy* and *precision*. A sensor measurement has imperfections that give rise to an *error* in the measurement result, as it was mentioned in 2.2.1 Moreover, we also have to consider carefully the uncertainty issues, which are introduced by each operator separately. Identifying such issues has as ultimate goal to minimize the false positive and false negative cases that may arise from the adjustment of the operators on uncertain data streams.

***Selection*:** During *selection*, data items are extracted for further processing based on the constraint evaluation of a certain measurement attribute. Tuples that do not satisfy the selection criterion are discarded from the data stream. In [33], a threshold control is introduced as the first step of the condition evaluation in selection. The incoming data stream is evaluated against a given threshold, resulting either in the boolean *true* if the threshold holds or *false* for exceeding a threshold. The accuracy and precision data quality parameters of a measurement value (α, ε), as well as a user defined threshold (a_b) define a new uncertainty range $\delta = a_b + a + \varepsilon$. In the context of selection, this approach reveals the following shortcomings for data items lying in the uncertain range δ:

(i) Sensor measurements in the uncertain range are selected, even though the true value may not exceed the threshold constraint.
(ii) Data items are not selected, although the selection condition may be met by the true value.

The false positives and false negatives may balance if there is a uniform data distribution in the uncertain range. The false selection leads to erroneous results if aggregation operators are applied during further data processing. The aggregated value is either too high because too many data items have been selected, or to low because relevant data items are missing.

Definition 1 (Windowed uncertain selection): *Assuming that a selection condition F will be applied to each state of a window W over stream S, the selection operator for uncertain data streams can be defined as*

$$\sigma_F^W\big(S(\tau)\big) = \sigma_F\Big(W\big(S(\tau)\big)\Big) = \Big\{s \in W\big(S(\tau)\big) : F(s)\ holds \wedge \Pr\Big\{dist\Big(avg\Big(W\big(S(\tau)\big)\Big), s\Big) \leq \varepsilon\Big\} \geq \alpha\Big\}$$

where $dist(.,.)$ *is the Euclidean distance function between two objects, avg(..) is the average of all values in the current sliding window* $W(S(\tau))$ *and* ε, a *are the precision and accuracy data quality parameters.*

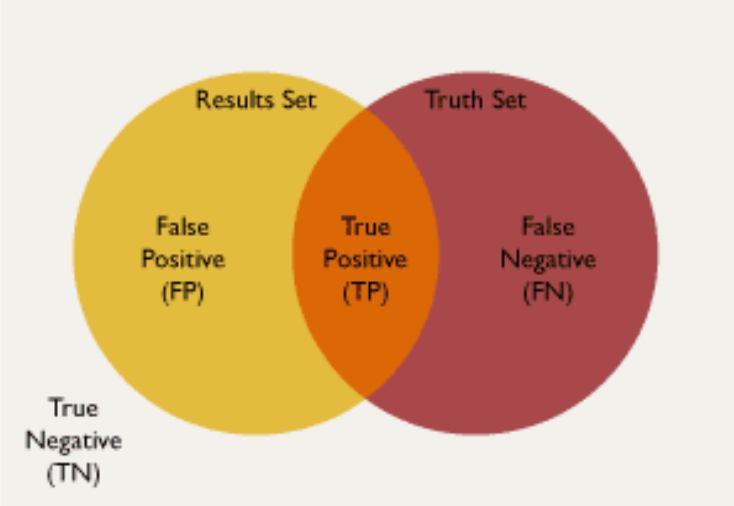


Figure 6: Definition of true/false positives and true/false negatives.

According to the previous definition of selection operation on uncertain data streams, for every object *s* within sliding windows, the distance between *s* and the average of all values in the current sliding window is computed. If the probability of that distance being less or equal to the threshold ε, is greater than threshold a and the condition *F* holds for object *s,* then *s* is reported to the selection answer. Note that with the constraint $\Pr\left\{dist\left(avg\left(W\left(S(\tau)\right)\right), s\right) \leq \varepsilon\right\} \geq \alpha$ the false positive results are discarded.

***Aggregation*:** During aggregation, each group of data items is summarized to compute a single data result, the *aggregate*. The aggregation operators compress the incoming data to one output value or create a synopsis consisting of several data items. This data value represents not only a certain point in time but a whole time interval. The time-stamp has to be adjusted to the form $[t_b, t_e]$ to represent this fact. The time-frame defining the grouping for an aggregation operator is independent from the window size *w* for data quality calculation. An aggregation operator takes *N* tuples modeled as *N* random variables, and performs an operation such as *sum* or *min*/*max* on these variables. The data uncertainty of one *aggregate* is calculated based on all incoming tuples' uncertainty information.

The result of the uncertainty of an aggregation will not be measured directly. For instance, what is the error in Z = A + B, where A and B are two measurements with errors ΔA and ΔB respectively? A first thought might be that the error in *Z* would be just the sum of the errors in *A* and *B*, that is, $(A + \Delta A) + (B + \Delta B) = (A + B) + (\Delta A + \Delta B)$. However, this assumes that, when combined, the errors in *A* and *B* have the same sign and maximum magnitude, that is, they always combine in the worst possible way. This could only happen if the errors in the two variables were perfectly correlated. We establish that the correlation structure among these variables determines appropriate techniques to compute uncertainty in aggregation results [36].

Table 4 introduces some simple rules for expressing the uncertainty in aggregation results.

If a variable Z depends on one or two variables (A and B) which have independent errors (ΔA and ΔB) then the rules for calculating the error in Z is tabulated in the following table for a variety of simple relationships[5]. These rules may be compounded for more complicated situations.

Table 4: Rules for calculating the errors in aggregation results

Relation between Z and (A,B)	Relation between errors ΔZ and (ΔA,ΔB)
Z = A + B	$\Delta Z^2 = \Delta A^2 + \Delta B^2$
Z = A - B	$\Delta Z^2 = \Delta A^2 - \Delta B^2$
Z = AB	$\left(\frac{\Delta Z}{Z}\right)^2 = \left(\frac{\Delta A}{A}\right)^2 + \left(\frac{\Delta B}{B}\right)^2$
Z = A/B	$\left(\frac{\Delta Z}{Z}\right)^2 = \left(\frac{\Delta A}{A}\right)^2 + \left(\frac{\Delta B}{B}\right)^2$
$Z = A^n$	$\frac{\Delta Z}{Z} = n\frac{\Delta A}{A}$

The support of (conditioning) aggregation operations on data streams involving continuous-valued uncertain attributes includes one more difficulty. Even if the input stream contains continuous-valued uncertain attributes, which are modeled by continuous random variables, conditioning operations (*e.g.,* filters and group) can introduce uncertainty about tuple existence, which needs to be modeled by discrete random variables. Hence, for complex queries involving conditioning and aggregation, the distributions for both continuous and discrete random variables must be computed, which is a hard problem [34].

Definition 2 (Windowed uncertain aggregation): *For each combination of values that belongs to* $W(S(\tau))^{W(S(\tau))}$*, an aggregation function f (such as SUM,MIN,MAX or AVG) is applied. The aggregation output is one stream of tuples of the form* $\langle f(s_i, s_j, \dots, s_n), \tau, e\rangle^{\langle f(s_i, s_j, \dots, s_n), \tau, e\rangle}$ *for each sliding window. "τ" is the smallest timestamp of the objects, while* $f(s_i, s_j, \dots, s_n)^{f(s_i, s_j, \dots, s_n)}$ *is the final aggregate value if it is smaller than a (user-specified) threshold ε, and e is the error in the aggregation result according to the rules of* Table 4. *More formally,*

$$\gamma_L^{f^W}(S,\tau) = \gamma_L^f\left(W(S(\tau))\right) = \left\{\begin{matrix}\langle f(s_i, s_j, \dots, s_n), \tau_m, e\rangle : \forall\, s \in W(S(\tau)) \wedge \\ f(a_i, a_j, \dots, a_n) \le \varepsilon \wedge \tau_m = \min \tau\end{matrix}\right\}$$

***Join*:** This symmetric binary operator may be applied between two streams. There is no restriction that windows of the same type or the same scope must be specified over each stream. Each newly arriving tuple within window W_1 of stream S_1 is checked for possible matches against the current state of window W_2 of stream S_2 and vice versa. Matching is performed according to the join condition J involving attributes from both streams (*e.g.,* $S_1.A_i = S_2.A_j$). If matching tuples are found, the

[5] http://teacher.nsrl.rochester.edu/phy_labs/AppendixB/AppendixB.html

resulting joined element must be assigned a new timestamp value. Joining is an important operation in queries that target data streams that have no navigable relationships to each other. A join of two data sources is the association of objects in one data source with objects that share a common attribute in the other data streams. Join operator can be used to represent or to detect complex events in sensor networks and it is the fundamental operation for relating information from different streams.

The problem of join processing is challenging in the context of uncertain data because the join-attribute is probabilistic in nature. Therefore, the join operation needs to be redefined in the context of probabilistic data. An important aspect of join operation is that the uncertainty model significantly affects the nature of join processing. The evaluation strategies of joins vary significantly with the nature of the join attributes. Recent research on probabilistic databases has mostly focused on join attributes modeled by discrete random variables. Another consideration supports joins based on the possible worlds' semantics. In every possible world, each random variable takes a specific value, thus a join can proceed just as in a traditional database. However, when data uncertainty is captured using continuous random variables, join methods based on possible world semantics hardly work because the possible values of a continuous random variable cannot be enumerated (the number of such possible values is infinite and each possible value has probability 0).

Next, we formally define the problem of join on uncertain data streams, which consists of three distinct components. Initially, we focus on the uncertainty impact of the time-stamp-based join of synchronous and asynchronous data streams and illustrate the handling of jumping windows during the window-wise data stream join execution.

Join of synchronous streams: The simplest join approach assumes synchronous sensor data streams, and builds one-to-one tuple pairs based on identical time-stamps, as shown in Figure 7. Equal data stream rates do not suffice for this approach. The sensor data could be measured shifted against each other, so that no identical time-stamps exist. During the join of two data streams D_1 and D_2, data uncertainties U_A and U_B are not affected but copied to the resulting data stream. This results in a memory limitation. To keep all this information requires an increased memory space, which, in turn, affects system's performance.

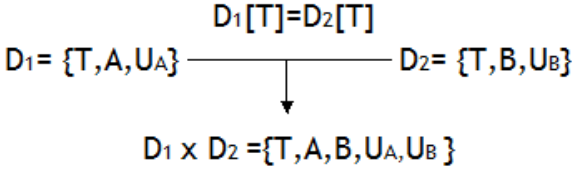


Figure 7: Elementary join of equal time-stamps

Join of asynchronous streams: The assumption of synchronous data streams does not hold for typical application scenarios. In [35], sampling and interpolation techniques are used to adapt the stream rates and overcome phase shifts in the data streams. Then, the complex operator has to be split up as shown in Figure 8 to allow the tracking of the uncertainty impact. The data streams D_1 and D_2 are sampled and/or interpolated to be joined afterwards using the time-stamp-based, synchronous join approach.

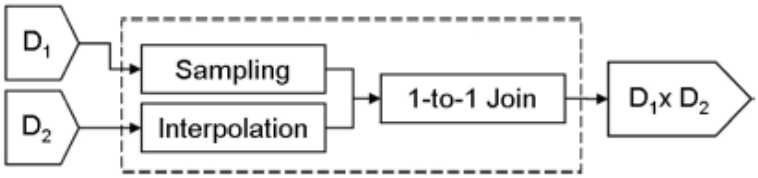


Figure 8: Joining asynchronous data streams.

Window join of data streams: Conceptually, a join operator must ensure that every tuple in one of its inputs is compared with every tuple in the other. When these input sets are unbounded, as is the case for infinite streams and continuous queries, we have the problem that the comparison of two infinite inputs would require infinite storage. The window-wise data stream join is recommended to comply with restricted memory and CPU resource constraints in data stream environments. In [33], the jumping windows are introduced to reduce the data overhead produced by the uncertainty transfer. The uncertainty information is propagated not for every single measurement value, but rather aggregated over a certain period of time.

While a sliding window join of two data streams is executed not all streaming tuples find join partners, independent from the specific join implementation. Thus, the window-wise join of data streams includes an implicit sampling on one or both affected data streams. To track the influence of this sampling on the data uncertainty component *precision*, the implicit sampling rate has to be recorded for each jumping window, while it overlaps with the sliding join window (Figure 9 a). As soon as the sliding join has left the jumping window (Figure 9 b), the precision can be updated, and the data quality can be propagated to the next operator in the processed query.

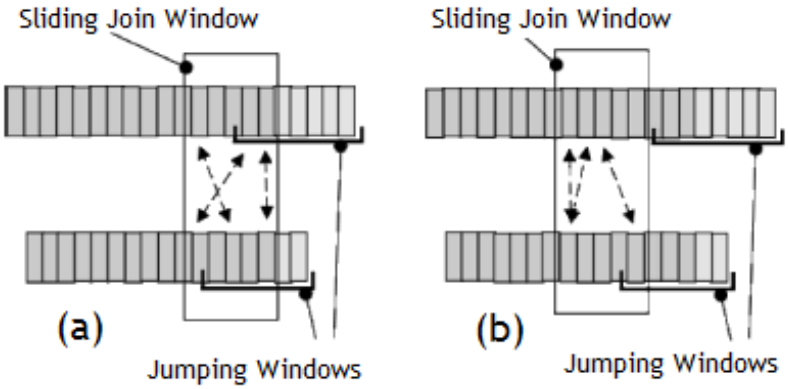


Figure 9: Sliding window join

Definition 3 (Windowed uncertain join): *At each timestamp* $\tau \in \mathrm{T}$*, the windowed uncertain join between two streams returns the concatenation of pairs of matching tuples taken from either window state. In particular:*

$$\begin{aligned} S_1(\tau) \bowtie_W S_2(\tau) = {} & W_1\big(S_1(\tau)\big) \bowtie W_1\big(S_1(\tau)\big) \\ = {} & \{\langle s_1, s_2, \tau_m\rangle : s_1 \in W_1\big(S_1(\tau)\big), s_2 \\ & \in W_2\big(S_1(\tau)\big) \wedge J(s_1, s_2) \wedge Pr\{dist(s_1, s_2) \leq \varepsilon\} \\ & \geq a \wedge \tau_m \, min \, \tau\} \end{aligned}$$

Given two uncertain data streams S_1 and S_2, a distance threshold ε, and a probabilistic threshold $a^{\epsilon} \in (0,1]$, the join on uncertain data streams continuously monitors pairs of uncertain objects s_1 and s_2. Each tuple within window W_1 of stream S_1 is checked for possible matches against the current state of window W_2 of stream S_2 and vice versa. Matching is performed according to the join condition J involving attributes from both streams.

According to the previous definition of join operation on uncertain data streams, for every object pair (s_1, s_2) within sliding windows $W_1(S_1(\tau))$ and $W_2(S_2(\tau))$ respectively, the joining probability of s_1 being within ε distance from s_2 is computed. If the resulting probability is greater or equal to a probabilistic threshold α, then this pair (s_1, s_2) is reported as the join answer, otherwise, it is a false alarm and can be safely discarded (users need to register two parameters, distance threshold ε and probabilistic threshold α) [37].

In this section, we presented an efficient way to propagate uncertainty in data streams. For a comprehensive evaluation of sensor measurements, we defined the uncertainty in the context of streaming data and proposed two uncertainty components: accuracy and precision. Operators retrieved from traditional data stream querying and the signal processing domain is applied to extract complex knowledge from raw data streams. We analyzed these operators to track the problems that uncertainty causes during propagating this in a raw data stream. Moreover, techniques and metrics are gathered and presented to calculate uncertainty in the output of operators' selection, aggregation and join. In the following section we describe the building block for the detection of extreme events. We present two modified extreme event detectors in order to account for the inherent data uncertainty.

3.3. *Uncertainty-aware detection of extreme events*

Concerning the design of mechanisms notifying for extreme events, the estimated uncertainty, in conjunction with appropriate assumptions for the prior probabilistic models, can be exploited in a statistical framework for the detection of extreme values. Extreme value theory allows, under specific conditions, to predict *rare events*, which diverge from a "normal" pattern because of their rareness. For instance, in the HYDROBIONETS framework, a typical extreme event is the detection of high chlorine concentration in the water, or a high concentration of biofilms on the desalination membranes. As mentioned before, early warning for abnormal behavior is crucial when working in large-scale industrial environments.
In our developed uncertainty aware data processing system, the identification of critical events is performed by means of two robust and computationally efficient methods. More specifically, we enhance the performance of two widely used techniques for extreme events detection by incorporating the underlying estimated data uncertainty. The first one, namely, the *compliance with operating limits*, performs simple comparisons of predetermined user-specified operating limits with the recorded measurements augmented by their estimated uncertainty. This modification maintains the computational efficiency of the original version, while improving its adaptivity to imprecise measurements. In a similar way, the second

approach, the so-called *Peaks-Over-Threshold* (POT) method, is modified accordingly so as to identify the time instants when the measurements (also augmented by the estimated combined or expanded uncertainty) exceed an estimated threshold.

Compliance with operating limits

The simplest way to exploit the estimated combined or expanded uncertainty to design an alerting mechanism is as shown in Figure 10. More specifically, let l_u denote an upper operating limit dictated by a manufacturer or a specification standard. Although, for convenience, we restrict ourselves in the case of an upper limit, however, the same remarks are straightforward when compliance with a lower operating limit l_m is required.

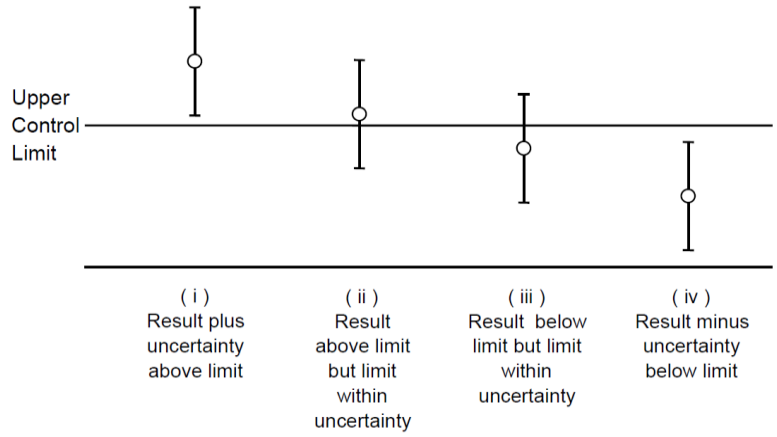


Figure 10: Compliance conditions for a measurement result.

As shown in Figure 10, there are four possible cases for a measurement and its associated expanded uncertainty interval, $y \pm U_b$ when compared with an upper limit l_u namely,

(i) both the measurement and the expanded uncertainty interval are above the upper limit *lu*,
(ii) the measurement is larger than *lu* and the expanded uncertainty interval contains *lu*,
(iii) the measurement is lower than *lu* and the expanded uncertainty interval contains *lu*, and
(iv) both the measurement and the expanded uncertainty interval are below *lu*.

Case (i) clearly triggers an alerting notification for the occurrence of an extreme event, while (iv) is the only one which is in compliance with the specifications. On the other hand, in cases (ii) and (iii) we could not infer with absolute certainty whether an alert should appear or not. However, in a socially "sensitive" application, such as the water treatment, a system operator should classify cases (ii) and (iii) as possible divergences from normal operation, and thus draw more attention on the associated monitored variables. Notice also that, in contrast to the original version of

this method, which supports only two cases (above or below *lu*), the modified one exploits two additional ones due to the presence of uncertainty.

Despite its simplicity, the main drawback of this method is that it can be very sensitive to an under- or over-estimate of the expanded uncertainty, as well as of the measurement value, increasing the probability of false alerts. However, with appropriate setup of the hardware (sensors) and continuous monitoring of the environmental conditions, we could increase our trust to this method.

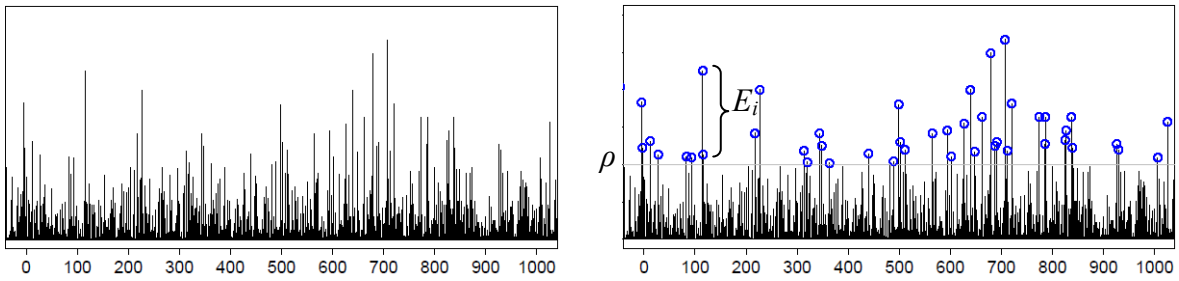
Peaks-over-threshold

Similarly to the previous method, we also extend the original POT method [38][39] in order to account for the underlying data uncertainty. More specifically, we consider $\tilde{y} = \{y_1,...,y_N\}$ to be a data stream with N measurement values, which are spread out by the corresponding estimated expanded uncertainty, that is, $\tilde{y}_i = y_i \pm U$ where we assume for the cumulative distribution function (CDF) F, that for $z > 0$, $F(z) = \Pr(\tilde{y} \le z) < 1$. Given a user-defined threshold ρ we study the statistical properties of the exceedances $\tilde{y}_i$ of over the threshold level ρ by fitting them with an appropriate distribution. In the following, we mainly rely on a threshold-dependent complementary CDF (or, equivalently, exceedance probability), which is given by

$$\bar{F}_p(z) = \Pr(\tilde{y} - \rho > z \mid \tilde{y} > \rho) = \frac{\bar{F}(\rho + z)}{\bar{F}(\rho)} \tag{15}$$

where $\bar{F}(z) = 1 - F(z)$, for $z \ge 0$, denotes the tail of F.

The above identities can now be used to estimate tails and quantiles, to be used as refinements of the threshold ρ, adapted to the measurements statistics, within a predetermined level of confidence. To this end, let $N\rho$ be the subset of indices $j \in \{1,...N\}$ for which $\tilde{y}_j > \rho$, that is $N_u = \{j \in \{1,...N\} : \tilde{y}_j > \rho\}$.



(a) Original time-series (b) Peaks over threshold

Figure 11: Original time-series and its peaks over threshold

Then we denote by $E_1,....E_{N_\rho}$ the excesses of $\tilde{y}_1,\tilde{y}_2,...\tilde{y}_N$, that is, the heights of the exceedances over ρ, as shown in figure 18b. $\bar{F}(\rho)$ is estimated simply as the

relative frequency, $\overline{F}(\rho) \approx \frac{N_\rho}{N}$, while $\overline{F}_p(z)$ is approximated by the generalized Pareto (GP) distribution as follows

$$\overline{F}_\rho(z) \approx \left(1+\gamma\frac{z}{\sigma(\rho)}\right)^{-\frac{1}{\gamma}} \quad , \; z \geq 0 \tag{16}$$

where the parameters γ, $\sigma(\rho)$ can be obtained via maximum likelihood (ML) estimation from the acquired sensor measurements directly. By combining (15)-(16) we obtain the overall tail estimator as follows,

$$\overline{F}(\rho+z) \approx \frac{N_\rho}{N}\left(1+\gamma\frac{z}{\sigma(\rho)}\right)^{-\frac{1}{\gamma}} \quad , \; z \geq 0 \tag{17}$$

Finally, for a given $p \in (0,1)$ we obtain an estimator for the p-th quantile, z_p, as follows,

$$\hat{z}_p = \rho + \frac{\sigma(\rho)}{\gamma}\left(\left(\frac{N}{N_\rho}(1-p)\right)^{-\gamma} - 1\right) \tag{18}$$

This quantile can be further employed as a refinement of the initial threshold ρ in subsequent time windows of the data streams. Another benefit of using a probabilistic framework, as is the case of POT, instead of the simple compliance with operating limits, is that we can also estimate the average time interval between successive extreme events of similar intensity. This elapsed time is called *return period*, and is defined as the inverse of the exceedance probability as follows

$$T_R = \frac{1}{\overline{F}_p(z)} \tag{19}$$

We notice here that this does not mean that if an extreme event with a return period *TR* occurs, then the next will occur in about *TR* time units (*e.g.,* days, months, years). Instead, it means that, in any given time unit, there is a 1/*TR* chance that it will happen, regardless of when the last similar event was.

Overall, we observe that the two techniques make different initial assumptions about the amount of information available for the uncertain time series, and have different input requirements. The POT method allows us to work with larger sample populations, which ensures better fits to a distribution function. However, this comes at the cost of assuming that the data are considered to be identically distributed, which may not be always the case in practice. This block-based method best suits to block-structured data (*e.g.,* yearly, monthly, weekly). On the other hand, the first method may be very sensitive to an under- or over-estimate of the expanded uncertainty, as well as of the measurement result, yielding to false alerts. Consequently, when deciding which technique to use, users should take into account the information available on the uncertainty of the time series to be processed.

4. Data Streams Correlation Frameworks

The processing, management and mining of data streams have attracted on increasing amount of interest recently. Data streams appear in a variety of settings, such as environmental and medical systems. Typical data stream applications include sensor monitoring and sensor data analysis. In all these situations, the data sources generate data with no end in sight, making it impossible to store all the historical data. The best approach is the data processing to be performed in an on-line fashion, to avoid the complete data storage and to "catch" abnormal behaviour of the applications. There are many fascinating research problems in such settings, like clustering [15], summarization [16] and forecasting [17][18]. The correlation analysis is a way of measuring the linear relationship between dimensional data streams. Here we focus on a less-studied problem, namely on computing correlations on *uncertain data streams*. Our goal is to monitor k numerical uncertain sequences, $X_1,\ldots,X_k$ and to determine automatically all the pairs of sequences that have a correlation above a specific threshold. That is we want to report all the pairs of streams X_i and X_j, for which stream X_i follows the stream X_j above a specific correlation threshold *t*.

The proposed data processing subsystem aims to support the HYDROBIONETS wireless sensor network infrastructure for multi-sensing and multi-actuation in water treatment and desalination plants. In our case, a desalination pilot plant is located in La Tordera, which is equipped with a number of various electrochemical sensors, scattered in distinct locations, for monitoring several physical and mechanical variables in the plant. In order to perform timely actuation and provide guarantees for the validity of a detected extreme event, we need to monitor continuously and online the correlations between predetermined pairs of data streams produced by sensors at different stages of the water treatment (pre-filtered, pre-treatment and reverse osmosis phases), as well as their inherent uncertainty. So, the problem we address in this thesis is defined as follows:

Problem: "*Given m co-evolving uncertain data streams of equal length n, detect at any point of time the occurrence of an extreme event, along with the top-k pairs of streams which are highly correlated.*"

Without loss of generality, we can assume that two data streams X and Y have the same length *n.* Intuitively, two data streams are highly correlated if they look very similar as involving in time. Figure 12 shows five data streams evolving over time. We can notice that the blue-red and the red-green data streams present "similar" behaviour across the time, so we consider that they are highly correlated. Besides, in the case of the red-green streams, after a time-point there is no correlation. If the data streams X and Y were static, the problem would be trivial: simply compute the Pearson's cross-correlation function. However, when X and Y consist of uncertain measurements, observe different dynamic data with different distributions, have big data volume and continuously increase in length, the problem is challenging. In this chapter we will develop the technique for fast computing correlations above a specific threshold in simple data streams and in section 4.3 we will propose the way to combine this technique with *uncertain* data streams.

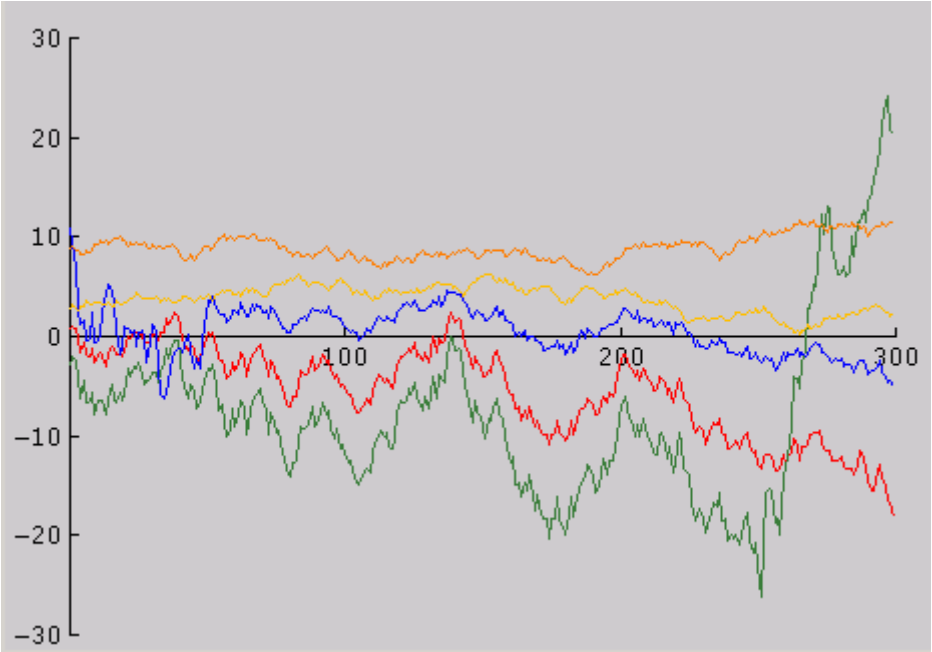


Figure 12: Example of correlated sequences.

We need a method which will monitor data streams and this method should determine whether there is a correlation above a specific threshold (the threshold is determined by the user) among a stream X and other streams. Specifically we need a method that has the following characteristics:

- *Computational efficiency*: Fast and robust computation for detecting any abnormal system behaviour in an almost time-continuous fashion.
- *Agility*: The memory space requirement should also be linear on the length *n*.
- *Accuracy*: Given that the exact computations require increased space and time resources, we rely on suitable approximations with minimal approximation error.

The technical problem we are focus on is "on the fly" monitoring of pairwise correlations between high-dimensional uncertain sensor data streams. As we mentioned in the section 2.2, there is a great complexity between correlations in sensor data streams we have to examine. Considering these correlations and data uncertainty we define correctly the HYDROBIONETS project alerters. We propose the way to define the correlations between two data streams, that don't measure the same physical phenomenon and therefore they are characterized by alterative distributions. Moreover, the physical phenomena follow distributions that are not known in advance, because they are evolving dynamically.

The framework we propose uses careful approximations, exploiting the compression property of the Discrete Fourier Transform. The net effect is that our framework has good performance in terms of speed and memory, while it maintains excellent accuracy. Our experiments on real and realistic streams provided by ACCIONA

Agua show that our framework is faster than the straightforward correlation computation, while maintaining relative error in low level.

This chapter describes the major functionality assigned to our uncertainty-aware data processing system: Finding correlations among pairs of data streams, which monitor dynamic phenomena and their distributions, may be completely different. Initially, in section 4.1 we describe the use of correlation extraction. The section 4.2 describes briefly the key ideas adopted by our method and in the section 4.3 we analyse the monitoring of pair-wise correlation. Lastly, the section 4.4 gives the related work on data streams correlation monitoring and presents their vulnerabilities, if we apply them in HYDROBIONETS project.

4.1. *Monitoring Stream Interrelations*

The most commonly used technique for investigating the relations between quantitative variables, is the *correlation computation.* The goal of a correlation analysis is to detect whether two or more variables co-vary, and to quantify the strength of the relationship between these variables. There are two main uses for correlation computation[6]:

(i) Testing hypotheses about cause-and-effect relationships. In this case, the values of the X-variable are determined and we observe whether variation in X causes variation in Y (for example, giving different values to water PH and measuring the biofouling). This kind of correlations, are exploited once during the design phase of a system to obtain the necessary information about data interrelations.

(ii) Detecting whether two variables are associated without necessarily inferring a cause-and-effect relationship. If an association is found, the inference is that variation in X may cause variation in Y, or variation in Y may cause variation in X or variation in some other factor may affect both X and Y. These correlations are computed and controlled throughout the system operation because they can be changed. These changes are monitored and used for the best outcome of the operation system.

In summary, correlation extraction from data streams is used to assess the strength and direction of the relationships between them. Correlation between data streams indicates a predictive relationship (e.g. to predict misleading values of a data stream) that can be exploited for further analysis in for-casting or simulation tools.

Depending on the monitored phenomenon and the environmental conditions, the behaviour of the recorded data streams may evolve significantly over time. Changes in data characteristics (*e.g.,* statistical distribution) may indicate anomalies in the "normal" behaviour of the monitored streams, or alterations in the data acquisition or transmission process. Quantification of the degree of interrelation between pairs of seemingly different sensors, in conjunction with the detection of behaviours variations, is crucial for a meaningful and reliable decision making in an industrial infrastructure, as is our case.

[6] https://explorable.com/correlation-and-regression

To be more specific, the use of correlation extraction in HYDROBIONETS data concerning the monitoring of water desalination is threefold:

1. *Continuous monitoring of a dynamic system*: In section 2.2 we discussed the sensing performed in different points in the plant. In Table 5 we summarize the most interesting patterns that we have defined for data streams interrelations in the HYDROBIONETS project:

Table 5: Description for data streams interrelation patterns in HYDROBIONETS project.

Pattern	Description	Type of data streams interrelation	Involved Sensors
P1	If we observe a high flow rate in the fresh water (this water property is tracked by electromagnetic flowmeter sensor) then we check the pressure in the saline water.	Analogous	(a) Electromagnetic flowmeter sensor (b) Pressure sensor
P2	If we observe low response time in redox sensor, then we check the chlorine concentration, because there is a sudden rise of this.	Inversely Analogous	(a) Redox sensor (b) Chlorine sensor
P3	Increase in pH or temperature measurements presage the growth of biofouling.	Analogous	(a) PH or Temperature sensors (b) Biofilm sensor
P4	If we observe reduce to the water salinity, then we check the quality of the water effluent from MBR membranes.	Inversely Analogous	(a) Conductivity sensor (b) Chemical sensor
P5	If we observe fluctuations in differential pressure, then we check the concentration of biofouling in the water.	Analogous	(a) Pressure sensor (b) Biofilm sensor
P6	If we observe increase of the water temperature near in membranes, we expect the increase of water pressure in nearby point	Analogous	(a) Temperature sensor (b) Pressure sensor

The correlations concerning the above patterns are monitored continuously throughout the water desalination process, since we observe the right performance of the dynamic phenomena performed in this process. Besides, as we can see in the following, these interrelations notify the existence of events of interest.

2. *Simultaneous monitoring of the system*: Via the data streams correlation extraction, we have the opportunity to observe the water treatment behaviour in specific points in the plant. Figure 13 presents the sensors distribution in the different stages of the water treatment (pre-filtered, pre-treatment-reverse osmosis) in La Tordera's desalination pilot plant.

The uncertainty-aware data processing system gives us the opportunity to monitor the system behaviour in specific points into the plant. The set of the available electrochemical sensors is divided into subsets of highly correlated sensors. In every different phase of the water treatment, different subsets of sensors are used. This clustering enables a more convenient and meaningful monitoring of the overall infrastructure, since we have the capability to identify the possible errors in the specific phases in water treatment. This way we can intervene directly in the desalination process and we benefit in *time* (since we know exactly where the error occurs) and *cost* (since we prevent the process of a wrong procedure).

3. *Distinguishing efficiently between occasional and extreme events* constitutes a major issue in the design of data management systems. This is the third major use of correlation extraction in our infrastructure. It is of great importance to ensure in real or almost real time, especially when we deal with massive data sets, that a true extreme event occurs and not some coincidence or system/network failure. On the other hand, the degree of correlation between two or more sensor data streams characterizes their interrelations and dependencies. For this, the identification of highly correlated streams can be exploited as a further guarantee to verify the existence of a detected extreme event.

For instance, consider the case of two data streams recorded by a pressure and a temperature sensor, respectively. When the two sensors are placed nearby, we expect that a high pressure is associated with an increased temperature, which means that the correlation of these two streams should be relatively high. Thus, we assume that a potential notification for an extreme temperature should be related with a high measured pressure. If this is not the case, this information can be further exploited by a system operator to focus more on that part of the industrial infrastructure and perform a more thorough examination. The only ambiguous point here is related to the determination of "high correlation". The degree of "high correlation" is related to the specific application and the end-user, who has the flexibility to define how much strict this degree will be.

Timely actuation is a crucial issue, while providing guarantees for the validity of a detected extreme event is also of high significance. The transparency of the results should be more assured, since the uncertainty of data is arising from hardware defections or environmental variation in our infrastructure. To this end, we need to monitor continuously and in an online fashion the interrelations between a number of distinct data streams produced by sensors at different stages of water treatment (*e.g.,* pre-filtering, pre-treatment and reverse osmosis), while accounting for their inherent imprecision expressed in terms of uncertainty.

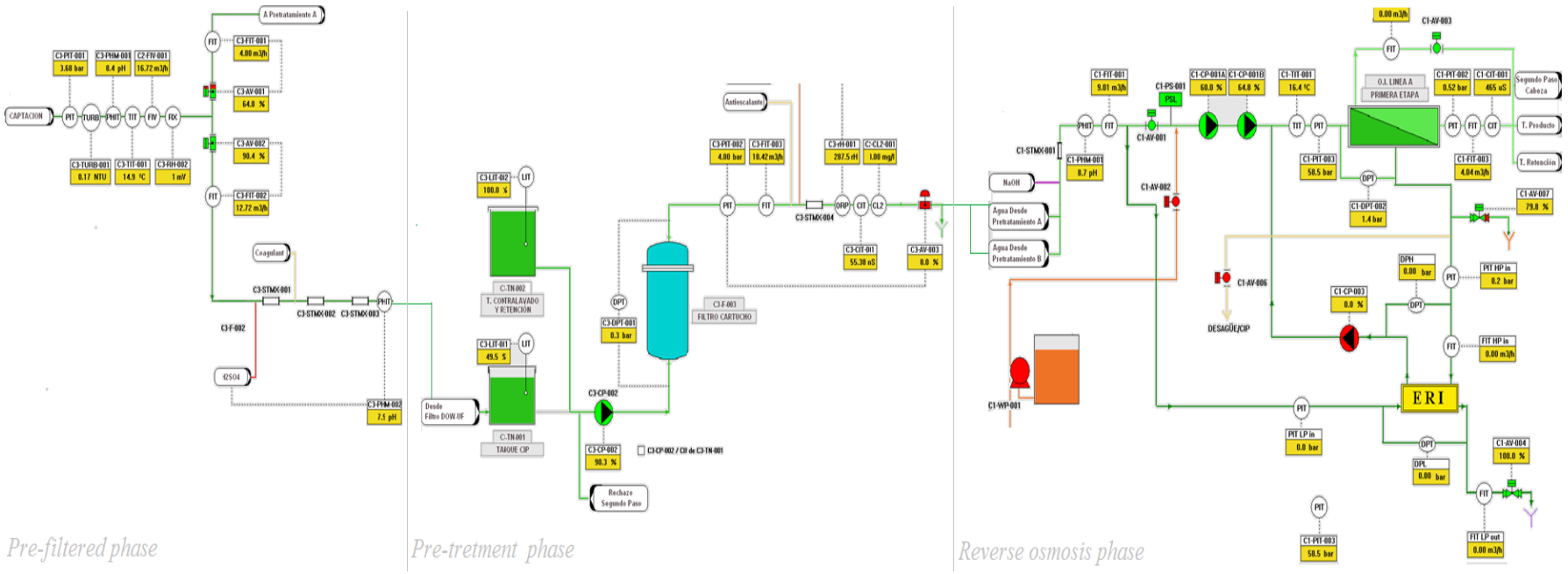


Figure 13: Sensor distribution in La'torderas desalination plant.

4.2. *Preliminaries*

In this section we introduce and analyse some concepts that are using in our approach, presented in section 4.3.

4.2.1. Data streams and sliding windows

A data stream D comprises a continuous stream of *m* tuples, consisting of *n* attribute values A_i ($1 \leq i \leq n$) and the timestamp *t*. For an efficient data uncertainty management, the stream is partitioned into windows each of which is identified by its starting point t_b, its end point t_e, the window size *w* and the sleep step *s*. A window contains the sensor measurements and each measurement is characterized from uncertainty information. This model is depicted in Figure 14.

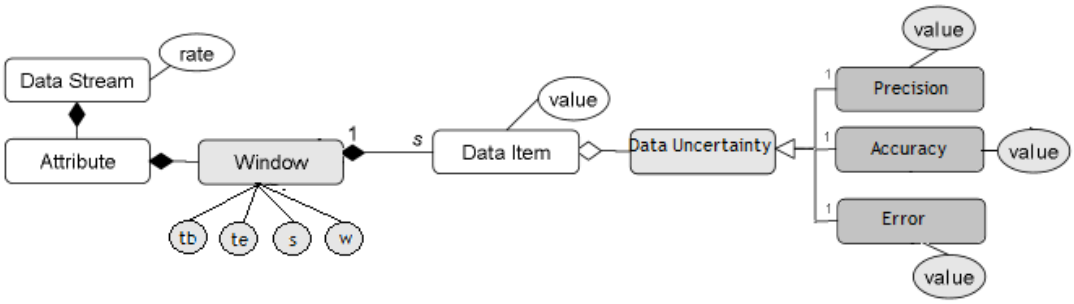


Figure 14: Data uncertainty model

Data streams must be handled either online or from databases, as data items flow rapidly into the system from our sensors. Over this dynamic data, the system must provide timely and incremental responses to multiple continuous queries, ideally keeping in pace with the data arrival rate. Since the size of the stream is potentially unbounded, the state of the data is not known in advance, so responses clearly depend on the set of stream tuples available during query evaluation. Streaming data is usually retained in memory and not physically stored on disk. Thus, it is not practically feasible to "remember" the entire history of rapidly accumulating stream elements due to resource limitations. The operators of the physical algebra keeping state information such as the join and aggregation usually cannot produce exact answers for unbounded input streams with a finite amount of memory. Besides, users can only be interested in the data recently arriving within a fixed time period.

To overcome such difficulties, *windows* have been introduced in query formulation. Such constructs generally emphasize on the latest data by taking advantage of an ordering among tuples, usually established through timestamp values attached to every item. Intuitively, at any time instant, a window operator (we will refer to it as "window") specifies a finite set of recent tuples from the unbounded stream; this finite portion of the stream will be subsequently used to evaluate the query and produce results corresponding to that time instant. As time advances, fresh items get included in the window at the expense of older tuples that stop taking part in computations. A window is generally considered as a mechanism for adjusting

flexible bounds on the unbounded stream in order to fetch a finite, yet ever-changing set of tuples, which may be regarded as a temporal relation.

There are two ways to physically build windows: (i) *attribute-based* windows, and (ii) *count-based* windows. In the first case, an attribute is designated as the windowing attribute (usually time), and consecutive tuples for which this attribute is within a certain interval constitute a window (*e.g.,* stock reports over the last 10 minutes). Here, tuples are assumed to arrive in increasing order of their windowing attributes. In the second case, a certain number of consecutive tuples constitute a window (*e.g.,* the last 10 readings from a sensor).

Definition 4 (Window over data stream): *Let W_E be a window with conjunctive condition E applied at time instant $\tau_0 \in \mathcal{T}$[7] over the items of a data stream S, i.e., over its current contents $S(\tau_0)$. Then:*

$$\forall\, \tau_i \in T, \tau_i \geq \tau_0, W_E\big(S(\tau_i)\big) = \{s \in S(\tau_i) : E(s,T)\ holds\}$$

provided that for any large, but always finite $n \in N$.

Therefore, each window is applied over the items of a single uncertain data stream S. The stream *S* consists of uncertain objects (denoted by *s*) and at every τ_i returns a concrete finite set of tuples $W_E(S(\tau_i)) \subset S(\tau_i)$ which is called the window state at this time instant. The conjunctive condition E relates to the type of sliding window (attribute/count based). For the rest of the document we will refer to W_E as *W*.

Figure 15 illustrates the scenario of sliding windows over one uncertain data stream *S*. Each uncertain data stream consists of a sequence of continuously-arriving uncertain objects at different timestamps, that is, $S = \{s[1], s[2], ..., s[t], ...\}$, where $s[i]$ is an uncertain object at timestamp *i*, and *t* is the current timestamp. Specifically, as shown in Figure 15 an operator always considers the most recent *w* uncertain data in stream, that is, $W_E(S) = \{s[t - w + 1], s[t - w + 2], ..., s[t]\}$ at the current timestamp *t*. In other words, when a new uncertain object $s[t + 1]$ comes in at the next timestamp $(t + 1)$, the new object $s[t + 1]$ is appended to *S*. Meanwhile, the old object $s[t - w + 1]$ expires and is evicted from the memory. Thus, operators at timestamp $(t + 1)$ is conducted on a new sliding window $\{s[t - w + 2], ..., s[t + 1]\}$ of size *w*.

[7] Time domain T is regarded as an ordered, infinite set of discrete time instants $\tau \in T$. A time interval $[\tau_1, \tau_2] \in T$ consists of all distinct time instants $\tau \in T$ for which $\tau_1 \leq \tau \leq \tau_2$.

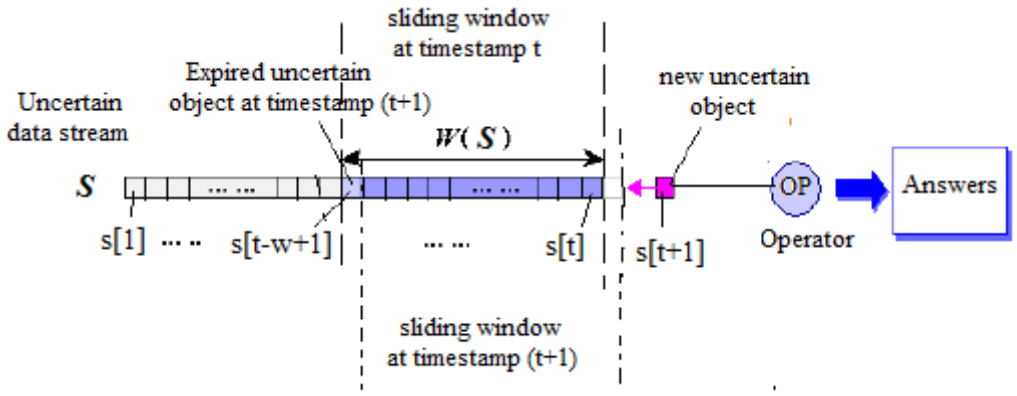


Figure 15: Illustration of sliding window on uncertain data streams

As expected, the separation of data streams in sliding windows brings problems of completeness and synchronization in the final results. These problems aren't qualified because it is out of the scope of this thesis. To conclude, the available electrochemical sensors may report a measurement within a predefined period of time, usually in the scale of a few seconds or minutes. Data processing of raw data streams is performed on the basis of *sliding windows*. In particular, a sliding window of recent measurement values is maintained, while the window moves with a predetermined step size when new measurements become available. Furthermore, as the contents of the sliding windows evolve over time, it makes sense for users to ask a query once and receive updated answers over time.

4.2.2. Pearson's Correlation function

A similarity measure is a relation between a pair of objects and a scalar number. Common intervals used to mapping the similarity are [-1,1] or [0,1], where 1 indicates the maximum of similarity. The most common similarity function used to perform complete or partial matching between time series is the *cross-correlation* function or *Pearson's correlation* function. The cross-correlation between two time series x and y of the same length N and same starting timestamp is defined as:

$$r_{xy} = \frac{\sum_{n=0}^{N-1}(x_n - \bar{x})(y_n - \bar{y})}{\sqrt{\sum_{n=0}^{N-1}(x_n - \bar{x})^2}\sqrt{\sum_{n=0}^{N-1}(y_n - \bar{y})^2}} \quad (1)$$

where $\bar{x}$ and $\bar{y}$ are the means of x and y respectively. The correlation r_{xy} provides the degree of linear dependence between two vectors x and y from perfect relationship (r_{xy} =1), to perfect negative linear relation (r_{xy} = -1). Equation (1) means that the cross-correlation coefficient can be computed by simple summation of the distinct data stream objects, which support incrementally computation. Based on this equation, we can design a straightforward approach to detect the correlation.

Intuitively, once the sliding window receives the new data objects, we incrementally update the basic summations (e.g., $\sum x_n^2$ and $\sum x_n$) of the subsequences x and y, within the sliding window. After processing the data objects in current window, if the correlation condition is satisfied, the current correlated subsequences should be kept, and when the following data objects come in, we make the incremental calculation of cross-correlation again.

It is obvious that the naive solution can continuously detect the correlations between the engaged data streams. However, for each new data object we have to recalculate the cross-correlation coefficient, which will result in high computation complexity. The major cost is produced by the sum of inner-product as described in Equation (1). In the area of signal processing and statistical analysis, the sum of inner-product is usually calculated by Discrete Fourier Transform (DFT) for efficiency purpose. Therefore, we propose to make use of the theoretical results of DFT to design a more sophisticated approach for correlation detection.

4.2.3. Data Reduction in Data Streams

Data streams are observations made in sequence and the relationship between its consecutive data items gives us the opportunity to reduce the size of the data without substantial loss of information. *Data reduction* [29] is often the first step to tackling massive time series data because it will provide a synopsis of the data. A "quick and dirty" analysis of the synoptic data can help us to spot a small portion of the data with interesting behaviour. Further thorough investigation of such interesting data can reveal the patterns of ultimate interest.

Data reduction techniques will reduce the massive data into a manageable synoptic data structure while preserving the characteristic of the data as much as possible. It is the basis for fast analysis and discovery in a huge amount of data. Data reduction is especially useful for massive data streams due to the high dimensionality of the data streams (we referred to the dimensionality in 2.2.1). Almost all high-performance analytical techniques for time series rely on some data reduction techniques. Because data reduction for data streams results in the reduction of the dimensionality of them, it is also called *dimensionality reduction* for data streams.

Many data reduction techniques can be used for time series data. In this subsection we will mention the most common of them. We analyze in details the data reduction with *Discrete Fourier Transform (DFT)*, which is the first proposed data stream reduction technique in the data mining community and is widely used in practice. *Discrete Wavelet Transform (DWT)* is a new signal processing technique based on Fourier Transform. It gains popularity in data streams analysis as it appears low computation cost. *Singular value decomposition (SVD)* is an optimal data reduction technique based on traditional principal components analysis. It is an attractive data reduction technique because it can provide optimal data reduction in some circumstances. A very new data reduction technique is the *random projection* technique. Random projection of time series has great promise and yields many nice results because it can provide approximate answers with guaranteed bounds of errors.

In Table 6, we summarize the characteristics of the above reduction techniques (the n parameter denotes the data stream length).

Table 6: Comparison of data reduction techniques

Data Reduction technique	DFT	DWT	SVD	Random Projection
Time complexity	$n \log n$	n	$\frac{m}{n} + n^2$	nk
Based on orthogonal transform	*Yes*	*Yes*	*Yes*	*No*
Approximation of data streams	*Yes*	*Yes*	*Yes*	*No*
Require existence of principal components	*Yes*	*Yes*	*Yes*	*No*
Compact support	*No*	*Yes*	*Yes*	*Not relevant*

The lower time complexity in computing the data reduction for each data stream with length n, is presented for DFT technique. DFT, DWT and SVD are all based on orthogonal transforms. From the coefficients of the data reduction, we can reconstruct the approximation of the time series. By comparison, random projection is not based on any orthogonal transform. We cannot reconstruct the approximation of the time series. To approximate a time series by a few coefficients, the DFT, DWT and SVD require the existence of some principal components in the time series data. Random projection, by contrast, does not make any assumption about the data. This makes random projection very desirable for data streams having no obvious trends such as price differences in stock market data.

Discrete Fourier Transform

Discrete Fourier Transform (DFT) converts a finite list of samples of a function into the list of coefficients of a finite combination of complex sinusoids, ordered by their frequencies, that has those same samples values[8]. It can be considered that DFT converts the sampled function from its original domain (in our case from time domain) to the frequency domain. Based on this assumption, we overcome the problem of *comparing dissimilar streams*. We mentioned in chapter 2 that data streams from different sensors should be compared for managing the HYDROBIONETS data infrastructure. One problem that we have to resolve in this case is the comparing data streams that they not be measured on the same scale. For example, suppose that we are interested in comparing the temperature and pressure data streams near the membranes (pattern P6 from Table 5). The temperature is measured in Celsius scale (50-100 ^{o}C) and the pressure is measured in Bars scale (0-1 Bars). To overcome this problem, we transform our data streams into frequency

[8] http://en.wikipedia.org/wiki/Discrete_Fourier_transform

domain. *Based on DFT we reduce, except from the stream length, the affection of the diversity in the data stream values.*

In the following, we will first introduce the basic knowledge of DFT, and provide important lemmas and properties in DFT theory, which based on them our DFT-based solution is elaborated.

Let $x = \{ x(0), x(1),\ldots, x(n),\ldots x(N-1)\}$ be a N-point sequence, and the Discrete Fourier Transform of x be $X = \{ X(1), X(2),\ldots,X(k),\ldots,X(N-1)\}$, we have

$$X(k) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}kn} \qquad k \in [0, N-1] \qquad (2)$$

The Inverse Discrete Fourier Transformation (IDFT) of X is

$$x(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X(k) e^{j\frac{2\pi}{N}kn} \qquad n \in [0, N-1] \qquad (3)$$

Note that x and X are of the same size and the DFT of a data stream is another data stream. In the following table, we summarize the most important properties of DFT:

Table 7: DFT theorems and properties

Property	Data stream representation	Transform stream representation
Periodicity	x(n) = x(n+N)	X(k) = X(k+N)
Linearity	ax(n)+by(n)	aX(k)+bY(n)
Symmetry	x(n):even x(n):odd	X(k):even X(k):odd
Convolution	x(n)*y(n)	X(k)Y(k)
Inner product	$\langle x(n), y(n) \rangle$	$\sum_{k=0}^{N-1} x(k) y(k)$
Parseval's theorem	$\sum_{n=0}^{N-1} x(n) y(n)$	$\frac{1}{N} \sum_{k=0}^{N-1} X(k) Y^*(k)$

For the most real data streams the first few coefficients contain most of the energy and it is reasonable to expect those coefficients to capture the raw shape of the data streams. Figure 16 shows a data stream and its corresponding DFT coefficients of the measurements of a temperature sensor, placed near to MBR membranes in pre-treatment phase. From the symmetry property of DFT, we know that for a real data stream, their *k*-th DFT coefficients from the beginning are the conjugates of its *k*-th coefficient from the end. This is verified in the same figure.

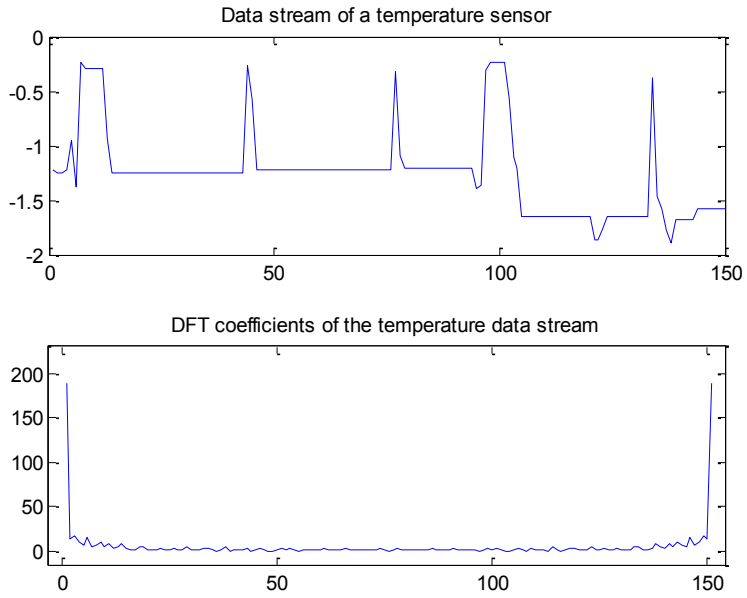


Figure 16: Temperature data stream and its DFT coefficients

The validity of our approach for using the DFT reduction for reducing the stream length and the affection of the data diversity is based on the compactness of the DFT representations. That is, the concentration of the main portions of the energy for a given stream in the first few significant (high-amplitude) DFT coefficients. Figure 17 illustrates this property for four data streams recorded in ACCIONA's plant, from which it is apparent that the main energy content of the streams is concentrated in the first few low-frequency DFT coefficients.

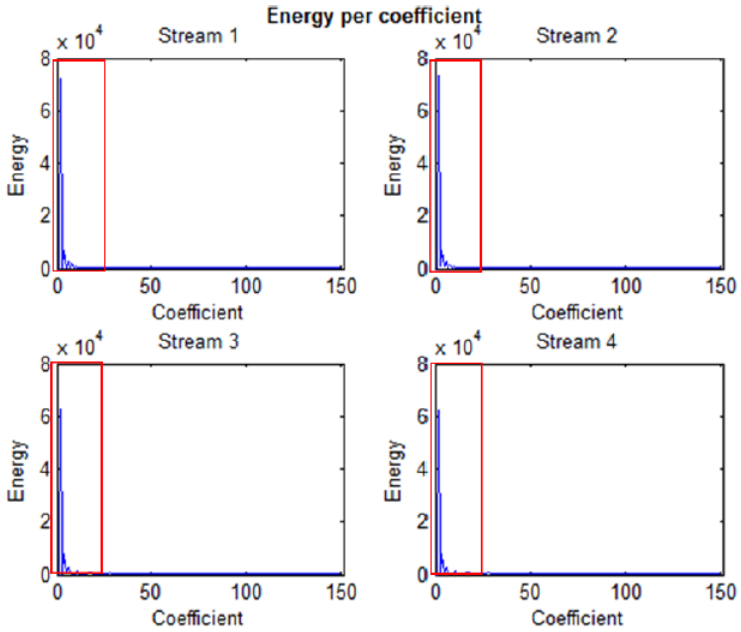


Figure 17 : Amplitudes of DFT coefficients for four real data streams acquired in ACCIONA's plant.

Exploiting the compression property of the DFT to concentrate the inherent energy content of a given signal in the first few high-amplitude coefficients we could reconstruct our data stream using only the first few coefficients. The first step for taking the approximation $\tilde{x}$ of our data stream is to compute the DFT of the data stream x. As second step, we have to define how many coefficients will be used for the data stream reconstruction. We notice here, that as we use more and more DFT coefficients, the DFT approximation gets better (Figure 18). After that, we have to compute the inverse DFT of the coefficients (we refer to this stream as $\tilde{X}$) that we decided to keep in the previous step. The final step of reconstruction is to get the real part of $\tilde{X}$. Figure 18 shows the DFT approximation of the temperature data stream using the first 10, 40 and 75 DFT coefficients (The initial temperature length was 150 measurements).

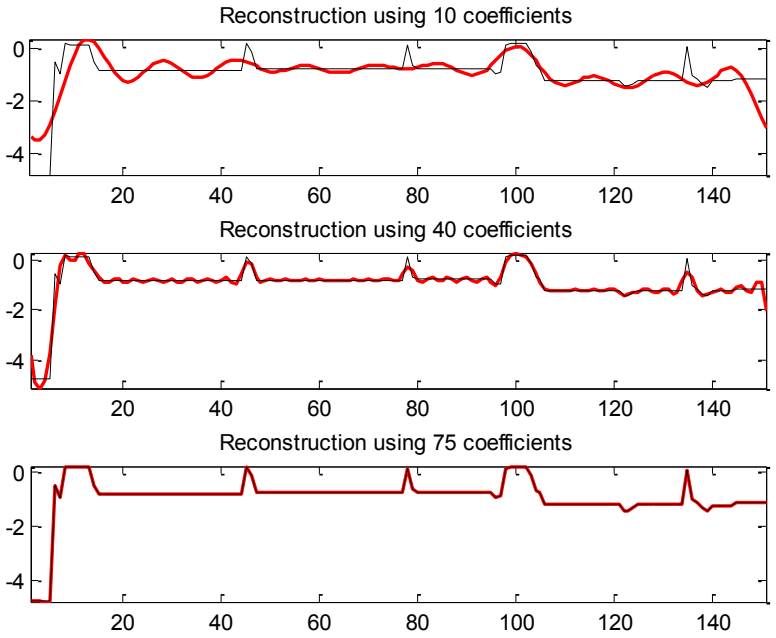


Figure 18: Approximation of temperature data stream with DFT reduction technique. From top to bottom, the data stream is approximated by 10,40 and 75 DFT coefficients respectively.

To conclude, *data reduction* based on the DFT works by retaining only the first few DFT coefficients of a data stream as a concise representation of the data stream. Note that the symmetry of DFT coefficients for real data streams means that the energy contained in the last few DFT coefficients are also used implicitly. The data streams reconstructed from these few DFT coefficients is the DFT approximation of the original data streams.

4.3. *Fast online pairwise correlation estimation*

The typical approach for extracting pairwise sensor stream correlations is by means of the Pearson's correlation coefficient. For two given streams $\boldsymbol{x}$, $\boldsymbol{y}$ of equal length N, which are time-synchronized in windows of size w, the correlation coefficient is given by (1). However, for each newly acquired measurement value, the correlation coefficient has to be recalculated, which yields an increased computational complexity, especially for high-dimensional data streams or for a large number of sensors. In particular, the major cost comes from the summation of inner products of the form:

$$\langle x_w, y_w \rangle = \sum_{i=1}^{w} x_i y_i \tag{4}.$$

Motivated by this limitation, in our proposed UADM system we implement a computationally efficient method for nearly real-time extraction of highly correlated data streams by combining discrete Fourier transforms (DFT) over sliding windows with a proper *stream similarity measure*. In order to account for the underlying uncertainty or other data ambiguities, we restrict ourselves on the detection of pairs of streams whose correlation is above a specific threshold.

DFT and Euclidean distance based approach

Let $\boldsymbol{x}_w$ and $\boldsymbol{y}_w$ denote two time windows of length w corresponding to the same time-interval. Working in a DFT framework, each sample x_i (similarly, y_i) can be expressed in terms of a linear combination of exponential functions

$$x_k \approx \frac{1}{\sqrt{w}} \sum_{f=0}^{N-1} X_f e^{i 2\pi f k / w}, k = 1,...w \tag{5}$$

where X_f is the set of N DFT coefficients, with $N < w$. In this way, the computational cost for computing the inner product between the two time windows (and subsequently the correlation coefficient) is reduced from w to N. The fast and efficient computation of the DFT guarantees that it can be used to compute inner products and, thus, correlations over sliding windows of any size.

The above DFT-based approach enables the fast monitoring of synchronized streams over a given time window, whose correlation exceeds a predefined threshold. This is dictated by the following lemma, which gives a correspondence between the correlation coefficient and the Euclidean distance between two data streams.

***Lemma 1 [30]*:** The correlation coefficient of two data streams $\boldsymbol{x}$, and $\boldsymbol{y}$, of length w is expressed in terms of a Euclidean distance as follows

$$corr(x, y) = 1 - \frac{1}{2w} d^2(\hat{x}, \hat{y}) \tag{6}$$

where $d(\hat{x},\hat{y})$ is the Euclidean distance between $\hat{x},\hat{y}$, that is, the original data streams normalized to mean zero and variance equal to one (for more details please see *Appendix A*).

By reducing the correlation coefficient to Euclidean distance, we can apply the techniques described in [20] to report data streams with correlation coefficients higher than a specific threshold:

***Lemma 2 [20]*:** Let the DFTs of the normalized data streams $\hat{x},\hat{y}$ be $\hat{X}$ and $\hat{Y}$, respectively. Then,

$$corr(x,y) \geq \varepsilon \Rightarrow d_M(\hat{X},\hat{Y}) \leq \sqrt{2w(1-\varepsilon)} \quad (7)$$

where ε is a given threshold and $d_M(\hat{X},\hat{Y})$ is the Euclidean distance between the corresponding truncated DFTs, which are derived by keeping the first $M \leq w/2$ DFT coefficients.

Lemma 2 implies that pairs of windowed sensor streams for which $d_M(\hat{X},\hat{Y}) > \sqrt{2w(1-\varepsilon)}$ cannot have correlation coefficients above threshold ε. By ignoring those pairs, we can get a set of likely correlated stream pairs. This approach which proposed in [20], is indicated for fast correlation monitoring, but it is not a good similarity measure of the data streams behaviour.

The method we propose for fast similarity computation among uncertain data streams compares the related data streams, driven by their behaviour across the time. In Table 5 we defined some basic patterns for monitoring the HYDROBIONETS cyber physical system. In all cases we observe the simultaneous behaviour (analogous/ inversely analogous) of the pre-defined data streams. For example, for the pattern p1, the related streams have analogous interrelation. This means that if the values of one stream are increasing, we expect that the values for the related stream are also increasing, under normal conditions. Otherwise, we have an unpredictable evolution in our system that needs our attention.

The approximation of the correlation coefficient via the Euclidean distance is not limited only in the computation of the above interrelations because this approach is very sensitive and leads us easily to incorrect conclusions. As we can see from Equation (6), the correlation coefficient is (inversely) related to the Euclidean distance between standardized versions of the data. This approach considers that data streams with small Euclidean distance are more correlated than other data streams with longer Euclidean distance. Based on this assumption, the data streams depicted in Figure 19 in case (b) are more correlated than the data streams depicted in case (a), even though that they have similar behaviour in both cases. The major difference in our approach is that we consider the data streams in both cases highly correlated, since they present similar behaviour across the time.

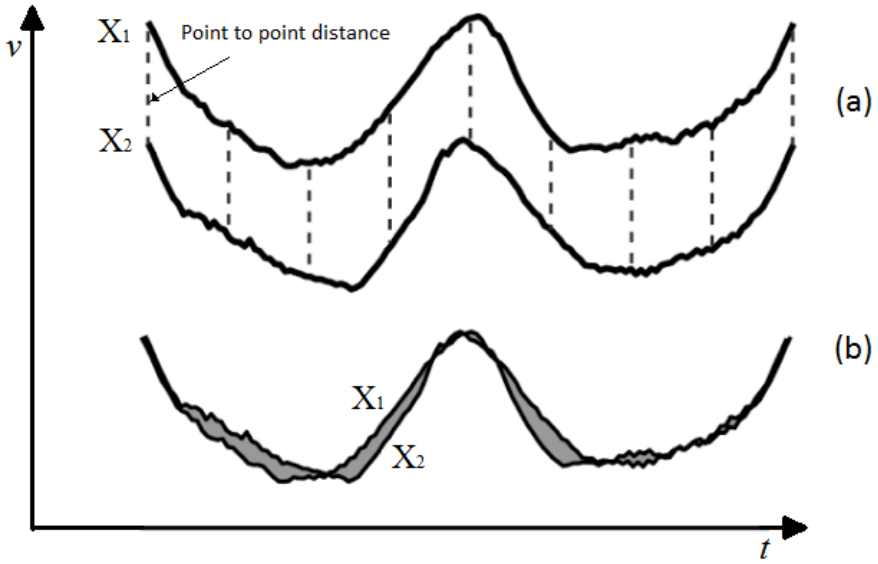


Figure 19: Euclidean distance is not a good similarity measure for data streams behaviour monitoring.

Fast pairwise similarity computation

In our data management system, we search for data stream pairs whose correlation is above a predefined threshold ε_{th}, in a fixed-sized sliding window. More specifically, let $\boldsymbol{s}$ be the reference stream, and $(\boldsymbol{y}_1,\boldsymbol{y}_2,..,\boldsymbol{y}_C)$ be the set of streams with which we compute the pairwise correlations in the current time interval. For a predetermined correlation threshold ε_{th}, the output of the process will be a subset of streams $\boldsymbol{y}_c$, for which the correlation with $\boldsymbol{s}$ is above ε_{th}.

In our proposed system, the problem of extracting highly correlated pairs of sensors' data streams is translated into a problem of identifying highly "*similar*" sensors' data streams, where the "similarity" is measured by an appropriately designed function. As in the previous DFT-based approach, the first step for each data stream values in the current window of length w, $x_1, x_2, \ldots x_w$, is to normalize to mean zero and variance one, that is,

$$\hat{x}_i = \frac{x_i - \bar{x}}{\sigma_x} \tag{8}$$

where

$$\sigma_x = \sqrt{\sum_{i=1}^{w} (x_i - \bar{x})^2} \tag{9}$$

As a second step, the corresponding DFT of the normalized windowed data is computed. The data compression capability of the DFT is exploited to reduce (i) the computational cost by approximating the original data by a highly reduced set of coefficients and (ii) the scaling in data stream values which monitor different quantities.

The final step towards our fast and robust extraction of highly "similar" sensor data streams is to identify those pairs $(\boldsymbol{s}, \boldsymbol{y}_c)$ with similarity above a given threshold ε_{th}. In order to avoid computing the similarity between all pairs of streams $(\boldsymbol{s}, \boldsymbol{y}_c)$, we reduce the set of candidate streams only to those streams that will be highly similar with $\boldsymbol{s}$ with high probability.

For this purpose, we introduce *peak similarity*, *psim*, as an appropriate similarity measure (in Appendix B- *Similarity measures* we mention three different measures for the estimation of similarity between data streams). More specifically, the similarity between two windowed data streams $\boldsymbol{s}$, $\boldsymbol{y}$ is computed by employing a truncated set of the first M high-amplitude DFT coefficients, where $M \approx w/2$, and the peak similarity measure is defined as follows:

$$p_{sim}(s, y) = \frac{1}{M} \sum_{i=1}^{M} \left[1 - \frac{|S_i - Y_i|}{2 \cdot \max(|S_i|, |Y_i|)} \right] \tag{10}$$

In order to account for the potential loss of information caused by the truncation of the set of DFT coefficients, the peak similarity measure does not employ the same threshold ε_{th} for finding the similar streams. Instead, we determine a new threshold $\varepsilon_{th,new}$, with our proposed method reporting as "highly-correlated" pairs those streams $\boldsymbol{s}$ and $\boldsymbol{y}c$ for which $psim(\boldsymbol{s},\boldsymbol{y}_c) > \varepsilon_{th,new}$. However, special attention should be given on the selection of the threshold value $\varepsilon_{th,new}$. From our experimental evaluation, employing data from a set of various distinct sensors, we observed that if we choose an "elastic" enough threshold $\varepsilon_{th,new}$, then the subset of streams $\boldsymbol{y}c$ with the highest peak similarity with $\boldsymbol{s}$ will also contain the highly correlated streams with $\boldsymbol{s}$ (that is, those with correlation coefficient above ε_{th}). In our implementation we set $\varepsilon_{th,new} = \varepsilon_{th} - e$, where e is a small positive number (in our experimental evaluation described below we set $e < 0.05$).

Uncertainty-aware fast pairwise similarity computation

Towards the design of an integrated uncertainty-aware data management system, we extend the above peak similarity measure in order to monitor similarities between uncertain data streams. For this, Equation (10) is not applied directly on the raw data streams, but on the original recordings by also accounting for their estimated uncertainty. We notice here, that the estimation of uncertainty in raw data streams is discussed in 3.1.2. Let U_i be the uncertainty value for the current window of each sensor data stream.

The similarity monitoring of uncertain data streams, also affects the choice of the thresholds used to decide whether two streams are highly similar or not. Specifically, the threshold $\varepsilon_{th,new} = \varepsilon_{th} - e$ is set based on the streams $\boldsymbol{s}_1 \pm U_1$ and $\boldsymbol{s}_2 \pm U_2$, where U_1 and U_2 are the corresponding estimated uncertainties of the two streams.

From the above, we derive an uncertainty-aware extension of *psim* which is given by

$$p_{sim,U}(s,y) = \frac{1}{M}\sum_{i=1}^{M}\left[1 - \frac{\left|\widetilde{S}_i - \widetilde{Y}_i\right|}{2 \cdot \max(\left|\widetilde{S}_i\right|, \left|\widetilde{Y}_i\right|)}\right] \quad (11)$$

where $\widetilde{S}$, $\widetilde{Y}$ are the truncated DFTs of the uncertain streams $\widetilde{s} = s + U_s$ (or $\widetilde{s} = s - U_s$) and $\widetilde{y} = y + U_y$ (or $\widetilde{y} = y - U_y$), respectively, with U_s and U_y denoting the uncertainties estimated in the current window of ***s*** and ***y***, respectively. The basic algorithm describing the previous infrastructure is presented in Figure 20:

Algorithm FindStreamCorrelations
Input:

- A reference stream S and its corresponding uncertainty value U_S ($S \pm U_S$)
- A set of data streams (denoting as $Yc = \{y_1, y_2 \ldots y_c\}$) and the corresponding uncertainty values U_i for each y_i. ($y_i \pm U_i$)
- The correlation threshold ε_{th}

Output:
A subset *C* of streams *Yc* for which the correlation with *S* is above ε_{th}

for each input stream (*S* and *Yc*) **do**
//Normalize to mean 0 and variance 1 based on (8) and (9)
$\hat{x}$ = NormalizationOf (*S* and *Yc*);
end for
for each normalized data stream $\hat{x}$ **do**
//Compute the Discrete Fourier Transform
X=DFT($\hat{x}$);
end for
//Determine the new correlation threshold
$\varepsilon_{th,new} = \varepsilon_{th} - e$
for each data stream y_i (from the input set *Yc*) **do**
//Compute the peak similarity measure with the reference
//stream *S* via (11)
peak_similarity = $p_{sim}(S \pm U_S, y_i \pm U_i)$;
//Decision making
if peak_similarity > $\varepsilon_{th,new}$ **then**
Add y_i to the output subset *C;*
end if
end for
Return *C*;
End of FindStreamCorrelations

Figure 20: Algorithm for detecting correlations in uncertain data streams, above ε_{th}

Finally, the steps implementing our proposed fast and robust uncertainty-aware similarity measure are shown in Figure 21:

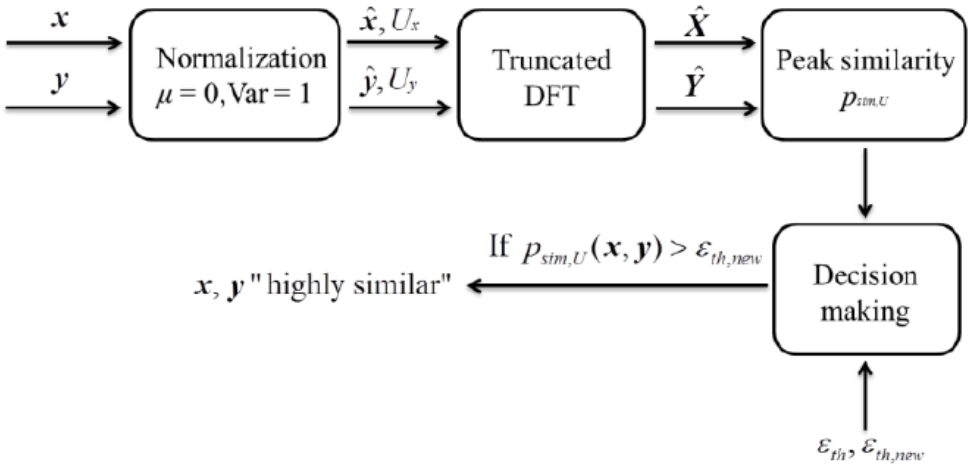


Figure 21: Flow diagram for fast computation of uncertainty-aware pairwise sensor stream similarity

4.4. ***Related Work***

In this section we present the related work recording the data stream correlation monitoring. We introduce the major points of each technique and we annotate the reasons we cannot apply these techniques in our application.

Yasushi Sakurai et al. [19] proposed BRAID algorithm which can detect lag correlations between data streams. It can handle data streams of semi-infinite length and they use careful approximations exploiting the Nyquist sampling theorem. They proposed to find the first maximum point on the global cross-correlation coefficient curve of two data streams, which takes times delay as a variant. They introduced an approximation by keeping a geometric progression of the lag values and using a base window they calculate the correlation coefficient. Their goal is to monitor *k* numerical data streams, $X_1,\ldots,X_k$ and at any point of time to determine two things: (i) which pairs of data streams have a *lag* correlation and (ii) what is the lag correlation length.

This approach cannot be applied in the HYDROBIONETS data processing infrastructure, since [19] focuses on *lag* correlations estimation on data streams. It monitors the stream correlations in different time intervals of each data stream. In our infrastructure we are interested in data stream correlations extraction in *same* time intervals, since through the data stream correlations in the same periods of time we:

(i) Monitor the evolution of the dynamic phenomena developed in different stages of the desalination process.
(ii) Detect-identify-localize sources and events of interest
(iii) Perceive any abnormal sensor behavior (coincidence or system/network failure)
(iv) Get the guarantees for the validity of a detected extreme event and provide the necessary notifications for the activation of the corresponding alerter.

Zhu and Shasha et al. [20] proposed the StatStream model based on time series data streams. Their system's goal is to compute in almost constant time the statistics for multi-stream analysis problems. The core function that they compute is Pearson correlation function (1) over sliding windows. They do this using Fourier transforms and random-vector based sketches primarily depending on how "cooperative" the data is. First of all, they get the discrete Fourier coefficients by applying the Discrete Fourier Transform (DFT) on a base window (e serial sequence of sensor measurements), then map data streams into a grid structure and calculate the correlation coefficients between streams in adjacent grids.

StatStream assumes that a data stream cannot be regarded as having a terminating point, and works on the data stream continuously, in order to meet real-time requirements. To be explicit, a data stream is regarded as a sequence, rather than a set. StatStream provides a basis for stating that any statistics presenting in the data stream at time t will be reported at time t+v, where v is a constant and is independent of the size and duration of the stream. StatStream establishes three major time periods: i) timepoints - the system quantum, ii) basic window - a consecutive subsequence of time points which comprise a digest, and iii) sliding window - a user-defined consecutive subsequence of basic windows that will form the basis for the time period over which a query may be executed. While this provides a great deal of flexibility in dealing with intervals, StatStream expects to have at least one value per timepoint and, if one is not present, an interpolated value is used. The interpolation in the face of missing data may insert a false reading in our data and lead us in false decisions. Moreover, because multiple values being reported in one timepoint, StatStream provides a summary value. The synthesis of summary values over time produces the same stream characteristics, so there is no clear indication that an irregularity has occurred, nor an action can take place to rectify the mistake. This summarization can also obscure the point where a value, or a set of values, has crossed the significance threshold.

The highly correlated stream pairs which are reported from StatStream are based in a hash technique, using a grid structure. The grid structure is geometrically and evenly partitioned into a number of cells. Each stream is located into one cell based on its DFT coefficients. The correlation coefficient is computed only for neighboring streams and in that way, StatStream discovers streams with correlations above a specific threshold.

We cannot apply the StatStream technique in our data, because our application monitors dynamic phenomena. For these dynamic phenomena we don’t know the distribution that they follow a-priori, so we cannot find a good hash function that puts whole streams with similar behavior (in the time) in nearby cells. There is one more weakness: In our uncertainty-aware data processing system, the user specifies the threshold for the strength of the correlated streams (how “similar” they are). A high threshold declares more correlated streams. So, the hash function should be adjusted to the user specified threshold, to put the highly correlated data streams in nearby cells. The main drawback of this technique to be integrated in our infrastructure is the difficulty to define an appropriate “similarity” function for data streams describing dynamic phenomena with unknown prior distributions, which is

normally the case in an industrial environment. Finally, the StatStream method monitors tens of thousands of data streams. Their approach performs well in dealing with large amount of data and it is designed for ad hoc[9] query rather than continuous query. This restriction has strict constraints on time delay. It wouldn't be effective for our application management, since we have less pairs of data streams to monitor.

Qing Xie et al [21] focus on *local* correlation detection, which may occur *in burst* in certain duration, and then disappear. They propose a framework to deal with the continuous detection of local Pearson correlation coefficient with time delay. Similar to conventional approaches for data streaming processing, a sliding window is applied in their framework. Given the maximal time delay allowed, and a minimal correlation value, they analyze the subsequences in the sliding window, and they find if there is any correlation occurring. Since the time delay factor is involved, their solution also employs the Discrete Fourier Transform (DFT). They take the advantage of the properties of DFT, and solve the cross-correlation coefficient with random time shift by reverse DFT on the inner product of two stream sequences. If the correlation is identified, incremental evaluation will be performed until the correlation is lost. Otherwise they can slide the window to the next candidate location.

Another characteristic of this work is that they apply a linear representation to approximate the data streams to accelerate the correlation analysis. They apply piecewise linear representation to use line segments for the approximation of data stream points and indicate the data stream from microscopic view. They are based on the feature of line segments and can make early pruning in the correlation candidates.

The main contribution of this work lies on the identification of *local* correlations in data streams that occur in burst. The proposed technique supports early pruning of correlation pairs. We cannot apply this technique in our processing correlation framework, because we are interested in the similarity of whole data streams and not for local changes in signal similarity that detects the local correlation. Besides, they use the Pearson's correlation coefficient to detect continuously the local correlation between the engaged streams. For each new data point and each possible time delay, they have to recalculate the cross-correlation coefficient which results in high computation complexity. It will be a problem for us, to recalculate the correlation coefficient every time new data of our sensors are arrived. Using the DFT approach, the sum of inner-product of the Pearson's correlation coefficient can be reduced for efficiency purpose.

In [22] the summary of two techniques [24] [26] is presented, that have been proposed for modeling the similarity matching problem for uncertain time series. One issue in this work is the data uncertainty modeling in time series and another is the methods they are using for the similarity matching in data streams. The general idea in these three techniques is that the data uncertainty is modeling with probabilistic methods (for a detailed description please see 3.1.1), which play the role

[9] An Ad-Hoc query is a query that cannot be determined prior to the moment the query is issued. It is created in order to get information when need arises.

of filters to reduce the signal noise. After the use of filters, they compare the time series data involving the concept of distance measures to solve the similarity matching problem in uncertain time series. Given a user-supplied query, a similarity search returns the most similar subsequences of the time-series, which satisfy the user-query, according to some distance functions. Our goal in HYDROBIONETS project is the interrelation and behavior monitoring of the data streams in same time intervals, since this information (i) helps us to the overall system monitoring and (ii) give us the guarantees for the validity of detected extreme events.

To be more specific, in [26] the uncertainty is modeled by means of repeated observations at each timestamp. Assuming two uncertain time series X and Y, the technique proceeds as follows. Firstly, the two uncertain sequences X, Y are materialized to all possible certain sequences: $TS_X = \{< u_{11}, ..., u_{n1} >, ..., < u_{1s}, ..., u_{ns} >\}$ (where u_{ij} is the j^{th} observation in timestamp i), and similarly for Y with TS_Y . Then they compute all possible distances between X and Y ($dists(X,Y)$). The result set of the user defined query is determined by a probability computation, which is formulated by the means of the counting distances: $\Pr(dist(X,Y) \le \varepsilon) = \frac{|\{dists(X,Y) \le \varepsilon\}|}{|dists(X,Y)|}$. The computation of this result set is infeasible, because of the very large space that leads to an exponential computational cost: $|dists(X,Y)| = s_X^n s_Y^n$, where s_X^n, s_Y^n are the number of samples at each timestamp of X,Y respectively and n, is the length of the sequence.

Inspired by the Euclidean distance, [24] resolves the similarity matching problem by computing the sum of the differences of the streaming time series random variables. Each random variable represents the uncertainty of the value in the corresponding timestamp. This formulation is statistically complex, since it works under the assumption that all the time series values follow a specific distribution, and we don't have such knowledge since we observe dynamic phenomena that evolve over time. Besides, with this technique we cannot compute an exact value of the Euclidean distance between two uncertain time series, since only the mean and the deviation of each random variable at each timestamp are available. The uncertain distance between two uncertain series is also a random variable, something that increases the uncertainty factor in our infrastructure.

To conclude, both [26] and [24] compute the similarity between two sequences of the same length, by summing the ordered point-to-point distances between them. In this sense, they assume that the comparing variables are measured exactly on the same scale (e.g all temperature data streams are scaled on a Centigrade scale).In our case, for HYDROBIONETS data processing, this assumption is not valid, since we have to monitor the similarity in different data streams, e.g between temperature and pressure data streams or water flow rate and pressure.

Whereas traditional statistical machine learning provides well-established mathematical tools for data analysis [23][24][25], their performance is limited when processing high-dimensional data streams. To sum up, existing techniques for

monitoring pairwise stream correlations exhibit several drawbacks: In the recent work of [27], the problem of maintaining data stream statistics over sliding windows is studied, with the focus being only on single stream statistics. On the other hand, [28] introduced an extension for monitoring the statistics of multiple data streams, but the computation of correlated aggregates is limited to a small number of monitored streams. In addition, StatStream [20] has been proven a successful data stream monitoring system, which enables the computation of single- and multiple-stream statistics. However, the main drawback of this technique is the difficulty to define an appropriate "similarity" function for data streams describing dynamic phenomena with unknown prior distributions, which is normally the case in an industrial environment.

To overcome the limitations of the previous approaches, our uncertainty-aware data processing system is equipped with a computationally efficient "*similarity extraction*" module, which enables the monitoring of pairwise correlations between high-dimensional and heterogeneous sensor data streams in a fast online fashion. To this end, instead of computing all pairwise correlations between the original full-dimensional data streams, we exploit the compressibility property of the discrete Fourier transform (DFT) to concentrate the inherent energy content of a given sensor stream in the first few high-amplitude coefficients, as in [20]. Then, an appropriate similarity measure, which incorporates the estimated underlying uncertainty, is defined and applied on the associated pairs of truncated DFTs as a proxy of the corresponding correlation coefficients.

5. Analysing Hydrobionet Data Streams

The performance of the proposed system, in terms of managing the underlying data uncertainty and providing early warnings, is evaluated on two real datasets provided by ACCIONA Agua:

- ***Data from electrochemical sensors:*** This dataset consists of measurements from 29 sensors of several types (pressure, temperature, conductivity, turbidity, pH, flow, and redox). The corresponding measurements cover a period of 1 month at a sampling rate of one measurement every three minutes. Full sensor specifications (such as, sensor precision, sensitivity, and resolution), along with the corresponding measurements were provided for each individual sensor.
- ***Data from Biofilm sensors***: This dataset consists of measurements from 4 Biofilm and neighboring temperature sensors, located in different stages on the water treatment plant. The corresponding measurements cover a period of 2 months with low sampling rate due to maintenance or non-operating purposes. Due to the limitation of this data set (we have at our disposal about 100 measurement values), we present some indicatively results using our approach on this data.

The overall inherent uncertainty of the recorded sensor data is quantified over sliding windows. If not stated explicitly otherwise in the subsequent results, the sliding window size is set to 80 samples, which corresponds to a time interval of approximately 4 hours, while the step size is fixed to 1 sample that corresponds to a time-step of about 3 minutes. The expanded uncertainty is computed by fixing the coverage factor to $k = 1.96$, which is equivalent to a 95% confidence level.

Our experimental evaluation attempts to answer the following questions:

- What is the response of our system when we monitor the behavior of similar or dissimilar streams? Does the accuracy of our results increase when the sliding window size increases?

- What are the results of our method when a stream presents values out of sensor measurement range and is this a coincidence or an extreme event?

- What are the results of our approximation method to choose the top-k highly correlated streams? Does these results correspond to the results from Pearson's correlation? What are the results when the uncertainty of the measurement values is computed?

- How many streams can this approach handle simultaneously and what is the time cost?

- What is the precision of our results using as benchmark the correlation factor?

- How good are the time savings when using our approach compared with other methods (Pearson's correlation, StatStream, BRAID)?

We perform the experimental evaluation on a 2.27 GHz Intel Core i5 PC with 4 GB main memory. Our approach runs in the high performance interpreted environment of MATLAB, using the language's powerful array-based computation and the local functions for computing the DFT and Pearson's correlation.

Stability of our approach

Concerning the stability of our proposed fast pairwise stream similarity monitoring approach, as a first step we examine the response of our approach when we monitor the behavior of similar or dissimilar streams. Figure 22 (a) and Figure 23 (a), show one stream pair with similar behavior and one stream pair with dissimilar behavior respectively. In both cases, we have 1000 stream values and we calculate peak similarity and Pearson's correlation values for increasing sliding window size (20:20:100) and overlapping factor 50%.

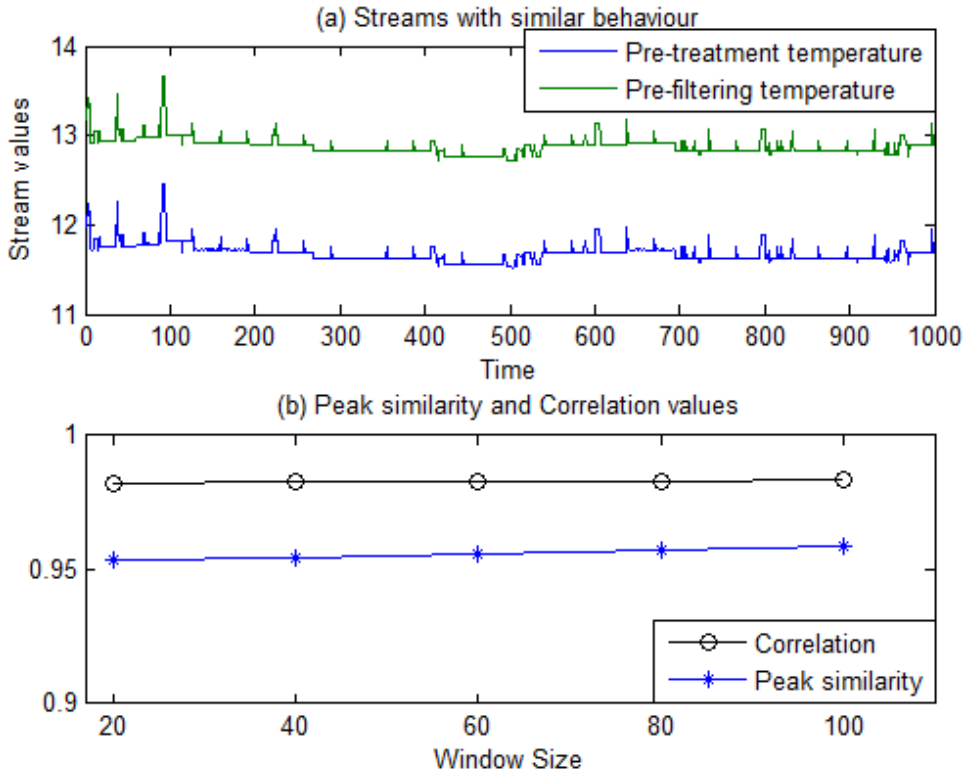


Figure 22: Comparison between peak similarity and correlation coefficient values for streams with similar behaviour, averaged over different sliding window sizes.

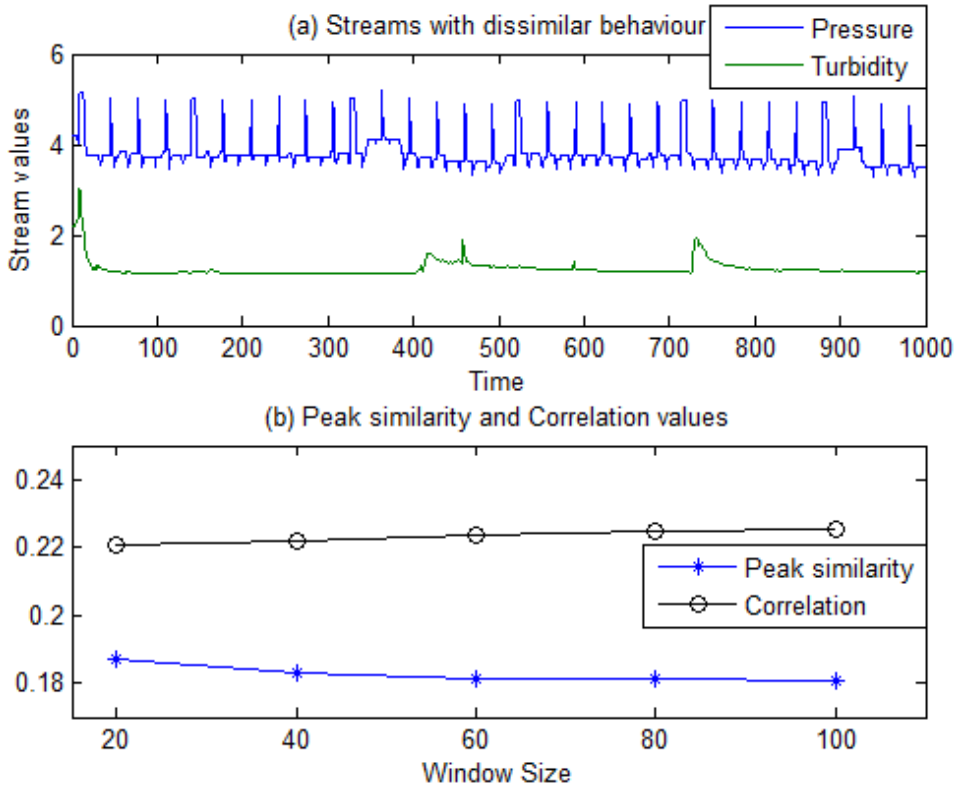


Figure 23 Comparison between peak similarity and correlation coefficient values for streams with dissimilar behaviour, averaged over different sliding window sizes.

Figure 22 (b) and Figure 23 (b) present the average Pearson's correlation and peak similarity values over different sliding window sizes. Indeed, both figures show that our approach is suitable for monitoring the stream behaviour. In case of similar streams, we have high peak similarity values (as the correlation values), in contrary with dissimilar streams that we have clearly lower values. In Table 8 we summarize these results for both cases.

Table 8: Correlation and Peak similarity (using DFT reduction technique) values, for measuring the behaviour of similar or dissimilar streams.

WS / Method	20	40	60	80	100	
Peak similarity	0,95380	0,95744	0,95750	0,95867	0,95908	**Similar streams**
Correlation	0,98170	0,98309	0,98321	0,98323	0,98326	
Peak similarity	0,22099	0,22163	0,22343	0,22487	0,22550	**Dissimilar streams**
Correlation	0,18710	0,18327	0,18158	0,18145	0,18055	

In addition, a main feature that we observe is that the peak similarity values are less than the corresponding correlation coefficient values. This deflection is expected, since our approach measures the similarity between streams by using a common similarity measure (peak similarity) in combination with the DFT reduction

technique. The size of sliding window doesn't affect our results, since we notice little changes after the third decimal digit.

A critical issue that arises by applying our method in HYDROBIONETS' project, is our method's behaviour, when (i) one stream presents some values out of measurement range and this is a coincidence and (ii) two streams detect an extreme event.

Figure 24 (a) presents two streams with similar behaviour and one stream records a value out of the sensor's measurement range. In this case, our method should guarantee that we have a coincidental event, by recognizing inverse behaviour than the expected. By assuming that the streams in Figure 24 (a) have analogous behaviour, we expect under normal conditions to be highly correlated. In case of an abnormal event (that is recorded by one stream), we expect low correlation. Actually, in Figure 24 (b) (or in Table 9) we can see the similarity values for the above case from 4 different methods. The compared streams have length 1500 stream values, and the results are the averages of the computations in different sliding window sizes (from 500 to 100 with step size 100) with overlapping factor 50%.

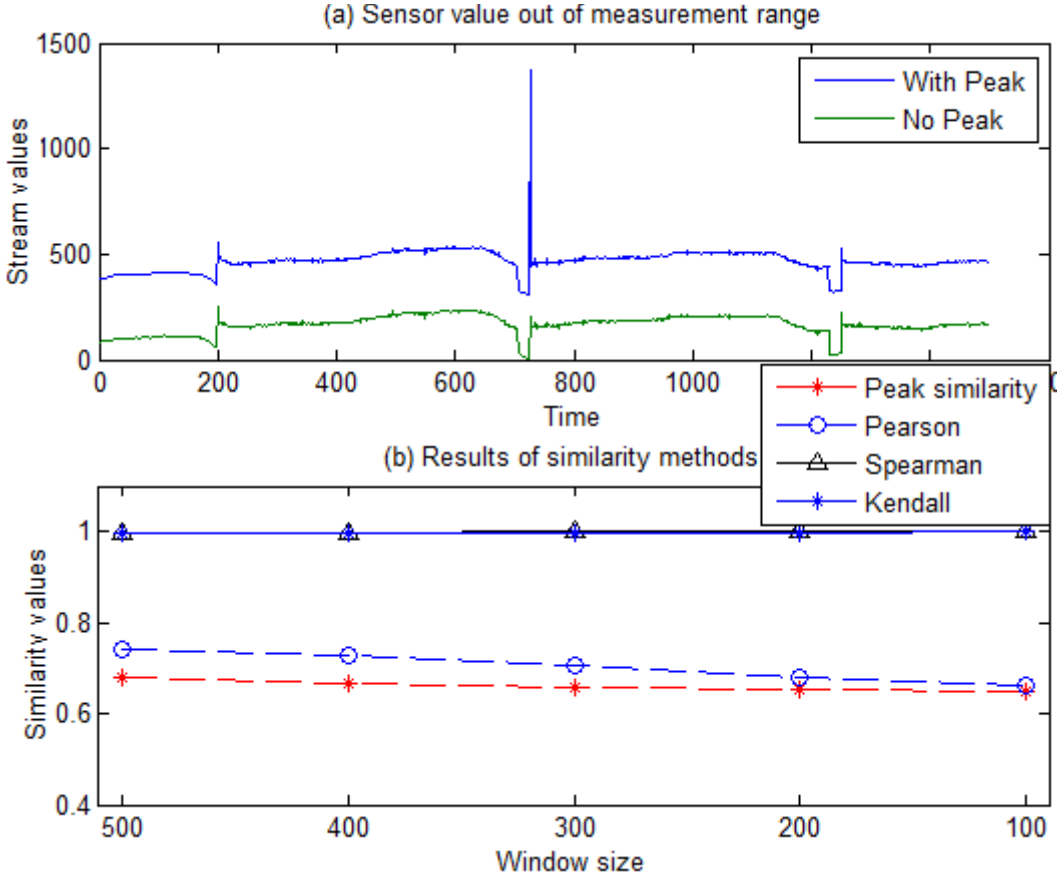


Figure 24: Averaged similarity values for one pair of data streams, with a value to be out of the sensor's measurement range, as the window size decreases.

These four different methods we used for monitoring the similarity between data streams in Figure 24 (a), are the Pearson's, Spearman's, Kendall's correlations and our Peak similarity approach. Both Spearman's and Kendall's similarity results present this pair of streams highly correlated, since both methods are not sensitive to

outliers. This behaviour is not suitable for our monitoring in Hydrobionets project. Instead, Pearson's correlation and Peak similarity approach recognize the existence of this outlier, since their results indicate low correlation between these two streams. In this case, we can realise that there isn't analogous behaviour between the two streams and we can ignore the outlier of the first stream. It worths to notice, that as the sliding window size decreases, the corresponding results of our approach and of Pearson's correlation also decreases. This behaviour is the expected, since the influence of the outliers is more as the size of the sliding window that includes the outlier, decreases. That is the reason the similarity values are smaller as the window size decreases.

Table 9: Similarity values from four different methods in case of the existence of an outlier.

WS / Method	500	400	300	200	100
Peak similarity	0,67788	0,66784	0,65703	0,65118	0,64986
Correlation	0,74104	0,72984	0,70803	0,68037	0,65316
Spearman's rho	0,99790	0,99857	0,99881	0,99959	0,99790
Kendall's tau	0,99658	0,99684	0,99667	0,99765	0,99658

Figure 25 depicts the output of the COL extreme event detector when a temperature sensor records an extreme event. As we can see, this method identifies as extreme events only those measurements which are strictly higher than 17. The temperature and pressure sensor streams have analogous behaviour (as defined in pattern P6 in Table 5), so both of them are increasing in the case of the extreme event. We expect that these two streams would be highly correlated.

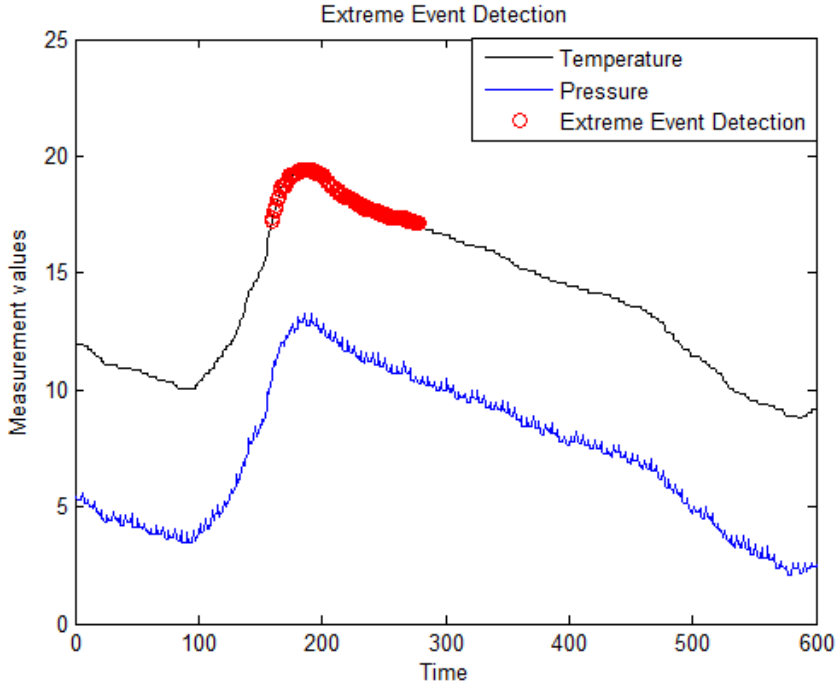


Figure 25: Extreme event detection from the COL method, when we monitor the temperature and pressure data streams.

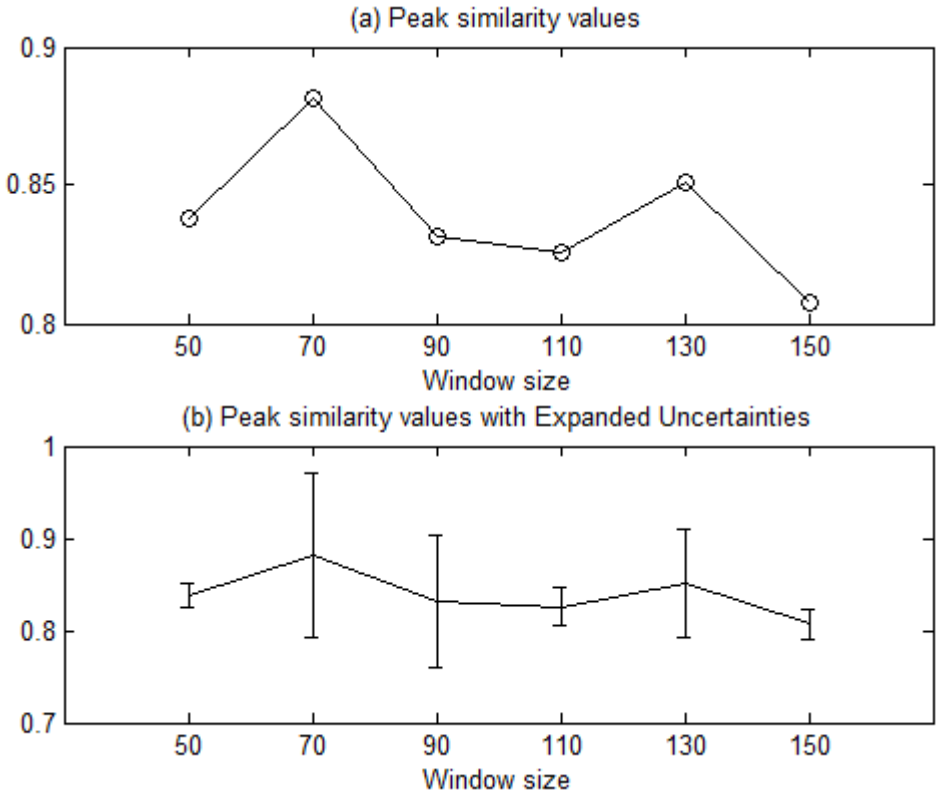


Figure 26: Averaged peak similarity values over different sliding window sizes (a) comparing original sensor measurements without incorporating uncertainty and (b) results with confidence intervals by incorporating the estimated expanded uncertainty.

Figure 26 presents the values of our approach, when we monitor a pair of streams which detects an extreme event. The streams we compare have length of 600 stream values. The plotted values are the corresponding averages of our approach over different sliding windows sizes. The window size ranges from 50 to 150, with step size 20 and overlapping factor 50% of the window size. Furthermore, in Figure 26 (b) we can see the averaged values when accounting for the underlying data uncertainty (with k=1.96). In both cases the response of our approach indicates that this pair of streams is highly correlated throughout the monitoring period. By this way we can guarantee the existence of the extreme event detected by COL method.

Performance of our pairwise stream similarity approach

A critical issue that arises from our approximation method in HYDROBIONETS' project is the valid match for the top-k highly correlated streams with regard to Pearson's correlation results. In Figure 27 (b) we can see the peak similarity and Pearson's correlation values between a reference pressure stream and other types of streams (the labels Prx, Tx, FFx, FLx, PHx, Cx, BFx, TRx denote pressure, temperature, feed flow, filtrate, flow, Ph, conductivity and turbidity sensor streams respectively). We computed the average of the similarity values between streams with length 2000 measurement values, with sliding window size 100 measurement values and with overlapping factor 50% of the window size. Taking the provision of highly correlated streams in both cases (for peak similarity and Pearson's correlation

methods) is the same: Fl_2, Fl_1, T_1, PR_1, T_2, TR_1, C_1, FF_2, T_3, PH_2, PH_1, TR_2, FF_1, BF_1, C_2.

An interesting point that we have to examine is whether the computation of uncertainty in our data streams affects these results. In Figure 27 (a) we can see the similarity values from peak similarity and Pearson correlation methods for the above streams after we have calculated the uncertainty in our data streams. The results about the provision of highly correlated streams are exactly the same: Fl_2, Fl_1, T_1, PR_1, T_2, TR_1, C_1, FF_2, T_3, PH_2, PH_1, TR_2, FF_1, BF_1, C_2. To conclude, the uncertainty estimation in our data streams, doesn't affect the final results for finding the top-*k* highly correlated data streams.

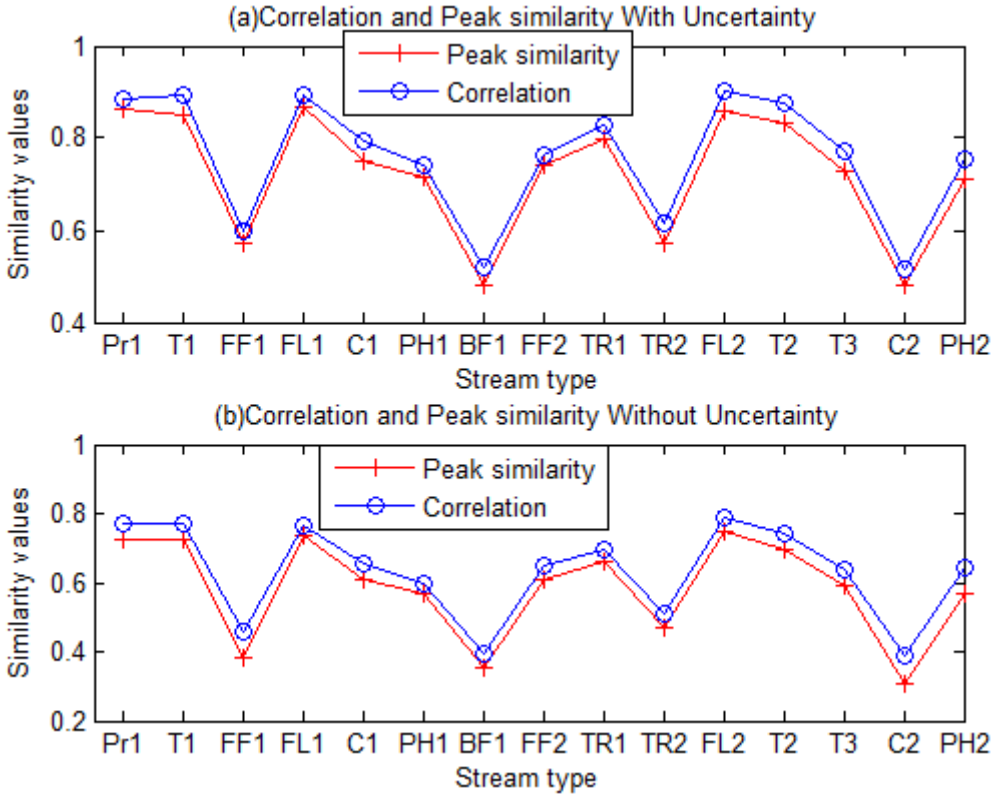


Figure 27:Peak similarity and Pearson's correlation values between one reference stream (pressure stream) and 15 other types of streams (a) including the uncertainty of data (b) without the uncertainty computation.

Our proposed approach computes the streams similarity at the end of each sliding window. We can increase the number of compared streams at the cost of increasing the delay in reporting results. Figure 28 shows the execution time by finding the pairs of streams which are related above a user defined threshold with a referenced stream, as the number of comparing streams increases. For this experiment the data streams are generated using the random walk pattern. For streams *s*,

$$s_i = 100 + \sum_{j=1}^{i} (u_j - 0.5),$$

where i=1,2,…n (n=stream length) and u_j is a set of uniform random real numbers in [0.1]. The streams have length 1000 stream values and the similarity results are the

averages of similarity computations in sliding windows with size 100 and overlapping factor 50 stream values.

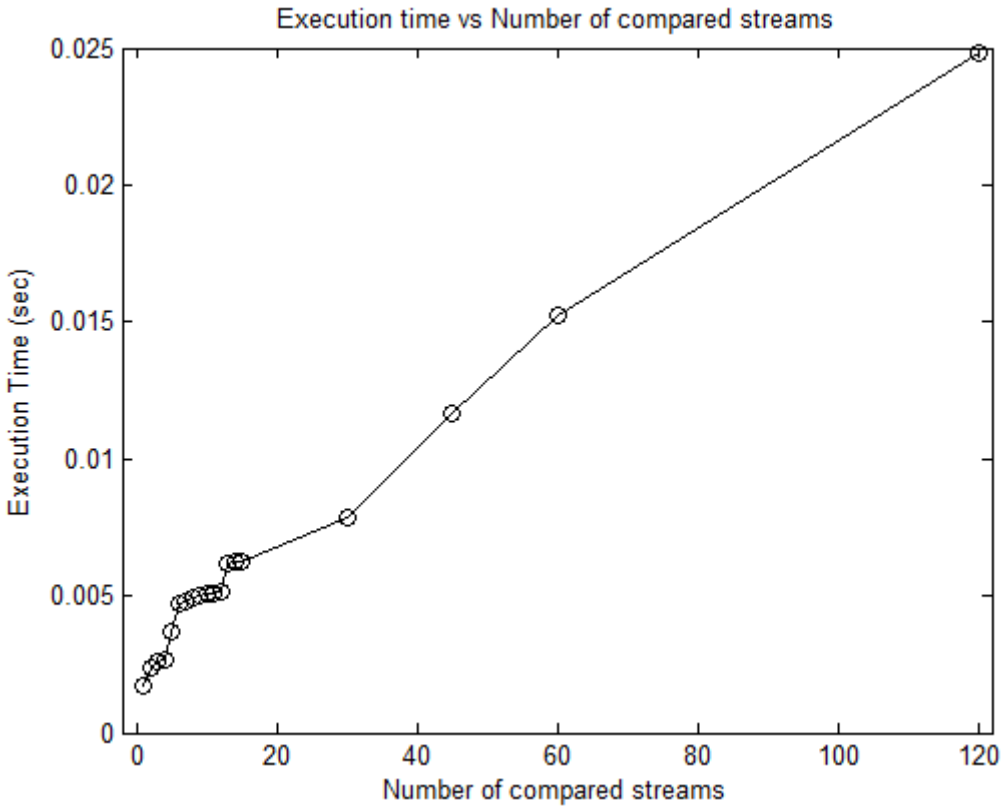


Figure 28: The number of streams that our proposed approach with peak similarity method can handle in an online fashion.

Comparing with other methods

As mentioned before, the computational complexity, and subsequently the execution time of our results is an important factor which affects the overall performance of our proposed uncertainty-aware data management system. To this end, we compare the performance of our proposed approach, in terms of execution times for increasing stream lengths against the typical Pearson's correlation coefficient and two other state-of-the-art methods, namely, BRAID [19] and StatStream [20]. BRAID can handle data streams of semi-finite length, incrementally, quickly, and can estimate lag correlations with little error. On the other hand, as mentioned before, StatStream resembles more our approach, by finding high correlations among sensor pairs based on DFTs and a three-level time interval hierarchy.

For the BRAID algorithm we set the correlation lag to be equal to zero. For the StatStream algorithm, a simple hash function is used based on the mean value of each stream. Keeping the integer part of the mean values, the streams are mapped to appropriate cells in a grid structure. Doing so, only the correlations between neighbouring cells are computed.

Figure 29 compares the execution times of our proposed method with the other three alternatives (Pearson's correlation, BRAID and StatStream), as a function of the stream length. The similarity values are computed over one pair of streams with

different stream length. The results reveal a significant improvement in execution time achieved by our method, which is more prominent for higher stream lengths. We observe that the execution time of our method remains almost constant over the whole selected range of stream lengths, in contrast to the naïve and BRAID methods, whose execution times increase rapidly as the stream length increases.

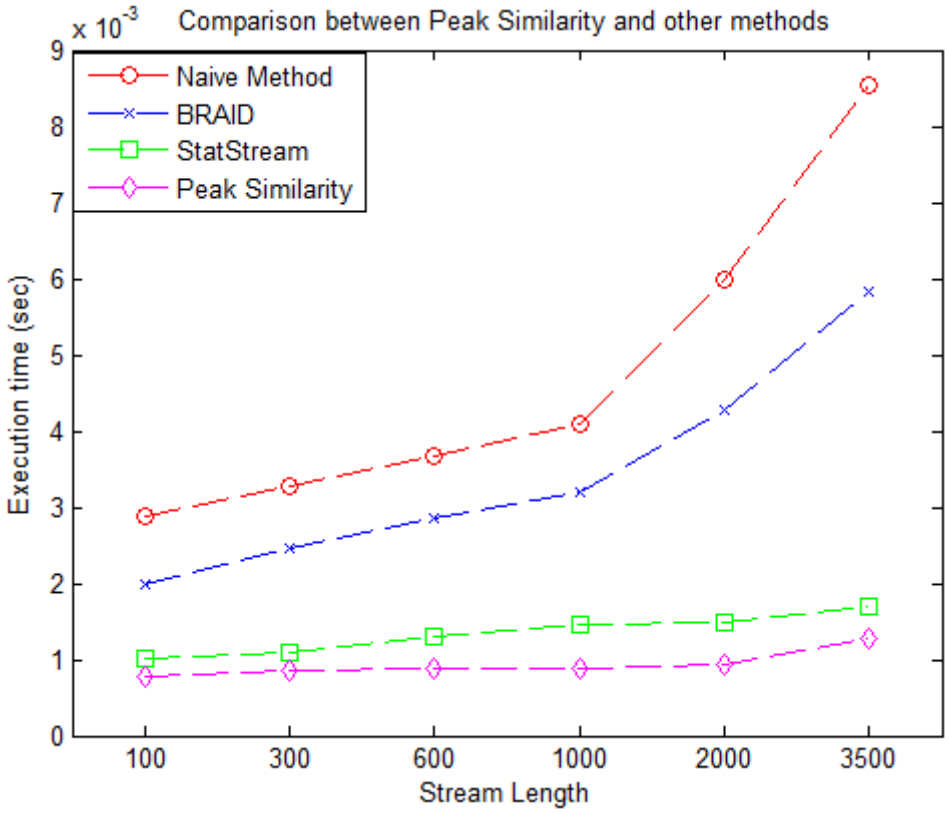


Figure 29: Comparison of execution times, as a function of the stream length for four methods: a)Peak similarity (our proposed), b) StatStream, c)BRAID and d) Naïve Method (correlation coefficient)

Our proposed approach and StatStream are performed with low execution time because the similarity values from both methods are estimated with few stream values due to DFT approximation. BRAID algorithm is characterized by gradual increase for increasing stream length, since it employs all the values of the recorded streams. The increased execution time of StatStream, compared to our approach is due to the hash function, which involves more computations for the stream mapping. We expect though that the performance of StatStream could be enhanced, by designing a more efficient hash function.

There are different similarity measures as presented in Appendix B, and one might wonder why we don't use one of them. We computed the precision results, by applying each different similarity measure to our approximation method. We use as true reference set the results from Pearson's correlation coefficient. Precision is the percentage of the similar pairs of streams above a pre-defined threshold identified by the different similarity measures, which are truly similar (we compare them with the true reference set). This experiment is performed between one reference stream and 300 different data streams (they were generated using the random walk pattern) with 1500 stream values length. The similarity values are computed over sliding windows

with size 100 and overlapping factor 50 stream values. For each 500 stream values we compute the averages of the recorded similarity values and we report the highly correlated ($\varepsilon_{th}>0.75$) stream pairs between the reference and the other streams. Based on these results, we compute the precision measure. In Table 10, we can see the precision results, by applying the different similarity measures referred to Appendix B- *Similarity measures* in our approximation technique.

Table 10: Precision results from different similarity measures

Common similarity	0.763	0.756	0.758
Mean similarity	0.773	0.766	0.774
Root mean similarity	0.791	0.788	0.793
Peak similarity	**0.961**	**0.954**	**0.937**

Our experimental evaluation is completed with the applying of our proposed approximation method for monitoring the data streams behaviour, between Biofilm and temperature sensor data streams from the second real data set provided by ACCIONA Agua. In Figure 30 we can see the behaviour of Biofilm-Temperature sensor data streams pairs for the few available samples of this data set.

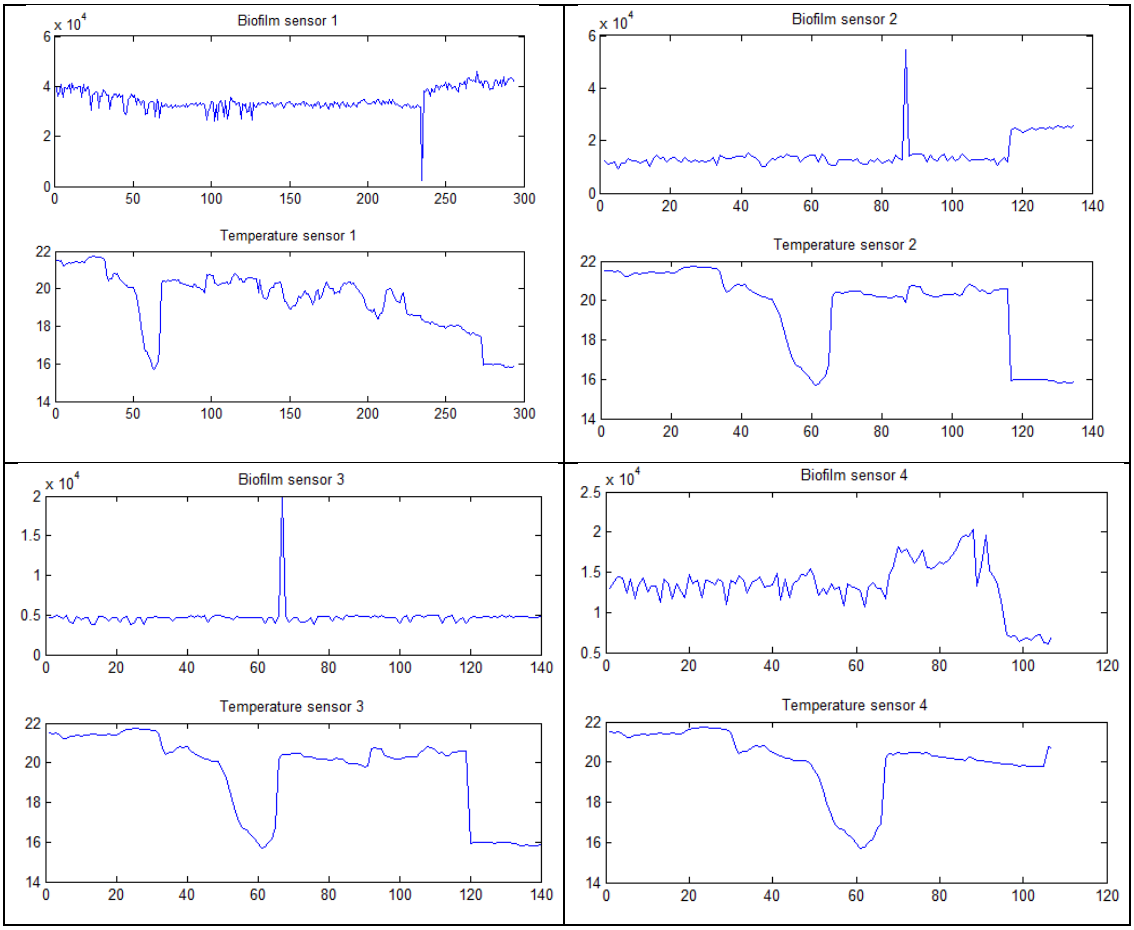


Figure 30: Behaviour between the pairs of Biofilm and Temperature sensor data streams.

Table 11 includes the similarity values and the corresponding errors for the behaviour of the four pairs of Biofilm-temperature sensor data streams. For this experiment we computed the average of the similarity results between the corresponding pairs of Biofilm and temperature sensors, concerning a sliding window with size 30 and overlapping factor 20 stream values. Unfortunately, we cannot conclude to valid inferences for applying our proposed similarity technique to bacteriological data streams. The available data set is limited and we cannot examine interesting cases.

Table 11: Similarity values for four pairs of Biofilm and Temperature sensor data streams with the corresponding error

	Pair 1	**Pair 2**	**Pair 3**	**Pair 4**
Correlation	0.56485	0.56846	0.51653	0.62063
Peak similarity	0.52092	0.53632	0.47652	0.59708
Error	0.04392	0.032134	0.04	0.02354

Furthermore, our future work includes the performance study of our system, by applying our approximation similarity approach to bacteriological data streams, since these streams present completely different distributions.

6. Conclusions and future work

Designing efficient data management systems capable of accounting for the inherent data uncertainty and providing early warning notifications is a challenging task in large-scale industrial infrastructures. The major issue of this thesis includes the proposition of an integrated uncertainty-aware data management system, which also supports timely detection of extreme events and fast online monitoring of pairwise sensor similarities in order to guarantee the validity of the detected extreme events. Comparison with state-of-the-art stream processing techniques revealed an improved performance of our proposed framework in terms of achieving accurate detection of extreme events, in conjunction with extraction of highly similar (correlated) pairs of possibly heterogeneous sensors, with significantly decreased execution times.

As a final outcome, we envisage to provide a set of data services to manipulate sensor measurements in large-scale industrial infrastructures, as well as to identify appropriate monitoring tools for the characterization of the generated data quality in real time. As a further extension, we will focus on the design of an automatic rule for the time-varying adaptation of the threshold $\varepsilon_{th,new}$, as well as the design of novel similarity measures of even lower computational complexity, while still approximating accurately the behavior of the correlation coefficient. In our future research we intend to concentrate on the application of the uncertainty propagation rules for quantifying the uncertainty between sensor data streams produced by heterogeneous data sources. Future work will involve the extension of peak similarity measure in a multiscale framework by employing more power transforms than the Discrete Fourier Transform. A characteristic example is the Wavelet Transform to extract the inherent frequency content of sensor streams. The next stage of our research includes the performance study of our system by applying our approximation similarity approach to bacteriological data streams, since these streams present completely different distributions.

References

[1] Buer T. and Cumin J. (2010). “MBR module design and operation.” Desalination 250(3): 1073-1077.

[2] W.M. Lu, Y.P. Huang, K.J. Hwang (1998). “Dynamic analysis of constant rate filtration data”, Vol. 31: 969-976

[3] M. Balazinska, A. Deshpande, M. Franklin, P. Gibbons, J. Gray, S. Nath, M.Hansen, M. Liebhold, A. Szalay, V. Tao. “Data Management in the Worldwide Sensor Web”. PERVASIVE computing April–June 2007.

[4] C. Aggarwal, ed. “Managing and Mining Uncertain Data”. Springer, 2009.

[5] M. Magnani, D. Montesi, “A Survey on Uncertainty Management in Data Integration”. Journal of Data and Information Quality (JDIQ), Volume 2 Issue 1, July 2010.

[6] P.Agrawal, A. Das Sarma, J.Ullman, J.M.Hellerstein and W.Hong, “Foundations of Uncertain Data Integration”. In Proc of VLDB 2010

[7] Todd J. Green. Models for incomplete and probabilistic information. In Charu Aggarwal, editor, Managing and Mining Uncertain Data. Springer, 2009

[8] T.Imielinski, W.Lipski. “Incomplete Information in Relational Databases”. J. ACM 31(4): 761-791 (1984)

[9] J. Cheney, L. Chiticariu, and W. C. Tan, "Provenance in databases: Why, where and how," Foundations and Trends in Databases, vol. 1, no. 4, 2009

[10] T.J. Green, G. Karvounarakis, and V. Tannen, "Provenance Semirings," in Proc of PODS, 2007.

[11] T.J. Green, G Karvounarakis, Z. Ives, and V. Tannen, "Update Exchange with Mappings and Provenance," in VLDB, 2007.

[12] G. Karvounarakis, Z. Ives, and V. Tannen, "Querying Data Provenance," in SIGMOD 2010.

[13] European Federation of National Associations of Measurement (Testing and Analytical Laboratories), “Guide to the evaluation of measurement uncertainty for quantitative test results,” Technical Report No. 1/2006, August 2006.

[14] B.Canton, “Mathematics of Data Management”, McGraw-Hill, 2002

[15] S.Guha, A. Meyerson, N.Mishra, R.Motwani and L.O’Callaghan. “Clustering data streams: Theory and practice”. IEEE TKDE, 15(3): 515:528, 2003.

[16] Y.Zhu and D. Shasha, “Statistical monitoring of thousands of data streams in real time”. VLDB, pages 358-369, Aug.2002

[17] B.-K.Yi, N.Sotiropoulos, T. Johnson, H.Jagadish, C.Faloutsos and A.Biliris. “Online data mining for co-evolving time sequences”. ICDE, pages 13-22,2000

[18] S.Papadimitriou, A. Brockwell and C.Faloutsos, “Adaptive, hands-off stream mining”. VLDB, pages 560-571, Sept. 2003

[19] Sakurai Y., Papadimitriou S., Faloutsos C., “BRAID: Stream mining through group lag correlations”. In SIGMOD (2005)

[20] Y.Zhu and D.Shasha, “StatStream: Statistical monitoring of thousands of data streams in real time”, VLDB, pages 358-369, Aug.2002

[21] Q.Xie, S.Shang, B.Yuan, C. Pang, X. Zhang, "Local correlation detection with linearity enhancement in streaming data", CIKM 2013, pages 309-318.
[22] M.Dallachiesa, B.Nushi, K.Mirylenka and T.Palpanas, "Uncertain Time-Series Similarity: Return to the basics", in Proceedings of the VLDB Endowment, 2012, pages 1662-1673.
[23] Tran, T. T., Peng, L., Li, B., Diao, Y., & Liu, A., "PODS: a new model and processing algorithms for uncertain data streams." In Proceedings of the 2010 ACM SIGMOD International Conference on Management of data (pp. 159-170).
[24] Yeh, M. Y., Wu, K. L., Yu, P. S., & Chen, M. S. , "PROUD: a probabilistic approach to processing similarity queries over uncertain data streams". In Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology (pp. 684-695). ACM.
[25] Tran, T. T., Peng, L., Diao, Y., McGregor, A., & Liu, A., "CLARO: modelling and processing uncertain data streams." The VLDB Journal—The International Journal on Very Large Data Bases, 21(5), 651-676.
[26] J.Aβfalg, H-P. Kriegel, P. Kröger and M.Renz, "Probabilistic similarity search for uncertain time series", in SSDBM, pages 435-443, 2009.
[27] Datar, M., Gionis, A., Indyk, P., & Motwani, R. "Maintaining stream statistics over sliding windows". SIAM Journal on Computing, 2002, pages 1794-1813.
[28] Gehrke, J., Korn, F., & Srivastava, D. "On computing correlated aggregates over continual data streams" , 2001, In *ACM SIGMOD Record*, pages 13-24, ACM.]
[29] D. Barbara, W. DuMouchel, C. Faloutsos, P. J. Haas, J. M. Hellerstein, Y. E. Ioannidis, H. V. Jagadish, T. Johnson, R. T. Ng, V. Poosala, K. A. Ross, and K. C. Sevcik. The new jersey data reduction report. *Data Engineering Bulletin*, 1997.
[30] Rafiei, D., & Mendelzon, A, "Similarity-based queries for time series data." In *ACM SIGMOD Record*, 1997, 13-25, ACM.
[31] A.Seliniotaki, G.Tzagkarakis, V.Christophides, P.Tsakalides, "Stream Correlation Monitoring for Uncertainty-Aware Data Processing Systems", In IEEE International Conference on Information, Intelligence, Systems and Applications, Greece, 2014
[32] C.Aggrawal, "Managing and Mining Uncertain Data", Springer-Verlag New York Inc, 2009
[33] A. Klein and W. Lehner, "Representing data quality for streaming and static data," in Proc. *International Workshop on Ambient Intelligence, Media and Sensing (AIMS)*, 2007
[34] T. Tran, A. McGregor, Y. Diao, L. Peng, and A. Liu, "Conditioning and aggregating uncertain data streams: Going beyond expectations," in Proc. *36th Intl. Conf. on Very Large Data Bases*, Sept. 13-17, 2010, Singapore.
[35] A. Klein and W. Lehner, "Representing data quality in sensor data streaming environments," *J. Data and Information Quality*, vol. 1, no. 2, pp. 1-28, 2009.
[36] A. Klein, "Incorporating quality aspects in sensor data streams," in Proc. *ACM 1st PhD Workshop in CIKM (PIKM)*, New York, 2007

[37] X. Lian and L. Chen, "Efficient join processing on uncertain data streams," in Proc. *18th ACM Conf. Inform. And Knowledge Management*, 2009.

[38] S. Coles, "*An introduction to statistical modeling of extreme values*," Springer, 2001.

[39] J. Beirlant *et al.*, "*Statistics of extremes: Theory and applications*," Wiley, New York, 2004.

[40] J.-P. Calbimonte, H.Jeung, O.Corcho, "*Querying semantically enriched sensor observations*", Heraklion, Greece, May 29-June 2, 2011

[41] McNeil, A., & Saladin, T (1997), *"The peaks over thresholds method for estimating high quantiles of loss distributions".* In Proceedings of 28th International ASTIN Colloquium.

[42] C.Cassisi, P.Montalto, M.Aliotta, A.Cannata, A.Pulirenti, *"Similarity measures and Dimensionality Reduction Techniques for Time Series Data Mining"*, chapter 3 of "Advances in Data Mining Knowledge Discovery and Applications".

Appendix A- *Correlation and Euclidean distance*

The purpose of a measure of similarity is to compare two lists of numbers (i.e. vectors), and compute a single number which evaluates their similarity. Most measures were developed in the context of comparing pairs of variables (such as income or attitude toward abortion) across cases (such as respondents in a survey). In other words, the objective is to determine to what extent two variables co-vary, which is to say, have the same values for the same cases.

One problem with comparing two variables is that they may not be measured on the same scale. The general principle is that a measure of similarity should be invariant under admissible data transformations, which is to say changes in scale. Thus, a measure designed for interval data, such as the familiar Pearson correlation coefficient, automatically disregards differences in variables that can be attributed to differences in scale. All valid interval scales, applied to the same objects, can translated into each other by a linear transformation. This means that to see how similar two interval variables are, we must first do away with differences in scale by either standardizing the data (this is what the correlation coefficient does), or by trying to find the constants *m* and *b* such that the transformed variable mX+b is as similar as possible to Y, and then reporting that similarity. Likewise, a measure designed for ordinal data should respond only to differences in the rank ordering, not to the absolute size of scores. A measure designed for ratio data should control for differences due to a multiplicative factor.

Euclidean Distance

The basis of many measures of similarity and dissimilarity is Euclidean distance. The distance between vectors X and Y is defined as follows:

$$d(x,y) = \sqrt{\sum_i^n (x_i - y_i)^2}$$

In other words, Euclidean distance is the square root of the sum of squared differences between corresponding elements of the two vectors. Note that the formula treats the values of X and Y seriously: no adjustment is made for differences in scale. Euclidean distance is only appropriate for data measured on the same scale. As we can see in the following (in the section on correlation), the correlation coefficient is (inversely) related to the Euclidean distance between standardized versions of the data.

Euclidean distance can be re-expressed in terms of the differences in level, scatter and shape of the variables.

$$d_{xy}^2 = \sum_i (x_i - y_i)^2 = \sum_i x_i^2 + \sum_i y_i^2 - 2\sum_i x_i y_i \quad \text{(Equation 1)}$$

The scatter or standard deviation of a variable x can be written as

$$s_x = \frac{\sum_i x_i^2}{n} - m_x^2$$

So,

$$n(s_x + m_x^2) = \sum_i x_i^2$$

Substituting this in the equation for distance squared, we get

$$d_{xy}^2 = n(s_x + m_x^2) + n(s_y + m_y^2) - 2\sum_i x_i y_i \quad \text{(Equation 2)}$$

The correlation between x and y can be written as

$$r_{xy} = \frac{\frac{\sum_i x_i y_i}{n} - m_x m_y}{s_x s_y}$$

Therefore,

$$n(s_x s_y r_{xy} + m_x m_y) = \sum_i x_i y_i$$

Substituting that into Equation 2, we get

$$d_{xy}^2 = n(s_x + m_x^2) + n(s_y + m_y^2) - 2n(s_x s_y r_{xy} + m_x m_y)$$

$$\frac{d_{xy}^2}{n} = s_x + m_x^2 + s_y + m_y^2 - 2s_x s_y r_{xy} - 2m_x m_y$$

$$\frac{d_{xy}^2}{n} = (m_x^2 + m_y^2 - 2m_x m_y) + (s_x + s_y) - 2s_x s_y r_{xy}$$

$$\frac{d_{xy}^2}{n} = (m_x - m_y)^2 + (s_x + s_y) - 2s_x s_y r_{xy}$$

So the average squared Euclidean distance is a function of the means, standard deviations and correlation between the variables.

Correlation

The correlation between vectors X and Y are defined as follows:

$$r(X,Y) = \frac{\frac{1}{n}\sum_i x_i y_i - \mu_X \mu_Y}{\sigma_X \sigma_Y}$$

where μ_X and μ_Y are the means of X and Y respectively, and σ_X and σ_Y are the standard deviations of X and Y. The numerator of the equation is called the

covariance of X and Y, and is the difference between the mean of the product of X and Y subtracted from the product of the means. Note that if X and Y are standardized, they will each have a mean of 0 and a standard deviation of 1, so the formula reduces to:

$$r(X^*,Y^*) = \frac{1}{n}\sum_i x_i y_i$$

Whereas Euclidean distance was the sum of squared differences, correlation is basically the average product. There is a further relationship between the two. If we expand the formula for Euclidean distance, we get this:

$$d(x,y) = \sqrt{\sum_i^n (x_i - y_i)^2} = \sqrt{\sum_i x_i^2 + \sum_i y_i^2 - 2\sum_i x_i y_i}$$

But if X and Y are standardized, the sums $\sum x^2$ and $\sum y^2$ are both equal to *n*. That leaves $\sum xy$ as the only non-constant term, just as it was in the reduced formula for the correlation coefficient. Thus, for standardized data, we can write the correlation between X and Y in terms of the squared distance between them:

$$r(X^*,Y^*) = 1 - \frac{d^2(X^*,Y^*)}{2n}$$

Appendix B- *Similarity measures*

As it was mentioned in [42], the similarity between two sequences of the same length can be calculated from different similarity measures. Let two data streams, $X = x_1, x_2, ..x_n$ and $Y = y_1, y_2, ..y_n$ some similarity measures are:

- Common similarity: $numSim(X,Y) = 1 - \frac{|x_i - y_i|}{|x_i| - |y_i|}$
- Mean similarity: $tsim(X,Y) = \frac{1}{n}\sum_{i=1}^{n} numSim(x_i, y_i)$
- Root mean square similarity: $rtsim(X,Y) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} numSim(x_i, y_i)^2}$

These measures provide values in range [0,1]. The upper boundary indicates that the vectors are exactly the same and the 0 value indicates the independence.