Trust Model for the Internet of Things

Tesis doctoral presentada por Francisco Javier Nieto de Santos

dentro del Programa de Doctorado Ingeniería para la Sociedad de la Información y Desarrollo Sostenible

Dirigida por:

Dr. Diego López de Ipiña

Y Dr. Unai Aguilera

Bilbao, Enero de 2023

Trust Model for the Internet of Things

Tesis doctoral presentada por Francisco Javier Nieto de Santos

dentro del Programa de Doctorado Ingeniería para la Sociedad de la Información y Desarrollo Sostenible

Dirigida por:

Dr. Diego López de Ipiña

Y Dr. Unai Aguilera

El doctorando

Los directores

Bilbao, Enero de 2023

Trust Model for the Internet of Things

Author: Francisco Javier Nieto de Santos

Advisor: Dr. Diego López-de-Ipiña

Advisor: Dr. Unai Aguilera

Text printed in Bilbao

First edition, January 2023

A mi familia, amigos y todos aquellos que me apoyaron en el camino y se acordaron de preguntarme qué tal iba con la tesis.

Abstract

The number of devices used in multiple fields is increasing more and more, in such a way that the Internet of Things is becoming one of the main sources of information. But the datasets produced by sensors, and consumed by applications and data scientists, are not perfect. They may have errors produced because of multiple issues (from faulty devices to vandalization of sensors). Therefore, it is necessary to put in place mechanisms that support the understanding of such datasets and that can determine whether there are problems in the data produced. Such mechanisms can provide a perspective of how much the sensor that produced the data can be trusted.

This work proposes to analyse the behaviour of a sensor based on the data it has generated, looking at two main aspects: how the data produced varies and to what extent the data contains faulty values that might be problematic. The proposed approach is focused mainly on the combination of some solutions based on statistics, in such a way that the outcome will be as much generic as possible.

This dissertation introduces all the heterogeneous data sources that have been used (and their particularities) and it performs a deep analysis of several statistical aspects for many types of sensors. It addresses the probability distribution of the data, as well as some statistical tests for studying if the data varies like white noise. Then, it looks at the statistical tests that analyse the presence of outliers and homogeneity, as they can be linked to certain types of errors in sensors.

Such analyses are used to define a set of solutions that can improve the transformation of the data for the tests and the analysis of the results of some statistics. All these results are combined, defining three models that analyse the variation of the data, the presence of outliers and the potential correlation with other sensors, giving as a result the implementation of a mechanism for sensors' data understanding and trust evaluation.

The evaluation of such solutions has shown that they are able to perform well with datasets and types of sensors not studied before, with an accuracy of around 0.8 and F0.5-score higher than 0.7 in most of the cases.

Resumen

El número de dispositivos que se utilizan en muchos campos no han parado de crecer durante los últimos años en el contexto del Internet de las Cosas, que se ha convertido en una de las primeras fuentes de generación de datos. Pero los datos que producen los sensores (y que consumen tanto aplicaciones como científicos de datos) no son perfectos y pueden contener errores producidos por múltiples causas (desde dispositivos estropeados hasta sensores vandalizados). Por lo tanto, es necesario implementar mecanismos que ayuden a comprender los datos generados, y que puedan indicar si existen problemas en los mismos. Esos mecanismos pueden dar una perspectiva sobre cuánto puede confiarse en el sensor que generó los datos.

Este trabajo propone analizar el comportamiento de los sensores basándose únicamente en los datos generados y mirando a dos aspectos principalmente: cómo varían los datos producidos y hasta qué punto los datos contienen valores erróneos que pudieran ser problemáticos. La teoría propuesta se enfoca, principalmente, en la combinación de soluciones basadas en la estadística, de forma que sea lo más genérica posible.

Esta disertación describe todas las fuentes de datos heterogéneas que se han utilizado, así como sus particularidades, y realiza un análisis detallado de varios aspectos estadísticas para un gran número de sensores. También analiza la distribución de probabilidad de los datos, así como algunas pruebas estadísticas para estudiar si dichos datos varían aleatoriamente. Luego se enfoca en otras pruebas estadísticas que estudian la presencia de valores anómalos y la homogeneidad de la serie temporal, ya que se pueden relacionar con diferentes tipos de errores encontrados en sensores.

Toda esta información se utiliza para definir una serie de soluciones que puedan mejorar la transformación de datos para las pruebas estadísticas y para analizar los estadísticos que producen durante el cálculo. Todos estos resultados se combinan en tres modelos que analizan la variación de los datos, la presencia de valores anómalos y la correlación con otros sensores, dando como resultado la implementación de un mecanismo orientado a la comprensión de datos de sensores y la evaluación del nivel de confianza en los mismos.

La evaluación de estas soluciones ha demostrado que funcionan bien con muestras de datos no utilizadas antes, incluso de tipos de sensores no utilizados anteriormente, con niveles de precisión alrededor de 0.8 y F0.5-score de más de 0.7 en la gran mayoría de casos.

Acknowledgements

This work represents research that took a lot of time to complete, and it would not have been possible without the support of many people that accompanied me in this trip. I want to thank my wife Olatz, for all her support during these years and her valuable knowledge about databases management. Special thanks to my kids (Ane and Xabi) for their patience and all those moments that made the way a bit more interesting (like those days 'playing' with an Arduino together). I do not forget my parents, grandmother, sister and friends, who asked me periodically about the thesis and encouraged me to go on. Thanks to my directors as well, for their guidance during the process and the support to complete this work (and its associated bureaucracy).

I also want to thank the colleagues of the projects in which some of this work was carried out, and of those projects that set the base of this dissertation (mainly to those from BETaaS, EUXDAT and HiDALGO), because of the interesting discussions we had and for their support.

Special thanks to all those colleagues and entities that provided me with the datasets that I needed to do the analysis and the evaluation of the solutions proposed. This includes Ports of Spain, Kunak, Meteoblue and the people from Széchenyi István University (SZE).

It was a pleasure to walk side by side with all of you these years.

Francisco Javier Nieto

Bilbao, January 2023

Contents

LIST OF FIGURES

LIST OF TABLES

Acronyms

ANNs	Artificial Neural Networks
ARIMA	Autoregressive Integrated Moving Average
AUC	Area under the ROC Curve
CAGR	Compound Annual Growth Rate
CEP	Complex Event Processing
CNN	Convolutional Neural Network
COV	Coefficient of Variation
CRISP-DM	Cross Industry Standard Process for Data Mining
CSV	Comma-Separated Values
DSRM	Design Science Research Methodology
DT	Decision Tree
ESD	Extreme Studentized Deviate
FPR	False Positive Rate
FTP	File Transfer Protocol
GA	Genetic Algorithms
HPC	High Performance Computing
IoT	Internet of Things
IP	Internet Protocol
IQR	Interquartile Range
kNN	K-Nearest-Neighbour
LR	Logistic Regression
LSTM	Long Short-Term Memory
MAC	Media Access Control
ML	Machine Learning
NaN	Not a Number
PCA	Principal Component Analysis
PM	Particulate Matter
POSIX	Portable Operating System Interface for uniX

Q-Q	Quantile-Quantile
RAE	Relative Absolute Error
RF	Random Forest
RMSE	Root Mean Square Error
ROC	Receiver Operating Characteristic
RQ	Research Question
SCADA	Supervisory Control And Data Acquisition
SMART	Simple Multi-Attribute Rating Technique
SNHT	Standard Normal Homogeneity Test
SQL	Structured Query Language
SVMs	Support Vector Machines
WSNs	Wireless Sensor Networks
YASA	Yet Another Segmentation Algorithm

1. INTRODUCTION

The scientist is not a person who gives the right answers, he is one who asks the right questions.

Claude Lévi-Strauss

The Internet of Things (IoT) has been growing during the last years, both from the economical and the technological aspect, but there are still some areas to address in order to implement the solutions the society demands and expects from this technological area. This work aims at proposing some solutions that support the development of the IoT area.

1.1. Context and Motivation

During the last years, Internet of Things (IoT) and its related technologies have gained an important momentum in the market. The number of available smart devices and sensors has grown every year, as well as their applicability to social and industrial areas and scenarios (already predicted by the Internet of Things initiative (Presser and Krco, 2011)). According to IoT Analytics (Wegner, 2022), the enterprise IoT market reached $157.9 billion in 2021 (growing 22.4%) and they forecast that it will reach a "CAGR of 22.0% to $525 billion from 2022 until 2027". IoT Analytics also published a forecast (Hasan, 2022) in which they mentioned that the number of connected IoT devices by 2025 is expected to be around 27.1 billion (active endpoints, not sensors/actuators). Others, like Statista (Vailshery, 2022), are more conservative and claim that the number of devices connected will be 24.1 billion by 2030. In any case, such huge amount of IoT devices is expected to generate around 73.1 ZB by 2025 (from the 17.3 ZB of data produced in 2019), according to IDC forecasts (Reinsel, 2019).

There are already many applications that use the data produced by IoT systems in order to implement their core functionalities in many domains (manufacturing, health, smart cities, logistics, emergencies management, etc.). This means that there is a strong dependency in the systems that could lead to key issues if the sensors and actuators used have an unexpected behaviour. In fact, some works, such as (Jeffery et al., 2006) and (de Bruijn et al., 2016), already reported that sensor datasets use to include erroneous information, as the data produced is not perfect. They identified two main issues: a) ***unreliable readings*** (faulty data collected from the sensor) and b) ***missed readings*** (that could not be transmitted for some reason).

Therefore, it is necessary, first, to carry out the adequate cleansing procedure before processing and using the data (as input for data analytics pipelines in an application or as

the base for building Machine Learning models). Secondly, it is important to go a step further and set up adequate procedures so it will be possible guarantee that the data produced can be trusted, automating as much as possible the data understanding and cleansing tasks. This is directly linked to the concept of *'garbage in, garbage out'* that claims that flawed input will produce flawed output.

These tasks require to deal with non-valid and missing values (filling the gaps whenever possible), the identification of those values that can be considered outliers (removing them in case it makes sense to do so), dealing with randomness and noise and avoiding duplication or constant values (in cases in which this should not be the case). Many of these tasks are related to the identification of anomalies in the data which may determine the quality of the dataset and the trustworthiness in the sensor that produced the measurements.

Considering that, according to the CrowdFlower Data Science report (CrowdFlower, 2017), data scientists spend 51% of their time "collecting, labelling, cleaning and organizing data", this is an important area to be addressed. This is in line with the Kaggle survey for data scientists (Mooney, 2018) that claims that data scientists spend 15% of their time on average (with a maximum of 24%) in data cleaning tasks, while Anaconda's survey in 2020 (Anaconda, 2020) reported that it takes 26% of their time instead.

In order to address these topics, it is necessary to understand the data in more detail, so data scientists can adapt their methods. It seems that there are some assumptions about datasets that are not always fulfilled by the data (such as the normal distribution of the data). Therefore, this work carried out a detailed analysis of several types of sensors, to understand how to identify the anomalies and map them with known errors.

This dissertation contributes with a novel solution that addresses the behaviour of the sensor through the data it has been generating, leaving apart network, security or privacy aspects. Analysing several aspects of the data (supported by statistical analyses and other techniques), it is possible to extract hidden characteristics of the data and provide a complementary vision about what is going on with the sensors, so it will be easier to prepare the data for its processing and to identify potential issues with the sensors. Such analysis can provide insights about how much developers can trust the values provided by sensors and integrate them in their applications.

Considering the current computational capabilities of the nodes at the edge, that can collect and process sensors' data, it is possible to implement efficient analyses at early stages, as close to the source as possible, in the data pipeline, embedding new features that could provide additional metadata that can be, later, collected and analysed even in real-time.

1.2. Research Questions and Hypothesis

The proposed PhD dissertation is focused on the area of trust models, especially in IoT environments, trying to apply it to provenance evaluation in the linked data field (since sensors data could be the base for constructing datasets providing measurements taken by those sensors). Therefore, the hypothesis proposed is the following one:

If Trust is evaluated using several aspects (such as the variation of the data and the presence of outliers), then the result will be accurate and representative of the real behaviour of the entity under evaluation, supporting those tasks related to data understanding and preparation.

Hence, this research work has aimed to understand the datasets generated by several types of sensors (used in several domains, such as smart cities and agriculture), identifying potential issues and particularities that, when cleaning and analysing the data, need to be addressed and considered. As stated in the hypothesis, this work considers two main topics that can be linked to some errors found in sensors' datasets and that can be related to the quality of the data: the **presence of some outliers** and **unexpected variation in the data**. Therefore, this dissertation proposes several research questions that may be addressed through experimentation:

- **RQ1: Which aspects of sensors are linked to the variation of data and the potential presence and role of outliers? Are we making some assumptions that are not met?**
- **RQ2: Which statistical solutions can provide valuable information about these aspects, so that we can analyse them and understand how they behave for different sensors? How can we use, combine, and interpret their outcomes?**
- **RQ3: In case outliers are present, how do they affect the basic characteristics of sensors' data? Are there alternatives that could mitigate the problem?**

Additionally, since correlation may be a way to compare the behaviour of similar sensors that could be exploited to detect misleading behaviours (mentioned in some existing solutions), there are some additional research questions that may clarify how such kind of solution would perform:

- **RQ4: How does correlation work with different types of sensors? What is the potential utility of correlation in understanding data?**
- **RQ5: How do the different solutions perform? Are they also affected by outliers? How?**

As a result, this dissertation proposes a solution that could facilitate all this analysis for different datasets, by addressing the following research questions:

- **RQ6: Is it possible to define a set of processes, to automate and formalize a data understanding process and trust evaluation, applicable to different domains?**
- **RQ7: Can we exploit computational resources in a better way, increasing the efficiency of the proposed solution?**

This research work acknowledges that aspects such as data buffering, synchronization and the management of missing values are important as well, and they have been analysed before in works like (Teh et al., 2020) and (Firat et al., 2012), but they require specific analyses and are beyond the scope of this dissertation (although the results could be useful to support solutions in some of those topics). This work assumes that such aspects are managed at the data collection level, in such a way that the data received contain measurements at regular intervals and that missing values are indicated with some specific

number (e.g., -99.9 in this case).

1.3. Objectives, Scope, and Assumptions

The aim of this dissertation is to determine how much users can trust in the data produced by sensors through the analysis of different aspects to be studied in early stages. There is an agreement on the inexistence of a generic definition of trust which fits perfectly to any situation. Such definition must be tailored case by case, depending on the context. Additionally, there is not a single solution for detecting issues in the data that fits with all the types of sensors and context.

Since trust has to do with relying and depending on other entities, it is right to say that **trust** is the *firm reliance on the capacity of a sensor or device to behave as expected whenever it is required to interact*, and such behaviour will be stable in time.

The main objective of the research work carried out has been to define a way to determine trust for sensors according to the two aspects mentioned in the hypothesis. Considering that coherence of data is key area to guarantee things provide what they are expected, the specific objectives covered in this work are:

- **O1: Analyse in deep the potential attacks and known errors in sensors' data related to the provision trust for IoT**, identifying the main points to be addressed by the models to de designed. Also, identify requirements for the models and the tools so the implemented tools will be really useful (i.e., consider efficiency);
- **O2: Provide an algorithm for analysing the variation of the data produced by sensors**, so it will be possible to determine whether there are some anomalies or not;
- **O3: Provide an algorithm for analysing those datasets generated by the sensors** under evaluation, with the purpose of identifying outliers and anomalous data, which might indicate certain data fragments should not be used because the devices that produced them cannot be trusted;
- **O4: Determine a meaningful way to combine different trust aspects**, so complementary models will be used together for providing a complete view about IoT devices behaviour in a coherent way;
- **O5: Validate the designed models**, checking they provide accuracy results and their robustness against certain errors and attacks to which the proposed algorithms could be vulnerable.

1.4. Research Methodology

The main research methodology followed during this work was the **design science research methodology (DSRM)**, as presented at (Hevner et al., 2004) and at (Peffers et al., 2007). The main objective was to build an artifact that would support data understanding phases in data analytics pipelines and sensor trust evaluation.

The steps followed during the research were those proposed by Peffers et al (Peffers et al., 2007), after analysing several methodologies and guidelines to implement DSRM:

- *Activity 1. Problem identification and motivation* – As stated in the introduction, it is necessary to formalize data understanding processes and trust evaluation, providing a tool for supporting such formalization, based on statistical tests (as they can be fast and flexible), and applicable to multiple domains. Several research questions related to these topics were defined, as well as the aspects to address (variation in data and presence of anomalous values). With such tools, it will be possible to understand how data are expected to behave, discarding solutions that might be affected by the nature of the data (e.g., as mentioned in Section 2, some ML-based algorithms could show problems when using datasets that are not following a normal distribution).
- *Activity 2. Define the objectives for a solution* – A set of objectives were defined previously, including the concrete aspects that must be addressed (variability, distribution of the data, presence of outliers and their effect), the application of multiple statistical solutions (exploring as many as possible), the applicability to different domains (using datasets from multiple sources and domains was a must) and the efficiency of execution (proposing a parallel implementation, as a way to experiment with execution in multiple cores, in line with the devices available in edge environments and Cloud solutions). The solution must provide enough information for the users to make decisions on how to use and process the data.
- *Activity 3. Design and development* – This activity was the most complex one, applying a quantitative research method approach, in which certain experiments were carried out to analyse different statistical solutions, with the purpose of observing which solutions were working, what results they were producing and how they could be used. As a result, the key aspects to include were identified and, therefore, solutions for data transformation, errors detection and data understanding (and trust evaluation) processes were designed. Then, these processes were implemented with R scripts, applying parallelization to the code.
- *Activity 4. Demonstration* – As a way to demonstrate the validity of the solution, this work carried out a complete analysis of new datasets (with a new sensor type not studied before and with new annotated datasets of a known sensor with specific errors) using the implemented R scripts. It was possible to observe the information generated by the scripts in two scenarios. The demonstration included the execution with different numbers of cores, showing how the implementation could be scaled up (10 executions were performed per script and core configuration).
- *Activity 5. Evaluation* – This work succeeded to observe that it was possible to quickly obtain valuable information about the characteristics of the data by using the implemented scripts. The evaluation process checked whether the information provided was accurate, including the identification of errors (by comparing the generated results with the annotations of the dataset and visual observation, as well as other graphs such as histograms). Metrics such as F-score were available in some cases (e.g., in errors detection). Additionally, some plots for showing speedup and execution time were generated as well, in order to observe the efficiency of the parallel execution.
- *Activity 6. Communication* – Once there were results to communicate, a scientific

article to explain the results of this research was prepared (and published).

It is important to highlight that the method followed in Activity 3 included research based on primary and secondary quantitative research methods. In some cases, the datasets were generated as part of the research work (as in the case of data from smartphones and the sensors attached to the Arduino board) and, therefore, there was direct control to influence the data obtained (such as generating certain types of outliers on purpose). In other cases (such as the Ports of Spain and air quality datasets) the datasets were provided by third parties and, therefore, the only possibility was to rely on the information they provided together with the data (such as annotations about data quality).

The experimental research was carried out by writing several R scripts (usually only focused on a concrete set of aspects, such as homogeneity, transformations, etc.) that analysed different statistical solutions in different time windows (from basic statistical properties, such as average, range, variance and coefficient of variation (Reed et al., 2002), to statistical tests for outliers, homogeneity or trend detection) with the available datasets, belonging to different domains (environmental monitoring, home automation, smartphones and smart cities). Such experiments randomly selected samples of different longitudes and were used to answer the research questions about the utility of the methods, the aspects of the sensors that may influence their behaviour, the data distributions, the utility of correlation, the basic detection of outliers, the utility of certain data transformations and their influence on other results. With the information generated, by induction, this work identified the statistics to apply, how to use them and how to make them complementary. The R scripts implemented were evaluated later (as explained) to determine if they were able to provide the expected results.

Taking these aspects into account, it is possible to confirm that this work fulfilled the six guidelines proposed by Hevner et al (Hevner et al., 2004):

- *Design as an Artifact:* A set of solutions and processes for data nature understanding, decision making and trust evaluation were produced, together with their corresponding (parallel) implementation;
- *Problem Relevance:* This work addressed an important business problem of understanding the data and identifying errors before carrying out complex data analytics. The implemented scripts could support researchers and practitioners when selecting the most appropriate data sources, data analytics and ML solutions;
- *Design Evaluation:* Since early stages, this work defined a way to evaluate several aspects of the outcomes produced, including not only the capability to provide relevant information and accuracy of results, but also the performance when executed in parallel;
- *Research Contributions:* During the research, several aspects that affect the characteristics of the sensors' datasets were identified, such as potential issues with variance, the reality about the probability distribution of the data, statistical tests that may be problematic, etc. This knowledge was used to implement a set of new models and useful scripts, that also demonstrated the utility of parallelization as a way to increase performance when analysing sensors data;

- *Research Rigor:* As explained, formal research methods were followed to identify key aspects of sensors' data and to design the R scripts, while the evaluation method was also formally defined;
- *Design as a Search Process:* All possible means were used to obtain a useful solution, adding as many sensor types and statistical solutions as possible, to contrast the results and gain more knowledge. That included datasets provided by third parties and datasets generated during the research in realistic scenarios;
- *Communication of Research:* The associated impact journal paper produced during this research work is a good representation of communication to a technology-oriented audience, as well as other papers published, related to this topic.

1.5. Thesis Outline

This dissertation is organized as follows: Section 2 analyses the current state of the art in those areas related to topics like the kind of errors and attacks that can be found, as well as existing models for trust provision; Section 3 discusses the importance of data sources and analyses the datasets used during this work; Section 4 analyses the sensors and their statistical properties, among others, to understand how to deal with the presence of anomalies and how they affect some properties of the data; Section 5 proposes some improvements to detect anomalies more effectively, as well as a set of procedures that are useful to understand the datasets, addressing trust evaluation through an aggregation mechanism; Section 6 describes how the validation was carried out and the results obtained; Section 7 describes the conclusions and future work around the different topics addressed.

2. RELATED WORK

The only sacred truth of science is that there are no sacred truths.

Carl Sagan

Trust models, as well as the detection of anomalies in datasets, is a topic that has been addressed in other works as well, although perhaps not exactly from the same perspective. This section introduces the types of errors that can be found in sensor-based systems, as well as trust solutions applied in other domains and some techniques already researched with respect to anomalies detection in IoT-based systems.

2.1. Errors and Attacks in Sensors

Since the purpose of this work is to propose a solution that can support the evaluation of the behaviour of sensors, it is important to understand which kind of problems may be faced with the data they generate, and how they could be affected by external malicious manipulations.

2.1.1. Errors and Faulty Values

Some works have analysed the types of errors produced by sensors when collecting data. This is the case in (Ni et al., 2009) and (Baljak et al., 2013), which identified a similar classification (random/malfunction, bias and drift/noise). In (de Bruijn et al., 2016), they used some of these solutions as the basis for defining a fault injection framework, producing benchmarking data.

These three works define taxonomies of data faults and, although they are not exactly the same, there is some consensus on the type of errors and the potential sources they could be causing them. Ni et al (Ni et al., 2009) analyse the faults from two perspectives that are closely related: the data-centric view (focused on the data produced) and the system-centric view (focused on the system construction properties and the malfunction of its components).

From the system-centric perspective, this work identifies five situations that may produce errors: calibration fault, hardware fault, low battery, environment out of range (a sensor measures a phenomenon outside its optimal designed capability/range, producing problematic values) and clipping (similar to the previous one and related to the limits of the sensor analogical to digital converter, that only produces constant values when reaching certain level).

The main interest in the context of this work is in the data-centric perspective, since it is just focused on the analysis of the data the sensor produced. There are four data-centric faults identified, that can be mapped usually with the system-centric ones:

- *Outliers:* An isolated measure that is out of range from the normal or expected values of a sensor (considered the most common one);
- *Stuck-at faults:* A list of constant (or almost constant) values registered for a period longer than expected;
- *Spikes:* Like in outliers, values are out of the expected range and change with more frequency than expected, and are not only isolated values;
- *High noise or variance:* Unexpected increase in the variation of the data, although it still produces measures in line with the phenomenon measured.

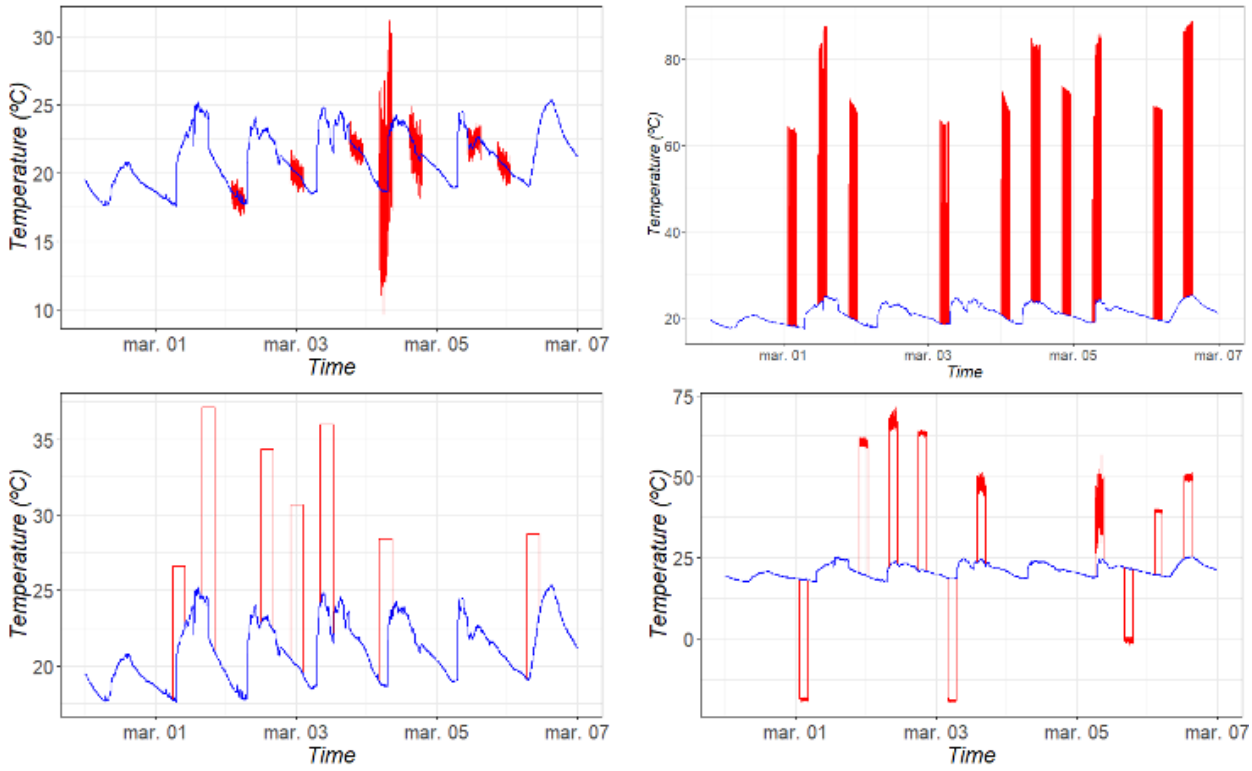

Figure 1 Types of faults in sensors: a) malfunction, b) random, c) bias, d) drift

The work presented by Bruijn et al (de Bruijn et al., 2016) provides a common view, in which the faults are quite similar, with small differences. In their case, they propose two categories: discontinuous (occurring from time to time and with limited effect) and continuous (there are inaccurate readings constantly and some pattern can be observed). According to those categories, the fault types defined are:

- Discontinuous:
  - *Malfunction:* Faulty readings that appear frequently in the dataset (similar to the spikes concept);
  - *Random:* Faulty readings that appear randomly and are not so frequent (they can be mapped to the outlier concept);
- Continuous:

- *Bias:* Unexpected constant values that may have some offset (mappable to stuck-at-faults);
- *Drift:* Deviation of data with high variance (similar to the high noise concept).

The figure shows the four types of faults proposed by Bruijn et al, showing the original measurements (in blue) and overlapping the faults injected (in red) according to their model.

The survey in (Erhan et al., 2021) also identifies types of problematic datasets, types of anomalies, types of faults and sources of anomalies (the environment, the system, the communication and attacks). Some of these sources are similar to the ones already mentioned in (Ni et al., 2009), adding two potential issues related to the phenomenon the sensors are measuring (e.g., unusual event like a natural disaster) and the existence of deliberate attacks to the system (e.g., vandalize physically the sensor or inject malicious traffic). As for the types of faults, they are like the ones used in (de Bruijn et al., 2016), with small variations in the name (spike, noise, constant and drift). Additionally, the survey in (Teh et al., 2020) identifies missing values as another failure to be considered.

2.1.2. Potential Attacks in Sensors Evaluation

Besides the existence of unexpected errors, it is necessary to be aware of specific attacks to IoT systems, focused on creating misleading measurements or even producing faulty ones that might affect negatively to any system or application relying on the metrics generated.

There are attacks that are related to trust and reputation solutions, focused on how to enforce them to produce misleading results and unfair alerts with respect to some concrete entity, such as good/bad mouthing or sibyl attack (generate misleading feedback about nodes, for ruining or boosting nodes reputation) (Bao and Chen, 2012) and conflicting behaviour attack (an entity has different behaviours with different groups of users, creating a conflict in ratings) (Sun et al., 2006). Since these problems affect mainly solutions calculating indirect trust, they are not so relevant for the scope of this work.

On the other hand, there are other attacks that can affect those solutions that aim at evaluating trust and reputation of entities through direct measurements (in line with this work). In this area, this work considers those attacks which have to do with identity management, variations over time and on-off attacks.

Identity management of the evaluated entities (trustees) and their evaluators (trustors) is a key aspect in trust models, since an inefficient management may lead to sybil and newcomer attacks, as explained by Sun et al (Sun et al., 2006) and Noor et al (Noor et al., 2013).

It is difficult to identify in a univocal way an entity, since there is a few information that can be used for generating a unique identity automatically (MAC addresses can be faked, IP addresses are dynamic, etc…), and this issue may affect the evaluation of entities by direct and indirect trust models. There is a lot of literature, especially in the Wireless Sensors Networks field, focused on providing solutions. In the context of this work, this may affect those systems that collect information from sensors connected through wireless mechanisms, or even to the systems that collect data from nodes that centralize the

information provided by several sensors, and then transmit this information to a Cloud-based solution.

In case identity management is not solid enough, malicious entities may use multiple (fake) identities for performing different attacks. In the case of newcomer attacks, they may use new identities for removing their historical data, acting as a new entity with no previous malicious behaviour. In the case of sybil attacks, multiple identities provide misleading feedback about an entity for self-promoting, for sinking the reputation of another entity or for taking the blame corresponding to another malicious entity. While the second case is not so relevant in the context of this work, the first one could be problematic, if there is an open system that registers new sensors automatically. Additionally, there may be cases in which a malicious entity impersonates the identity of an existing sensor, injecting malicious measurements in the network.

Another aspect to consider is how the behaviour of a sensor or entity may vary in time, since that could affect the perception of trust with respect to them. It is usual that some entities offer different levels of quality depending on the requirements received and on some management decisions. It may also happen that other factors affect the trust of entities in the IoT (i.e., availability because of low battery levels, vandalized sensors/things). Trust models need to take this into account, giving the appropriate weigh to past situations. Usually, as reported by Jøsang et al (Jøsang et al., 2007), forgetting factors are a way to get equilibrium in the evaluations.

But it is also very important to consider that variations in the data evaluated may be happening because of an on-off attack, in which an entity alternates good and bad behaviour in such a way a trust model or a monitoring system cannot react adequately and tag the entity as a risky one, as explained by Sun et al (Sun et al., 2006). In open systems, this may happen because a malicious entity imitates the behaviour of a sensor and injects errors in a deliberate way, or in cases, as mentioned before, in which an external entity is impersonating an existing sensor connected remotely. Therefore, trust models should apply forgetting factors in such a way they can detect these attacks and adapt the trust evaluation, focusing on what the sensor/entity is doing in a shorter term, and not on what such sensor/entity did some time ago.

2.2. Trust Models

2.2.1. Concepts around Trust

Trust evaluation is a topic which has been widely analysed, although it is a bit newer in the IoT field. It is possible to apply some of the approaches defined for the WSNs field, but they lack some aspects related to the usage of things by applications and the decision-making mechanisms involved.

First of all, it is important to understand the main concepts around trust evaluation. The survey done by Jøsang et al (Jøsang et al., 2007) proposed some widely accepted definitions, not only for some concepts, but also listing different types of trust.

It differentiates between the concepts **trust** and **reputation**. Two definitions for trust are

provided: one related to *reliability* and another one related to *decision*. The one relevant for this work is the one focused on decision trust, which is rather broad, and can be expressed as the willing of a party to depend on a third one, feeling secure and not worrying about the potential consequences.

On the other hand, it also introduces the concept of reputation, as the opinion that others have about a party. It is what others believe about that party. This concept is different from trust, as people will usually trust on those with good reputation, but it may also happen that a party trusts another one, even if its reputation is bad (e.g., because of a personal or deeper knowledge about that party).

In the context of this work, Section 1 defined **trust** as *the firm reliance on the capacity of a sensor or device to behave as expected whenever it is required to interact*. A more complete definition could be the ***extent to which an application or somebody is willing to rely on the capacity of a sensor to provide the right data whenever required and in a stable way***. The concept of reputation is not applied in this work, since there is not a collection of trust evaluations to generate reputation and the reputation provided by third parties is not used neither.

The different types of trust listed are the following:

- *Provision trust:* It evaluates a service or a resource provider to determine if it will work as expected or agreed, so the evaluator can rely on it;
- *Access trust:* It looks at the access mechanisms set in order to access resources from a provider, evaluating whether they are secure enough;
- *Delegation trust:* In systems in which a provider is delegating some actions in a third party, it evaluates the behaviour of the delegated entity and the overall mechanism;
- *Identity trust:* It evaluates if the identity of an agent is the one claimed by the agent, and it is closely related to authentication mechanisms;
- *Context trust:* It is related to the evaluation of the environment around the analysed party (like for infrastructures or legal systems), checking if there are mechanisms that will provide stability in case something goes wrong.

In the context of Internet of Things, the analysed models are focused on **provision trust** mainly, since they evaluate whether the sensors are available and they are providing data as expected. Access trust and identity trust may be interesting in the IoT domain as well, although these aspects are usually expected to be addressed by basic features of the system.

Additionally, another way to classify trust models is the type of metrics used:

- *Subjective:* They are measures obtained by means like surveys, in which users give their opinion, being specific (evaluate concrete aspects) or general (aggregate several opinions);
- *Objective:* There are clearly measurable metrics that determine the evaluation. They may be specific (concrete) or general (aggregating several measures).

Most of the models in the IoT domain (as well as this work) aim at implementing objective models, mainly because there are many aspects that can be measured with concrete metrics, such as the network, the energy levels, the data received, etc. This is interesting

because some of the known attacks to trust and reputation models can be avoided.

The work at (Alghofaili and Rassam, 2022) mentions different aspects to take into account when defining trust models for IoT. In this case, they identify the following properties:

- *Trust Composition:* It may be Quality of Service (evaluating quality aspects like reliability and tasks completion) or social trust (evaluating other aspects related to the relationship between parties like privacy, connectivity and honesty)
- *Trust Formation:* It may be based only on the evaluation of one aspect (single-trust) or several aspects (multi-trust)
- *Trust Propagation:* It may be distributed (trust evaluation is shared between IoT nodes) or centralized (central entities where trust can be accessed)
- *Trust Aggregation:* It represents the type of aggregation method that is used for the evaluations, that could be static or dynamic (when using belief theory, weighted sums, fuzzy models, etc.)
- *Trust Update:* It determines when the trust evaluation is updated, and it may happen periodically or when a certain event triggers the re-evaluation.

Although all of them have some interest, not all the models address specifically the complete list of properties, like the propagation aspect. In any case, this list, as well as the other aspects mentioned before (about the type of trust to evaluate and the type of metrics to use) should be used as a reference to start the design of new trust models, as they support the creation process, focusing on what is going to be measured and how, and setting the basic objectives of the model.

2.2.2. Trust Models Related to IoT

Taking a look at solutions for the IoT field, Chen et al (Chen et al., 2011) propose a trust model based on three aspects: End-to-end packet forwarding ratio, energy consumption with respect to packets received/sent and package delivery ratio. They provide an aggregation mechanism based on weights for those aspects. As well as a mean to use recommendations provided by other nodes, taking into account they might provide opposite evaluations for the same node and using thresholds of minimum recommendations. Although the model has been validated including situations with malicious nodes, no trust attacks have been taken into account and, therefore, the robustness of the model has not been demonstrated.

Bao and Chen (Bao and Chen, 2012) present a trust model based on three main pillars: honesty, cooperativeness and community-Interest. It shows how recent and old evaluations affect the final result of the trust evaluation, including indirect trust provided by third parties. While cooperativeness and community-interest introduce some social aspects in the IoT, honesty determines anomalies in the behaviour of a certain node (based on rules for retransmission, repetition, delay, etc). Although it has been designed taking into account three attacks related to indirect trust, it is more focused on the aggregation part, restricting the individual aspects to network related measurements.

Focusing on the security aspect, Leister and Schulz (Schulz and Leister, 2012) propose a

very theoretical model which analyses communication channels when interacting with other nodes. It determines á priori trust (before sending messages) and á posteriori trust (after receiving a message) based on measures for evaluating security, privacy and availability of channels and things. Although they mention several facets for determining trust, this is not clarified and there is not a clear statement about their aggregation. Even if they take into account potential security attacks, they do not consider trust attacks.

Finally, Gu et al (Gu et al., 2014) present a complex model which comprises three layers: Sensor, Core and Application. For the sensor layer, the model is based on a tree representing evaluated factors, which calculates trust by aggregating evaluated attributes at the same level of the tree by using fuzzy sets and attributes weights (for overhead, direct scoring, etc.). The core layer analyses network and routing in a similar way (but using other attributes such as capability, network price, anti-attack mechanisms…) and the application network uses the same calculation and attributes (i.e., service efficiency, risk and history).

From those models oriented to the WSNs field, the Energy Prediction based Trust Management model (Shen et al., 2010) is interesting for IoT. Trust is determined by calculating the packet successful delivery rate, complemented with an intrusion detection mechanism which predicts energy consumption of sensors, using Markov chains, and compares it to the real consumption. If packet loss is high and energy consumption increases too much, it infers the network is under a 'hello flood' attack. Although the model takes into account two aspects, it uses them only for attack detection, not trust evaluation.

Coming from the same field, the model defined at (Javed and Wolf, 2012) provides an interesting approach for identifying outliers in data generated by sensors, applying multiple regression techniques. It is not even focused on trust and, instead of using the result for predicting values, regression is used for identifying those sensors which are providing not expected values. Although it is interesting, it is hard that the algorithm considers a sensor fully in line with the expected values.

The work at (Al-Rakhami and Al-Mashari, 2021) also addresses trust in IoT related to the communication and integrity of data. In this case, the authors propose a way to apply the blockchain technology in the logistics domain, where IoT is used to monitor and manage supply chains (especially for the traceability of products). They address trust from the perspective of sharing data between the involved parties and guaranteeing its integrity. The role of the IoT part in this solution is to collect data from sensors at the involved parties, encrypting it and managing it using blockchain technology, so it will be kept logged and it will not be altered. Then trust is determined based on the integrity of the data and the transmission time. It is assumed that the data produced at the IoT nodes is right.

Another recent work (Alghofaili and Rassam, 2022) proposes an IoT trust model which is more focused on the evaluation of IoT nodes. In line with other solutions, it monitors the traffic of the sensors, extracting three features: packet loss (percentage of packets that failed to arrive), delay (the latency measured when transmitting data) and throughput (the bandwidth that was measured when sending the data). Once this information is available, the model proposes to use simple multi-attribute rating technique (SMART) to evaluate trust. SMART defines dynamic weights to the three criteria defined (based on Shannon's entropy methods), obtaining a weighted sum (with all values normalized between 0 and 1) and then it compares the result with a threshold. Then, the long short-term memory

(LSTM) technique is used to predict the behaviour of an IoT node and detect changes.

The work at (Adams et al., 2018) is one of the solutions that relies on the data produced by the sensors. It proposes to generate a model that can predict the sensors' values. Then it proposes to generate a forecast and calculate the residuals, that are used to determine the trust value by using the probability density function of the Student's T distribution. This way, low trust scores, that are represented by values in the tails, are linked to unexpected values that generate high residuals.

There are also trust models that look for some consistency between the data generated by a sensor and the one produced by other sensors, like in the case of (Liu et al., 2020). This work proposes an ensemble-based method in which two scores are calculated: one about the consistency of the generated value with respect to the rest of the sensor network and another one for calculating the historical reputation of the sensor by looking at previous values. The Internal Consistency Score is calculated based on a forecasting model (trained with trustworthy data), and an average of p-values obtained with a formula from the standard normal distribution function. The reputation score is obtained from the average of Internal Consistency Scores calculated. Finally, these scores are multiplied to obtain the trustworthiness score.

As a conclusion, unlike trust models for other fields before, models for the IoT environment tend to be multi-faceted, trying to analyse several points of view around things, although the aggregation mechanisms used might not be user friendly enough (general trust is usually provided, without details about underlying trust aspects or attributes). Moreover, none of the models focuses on provision trust and most of the analysed facets are focused on network related characteristics, only dealing with energy aspects in a few cases (related to efficiency and not to availability), and with a very small number of models analysing in depth the data coherence topic. Another topic to explore is the usage of forecasting techniques, so trust models could predict malicious behaviours earlier and become more proactive, instead of reactive.

As mentioned before, trust models have always to face certain problems not only because of their nature and the algorithms they apply, but also because of well-known attacks designed to undermine these systems. In the case of Internet of Things, assuming the ratings being generated are not provided by humans, it has been possible to reduce the list of potential problems, such as impact of incentives usage and positive/negative bias. In fact, in the kind of system considered in this work, problems like unfair ratings and evaluations are not applicable, since all the information should be available through the system that collects the data from the sensors.

Still, it may happen that there is a problem related to the lack of information that is necessary to build a model. In many cases, it is necessary to build a whole system around the model in order to enable the collection and analysis of certain data. This means that it is not easy, or even possible sometimes, to integrate a trust model in existing systems and the best option is to reduce the inputs needed (e.g., limiting it to the measurements received).

Additionally, as many models presented do not consider the data coherence, they may be able to detect some problems, but it is very hard (or not possible) for them to detect newcomer attacks and on-off attacks.

2.3. Outliers and Errors Detection in Sensors

2.3.1. Types of Solutions for Anomalies Detection

Besides those solutions that design and implement trust models, there are other solutions that are only focused on the detection and removal of anomalies in the data obtained from sensors. They are considered for this work, since the proposed solutions are directly related to the analysis of the data produced by the sensor, as it is the only 'trustable' data that is available, and the one that should be used by applications and data scientists.

Anomalies are considered *those values that deviate from other measurements in the same dataset, in such a way that they are suspicious to have been generated by some external event*, instead of the natural element or event observed with the sensor.

The work in (Erhan et al., 2021) categorizes the solutions for anomalies detection in two main groups:

- *Conventional Techniques:* This group contains solutions that are based on the numerical analysis of the data:
    - *Statistical methods:* It includes parametric (like regression models and Gaussian models) and non-parametric statistical models (like some clustering techniques, Hidden Markov Models and hypothesis testing);
    - *Time series analysis:* They use solutions like auto regressive integrated moving average (ARIMA) and Kalman filtering to obtain a model and compare forecasts with the obtained values;
    - *Signal processing:* It proposes to use solutions like Fourier transforms, filters and other wavelet-based transformations to de-noise the data and to detect outliers and changes in frequency;
    - *Spectral techniques:* It uses Principal Component Analysis (PCA) to build projections of the data in subspaces and detect anomalies in fewer dimensions;
    - *Information theory:* Techniques like entropy, information gain and relative entropy are used to compute reference values with the complete dataset, detecting anomalies in smaller samples of the data.
- *Data-driven Techniques:* It groups those activities based on Machine Learning and Deep Learning techniques:
    - *Supervised learning:* They classify the anomalies using full labelled data for training the model with techniques like decision trees, Support Vector Machines (SVMs) and rule-based classifiers;
    - *Semi-supervised learning:* Only part of the datasets is labelled, and the training is done with 'normal' data, with techniques like auto encoders and One Class SVM;
    - *Unsupervised learning:* No data labelled is used, with techniques like clustering (determining if the data does not belong to known clusters) and

Bayesian networks (estimating the likelihood of the new values);

- *Reinforcement learning:* Agents are used with specific reward functions to find anomalies through specific patterns, using techniques like Inverse Reinforcement Learning as well.
- *Deep learning:* This category includes models based on Convolutional Neural Networks, Generative Adversarial Networks, Sequential Networks (like Recurrent Neural Networks), autoencoders, Restricted Boltzmann Machines and hybrid models that combine some of the mentioned techniques.

The survey done in (Teh et al., 2020) does not group the type of solutions in the same way, but it also identifies different types of solutions for anomalies detection, that are quite similar to the subgroups mentioned before: Principal Components Analysis, Artificial Neural Networks, ensemble classifiers, support vector machine, clustering, ontology/knowledge-based systems, univariate autoregressive models, statistical generative models, grey prediction models, particle filtering, association rule mining, Bayesian network, Euclidean distance and hybrid methods.

In both cases, numerical solutions are considered as an important part of the literature, although another survey (Ramotsoela et al., 2018) only considers methods that are based on ML techniques, adding Genetic Algorithms as another type of solution that can be found.

2.3.2. Anomalies Detection Models

During the last years, there have been several works analysing how to clean data from different sources (many of them focused on climate data). They aim at identifying some outliers or gaps in the data that are removed. Works such as (Firat et al., 2012) and (Che Ros et al., 2016) discuss how to fill gaps in weather data and the usage of certain tests (such as the standard normal homogeneity test (Alexandersson, 1986), Pettitt test (Pettitt, 1979), Buishand range test (Buishand, 1982), etc.) to determine outliers in the data. The proposed solutions are not focused on linking the problems to sensors and they use aggregated data that has nothing to do with the concrete metrics collected in real-time by sensors. In the end, they deal with datasets that are normalized and the conditions of applying the tests to sensors' datasets are different.

There are solutions that have been designed to automatically clean data from sensor networks. In the case of (Jeffery et al., 2006), the authors proposed a system in which cleaning follows five stages (point, smooth, merge, arbitrate and virtualize) using simple queries. In the case of outliers and faulty readings, they propose to eliminate measurements beyond a certain threshold and to use the mean and the standard deviation of nearby sensors.

Other solutions (Zhang et al., 2016) propose a cleaning method of calculating an influence mean, which gives weights to sensor measurements (or removes them) depending on their reliability, based on the similarity of measurements. Such work mentions issues with Spearman correlation in other solutions and does not take into account that similarity might not need to be so high in terms of current value.

In the field of real-time streams, (Kenda and Mladenić, 2018) proposes an improved

method based on a Kalman filter applied to time series, such that it corrects the received values with the predicted ones from the filter when a certain threshold is reached. It is focused on additive outliers and assumes that the sensor produces continuous values, and no big jumps are expected in the data. Moreover, variance is at the core of the model, as a key parameter to determine the acceptable boundaries. The work in (Erhan et al., 2021) also mentions other approaches in the area of time series analysis, such as auto regressive integrated moving average (ARIMA), but obtaining good models is very complex and problem specific. Moreover, when detecting anomalies, they need to assume that the estimation is more accurate than the real value and, therefore, values far from the estimation are considered outliers.

Finally, there is an important group of solutions that attempt to solve the problem by using Machine Learning and Deep Learning techniques. The work in (Ramotsoela et al., 2018) classifies solutions for anomaly detection focused on parametric and non-parametric machine learning solutions. While the first group was mainly focused on detecting attacks and loss of data (Magán-Carrión et al., 2015), the second one was focused on the detection of abnormal values, including solutions based on K-nearest neighbours (kNN) (Liu and Deng, 2013), support vector machines (SVMs) (Martins et al., 2015), artificial neural networks (ANNs) and genetic algorithms (GA). According to the study in (Hasan et al., 2019), it compared logistic regression (LR), SVM, decision tree (DT), random forest (RF), and ANN, finding as a result that the solution based on RF performed better in general.

The applicability is huge, especially in the context of the Industrial IoT, where there are solutions using some statistics for dispersion together with an unsupervised ML algorithm for anomaly detection (Maseda et al., 2021) (applied to manufacturing), as well as solutions applying yet another segmentation algorithm (YASA) with a one-class SVM (Martí et al., 2015), applied to the oil industry. This kind of solution also has been applied in the field of autonomous vehicles (Oucheikh et al., 2020), detecting anomalies in sensors (in real-time) using a long short-term memory (LSTM) autoencoder that extracts data stream features and feeds a convolutional neural network (CNN) for classifying the anomalies.

The problem with ML-based solutions is that they tend to be too problem-specific, not being able to generalize for other systems and types of sensors. Many of the problems and constraints of the models are related to the low quality and imbalance of the data, as in the IoT environment it is difficult to get large and heterogeneous datasets with adequately labelled information.

Another interesting concept is that applying ML techniques to deterministic problems may not be the best solution, since ML is considered stochastic. A neural network cannot understand the physics or mathematics behind the sensors' behaviour and their data. For that reason, even if it is possible to obtain good models for specific problems, obtaining models that can maintain their efficiency when applied to other types of sensors and environments may require much more complicated models.

Additionally, parametric classifiers such as Gaussian Naive Bayes, linear discriminant analysis and quadratic discriminant analysis assume a normal distribution of the data (because of the way they use statistics such as the mean and the standard deviation). Therefore, it is important to understand if the data obtained from the sensors can fulfil such assumption. Otherwise, it may be necessary to apply some transformation to the data (e.g.,

Box-Cox (Box and Cox, 1964) and Yeo Johnson (Yeo and Johnson, 2000) transformations).

2.4. Discussion

It is possible to observe that researchers are trying to understand the problems around those systems based on sensors, as a first step to mitigate those problems by looking for the root of the problem and adapting the data generated. Besides the usual malfunction problems, there are several ways in which malicious external entities may attack IoT systems or just offer data coming from sensors that might be manipulated. Fortunately, it seems that, if the system has a robust identification system, the most important aspects of the sensors can be measured directly (e.g., the data they produce), so it is possible to determine if there is a problem, not needing to rely on reputation information provided by third parties (that could introduce another level of potential attacks).

In fact, most of the typical trust systems available in other domains depend on collecting third parties' opinions about the behaviour of the entity evaluated, because it is the only way to obtain relevant data for the evaluation.

Although many solutions for determining trust in IoT environments are based on analysing what happens at the network layer, this work proposes that focusing the trust evaluation mainly on the data generated is the best way to avoid additional issues and to obtain a closer idea of the real behaviour of a IoT system, and it could be complementary to those solutions that analyse the network behaviour.

The analysis of data is the line followed by several solutions focused on anomalies detection for systems based on sensors' data. Such anomalies detection approaches are based on the detection of outliers through different techniques that the literature in the field groups in six main categories: statistical-based solutions, time-series analysis (usually, ARIMA-based solutions), signal processing, spectral techniques, information theory and Machine Learning-based solutions.

In the case of ML-based solutions, it is possible to train very accurate models that will work efficiently in those environments they were implemented for, but it is hard to make them generic, not only because of the data used for training (as models can be re-trained), but also because of the variety of sensors they should address. There are many solutions that work fine in their own environment but porting the same solution to other environments would require an important customization.

As mentioned before, the fact that ML models cannot understand the mathematical properties of the data suggests that generic models should include more inputs and would require a large number of heterogeneous (and annotated, if possible) datasets that are not easy to collect.

Moreover, the impact that outliers have in the mean and standard deviation of the datasets are problematic for some ML techniques that should be discarded.

Still, it would be very interesting to apply ML models based on the mathematical properties of the data already extracted from the original dataset. Once the data has been pre-analysed, a ML model could identify which mathematical properties of the data could indicate anomalies and which thresholds would trigger the classification of data as

anomalous.

The problem of ARIMA-related solutions has been mentioned before. It is hard to get accurate models and, in the end, such approach assumes that the forecast is always better than the real data analysed. The models must be so accurate that it would be necessary to build one customized model for each sensor or, at least, one model for each group of sensors with a similar behaviour.

It seems the statistical-based solutions could be interesting for obtaining a model that would be applicable to multiple environments with enough accuracy (e.g., not producing too many false positives). Still, the way in which outliers alter the mean and standard deviation, again, is a major issue that must be addressed.

3. ACCESSING SENSOR DATA

Too often we forget that genius, too, depends upon the data within its reach, that even Archimedes could not have devised Edison's inventions.

Ernest Dimnet

One of the initial challenges when doing this work had to do with the collection of datasets that would be representative of a big enough diversity of sensors (if possible, with annotations about the failures detected), so it would be possible to analyse different behaviours and to understand how these types of datasets must be addressed. This section deals with the aspects of data provenance, and it lists the datasets that were used, together with their particularities, that should be taken into account as a previous step to the analysis of the data.

3.1. Understanding Data Provenance

Since this work aims at understanding how different sensors behave and how to determine an expected behaviour, it is crucial to understand first where they come from and the kind of metrics they deal with. It is clear that the nature of the data analysed has a direct impact in how its data varies and the potential presence of outliers (RQ1). For instance, sensors located outdoors are more exposed to unexpected variations, while those indoors are more protected (e.g., against weather) and are expected to behave in a more stable way. Additionally, metrics like temperature are completely different from others like water salinity or luminosity.

As this work wants to provide a solution as much generic as possible, it was important to access multiple types of sensors from multiple sources and analyse how they behave. But such activity is problematic, since not so much data is available in an open way. Even if during the last years more datasets have been made available, they are not always the kind of source needed. For instance, datasets with climate-related information (e.g., temperature, humidity, etc.) use to provide coarse grained information, giving average temperatures per day (or hour), instead of real-time values. Moreover, it is very unusual to find annotated datasets that provide information about potential outliers, so they can be used to perform a deep analysis and validation.

Finally, understanding data requires some preparation, since datasets are heterogeneous, they come in different formats and organization, so it is important to take into account that some pre-processing may be needed as part of a process for understanding the data and analysing it (RQ6). Therefore, this chapter also addresses how the data was adapted, so

metrics from different data sources could be used in the context of this work.

3.2. Data Sources

With the aim of covering as many data types as possible several data sources have been used with three main purposes: i) Understand the behaviour of some sensors and their associated metrics; ii) Build the models, so they fit the real-world behaviour; and iii) Validate the proposed solutions.

The datasets used cover different types of metrics (from temperature to significant wave height), different models of sensors and different locations. With this approach it is possible to get a wider view of the mentioned aspects, since certain metrics (such as temperature, for instance, may behave a bit different in different places).

From the beginning, the idea was to use real datasets coming from large IoT deployments, instead of only setting up a laboratory environment with several sensors generating data. The main reason is that these systems provide real-world datasets, already validated, avoiding any kind of bias in the data. They are heterogeneous IoT environments which provide valuable information, even if, sometimes, they provide the same kind of metrics.

This section lists the main data sources used, describing the kind of information they provided, their format, how they were pre-processed (when necessary) and the main information extracted from the datasets analysis.

A few additional small datasets were used in order to do further analysis of known sensors or to analyse additional types of sensors. This was the case of the datasets provided by the Meteoblue and the Kunak companies. In the first case, they provided a CSV file with metrics measured from a weather station in a farm (for analysing precipitation), while the Kunak dataset (in CSV as well) contained air quality measurements.

Some of these datasets can be found in the GitHub repository which contains the materials of the research done: https://github.com/fjaviernieto/Sensor-Trust-Tools. Not all the datasets are published because of licensing limitations.

3.2.1. Ports of Spain

Ports of Spain is a Spanish public entity that manages a set of public data sources around the Spanish coast, among other activities. They have several devices (like buoys and weather stations) that collect data in real time, so they can monitor the coast, currents and weather conditions that may affect the navigation of ships.

They manage a large list of heterogeneous devices that provide valuable data, although not all of them provide the same metrics and the frequency of data collection also varies (even for the same type of sensor). Since it is possible to request access to historical data, a list of data sources was selected (located in the area of Algeciras) and the historical data was requested (through a formal procedure), that was provided by Ports of Spain as a group of datasets. It is an interesting source of data because of the variety of sensors used and because of the possibility to find equivalences between similar sensors.

3.2.1.1. Information Provided and Data Acquisition

Ports of Spain can provide a lot of data from different datasets groups (depending on the data owner and the kind of devices which generated the data). It is possible to access a map of devices and real-time data from the 'Oceanography' option. The main groups of datasets available are the following:

- REDEXT [https://bancodatos.puertos.es/BD/informes/INT_2.pdf]: Measurements coming from the network of buoys which are located in deep waters (200 meters);
- REDCOS [https://bancodatos.puertos.es/BD/informes/INT_1.pdf]: Measurements coming from the network of buoys located close to the coast and the ports (100 meters deep);
- REDMAR [https://bancodatos.puertos.es/BD/informes/INT_3.pdf]: Measurements coming from the Spanish tide gauges (up to 30 stations) controlling water level in the ports;
- REMPOR [https://bancodatos.puertos.es/BD/informes/INT_4.pdf]: Measurements coming from the Spanish network of meteorological stations installed in the ports, for monitoring atmospheric conditions;
- EXTERNAL [https://bancodatos.puertos.es/BD/informes/INT_12.pdf]: A group of datasets coming from external entities, mainly coming for buoys, for completing the available information.

In each case, the information available per dataset is different (specifications are available in the links, in Spanish only). While some devices are focused on measuring water and waves level, other are focused on retrieving information about temperature, wind speed and direction, atmospheric pressure, etc. The list of all the metrics used is the following:

- Sea level / waves altitude (in some datasets, it is an average of the last 24-26 minutes);
- Waves direction (in some datasets, it is an average of the last 24-26 minutes);
- Water temperature in the surface;
- Speed of the ocean current (average of last 10 min);
- Direction of the ocean current (average of last 10 min);
- Atmospheric pressure;
- Air temperature;
- Wind speed (average of last 10 min);
- Wind direction (average of last 10 min);
- Water salinity.

Those metrics which are calculated indirectly from others with some additional processing (i.e., tide, maximum wave altitude, etc...) have not been used, since the objective was to focus on the raw data taken from the sensors.

There are datasets available for all the Iberian Peninsula coast, although for the experiment, only a few of them were requested. It was mandatory to sign a form declaring the datasets

requested and accepting the indicated usage conditions. After sending it by email, an FTP location was made available for downloading the datasets. This is the list of requested historical datasets:

- Meteorological station Dique Exento_Sur (REMPOR)
- Meteorological station Dique Exento_Norte (REMPOR)
- Buoy in Algeciras – Pta. Carnero (REDCOS)
- Meteorological station Dique de Abrigo (REMPOR)
- Tide gauge Algeciras (REDMAR)
- Meteorological station Endesa (REMPOR)
- Meteorological station Campamento (REMPOR)
- Buoy in Golfo de Cádiz (REDEXT)
- Tide gauge Bonanza 2 (REDMAR)
- Tide gauge Huelva 5 (REDMAR)

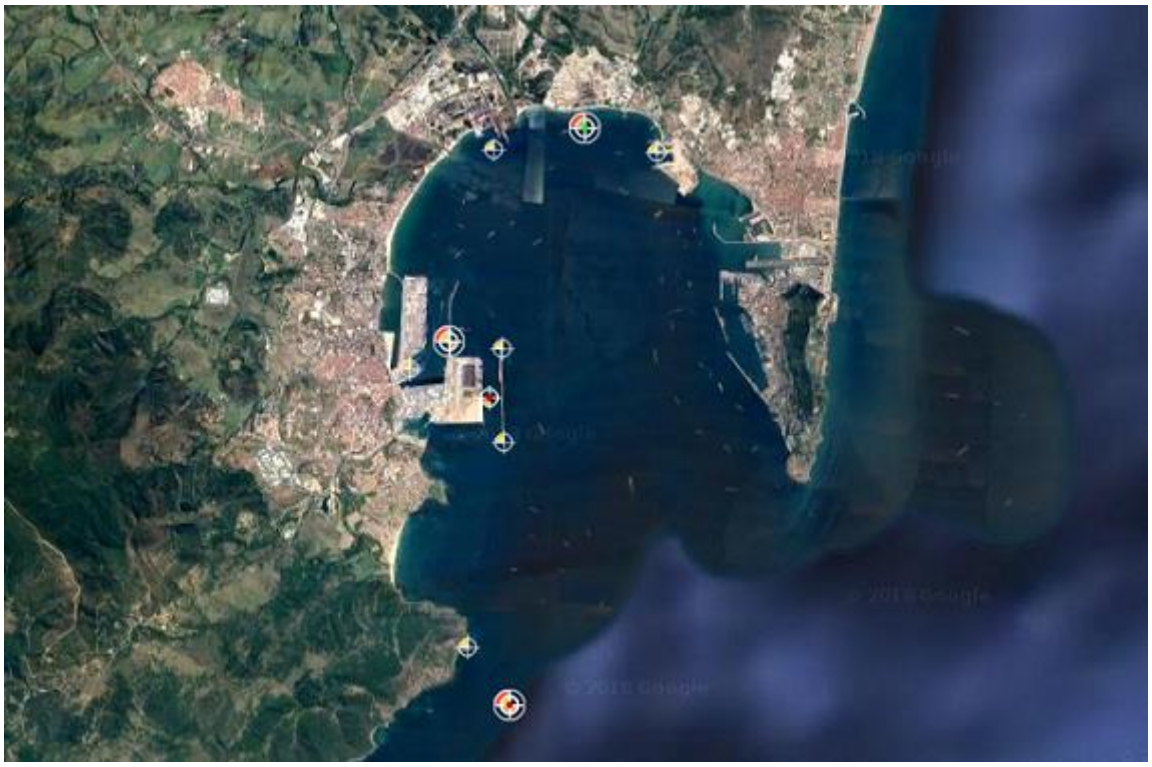

Figure 2 Map of the datasets used in the area of Algeciras port, from the Ports of Spain website

The reason for doing so is that it was possible to get heterogenous metrics from the available devices while, at the same time, since some of them measure the same aspects (i.e., atmospheric pressure, wind speed, etc), it would be possible to check whether it is possible to consider some of them equivalent, due to their relatively close locations.

3.2.1.2. Format of the Data

All datasets were provided in plain text files (with the extension '.dat') which are editable with any text editor. They contain an introductory text which explains the source of the dataset, the metrics included in the dataset and how to interpret the data. It is always

mentioned that directions are always expressed as '0 = North, 90 = East'. Null values are set to -9999 or to -99.9, depending on the dataset.

It describes that the unit of measurement per each metric is the following:

- Sea level / waves altitude (m or cm, depending on dataset);
- Waves direction (0=N, 90=E);
- Water temperature in the surface (ºC);
- Speed of the ocean current (cm/s);
- Direction of the ocean current (0=N, 90=E);
- Atmospheric pressure (Hpa);
- Air temperature (ºC);
- Wind speed (m/s);
- Wind direction (0=N, 90=E);
- Water salinity (psu).

After the description of the dataset, the data itself is included in plain text, organized in a kind of table in which there is one column per metric (sea level, air temperature, wind speed, etc.) and one row per metric taken.

```
67
68  Datos METEOROLOGICOS calculados sobre periodos de 10 min.
69  para una altura  de 3 m. sobre la superficie libre.
70  (Ps y Ta son medidad instantaneas)
71
72  Ps      : Presion Atmosferica                      (Hpa)
73  Ta      : Temperatura Media de Aire                (C)
74  Vv_md   : Velocidad media del Viento               (m/s)
75  Dv_md   : Direccion media de PROCEDENCIA del Viento (0=N,90=E)
76  Pro_Met : Canal de obtencion de los datos
77
78
79  DIRECCIONES: El criterio es  0 = Norte, 90 = Este
80  DATO NULO  : Es representado por -99.9
81
82
83  Los campos Pro_Oe, Pro_Od, Pro_Oce, Pro_Met especifican
84  el canal de procedencia de cada conjunto de datos
85  Pueden tomar los siguiente valores
86    0)   No existe informacion para dicho conjunto
87    1)   Datos Procesados por Puertos del Estado
88    2)   Datos Procesados y Almacenados en la Boya
89    3)   Datos Trasmitidos por Satelite en Tiempo Real
90
91    El campo  Qc_e, se refiere al control de calidad
92    puede tomar distintos valores, siendo aceptables 1,2 y 3
93    - 1 y 2 registros de buena calidad
94    - 3 calidad dudosa (el usuario debe valorar la posibilidad
95      de incluirlo en su estudio o no)
96
97  LISTADO DE DATOS
98
99    AA MM DD HH    Hm0   Tm02    Tp   Hmax  Thmax  Pro_Oe   Dmd   Dmd_P  Ds_P  Pro_Od   Ts2    Sa2   Vc_md  Dc_md Pr
100
101 1996 08 27 18    0.6    3.3   5.8    0.9    3.2       1  -99.9  -99.9  -99.9       0  -99.9  -99.9   42.8  174.0
102 1996 08 27 21    0.6    3.2   3.7    0.9    3.7       1  -99.9  -99.9  -99.9       0  -99.9  -99.9   33.0  147.0
103 1996 08 28 00    1.0    4.4   5.7    1.5    4.7       1  -99.9  -99.9  -99.9       0  -99.9  -99.9   29.2  158.0
104 1996 08 28 03    0.9    4.4   5.6    1.3    5.7       1  -99.9  -99.9  -99.9       0  -99.9  -99.9   21.8  170.0
105 1996 08 28 06    0.8    3.7   5.1    1.3    5.4       1  -99.9  -99.9  -99.9       0  -99.9  -99.9   17.0  163.0
106 1996 08 28 09    0.8    3.8   5.1    1.2    6.3       1  -99.9  -99.9  -99.9       0  -99.9  -99.9   20.2  132.0
107 1996 08 28 12    0.7    3.9   5.2    1.0    5.2       1  -99.9  -99.9  -99.9       0  -99.9  -99.9   19.8  116.0
108 1996 08 28 15    0.6    4.1   4.3    0.8    4.5       1  -99.9  -99.9  -99.9       0  -99.9  -99.9   42.0  151.0
109 1996 08 28 18    0.6    3.6   3.7    0.9    5.2       1  -99.9  -99.9  -99.9       0  -99.9  -99.9   31.3  159.0
110 1996 08 28 21    0.6    3.6   7.3    0.7    4.3       1  -99.9  -99.9  -99.9       0  -99.9  -99.9   21.0  133.0
111 1996 08 29 00    0.6    3.8   5.3    0.9    4.2       1  -99.9  -99.9  -99.9       0  -99.9  -99.9   25.3  121.0
112 1996 08 29 03    0.8    4.2   4.8    1.2    5.1       1  -99.9  -99.9  -99.9       0  -99.9  -99.9   26.7  168.0
113 1996 08 29 06    0.8    4.1   5.7    1.2    5.7       1  -99.9  -99.9  -99.9       0  -99.9  -99.9   26.1  190.0
114 1996 08 29 09    0.7    3.8   6.1    1.0    6.1       1  -99.9  -99.9  -99.9       0  -99.9  -99.9   27.4  195.0
Normal text file
```

Figure 3 Format of a Ports of Spain dataset

Some datasets contain data since 1996, while others may have less data (i.e. 2011, 2009...), depending on the date in which the device became available. Therefore, files size may go from 4MBs (the smallest one, starting in 2013) to 125MBs (the largest one, starting in 1996).

3.2.1.3. Pre-processing of the Data

Since the implementation of the data analysis application was done in such a way it loads only CSV files, it was necessary to extract the relevant data from the .dat files received and to transform them so it would be possible to export everything to a CSV file.

In most of the cases, the content of the tables with the data was copied and imported in Microsoft Excel, creating a XLSX file with the data organized in columns (it is straight forward by indicating that the columns were using blank spaces as separator when importing). Since date information was split in several columns, a new column was used in order to put together these fields, building dates with the format 'yyyy-mm-ddThh:mm' (this is done with some operations concatenating the content of several cells and certain characters). After that, the relevant columns were selected, copied and pasted in another tab of the Excel file, adding a line with the adequate headers (basically, 'Date' and one additional name per metric included). Finally, the content of that tab was exported to a CSV file, using just the saving function in Microsoft Excel.

The result is a CSV file which can be imported by the specific code written for the analysis of the data, as well as by the scripts that evaluate the trust of the sensors.

One of the issues with the datasets is that some of them contain one row per each hour, in which some values are calculated as an average of the metric during the last hour. But there are other columns that contain concrete values per each 5 minutes. This is the case with the datasets with information from the tide gauges, when measuring the sea level. These datasets contain a column with the sea level average per hour and 12 columns with sea level measured at minute 0, minute 5, minute 10, etc...

Instead of using the average value per hour, it is better to use the actual value measured per each 5 minutes, since these values provide more accuracy about what is going on with this concrete metric. Therefore, it was necessary to perform a transformation in the dataset, so one row would be created per each metric taken. Such transformation required to read the value of the corresponding column, generate the date and put all the data as a new row.

Initially, the idea was to use a macro in Microsoft Excel, which was taking the values, copying them in the corresponding cells and building the corresponding date, in a new tab. It had an iteration taking the 12 values and generating 12 dates for each row of original data. Even if the number of operations to perform was not so high, this solution was taking too long to execute. The dataset used (from the tide gauge in Algeciras) contains more than 77000 records (metrics since 2009 to 2018). Attempts to run the macro with several months of data was taking around an hour, but trying to run the macro with the complete dataset was not possible. After more than 13 hours computing, the process was stopped, since the computer was overloaded and seemed to be blocked. The time to compute rows was not growing linearly in this case, so it seems the system was getting overloaded because macros code in Excel is not efficient enough (and especially with large amounts of data).

The solution was to import the data to a SQL Server database and perform some SQL

operations creating a new table with the dataset with the re-arranged metrics. For doing so, it was necessary to transform the .dat file to a XLSX one using Microsoft Excel (adding an additional column for building the date and importing just the lines with the data). After creating a new database (named REMPOR), such XLSX file was imported in the database as a new table with data. Later on, a SQL script was applied and, finally, the resulting table with the data was exported to CSV. The script contains one INSERT clause and 12 SELECT clauses with the UNION operation. It takes just 7 seconds to run, and it generates a table with more than 927000 records with the Algeciras tide gauge dataset. The records are not stored sorted out by date so, when exporting the dataset, it was necessary to indicate to SQL Server to export the data to a plain text file (in CSV) through a SQL sentence which orders all records by date.

The same process was carried out with the Bonanza and Huelva tide gauge datasets. In the case of the Bonanza dataset, 2708772 records were created (in 14 seconds), while in the case of Huelva, 2275356 records were created (in 11 seconds).

Before using the CSV files created, it was important to check that the character used for the separation of metrics is the adequate one (',' instead of ';') in order to avoid errors. Therefore, in some cases, it was necessary to replace the character (due to the wrong correction of character selection when exporting the data).

3.2.2. Benchmark Datasets

This group of datasets comes from the work done in (de Bruijn et al., 2016). It contains real metrics measured in different locations and with different systems (indoor Intel system, measures from Smart Santander and outdoor measures with a SensorScope system), and it contains adapted datasets with failures injected according to the hypothesis of the presented work. This is an interesting dataset because it provides real measures from different sensors, and it also contains annotated datasets that contain all the errors that we may expect in an IoT system. Therefore, it is good for analysing the applicability of different solutions and how each one of them work for each kind of error.

3.2.2.1. Information Provided and Data Acquisition

This group of datasets provide not only the raw data that was collected from several systems, but also some annotated datasets where concrete errors were injected.

In the case of the metrics collected with the Intel system, it was connected to 10 indoor nodes (able to measure temperature and light intensity) providing data every 30 seconds, for one week. Each file provides information about the following aspects:

- timestamp: the timestamp of the metrics provided, in standardized format (yyyy-mm-ddTHH:mm:ss)
- mote_id: identifier of the mote generating the data;
- has_fault_type: value describing whether it is a faulty value and which kind of fault;
- temperature: the temperature measured in ºC;
- light: light intensity measured in lux.

Another block of datasets comes from metrics taken from the Smart Santander system. There is data from 16 outdoor nodes measuring temperature. The raw data contains metrics generated every 5 minutes, during almost one month. Each file provides information about the following aspects:

- timestamp: the timestamp of the metrics provided, in standardized format (yyyy-mm-ddTHH:mm:ss)
- mote_id: identifier of the Santander node generating the data;
- has_fault_type: value describing whether it is a faulty value and which kind of fault;
- temperature: the temperature measured in ºC.

The last group of datasets contain metrics coming from an outdoor SensorScope system (Ingelrest et al., 2010). Such system was connected to 23 meteorological stations that were producing data every 2 minutes, for one month and a half. It is interesting to highlight that the raw datasets contain the following information:

- Ambient temperature, measured in °C;
- Surface temperature, measured in °C;
- Solar radiation, measured in W/m^2;
- Relative humidity, measured as a percentage;
- Soil moisture, measured as a percentage;
- Watermark, measured as kPa;
- Rain meter, measured as mm;
- Wind speed, measured as m/s;
- Wind direction, measured as degrees.

Only 10 of these meteorological stations were used for preparing the benchmarking datasets, taking only the data from the ambient temperature metric. Therefore, the information contained in the files is the following:

- timestamp: the timestamp of the metrics provided, in standardized format (yyyy-mm-ddTHH:mm:ss)
- mote_id: identifier of the meteo station generating the data;
- has_fault_type: value describing whether it is a faulty value and which kind of fault;
- temperature: the temperature measured in ºC.

In this case, although this work is focused on using the datasets prepared for fault analysis, it would be possible to use the raw data from the SensorScope system to analyse additional types of sensors, that are very relevant for the agriculture domain.

3.2.2.2. Format of the Data

This is a group of datasets in CSV format (although their extension is set as .txt), each one representing the measures of a single mote (or sensor). They are separated in several folders (Intel, Santander, SensorScope), each one containing only the measures from the

corresponding system. Inside each folder, there are other three folders, representing the kind of data we can find:

- raw: it contains the data as it was collected directly from the system;
- interpolated: it contains the full data clean (applying interpolation), as well as datasets with errors injected (random, malfunction, bias, drift, polynomial drift and a mixture of errors);
- non interpolated: it contains the full data clean (without interpolation), as well as datasets with errors injected (random, malfunction, bias, drift, polynomial drift and a mixture of errors).

Each folder contains another folder with the clean data and one folder per type of error injected. The names of the files describe the type of error injected and the identifier of the mote (e.g., mote='2'_sensortype=temperature_faulttype=drift.txt). Each file contains the header row (except in the files collecting raw data) with the names of the fields and then the values separated by commas, with one row representing one set of metrics received, the corresponding timestamp and information about the faulty values.

The information of faulty values is encoded as follows (according to the readme document delivered with the data):

- No fault = 0
- Random = 1
- Malfunction = 2
- Bias = 4
- Drift = 8
- Polydrift = 16
- Polydriftnoise = 32

3.2.2.3. Pre-processing of the Data

Since the datasets provided are in CSV format, it was not necessary to do any specific pre-processing of the data. It was possible to use them in the same way as they were provided.

3.2.3. Custom Arduino System

Another data source that has been used is a small home IoT installation, based on Arduino, focused on home monitoring, able to collect information about temperature, humidity and moisture of a small plant. The main purpose of this experiment was to have a controlled environment in which it was possible to test new types of sensors (like the hygrometer for soil moisture and air humidity). The controlled environment allowed to annotate the dataset created when certain changes in the room happened. For instance, it was possible to control the air conditioning machine, or when the plant was irrigated, and see how the sensors were reacting in a context of normal operation. The changes that were registered should be useful to validate the solutions, differentiating 'natural' outliers from 'artificial' ones.

3.2.3.1. Information Provided and Data Acquisition

The custom system that was built contained a few sensors that would be useful in the case of a homemade system focused on monitoring some plants and their context. The system was active collecting data for one week, working 24/7. A metric is generated every 2 minutes for all the sensors in the system. The system was composed by:

- 1 x Arduino Mega 2560 R3 board
- 1 x protoboard
- 1 x Hygrometer sensor and module with LM393 chip
- 1 x DHT11 sensor for temperature and humidity

The Arduino application has a loop that configures the sensors and the serial connection with a laptop through the USB cable. Every two minutes, the system collects the current metric of all the sensors, creates a line of text containing the information of the metrics with a timestamp and sends such line through the serial connector, sleeping for other two minutes.

At the laptop side, the Realterm application is continuously monitoring the serial port of the corresponding USB connector, capturing all the text that arrives and writing a CSV file with the information received. Each time it received data through the serial connection, it wrote a new row in the CSV file.

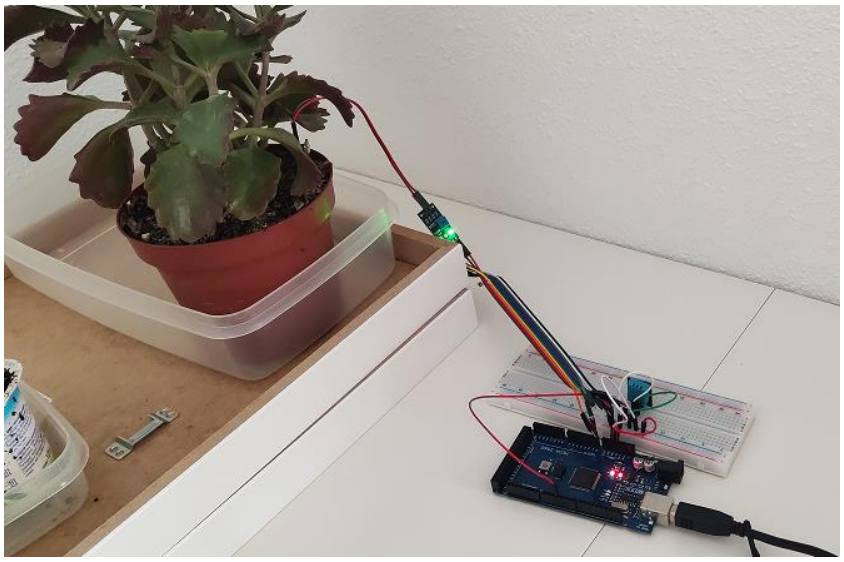

Figure 4 Custom Arduino System

The dataset generated contains the following information:

- time: the timestamp of the metrics provided, in standardized format (yy-mm-ddTHH:mm:ss)
- hygrometer: raw value of the hygrometer;
- moisture: moisture percentage calculated based on the maximum, minimum and actual value of the hygrometer (calculated by the Arduino program);
- air_humidity: air humidity percentage as provided by the library that collects the data from the sensor;

- tempC: air temperature as provided by the library that collects the data from the sensor (in Celsius);
- tempF: air temperature as provided by the library that collects the data from the sensor (in Fahrenheit);
- error: flag indicating whether there is an error or not in the metrics;
- annotation: text explaining if there was any special circumstance that could affect the measurements taken (such as manipulation of the sensor).

While the data was gathered, the context of the experiment was also recorded. This included whether the air conditioning machine was turned on and off and those moments in which the plant was irrigated. In one of the cases, the hygrometer was moved from its original position, so it would be less exposed to the water when irrigating. Also, the sensors were vandalized for a few minutes, in order to generate some outliers that could be useful for the study. Such actions were recorded using the 'annotation' field.

3.2.3.2. Format of the Data

The dataset is captured in CSV and, since the data is sent through the serial connection in the adequate format (with comma separated values), the file that is generated by Realterm is a normal CSV file that does not require specific adaptation. It contains the header row with the names of the fields and then the values separated by commas, with one row representing one set of metrics received (with the corresponding timestamp).

3.2.3.3. Pre-processing of the Data

Since the dataset already generates a CSV file, it was not necessary to do any specific pre-processing of the data. It was possible to use it as it was generated.

3.2.4. Physics Toolbox Suite

Our mobile phones are machines full of sensors that are used by the different applications in order to show information correctly, to know if we are moving (and how), to know our position, to adapt the intensity of the screen or even to remove noise in the background when we are talking. Therefore, they are a very interesting source of data about different types of sensors which are relevant to analyse, since they are present in many other systems and because they might represent a good case for applying the solutions proposed in this work.

There are multiple mobile applications for Android phones that can be used in order to collect information from the sensors integrated in the phone. Each one has its own limitations (due to the large heterogeneity of devices) and, after some analysis, Physics Toolbox Suite was selected. Such tool is able to collect metrics from most of the sensors, it generates good graphs, and it can generate CSV files for free.

3.2.4.1. Information Provided and Data Acquisition

The Physics Toolbox Suite application has been used with two different models of Android phones: a Samsung Galaxy S6 and a Xiaomi Mi Lite 11.

The metrics that the application can collect are the following:

- G-Forces
- Accelerometer
- Gyroscope
- Sound
- Spectrogram
- Luminosity
- Color detector
- Magnetometer
- Barometer
- GPS
- System temperature

Since the purpose of this work is to cover several sensors, the data collection for this case was focused on the data provided by the accelerometer (in three axes), the luminosity and the sound. The devices used do not include a barometer, so it was not possible to collect air pressure information.

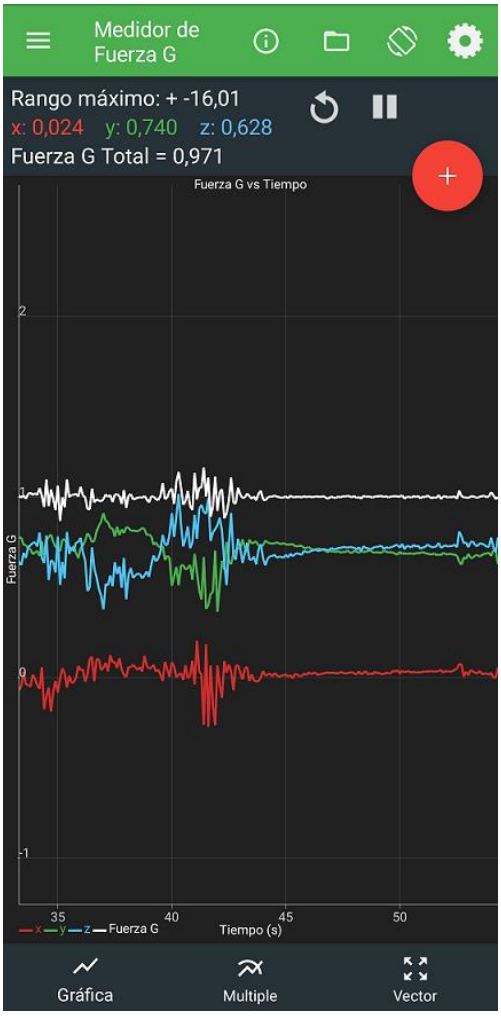

Figure 5 Physics Toolbox Suite Screenshot

For those metrics collected, the data collection feature was activated during several hours. Sometimes, data acquisition was done indoors and in other cases outdoors, since some of the sensors would behave differently (e.g., luminosity and sound). In the case of accelerometer, the device was still on a table most of the time, moving it in a natural way, as it would happen normally. In all cases, there were situations with 'natural' outliers (i.e., a message arriving that generates movement or a shadow of a person moving close to the mobile phone), while in other cases some outliers were generated on purpose (shake the phone, talk directly to the phone or cover the luminosity sensor for a few seconds). Such outliers, although 'generated', represent realistic situations that may happen.

In those cases in which some outlier was happening, the dataset was annotated, so it is possible to distinguish what happened to the dataset.

3.2.4.2. Format of the Data

The dataset is generated in CSV directly by the application. It contains the header row with the names of the fields and then the values separated by commas, with one row representing one set of metrics received (with the corresponding timestamp).

3.2.4.3. Pre-processing of the Data

Since the dataset already generates a CSV file, it was not needed to do any specific pre-processing of the data format itself. Still, because of the high frequency in some sensors, it is necessary to deal with the timestamps of the dataset. It may happen that there is more than one metric per second for some cases, but certain functions in R, when using the metric together with the timestamp (for instance, with the ggplot2 library to create graphs), require that the time field is adapted, so it will be compatible with the POSIX format.

The way to solve this was to create R code that sets the time with milliseconds in the right format (yyyy-mm-dd HH:MM:ss.ms) and then, using the ymd_hms() function, it is converted to POSIXct object representing the full date.

Additionally, since some of the metrics had a lot of zero values (e.g. indoors luminosity), some of the mechanisms used for analysing data variability were failing. For instance, determining how data might fit with a gamma distribution fails because of these values. Therefore, in those cases in which the data was showing a lot of constant zero values, one unit was added to the full series. Applying such modification does not alter the properties of the time series at all (since it is only moved up as a block), so its variation is the same while it solves the calculation issues in the numerical libraries.

3.2.5. Gyor Air Quality Sensors

In the context of the H2020 HiDALGO project (https://hidalgo-project.eu/), one of the pilots focused on simulating Urban Air Pollution in several cities. As part of the validation, some measurements about air quality in the city of Gyor were collected, so they could be compared with the simulated values. Some of the metrics collected have been used to analyse the behaviour of sensors related to air quality.

The data was collected with Bosch sensors. The dataset available provided information from five sensors located in the city during one week. Sensors are separated around 900

meters each other, and they are located in different areas of the city in a homogeneous way (so not many sensors are concentrated in an area).

3.2.5.1. Information Provided and Data Acquisition

The metrics provided by the Bosch station were collected every minute. Measures for the following metrics are provided:

- Temperature average
- Air pressure
- NO_2
- O_3
- PM10
- PM2P5

These sensors have been working in an operational way for many months, so they provide information about what happens in the city and there are no artificial outliers generated. There are seven separated files with data, each one providing data for an entire day (from 1st January 2021 until 7th January 2021), for the five sensors available.

3.2.5.2. Format of the Data

The dataset is provided in CSV format, without additional information. It contains the header row with the names of the fields and then the values separated by commas, with one row representing one set of metrics received (with the corresponding timestamp). The last column of each row includes the device identifier, so it is possible to distinguish and filter the source of the data.

3.2.5.3. Pre-processing of the Data

As the data was separated in several files, first of all, it was necessary to put all the data together in a single CSV. Since the implementation of the R scripts were considering files produced by a single sensor, the data was separated in different files, each one for each of the five sensors available. A simple filtering by 'device identifier' in Excel was enough to separate the data as needed.

Finally, in order to facilitate the usage of the data in R, another field was added to the CSV files, including just an integer representing the number of the metric in the full list of metrics. Although the 'UTC' column could be used for this purpose (as it represents the timestamp), the new column facilitates some operations related to the date processing (such as when creating graphs with the ggplot2 library).

3.3. Discussion

This chapter has shown that the access to datasets including information for different types of sensors is not simple and it requires to be addressed specifically. It is impossible to obtain all the information that this work may need from the same source, so the only way to proceed is to collect information from multiple sources (that also represent different application contexts). Some sources were related to the agriculture domain, while others

were related to maritime monitoring or air quality in urban areas, even if they were taking some common metrics (e.g., temperature and air pressure). In all cases, data was presented in a different way (although the usage of CSV is extended, it did not happen in all cases and some transformation was necessary).

Heterogeneity is not only in the format side, but also in the way the information was collected. While some systems were collecting data every 30 minutes, others were collecting data every second (some had even higher frequencies). This is an aspect to take into account as well, since the number of measurements available for analysing the data is relevant. In some cases, it may limit the kind of approach to apply (since a minimum number of metrics is required) while, on the other hand, it also affects the results of some statistics.

Another interesting aspect is the location of the sensors. Although this is usually known, there may be datasets in which this is not the case. In the case of sensors located indoors, this might not be so relevant (the main interest would be whether more sensors are located in the same room or if there were elements that might affect the sensor behaviour, like air conditioning systems), but it may be more relevant for outdoor sensors, since they might be exposed to more elements altering their behaviour and it is a crucial information to determine whether two sensors could be considered equivalent. This variety enables the analysis of multiple contexts and the study of the applicability of correlation as a valid aspect for determining how a sensor is working.

All in all, it is also important to highlight the number of types of metrics available, so it is possible to carry out a deep analysis with a good variety of sensors. In some cases, there are even metrics taken for the same type of sensor indoors and outdoors, so it facilitates the possibility to understand potential differences when the context changes. This is necessary in order to design a solution that will be applicable to multiple environments, unlike other solutions that are focus on a single context (e.g., ML models tailored to a concrete industrial context).

Still, even if there is a good collection of measurements, it is important to admit that not all the environments are covered with the datasets available, so it is not possible to guarantee that the solutions proposed are generic for any context. As an example, temperature may behave different in an industrial environment working at very high temperatures. Since this type of dataset was not available, although the evaluation will look at the possibility to extrapolate the models, it is hard to determine to which extent the models can be applied successfully in any environment.

4. ANALYSIS OF SENSORS' DATA

We are drowning in information but starved for knowledge.

John Naisbitt

This section analyses different sensors and their characteristics, trying to understand how they behave and how it is possible to interpret information that is extracted from the data. Not all the sensors are influenced by the same factors, and the same kind of sensor (e.g., temperature) may behave different indoors and outdoors. Even having the same sensor outdoors but in different locations may produce different datasets.

As explained in Section 2.1, there are different types of errors in sensors and these might affect different properties of the dataset, such as the variation of the data and the role of the outliers that could be found. Therefore, it is necessary to understand the normal behaviour of different sensors, as well as how they are affected by the presence of outliers. Whenever possible, it is also important how certain statistics are influenced by the presence of different errors, so it will be possible to determine how to address the problem of detecting abnormal behaviours.

As part of the experimental research, a visual analysis of the data for different samples size was done, together with some basic statistics (RQ1). The objective of such analysis was to understand how the metrics evolve in time (if there are large jumps, or constant values, etc.), looking at some basic statistics that usually affect many calculations and tests (such as mean and variance).

Then, an analysis of the variation in the datasets was carried out, as well as of the kind of distributions they follow (RQ1). Many sensors do not seem to follow a normal distribution and that might be an issue for applying certain solutions. Additionally, it was necessary to understand how some statistical tests work with the sensors, to determine which ones could be applied and how, and how they are influenced by the presence of certain errors (RQ2, RQ3, RQ5).

A similar approach has been followed for the detection of outliers. Part of the experimental research was to go through different solutions for outliers detection that could be generic, but perhaps not applicable (or not accurate) in the case of datasets generated by sensors (RQ2).

As the analysis of correlation may be promising, this approach was included, as well as some specific tests in the experimentation, looking at how it would work in the presence of errors and outliers, as a solution that could be complementary to others (RQ4, RQ5).

4.1. Basic Information

This subsection is focused on analysing visually the metrics, as well as some basic statistical aspects of the datasets (like average, variance, maximum value, minimum value, mode, coefficient of variation and interquartile range) for different types of sensors, with the purpose of understanding how they vary when using different sizes of the dataset and what can be expected. With such information, it is possible to extract some reference values that can be used to configure the detection of some anomalies, while it is also possible to understand which periods could be more adequate for the detection certain problems in the data. Additionally, it is also interesting to compare the results when analysing datasets providing the same metric, but from different sources so, for instance, this work compares the temperature measures of a sensor in a buoy against an indoor sensor in a homemade system.

4.1.1. Air Humidity

The information from air humidity has been extracted from the Arduino system, which is located indoor. It is possible to observe that there are moments in which the values are varying at a large scale. Usually, this is related to the hours in which the air conditioning machine is working, or when windows are open.

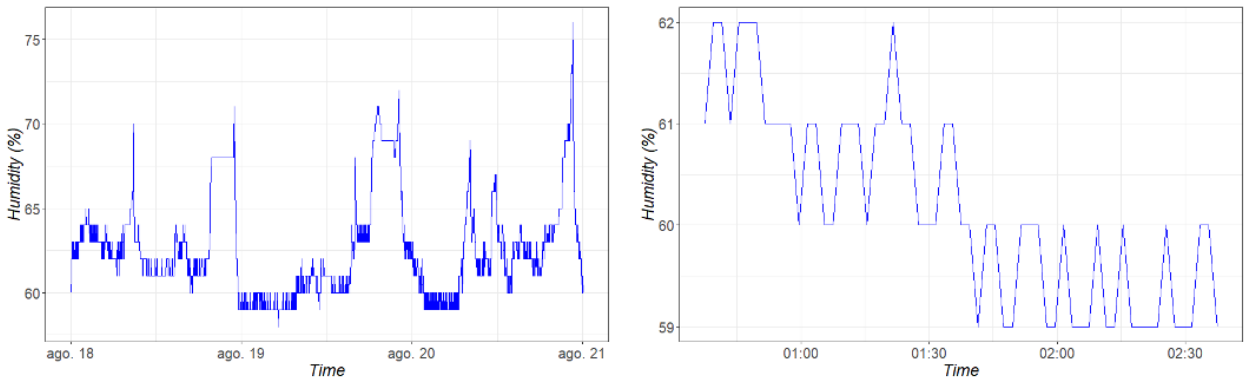

Figure 6 Air humidity sensor measurements plot from the Arduino system

The basic statistical analysis shows that the values do not vary too much when analysing data for a sample of 6 hours or less and the mean is quite stable in general. Observing the data, depending on the sample selected, it may happen that some values are identified as outliers in some concrete moment, although there are not many jumps in the data, and it is continuous in general (as it may happen with temperature). Since some of the situations that generate variation in the data would not happen outdoors (e.g., switch on air conditioning), we could expect even a more stable behaviour in such context. The Coefficient of Variation (COV) is lower than 0.7 and the Interquartile Range (IQR) is very stable between 5 and 0.5 as well.

4.1.2. Air Temperature

Temperature is a very common metric taken in many systems. Therefore, it was possible to collect temperature metrics from several sources. As the data sources are heterogeneous (in

location, sensor models and frequency), each one has been analysed separately.

4.1.2.1. Ports of Spain

In the case of Ports of Spain, there are two datasets providing information about air temperature: REDEXT Golfo de Cádiz and REDCOSM Puerta Carnero. As mentioned in the previous section, these sensors are outdoors, and they might be influenced by the buffer effect from the sea water.

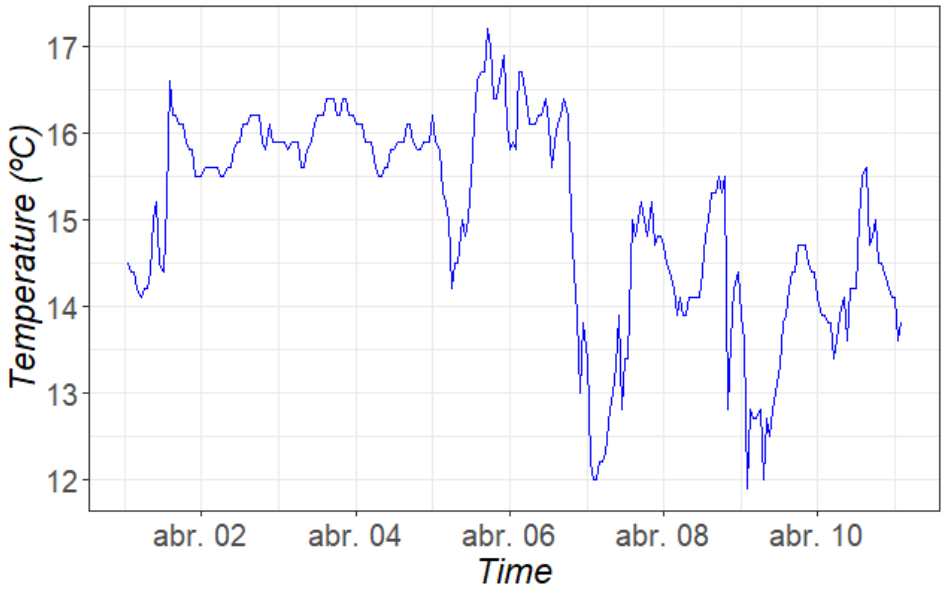

Figure 7 Air temperature measurements from Ports of Spain

The observation of the basic statistics shows a rather low variance, especially in the time frame of one week and less (in one of the cases, 3 days or less). Then, the variance increases in a meaningful way. Because of the important influence of the meteorology and the expected seasonality in general, it is normal such variance. It is not usual to find abrupt jumps in the data since the temperature varies smoothly in time.

4.1.2.2. Benchmark Datasets

The benchmark datasets include air temperature measurements from several nodes and platforms. When analysing their behaviour, one indoor sensor (from the Intel platform) and two outdoor sensors (one from the SensorScope platform and another one from the Smart Santander platform) were selected. Since the same datasets are available, but with errors injected, the statistics of the original dataset were compared with the statistics of the adapted ones. For the Intel platform measures, the dataset with drift error was used, for the Smart Santander the one with bias error and for SensorScope the one with malfunction error. The figure represents the Intel sensor measurements, highlighting in red the injected errors in the dataset with failures.

The sensor from the Intel platform shows lower variance, since those sensors indoor are more protected than those outdoors and the range in which they work is smaller. Temperature is not a metric that shows large variances for medium and short samples. Still, it is interesting to see how the number of measurements also affect the variance itself, as it is observed when comparing with the dataset from Ports of Spain (that has a variance of 0.286 for 3 days with a range from 17.2ºC to 19.1ºC, while the indoor sensor shows a

variance of 4.248 and a range between 17.62º and 17.95º). Therefore, it is necessary to take into account the frequency in the collection of data when using variance.

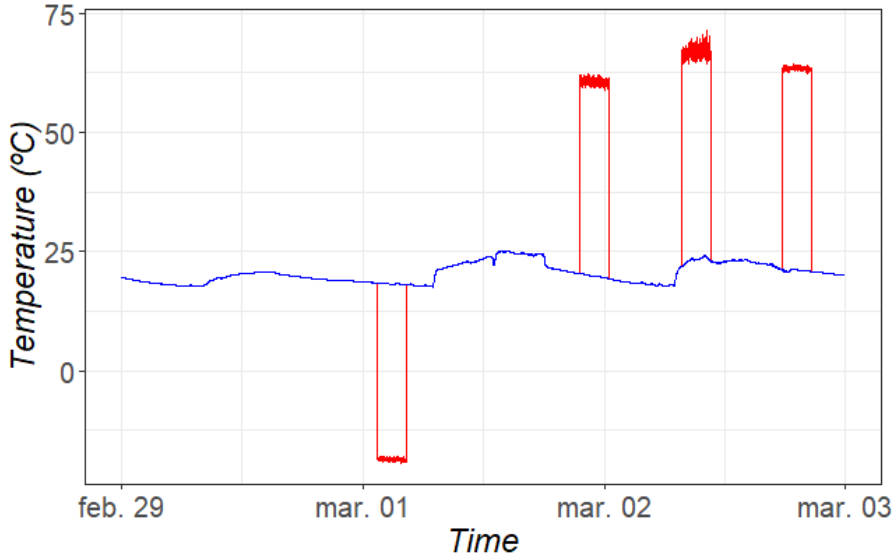

Figure 8 Indoor air temperature measurement from the Intel platform, with drift error

In this case, the coefficient of variation requires some transformation in the data (because of the negative values). Measures in Kelvin degrees are compliant with the coefficient of variation, so it is possible to move all the values up in the graph, just summing up the adequate value (similar to what is done to transform from ºC to K). After transforming the data, the coefficient of variation is low for the indoor sensor (less than 0.047, except in some concrete cases), while for the other sensors is not high neither (less than 0.138 and 0.60).

Finally, it is very interesting to analyse the results when a failure is injected in the data. In the case of Intel and Smart Santander datasets (injected with drift and bias errors) there was an evident modification in the variance of the data. In the case of drift error, variance raised from 4.24 to 276.6 for 3 days of data (with coefficient of variation going from 0.1 to 0.38), while the bias error raised variance from 7.07 to 45.26 in one week data (with coefficient of variation from 0.13 to 0.29). In the SensorScope platform there was not an important difference when the malfunction error was present. It was adding outliers, but it was not displacing the measurements too much (like in the other cases), so the change was rather small (e.g., variance raising from 21.99 to 22.13 for one week of data).

4.1.2.3. Custom Arduino System

In this case, there was only one dataset produced by the custom Arduino system that was collecting data during several days. Still, it is interesting, since the system stored the data in ºC and ºF, so it is possible to observe the differences in the statistics when changing the unit of the measurement.

It is possible to observe that there is a clear seasonality in the data for a few days, since it represents periods in which the air conditioning was switched on (it was activated most of the days at the same time), periods at night and other moments during the day in which the air conditioning was switched off.

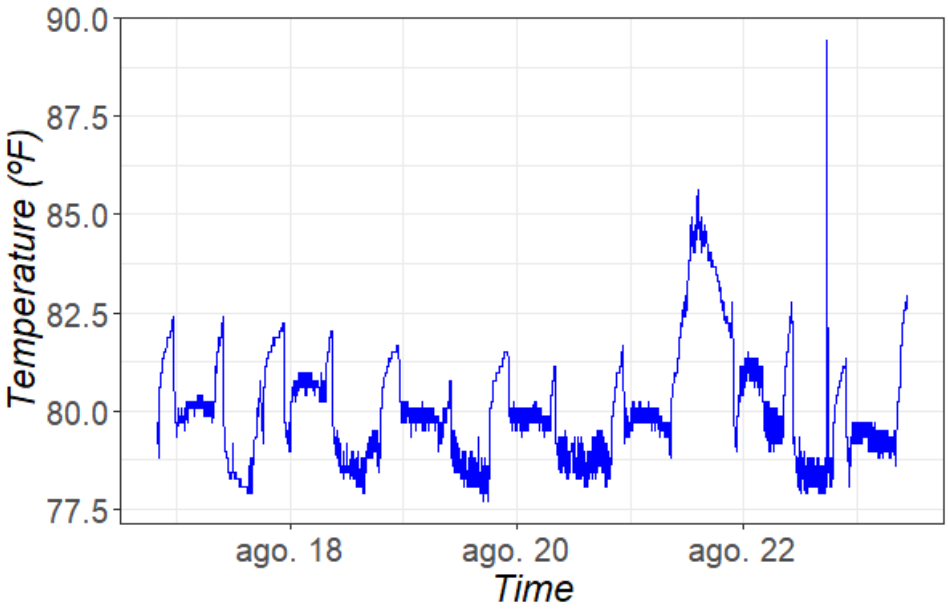

Figure 9 Air temperature measurements from the Arduino system (ºF)

Even if it is clear that there were moments with less stability (as the temperature was increasing because of the temperature outdoors heating the room or the air conditioning at full power cooling the room), in general there is not much variance (less than 1 even for 3 days of data), but it is interesting to observe how the unit used for the measurement affected this statistic. In general, the variance measured in ºF was between 3 and 4 times higher than when it was measured in ºC. On the other hand, COV and IQR show similar values, no matter the unit used with values below 0.03 and 1.8 respectively.

4.1.3. Atmospheric Pressure

The atmospheric pressure is a meteorological aspect that seems to be quite stable in time and in large areas. In this case, all the datasets available were part of the Ports of Spain datasets.

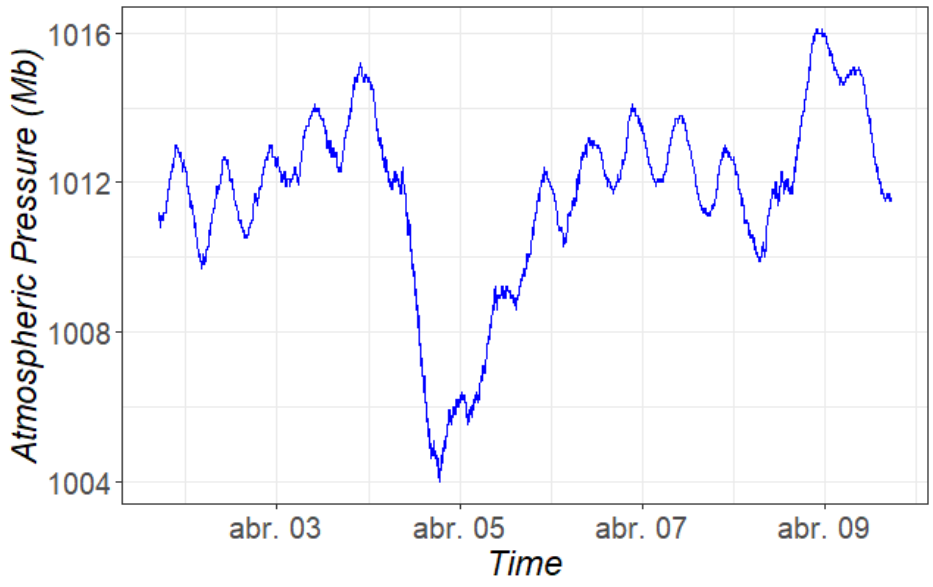

Figure 10 Atmospheric pressure measurements from Ports of Spain

This kind of metric shows a lot of stability, with low variance in time, even for an entire week (with a value of 0.6157). There is some variation in time, but there are very smooth transitions in values going up and down, being always in a limited range of values (so IQR is around 3 and COV around 0.003).

It is very interesting to highlight how the units used for expressing the measurements affect its basic statistics. In the case of Dique Abrigo and Golfo de Cádiz, measures are in Mb, while in the case of Pta Carnero measures are in Hpa (that corresponds to Mb/10). When looking at the basic statistic tables, while Pta Carnero shows a variance of 0.19 for a year, Dique Abrigo shows a variance of 13.35. Even if it could be more stable in the first location, still it is several orders of magnitude higher, and this happens as well comparing with Golfo de Cádiz measures.

4.1.4. Light Intensity

Light intensity is a kind of sensor that is possible to find in smart city platforms and solutions focused on domotics. This work analyses several datasets that collected data about light intensity. From one side, the Physics Toolbox Suite captures light intensity with one of the sensors in the mobile phone. Two datasets are available, one measuring indoor light intensity and another one outdoor intensity. On the other hand, in the case of benchmark datasets, one of the platforms (the one from Intel) also contains light intensity data, together with datasets in which failures were injected.

The figure shows the values measured by the indoor light sensor connected to the Intel platform. The blue line represents the original measures, while the red line represents the injected errors (in this case, a random error, similar to a malfunction one).

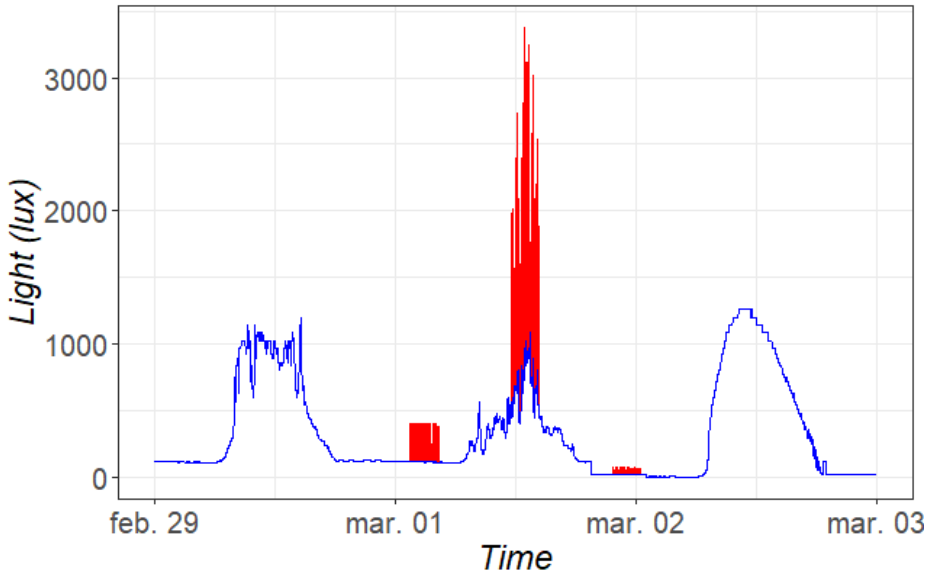

Figure 11 Light intensity measurements from benchmarking datset with random error

When looking at the basic statistics, this is one of the sensors that shows the higher variance in time, that decreases a lot when looking at samples of six hours or less. This happens because of the difference of light intensity between day and night, as the average value is very far from both the normal values in night and the values during the day. Looking at the

coefficient of variation, while large samples have a coefficient around 1 (between 1.11 and 0.95), short samples have a small coefficient (between 0 and 0.02). Therefore, the analysis of datasets will be more accurate when using short samples for light intensity.

In fact, values are very stable during the night, while during the day it is normal to find some trend and more potential outliers (depending on where the sensor is located). It is true that also some outliers can appear during night (e.g., a light is switched on) but, due to the stability in the data, they are easier to detect.

When looking at the effects of outliers, it is possible to see how they increase the variance drastically (in the case of the coefficient of variance the increase is not so high, but it shows a difference, reaching 1.21). It becomes more evident when the error happens during night, as the variability introduced by the faulty data is large. For instance, when calculating only variance and coefficient of variation for the first error in the figure, variance increases from 5.686 to 6881.912 and coefficient of variance goes from 0.021 to 0.574.

4.1.5. Particulate Matter (PM) 10 in Air

Particulate matter (PM) is also known as particle pollution that are in the air, including small solid particles and liquid droplets. In the case of PM10, these are small inhalable particles with a size of less than 10 micrometres, such as pollen, molds or dust. This is an unstable metric that varies a lot in time, as it can be seen in the figure (that shows only 12 hours of data).

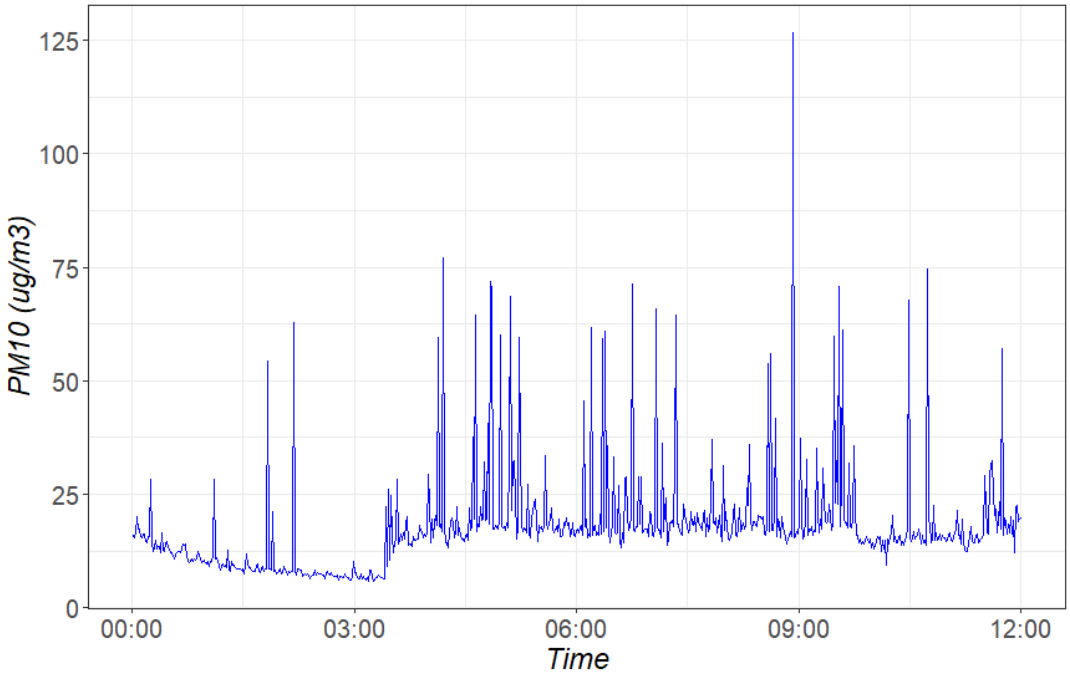

Figure 12 PM10 measurements from the city of Gyor

While the mean does not vary so much, the variance goes from 9.25 (30 minutes) to 391.13 (1 week), so it is very difficult to deal with its variability, although looking at the coefficient of variation the information becomes more stable. Its values are between 0.73 and 0.58, except for short samples (one hour or less) that gives values around 0.20. PM10 also shows clear peaks from time to time. Some of these peaks could be detected as outliers and,

although it might be natural, it is hard to determine if they were caused by a problem. In any case, raising an alert with these outliers is something that may be useful in all cases.

4.1.6. Precipitation

Precipitation is an interesting metric for agriculture and, together with soil moisture, can provide farmers valuable information. This is a very particular metric, since most of the time it takes the value 0.0 (depending on the area and the month of the year, it may not rain for days or weeks). Therefore, it shows very low variance (e.g., 0.1123 for one week of data), although the coefficient of variation is rather large (with values between 6.05 and 6.8 for large samples, and even 13.4 in some case).

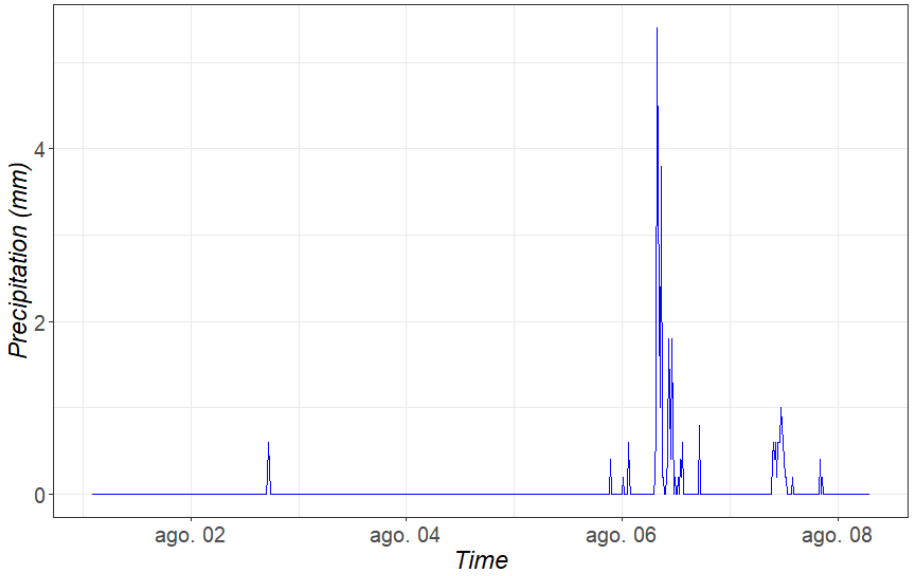

Figure 13 Precipitation measurements from the Meteoblue dataset

Moreover, since rainfall appears suddenly and it is irregular, every time the sensor detects values, they look like outliers (there are clear peaks in the graphs). In some cases, this is maintained in time (a few hours), but in the usual case, there might be problems to distinguish certain failures from the normal outliers. Perhaps drift and bias errors can be detected correctly, but malfunction errors are difficult to categorize. Using short samples could be a good option, so the measurements in rainy days are amplified with respect to the rest of the dataset, reducing their aspect of outliers.

4.1.7. Sea Level

Ports of Spain has provided several datasets that include information about sea level. As we appreciate in the *Figure 14* (showing two weeks of data), there are clear cycles that we can associate to the natural tides, that also vary according to the Moon. Therefore, it is necessary to assume that there is a variability that requires some specific understanding (e.g., calculating the coefficient of variation, instead of the variance, as it is stable between 0.20 and 0.39, in general). IQR is not small (between 29 and 43), but such ranges are stable. Also, outliers detection may be problematic, because of the clear seasonality of the data. Any analysis would require transforming the data to remove such seasonality.

When looking at the numbers from the two datasets available with such metric, even if the both of them are measuring the metric with the same unit, they show very different values for mean, maximum, minimum and variance. Therefore, the conclusion is that the location of the sensor is very relevant for this kind of metric, as some places (like ports) could be more protected than others from winds and currents that could affect the sea level.

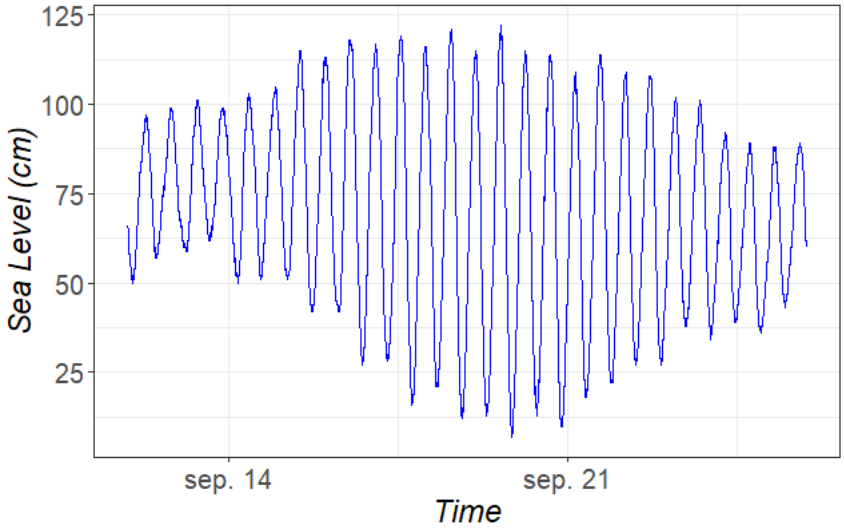

Figure 14 Sea level measurements from Ports of Spain

4.1.8. Soil Moisture

The measurements analysed for soil moisture were obtained with the Arduino system connected during several days, controlling the soil for a small plant that was irrigated periodically. The figure clearly shows the periods in which the soil was irrigated, and how the moisture was decreasing as time was passing.

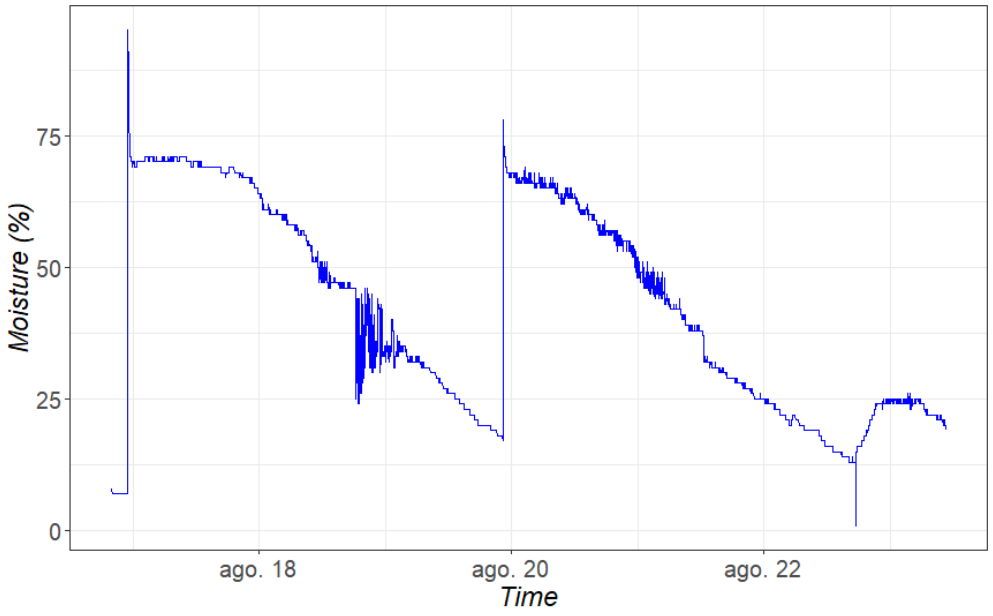

Figure 15 Soil moisture measurements obtained from the Arduino system

This kind of sensor shows a clear seasonality, but which depends on the moment in which the soil is irrigated. If this is an automated process, clear cycles are expected to appear. Otherwise, it would not be so easy to determine the periods in the data.

As for the basic statistics, the variance for long periods is quite large (as the metric will take all the possible values, from 1% to 95%), but for periods of less than six hours, the variance is really small. The mean is almost always very stable, around 66%. Therefore, although it will vary all the time (normally decreasing its value), it is rather stable, except when irrigation is done, as such action generates a peak that could be interpreted as an outlier.

4.1.9. Water Current Speed

This is a rather uncommon metric, measured only in a few systems dedicated to sea monitoring. Therefore, only two datasets provided by Ports of Spain were analysed. The figure shows metrics for one complete week, so it is possible to observe that it is not a stable metric, showing peaks up and down from time to time. Since it seems there are no clear cycles, it is very difficult to eliminate seasonality. It looks like white noise, and it may be problematic for detecting outliers, except for short samples (one of the datasets shows low variance for one day of data).

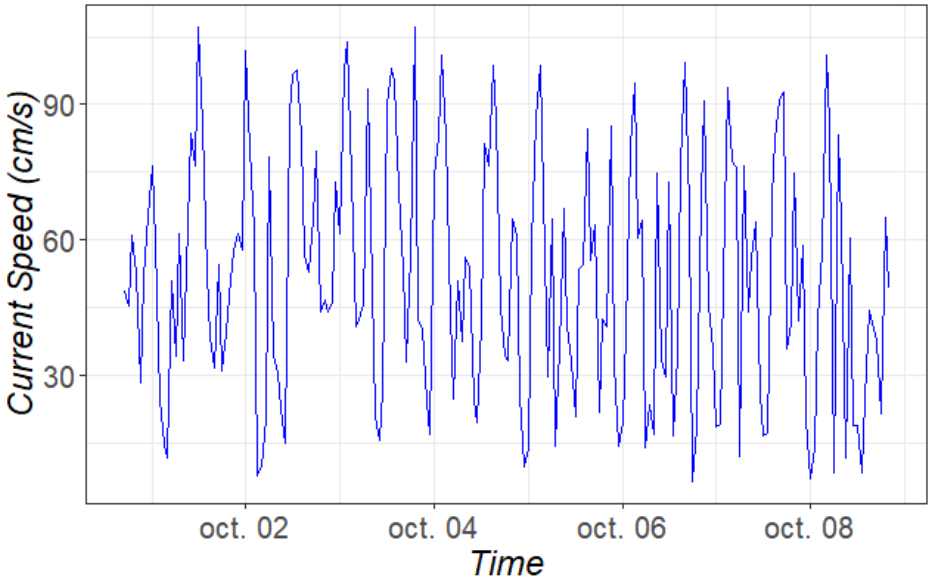

Figure 16 Water current speed measurements obtained from Ports of Spain

Variance, anyway, is not a clear reference value. Currents are faster in some areas and the datasets available show such difference. Therefore, they both vary a lot, but one of them shows higher variance for certain periods (like one year or in short samples like those of more than 2 hours). As variance increases a lot when adding measurements, coefficient of variation, again, could be more adequate as it is between 0.54 and 0.63 (and IQR is between 18.95 to 26.12.

4.1.10. Water Salinity

As with other metrics, water salinity is measured only in a few systems focused on sea monitoring. Therefore, only one dataset provided by Ports of Spain was analysed.

While this metric shows a very low variance in general (0.004 is the largest one calculated), it seems it takes very few values in time, having long period of constant measures, that then jump to another one with some constant values again. COV and IQR are stable all the

time around 0.001 and 0.09 respectively, only increasing when failures are present.

It is very stable in time, but it may give problems for some analyses in case we analyse a sample which only contains constant values. On the other hand, it might be easier to detect outliers, due to its stability, although some jumps in the data could be categorized as outliers when they are not.

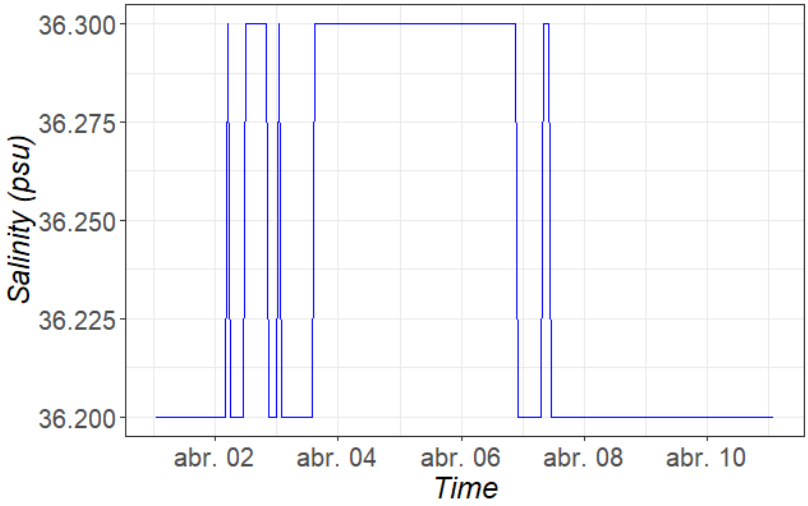

Figure 17 Water salinity measurements obtained from Ports of Spain

4.1.11. Water Temperature in the Surface

All the datasets analysed providing information about water temperature are from the datasets provided by Ports of Spain. Since the water acts as a kind of buffer for temperature, we expected the metrics to be more stable than those from air temperature.

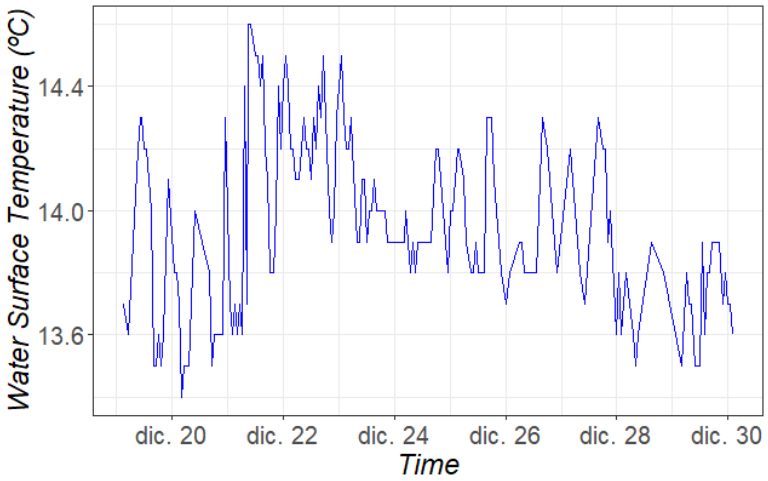

Figure 18 Water temperature measurements obtained from Ports of Spain

Unlike the air temperature, the measurements are very stable, to the extent that they may remain constant for many hours. In fact, when looking at the tables of the basic statistics, variance is very low even for samples with a month of data (0.5019 in the highest case). Even with six months of data, variance is not high (2.2087), while in the case of air temperature, for the same buoy, it almost doubles the variance (4.1171). As expected, their maximum and minimum values are also different, in such a way that a lower IQR is observed (less than 0.5), although COV is similar in general.

4.1.12. Wind Speed

All the datasets analysed providing information about wind speed come from Ports of Spain. The datasets providing information about this kind of sensor are: REMPOR Dique ExSUR, REMPOR Dique ExNORTE, REMPOR Endesa, REMPOR Dique Abrigo, REMPOR Campamento, REDEXT Golfo de Cádiz and REDCOSM Puerta Carnero.

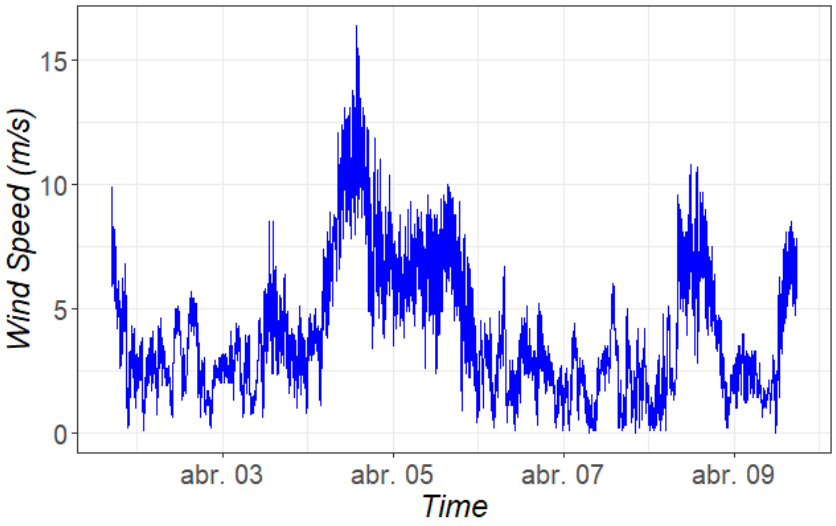

Figure 19 Wind speed measurements obtained from Ports of Spain

Even for one week of data this type of metric shows that it varies a lot. Variance grows more than in other sensors beyond one day of data. When using less than one day of data, variance is around 1. On the other hand, COV (from 0.45 to 0.66) and IQR (from 2.3 to 3.6) are quite stable even for a month of data. Additionally, it seems that there are not large jumps in the data, so the outliers detection mechanisms could be applied not expecting too many false positives.

4.2. Data Distribution and Variability

4.2.1. Analysis of the Data Distribution

Analysing normality of the different sensor types is important, since some of the statistical tests to be used are based on the assumption that the metrics taken follow a normal distribution. In fact, in some cases, it is necessary to use different statistical tests when data is not following the normal distribution (i.e. when analysing means, Wilcoxon test should be used instead of a T test). Additionally, this information can be also useful in order to identify the kind of sensor we are dealing with, whenever this information is not available (as a kind of profiling activity).

Therefore, this subsection addresses the analysis of the normality of data for the different sensor types that have been studied. In order to do so, this work uses several tests, so it is possible to confirm or discard normality using different approaches.

First of all, taking a look at the visual aspect of normality is very useful, since it provides a very good idea about how close the dataset is to a normal distribution by using Q-Q (Quantile-Quantile) plots, as well as a histogram of the dataset (together with a normal

probability curve).

Then, it is also possible to use a numeric approach by using several statistical tests. According to (Jeffery et al., 2006), the Shapiro-Wilk (SHAPIRO and WILK, 1965) test is the most robust one, followed by the Anderson-Darling one (Anderson and Darling, 1954), when comparing them with Kolmogorov-Smirnov and Lilliefors approaches. It is important to take into account their limitations, since Shapiro-Wilk test is intended for datasets up to 5000 values, while Anderson-Darling requires, at least, 7 values in the dataset.

The datasets available make a bit complicated the analysis, mainly because of the amount of data to analyse. In some cases, the values provided are generated every hour, while the periods of data to be analysed are of a few hours. That means that it is necessary to analyse seven hours of data for the normality analysis, but maybe only the last six hours of data would be used for applying the proposed models.

The fact that some datasets are quite big is not so relevant, since the periods of data to be analysed are shorter, in line with the size of the samples to be used for outliers detection. In any case, this is a positive factor, since the approach followed is to take, randomly, several parts of the dataset in order to analyse them separately and to see if it is possible to confirm the results of the analyses.

The experimentation done included the analysis of, at least, one dataset per type of sensor, taking long and short samples, in order to check how this property of the data would behave depending on the observations available. A minimum of 20 samples of each size were taken for the analysis, not only determining whether the data was following a normal distribution, but also applying a fit function to determine which distribution was closer to the data under analysis (comparing with Burr, Weibull, gamma, lognormal, normal, uniform and one-point).

As a result, it was possible to observe that sensors' data does not follow a normal distribution most of the times. For instance, in the case of air temperature, there is a lot of literature that claims the temperature follows a normal distribution, but this is referred to datasets that provide monthly average values for several years, instead of real-time captured datasets.

One of the analyses was carried out by using the benchmarking data provided by (de Bruijn et al., 2016). To be more concrete, the analysis was done by using the raw data of the Intel node number 22, which provided indoor information for temperature and light.

The histogram and the Q-Q plot show that the distribution of the full data is not normal at all. In the histogram, without the group of highest temperatures, it may happen that the dataset would follow a normal distribution. On the other hand, the Q-Q plot is light tailed (values in the extremes are out of the expected part of the graph), and something happens with the values around 30ºC (the values are more spread than expected). This was confirmed by the Anderson-Darling test with the full dataset, providing as a result A = 716.95 and a p-value < 2.2e-16 (considered 0).

When considering two days of data (a sample of 5000 observations), the result was not so different. The Q-Q plot keeps showing a large variation of values in the extremes (it shows

a clear light tail). When running the Shapiro-Wilk test in several chunks of the dataset, the result showed that the dataset does not seem to follow a normal distribution (the p-value is always close to 0). After running the test several times, the results showed a p-value < 2.2e-16 all the time with W values between 0.75436 and 0.96058. Anderson-Darling tests provided p-value < 2.2e-16 all the time as well, with the statistic A values between 49.977 and 349.26. And this was the case for 24 hours of data as well, although with samples of six hours of data the result improved a bit (with Q-Q plots and histograms closer to a normal distribution, although with some left skewed behaviour, and Shapiro-Wilk test providing very low p-values, but not as close to 0).

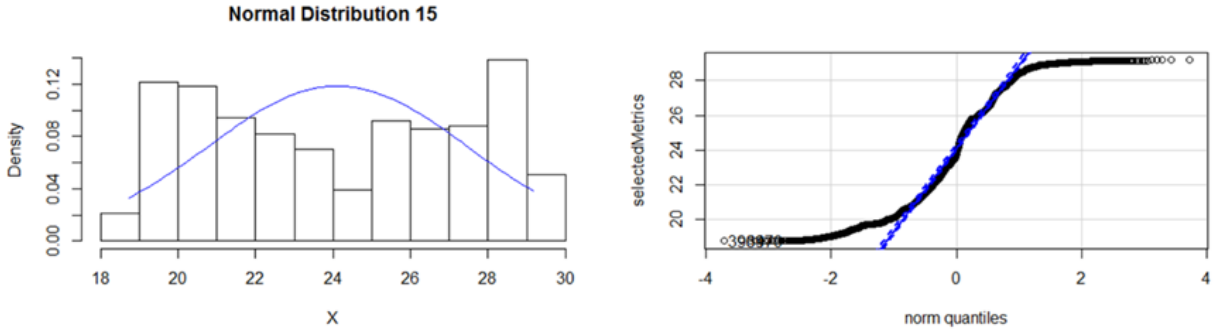

Figure 20 Distribution analysis (histogram and Q-Q plot) for temperature (2 days)

As the number of values was reduced, it was possible to obtain more stable outcomes that determined that temperature may follow a normal distribution. When taking values for 10 minutes of data (around 17 values), in 75% of the data groups selected randomly, tests and visual analysis were confirming that the data was following a normal distribution. When taking data between 30 minutes and 10 minutes, in 60% of times the data values selected were following a normal distribution, according to the graphs and to the tests.

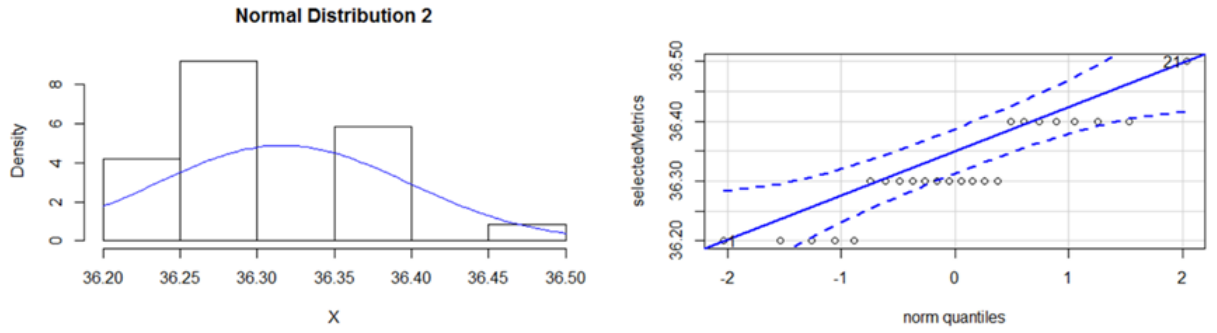

Figure 21 Distribution analysis (histogram and Q-Q plot) for water salinity (24 hours)

But this is not common to all types of sensors. For instance, in the case of water salinity, reducing the size of the samples made the distribution to converge to a one-point distribution, since this kind of sensor tends to measure the same value during long periods of time (this is, it provides a constant value).

In general, analysing the multiple datasets, although some histograms showed that the distribution could be approximately normal (as in air pressure), the Q-Q plots and distribution fit experiments confirmed that this was not the case. In some cases (water temperature, soil moisture and water salinity), the distribution will be light or heavy tailed

(values in the extremes are out of the expected part of the graph, and it seems the distribution is closer to a uniform or one-point one). Other sensors show a skewed right distribution, as values tend to concentrate at the right part of the mean (i.e., air temperature, sea level, water current speed, wind speed, precipitation, humidity). In many cases, they are close to a gamma, Burr or lognormal distribution (even Weibull in some cases).

When reducing the size of the sample and, therefore, the number of measurements used, the data distribution was somewhat closer to a normal one, as in the case mentioned about temperature. Therefore, depending on the type of analysis to perform and the sample size that is relevant for such analysis, it might be important to choose other ways to calculate mean and variance, and non-parametric solutions should be selected as the preferred ones for statistical tests.

When errors appear in the data, the distribution may be affected, depending on the magnitude of the errors and outliers. When malfunction error is present, the normality may even improve (as observed with the datasets provided by (de Bruijn et al., 2016)), because it adds a few extreme values that support the Gaussian shape of the mean. On the contrary, bias errors are exemplified by the existence of many constant values that are overrepresented, making the dataset less normally distributed and skewing the distribution (as in the example of the next figure). When drift error is present, the data become heavy tailed, increasing the weight of extreme values.

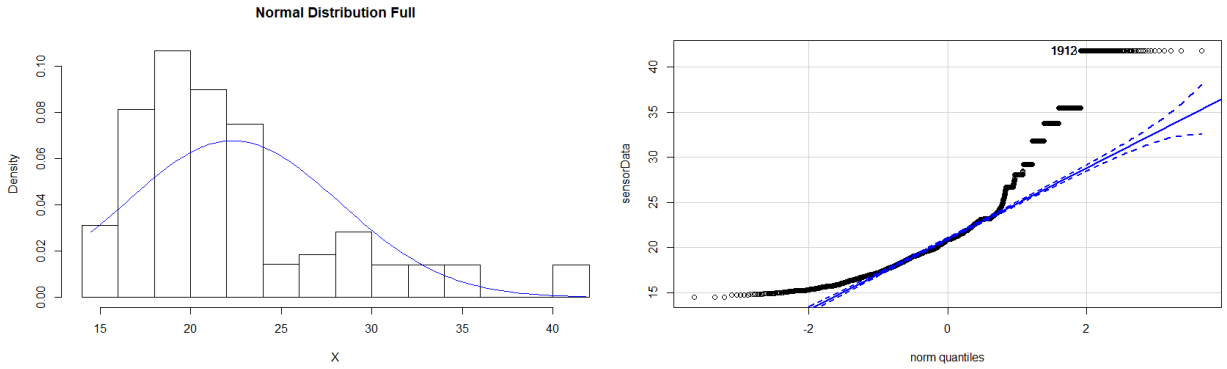

Figure 22 Data distribution analysis for temperature dataset with a bias fault injected

Consequently, even when using sample sizes in which the data may follow a normal distribution, if drift or bias errors are expected, non-parametric solutions should be selected.

4.2.2. Analysis of Data Randomness

There are some types of errors (like malfunction) that alter the data in such a way that it includes some random values during certain time. Therefore, detecting such inclusion of random values in the data would support the detection of some concrete errors that, combined with some additional information about the outliers analysis, could determine that something is wrong in the data.

There are some statistical tests that can provide information about the existence of a set of random or unexpected values in the data sample analysed: runs test and Ljung-Box test.

The Runs test aims at determining whether the dataset under analysis comes from a random process. In order to do so, it counts the 'runs' present in the dataset, dividing the values in two groups by using the mean or the median as reference. Based on that, it determines the expected runs and calculates the statistic:

$$Z = \frac{R - \bar{R}}{s_R} \tag{4–1}$$

While R is the number of real runs, $\bar{R}$ is the number of expected runs and s_R is the standard deviation of the number of runs. They can be computed according to the following equations:

$$\bar{R} = \frac{2\,n_1\,n_2}{n_1 + n_2} + 1 \tag{4–2}$$

$$s_R^2 = \frac{2\,n_1\,n_2\,(2\,n_1\,n_2 - n_1 - n_2)}{(n_1 + n_2)^2\,(n_1 + n_2 - 1)} \tag{4–3}$$

The hypotheses given by the test are the following:

- H_0: The time series was produced in a random manner;
- H_a: The time series was not produced in a random manner.

On the other hand, the Ljung-Box test is used for detecting the existence of white noise in the time series. White noise is a sequence of non-correlated random variables with a mean of zero and a finite variance. Therefore, the test works based on the autocorrelation of the dataset, proposing the following hypotheses:

- H_0: The dataset can be considered white noise;
- H_a: The dataset cannot be considered white noise.

It provides better results with small samples and its statistic is calculated with the following equation:

$$Q(m) = n(n+2)\sum_{j=1}^{m}\frac{r_j^2}{n-j} \tag{4–4}$$

In such equation, r_j^2 is the accumulated sample correlation and m represents the lags to be tested.

These tests have been used experimenting with several datasets and types of errors that we can find in the datasets. In principle, it seems it could be useful, mainly, for supporting the detection of malfunction and random errors, but only when using small samples (depending on how many observations are affected by the error), so they need to be used under specific circumstances.

4.2.2.1. Errors in Temperature with Injected Faults

One of the experiments has been done using the benchmarking datasets provided by Bruijn et al (de Bruijn et al., 2016), as they contain multiple types of errors. In order to compare

results later on, the Intel devices mote 4 and mote 33 were selected, using seven long samples (representing ten hours of temperature data) and seven short samples (representing one hour and 40 minutes) for each type of error. In those groups, five samples had errors, while the other two did not.

In the case of bias and drift errors, none of the tests detected any issue in the dataset. They reported that there was no evidence of the datasets being produced in a random way or representing white noise. In all cases, the number of runs detected was very low (no more than 13 for long windows and no more than 9 in short ones, except for one case in which it was 22). Ljung-Box test reported p-value = 0 in all cases, while the runs test provided also very low p-values (that could be considered 0, in fact), between 1.1188e-264 and 2.6477e-253 for long windows and 2.07822e-45 and 1.27622e-29 for short windows.

On the other hand, malfunction errors provided slightly better results. In the case of long windows, the tests reported no evidence of random values and white noise (with Ljung-Box always reporting p-value = 0 and Runs test reporting very low p-values with a maximum of 1.16629e-96), although the number of runs detected by the runs test was now much higher (reaching values higher than 200 in some cases). On the other hand, short samples were able to detect some problems. Ljung-Box reported three (out of 10) samples as white noise, while Runs test reported always very small p-values again.

When looking at random errors, the accuracy of the models improved. With long samples, the Runs test detected one of the random errors, while Ljung-Box almost reported the same error (although it did not reach a p-value of 0.05). When looking at the results of short samples, the Runs test reported a very low number of runs for the datasets without errors, but large numbers for the rest (between 15 and 58 runs). Also, it detected random behaviour in six out of 10 datasets, without false positives (samples without errors showed a p-value around 2e-45, clearly differentiated from the rest, closer to 0.05 and 0.01). Ljung-Box test found also six out of 10 errors, reporting 0 only for the samples without errors.

Looking at the results, it seems it was more feasible to detect malfunction and random errors with smaller samples, while bias and drift errors cannot be detected. This is normal, since the anomaly created by bias and drift make runs to be clearly differentiated between the normal data and the error itself (so the reference value for the runs is altered and a very few runs are reported).

On the other hand, malfunction and random errors seem to be detected when they are an important part of the sample, but not when they are represented in just a few observations of the sample. Therefore, another experiment was carried out using even smaller samples, representing 70 minutes of data.

4.3. Detection of Anomalous Values

Instead of relying only on the final result of the tests, it would be better to go through the list of statistics calculated, looking for certain figures in the statistic value that may indicate more accurately where one or multiple outliers are present. Moreover, tests such as Grubbs, ESD (extreme studentized deviate) and Dixon are better for detecting a small number of outliers (malfunction errors), while homogeneity tests are useful when the

number of outliers is larger, and they are concentrated (drift errors). Therefore, the tests can complement each other.

4.3.1. Analysing Tests for Outliers

One of the most important aspects is the identification of outliers, because these are values that could alter the analysis of the data (i.e., with a relevant impact on means, variance, ranges, etc.). There are specific statistical tests for detecting outliers, although they have some limitations. The best known are Dixon's Q test (Dixon, 1953), Grubbs test (Grubbs, 1969) and the extreme studentized deviate test (Rosner, 1983), a type of sequential application of the Grubbs test for more than one outlier. Although Dixon and Grubbs can detect one outlier, ESD was designed to detect multiple outliers (the Tietjen–Moore test (Tietjen and Moore, 1972) also can do it, but it requires being provided with the number of outliers beforehand, so is not useful for our case).

The Grubbs test aims at finding those values that stand out with respect to the rest of the time series, based on the distance to the mean and taking into account the variation of the data. It is useful for detecting a single outlier, and it should be used only in datasets following a normal distribution. The hypotheses proposed are the following:

- H_0: There is no outlier in the dataset;
- H_a: There is one outlier in the dataset.

It calculates the statistic G following the formula:

$$G = \frac{\max_{i=1,\ldots,N} |Y_i - \bar{Y}|}{s} \tag{4–5}$$

In such equation, $\bar{Y}$ is the mean and s the standard deviation of the sample. The critical value is obtained from a specific formula that depends on Student's T:

$$G > \frac{N-1}{\sqrt{N}} \sqrt{\frac{t^2_{\alpha/(2N),N-2}}{N-2+\ t^2_{\alpha/(2N),N-2}}} \tag{4–6}$$

Here, N is the number of elements in the sample analysed and the value used from the Student's T is the upper critical value of the distribution with N – 2 degrees of freedom.

If the p-value obtained from the test is minor than the alpha value used (in this case, 0.05), the interpretation of the test says that the null hypothesis should be rejected meaning that there is, at least, one outlier.

In case we want to detect more outliers, we have to use the ESD, that removes a detected outlier in order to apply again Grubbs for detecting the next outlier. This sequence is repeated for as many outliers as we suspect there might be in the dataset.

On the other hand, Dixon's test is based on the difference between biggest and smallest values of the dataset, to find a kind of proportion between the gap found in small values and the full range of the dataset. It is for a single outlier and it assumes the data has a normal distribution. The hypotheses proposed are:

- H_0: There is not a significant difference between the analysed value and the rest of values;
- H_a: There is a significant difference, and the suspected value is an outlier.

The formula for the Q statistic is:

$$Q_{exp} = \frac{Y_2 - Y_1}{Y_n - Y_1} \quad (4–7)$$

In order to calculate it, the dataset values are sorted, so Y_1 is the smallest value of the dataset, Y_2 the second smallest value and Y_n the largest value of the dataset. For checking the largest values, the Y_2-Y_1 part is substituted by Y_n-Y_{n-1}. The Q_{exp} calculated is compared with a table of reference Q values.

Several experiments have been performed in order to determine if the presence of outliers is detected and if we may receive false positives. Those experiments include the different cases we may find with several types of sensors and several types of errors.

Dixon has an important problem that has to do with the way it is formulated. First of all, it is limited to 30 points (because of the way the critical value for the Q statistic is calculated), so it can be only used with very small samples. On the other hand, the main problem is that its Q statistic becomes 0 when the smallest points in the sample have the same value (something that is very usual in the samples analysed), because the numerator of the Q Equation (4-7) equals 0. Therefore, Dixon has been discarded and the proposed approach focuses on Grubbs and ESD tests for detecting outliers, transforming the dataset to eliminate trends and seasonality whenever necessary.

As a general remark, Grubbs was quite robust detecting outliers (even in complex metrics like water salinity in the sea), although it showed some clear issues when dealing with errors like bias and drift. When the number of outliers is not high, it works very well, like in short malfunctions and random errors. Since its statistic is based on the mean and standard deviation, (and these values are altered when the number of outliers is large and when there is a strong trend in the data) it may provide false positives sometimes (Leys et al., 2013), especially when dealing with non-stationary samples. In the case of strong trend, that may mask outliers, it detects as outliers the lack of normality in the data, although there are samples without a normal distribution for which the test works fine as well.

For trend-related transformation, a differences function (implemented with 'diff' in R) has been used (with the lag difference parameter set as 1), since it has shown a good capability to remove the trend and is widely used. Such transformation, basically, subtracts the previous value (observation) from the current value, for each observation in the dataset.

$$difference\ (t) = observation(t) - observation(t - lag\_difference) \quad (4–8)$$

Other transformations, such as polynomial regression, may provide good results, especially when we have different trends in the same dataset, but they were not used at this stage (as this aspect requires specific research). Since the samples used are not so long, it was not necessary to deal with seasonality of the data.

4.3.1.1. Outliers in Ports of Spain

Some of the datasets provided by Ports of Spain contain some errors (because of observations that were not available). Since these observations are represented as -99.9, depending on how they appear in the datasets (how much they last), it is possible to interpret them as bias or random errors. Additionally, a few outliers were injected, in order to represent small malfunction and drift errors.

The dataset produced by the buoy named 'Golfo de Cádiz' was selected in order to do some analysis by selecting samples with six days of data. This dataset has only one observation per hour, so the samples only contained 144 measurements per each metric in the dataset. Six samples were selected, with four of them containing errors and two of them without errors. The metric selected for the analysis was atmospheric pressure, which is expected to be very stable in time.

In all cases, the Grubbs test was able to detect the outliers correctly. All the samples with some errors were reporting p-values between 1.83145e-08 and 2.30043e-11. Those samples without errors reported p-value = 1 and p-value = 0.757405.

As it may be necessary to apply some transformation to the data for other sensors, the difference transform was also applied to these samples, in order to see if the results of the tests kept on providing good results.

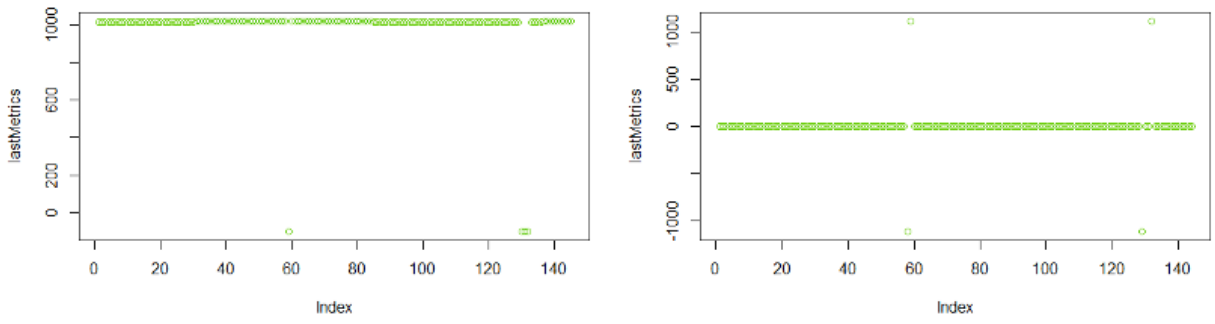

Figure 23 Atmospheric pressure (with error) transformed with the 'difference transform'

In such case, the Grubbs test also worked perfectly fine. Again, those samples with outliers reported very small p-values, between 9.11505e-07 and 0, while the samples without errors reported p-values higher than 0.05 (0.4556, 0.754232). Therefore, even after the transformation, the high stability of the atmospheric pressure did not report any false positive.

4.3.1.2. Errors in Temperature with Injected Faults

Thanks to the framework developed by Bruijn et al [x], it is possible to use a group of benchmark datasets to analyse different types of issues that may appear with sensors. Since the types of errors and measurements are similar among the datasets, the analysis of the outliers has been performed using the temperature measurements from two Intel devices: mote 4 and mote 33. Seven long samples (ten hours of data) have been taken from each mote dataset, from which five have errors and two have no errors. Also, seven short samples (one hour and 40 minutes) have been taken in the same way, for analysing other kind of sample.

When looking at the results of long samples, the Grubbs test reports a value of 1 for all the cases with bias and drift errors, including those samples without outliers. This happens for both motes because drift and bias errors alter a lot the mean and the standard deviation. Sometimes, they represent an important part of the sample, so the mean is between the normal measures and all the measures at the level of the error, altering the calculation of the G statistic. With short samples the result is very similar (except for one sample without outlier reporting correctly).

n the case of malfunction, the test works very well for all those long samples without the error, since the p-value generated is 1. But the result is not so good for the samples with errors. In the case of mote 4, three out of five errors are detected correctly, but in the other two we find that p-value = 1 and p-value = 0.06308 (this last one is close to detecting the error). For mote 33, only one error is detected correctly, while the other four fail to detect the anomaly, with three of them providing p-value = 1 and only one providing p-value = 0.07205. For mote 4 short samples, those without error provide the expected outcome (with p-value = 1), while two out of five errors are not detected in the rest of samples (with p-value = 0.190422 and p-value = 0.063366). Short samples for mote 33 provide only fail to detect one error, with p-value = 1, working fine for all the others.

Finally, when looking at the long samples with random error, the result is perfect for mote 4, since the samples without error report p-values higher than 0.05 and all the samples with error report very small p-values (between 0.00016 and 1.5174e-05). This is also the case for mote 33, with p-values between 0.00034 and 9.2531e-06 for the samples with random error. On the other hand, with short samples, in mote 4 it fails to detect only one of the errors (with p-value = 0.54947), while for mote 33 it fails to detect two errors (with p-values around 0.12).

Since it seems that most of the cases detect some trend in the data, the difference transform was applied to all the samples, expecting some improvement in the error detection for bias and drift errors. Interestingly, when applying the transformation, the samples show a rather plain graph, with two clear outliers that correspond to the position in which the error start and the position where the error ends.

In the datasets with bias errors, Grubbs reports p-value = 0 (also reported as p-value < 2.2e-16) for all those samples with errors (for both motes, including short and long samples), although it also reports the existence of outliers in those samples that contain no errors (except in one case). In those cases, all p-values are higher than 2.90893e-09. Even if there are some small peaks in the original samples, they should not be considered as outliers, according to the plots.

When looking at the datasets with drift errors, the result is similar, in the sense that all those samples with error report a p-value = 0. In this case, the long samples without error also have very low p-values (again, around values like 1.5086e-09), with some of them also reporting 0, so it becomes complicated to distinguish them from the ones with errors. With short samples, only one reports fine, while the others have small p-values, with the lowest one reporting 3.50196e-05.

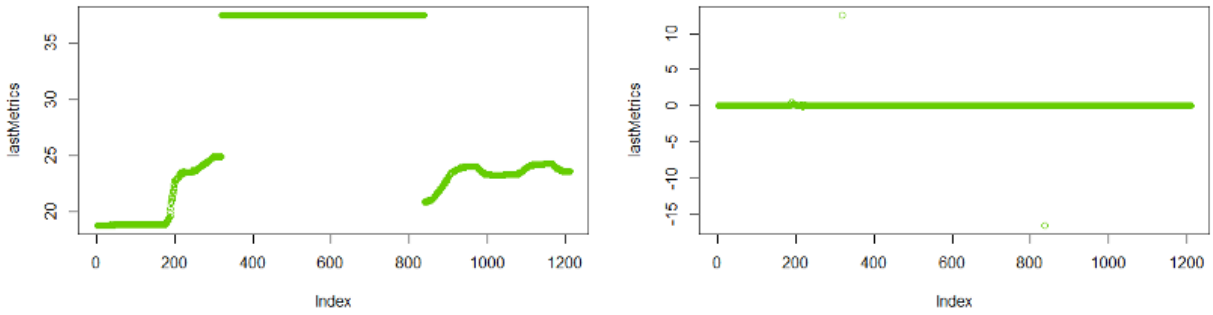

Figure 24 Temperature measurements (bias error) transformed with the 'difference transform'

For those datasets with malfunction and random errors, the transformation does not work very well, reporting almost all the samples as containing an outlier, even for the samples without errors (it only worked correctly for short samples in mote 4). The p-values vary a lot, and it is not possible to distinguish a clear pattern in the new p-values produced, so it would not be possible to differentiate samples with and without errors.

4.3.2. Effect of Outliers in Homogeneity

Another type of tests that can be used to detect the presence of some anomalies. This is the case of homogeneity tests, which are able to determine if some property is changing in different parts of the time series. This means that these tests should be able to detect 'jumps' or values that somehow 'break' the previous properties of the dataset.

This work has considered several homogeneity tests, which could be complementary because of the different aspects they address. These tests are the Standard Normal Homogeneity Test (SNHT) (Alexandersson, 1986), Pettitt's test (Pettitt, 1979), Buishand's range and U tests (Buishand, 1982)(Buishand, 1984), and Lanzante test (Lanzante, 1996). Some of them have been widely used to determine the homogeneity of datasets related to temperature and rainfall, as a way to detect if the dataset requires some transformation in order to improve forecasts. But, in this case, the intention is to detect which parts of the datasets show unexpected changes that should be considered. From this list, only the SNHT and Buishand claim to assume a normal distribution of the data; still, SNHT seems to yield good results with non-normal datasets, according to the experiments.

In fact, they can be considered complementary, as the way to calculate them is different and one test could detect errors in the data while others do not. In that sense, it is interesting to observe the outcomes of the tests, to identify those which are similar.

In the case of SNHT, the test assumes a model in which we consider there is a single change point m in a normally distributed dataset, in such a way that the mean changes in that point. It defines these hypotheses:

- H_0: Data follows a distribution $N(\mu, \sigma)$ in the whole dataset;
- H_a: From 1 to m, data follows a distribution $N(\mu_1, \sigma)$, but from m to the end of the dataset, data follows a distribution $N(\mu_2, \sigma)$.

This is done by calculating the following statistic:

$$T_k = k\, z_1^2 + (n-k) z_2^2, for\ k = 1, \dots, n \quad (4–9)$$

In this equation, k is the current position calculated for the statistic. The z_1 and z_2 values are calculated using the following equations:

$$z_1 = \frac{1}{k} \sum_{i=1}^{k} (Y_i - \bar{Y})/\sigma \quad (4–10)$$

$$z_2 = \frac{1}{n-k} \sum_{i=k+1}^{k} (Y_i - \bar{Y})/\sigma \quad (4–11)$$

The change point is obtained by finding the maximum value of the statistic: T = max(T[k]).

On the other hand, Pettitt's tests is a non-parametric test that can detect if there is a change point in which the time series changes its distribution. The test is able to indicate the location of the change point. The hypotheses defined are:

- H_0: There is no change point, and the distribution is the same for the dataset;
- H_a: There is a change point in which the data changes its distribution.

This is calculated through the statistic $U_{T,k}$:

$$U_{t,T} = \sum_{i=1}^{t} \sum_{j=t+1}^{T} sgn(X_i - X_j), \qquad 1 \le t < T \quad (4–12)$$

$$sgn(x) = \begin{cases} 1, if\ x > 0 \\ 0, if\ x = 0 \\ -1, if\ x < 0 \end{cases} \quad (4–13)$$

The expected change point is obtained by determining:

$$K_\tau = U_{\tau,T} = \max|U_{t,T}|, \quad 1 \le t < T \quad (4–14)$$

For Buishand's test, there are two types (which are similar): Buishand range and Bishand U. The Buishand range test, like SNHT, aims at determining if there is a change point in the time series where the data distribution changes. Its hypotheses are:

- H_0: The time series follows one or more distributions with the same mean;
- H_a: There is a change point in which the mean of the data changes.

The statistic is calculated with the formula:

$$R_b = \frac{\max(S_k) - \min(S_k)}{\sigma \sqrt{n}} \quad (4–15)$$

And the calculation of the rescaled adjusted partial sums is done through:

$$S_k = \sum_{i=1}^{k} (x_i - \bar{x}) \ , \quad 1 \leq i \leq n \tag{4–16}$$

The hypotheses are the same for Buishand U, but with the particularity that it assumes a normal distribution of the data. The adjusted partial sums are calculated with the same formula, but not the statistic, that is calculated as follows:

$$U = [n\ (n+1)]^{-1} \sum_{k=1}^{n-1} (S_k / D_x)^2 \tag{4–17}$$

With D_x calculated as follows:

$$D_x = \sqrt{n^{-1} \sum_{i=1}^{n} (x_i - \bar{x})} \tag{4–18}$$

Finally, Lanzante's test is similar in its purpose, looking for a shift in the time series (modelled with a Theta-coefficient) by using ranks. It is able to detect shifts in the central tendency of the sample. As the previous ones, the hypotheses defined are:

- H_0: There is no shift in the time series data;
- H_a: There is a shift in the time series data.

It calculates an adjusted rank sum using the following equation:

$$U_k = 2 \sum_{i=1}^{k} r_i - k\ (n+1) \ , \quad k = 1, \dots, n \tag{4–19}$$

The change point is calculated as the maximum value of the statistic:

$$m = k\ (\max|U_k|) \tag{4–20}$$

Some annotated datasets have been analysed in order to understand how the tests behave. Such datasets contain different types of outliers and errors, indicating those measurements in the time series that represent the error (so it is possible to determine if the tests are detecting some change).

Such experimentation does not only analyse the p-value provided by each test, but also the statistics calculated by each test, looking for some patterns that could be used for recognizing the existence of outliers and other anomalies. The proposed idea is to combine this information with the one produced by those tests focused on the outliers detection (such as Grubbs test).

After using these tests with several types of sensors, it is possible to observe that, while SNHT and Buishand tests provide similar outcomes, Pettitt and Lanzante do the same, agreeing on the change location in most of the cases. When looking at the plots of the statistics calculated, Pettitt and Lanzante still look very similar, but this is not the case for

SNHT and Buishand tests. And it is important to highlight that some local maximum and minimum values seem to provide relevant information about the location of outliers.

4.3.2.1. Outliers in Ports of Spain

Ports of Spain datasets contain measurements from several types of sensors and one of the interesting points was to understand whether some outliers would be highlighted by homogeneity tests in a very stable dataset. On the other hand, as it was reported in section 4.1, some sensors show a behaviour that could be misleading for the tests, because of the natural jumps in the data. Therefore, it was interesting to analyse formally if this was the case.

In order to do so, some experiments took samples from a dataset (the one named Golfo de Cádiz), and added small groups of outliers, that were representing errors like drift and malfunction (with a low number of outliers). It is important to highlight that, in almost all the cases, the homogeneity tests reported that some change was present, being too sensitive and causing false positives (e.g., when values of temperature vary in long windows). This might be because of the thresholds defined for the tests, which could need some customization. In that sense, it was possible to observe that the ranges of values of the statistics vary significantly depending on the type of sensor and the time window. While in some cases Pettitt yielded values between 1500 and -1500 (6 days of data on atmospheric pressure), in other cases it was between 0 and 200 (one-and-a-half days of atmospheric pressure data) or between 100 and -3000 (6 days of water salinity measurements). The same applied to all the statistics. Therefore, some normalization solution might be useful for generalizing the results.

As a concrete example (in the figure), when testing different outlier injections and only three outliers were injected in the drift error (the one detected at K = 133), the SNHT was not able to detect the anomaly (with a p-value of 0.387), while Pettitt and Lanzante still detected that something was wrong (although in K = 15). The same happened in other experiments, in which the statistical tests pointed to different locations (the K result) as the position of the outlier.

In the figure, showing one of the samples with atmospheric pressure, SNHT, Pettitt and Lanzante detect the outliers (SNHT reports T = 11.043 and p-value = 0.02167, Pettitt reports U* = 1712 and p-value = 0.006501, and Lanzante reports W = 1337 and p-value = 0.00021). While SNHT reaches its maximum value in the second graph (K = 133), Pettitt and Lanzante reach their maximum much earlier (K = 102), although in that position there is no change yet. This is because of the limitation associated to how the statistic is calculated and the position is selected, but it is possible to observe peaks in the positions where the outliers happen, with very close angles, compared to other local maximum/minimum values, where the angle of change is higher.

In this concrete case, Buishand range and U tests failed to report the anomaly (Buishand range reported R / sqrt(n) = 1.336 and p-value = 0.2403, while Buishand U reported U = 0.38263 and p-value = 0.07867), with a probable change point at k = 68. Looking at the plots, it is still possible to observe some local minimum values in those positions where the outliers are located.

It is interesting to mention that, when testing different outlier injections and there were

only three outliers in the drift error (the one detected at K = 133), the SNHT was not able to detect the anomaly (with a p-value of 0.387), while Pettitt and Lanzante still detected that something was wrong (although in K = 15).

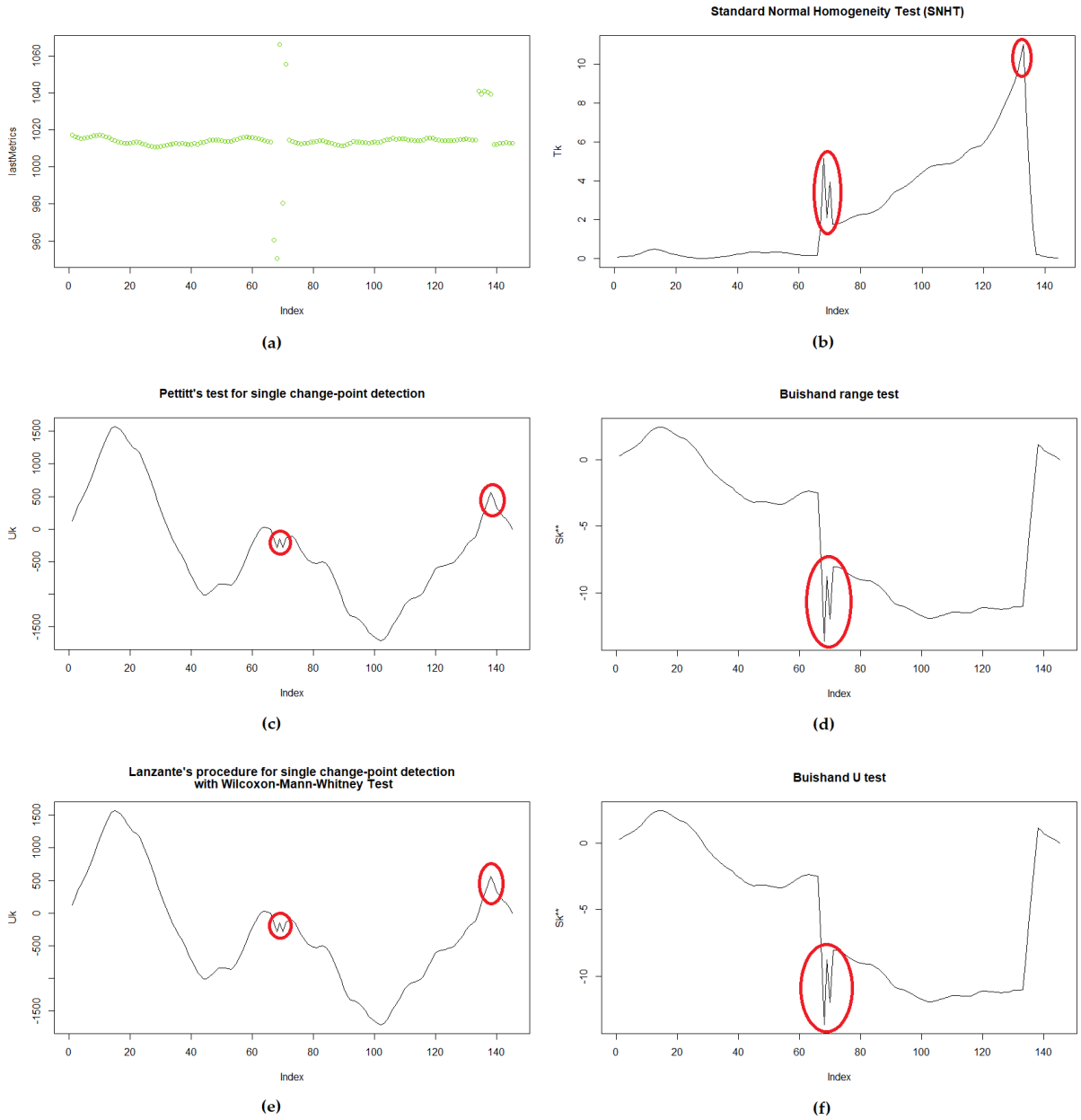

Figure 25 Results of the statistics calculated for homogeneity tests using atmospheric pressure

Finally, it is important to highlight that the presence of outliers does not mean that there is a problem with the data. As expected, in the case of water salinity, because of the characteristics of the data, some changes in the time series were considered a "jump" and reported as an outlier by the homogeneity tests (all of them, with p-value < 2.2e-16 in most of the cases). Although this is fine from a statistical point of view, the particularity should be considered, without removing or modifying such values.

4.3.2.2. Errors in Temperature with Injected Faults

One of the main advantages when using these datasets is that they are all correctly annotated and different types of errors have been injected following a formal method. This

work has analysed the effect of the homogeneity tests when looking at samples with bias, drift, malfunction and random errors. As mentioned in Section 2, all of these errors include the existence of abnormal values that can be considered outliers. Therefore, we should expect these tests to provide some indication about the existence of those changes in the time series.

The dataset selected are the ones corresponding for mote 4 of the Intel device measurements, focusing on the temperature metric. Two time windows have been selected, representing the 6% (representing 10 hours) and the 1% (representing 1 hour and 40 minutes) of the full dataset, in order to check how the amount of samples affect the analysis. Several samples were selected in order to check if the results were consistent in different parts of the time series.

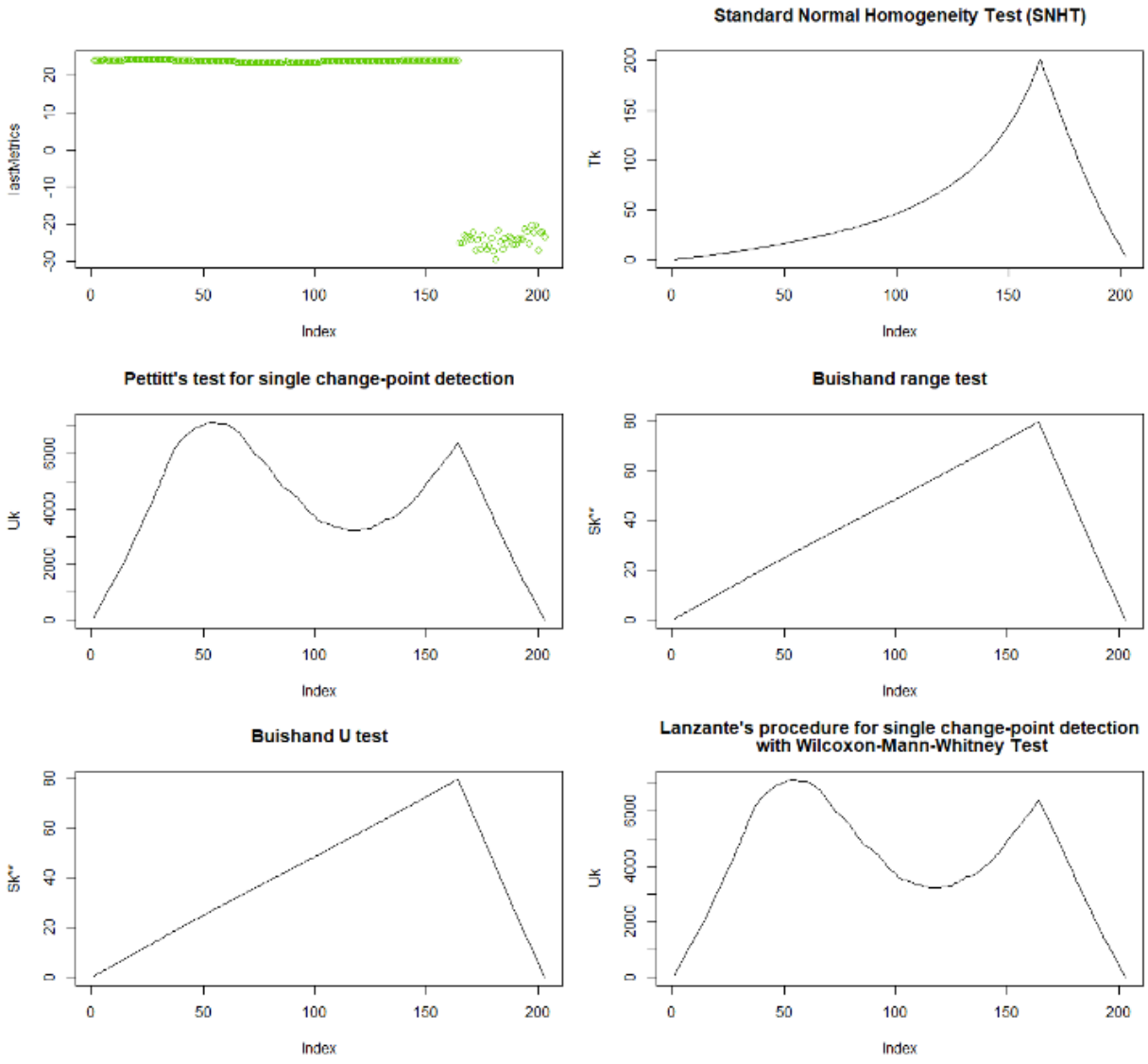

Figure 26 Results of the statistics calculated for homogeneity tests with drift error

When looking at the values produced by the different tests for a dataset including the bias error (for long window), all the homogeneity tests reported that they detect a change in the dataset (since they all report p-value < 2.2e-16, which is considered 0). Additionally, in a concrete example, all of them reported that the change happened at the location K = 928, that is the point in which the bias error ended, going back to the original values of the

sample. Although Mann-Kendall test (Mann, 1945) detected some decreasing trend, the tests were not affected.

When looking at the plots of the time series representing the statistics generated by the homogeneity tests, it is possible to observe that all of them generated clear 'peaks' when the bias error started and ended. They are local maximum values and, therefore, these values can be used to determine where the outliers and errors are present. In this concrete case, instead of indicating the end of the error, it would be possible to determine where it started, so alerts might be raised earlier and the causes could be found.

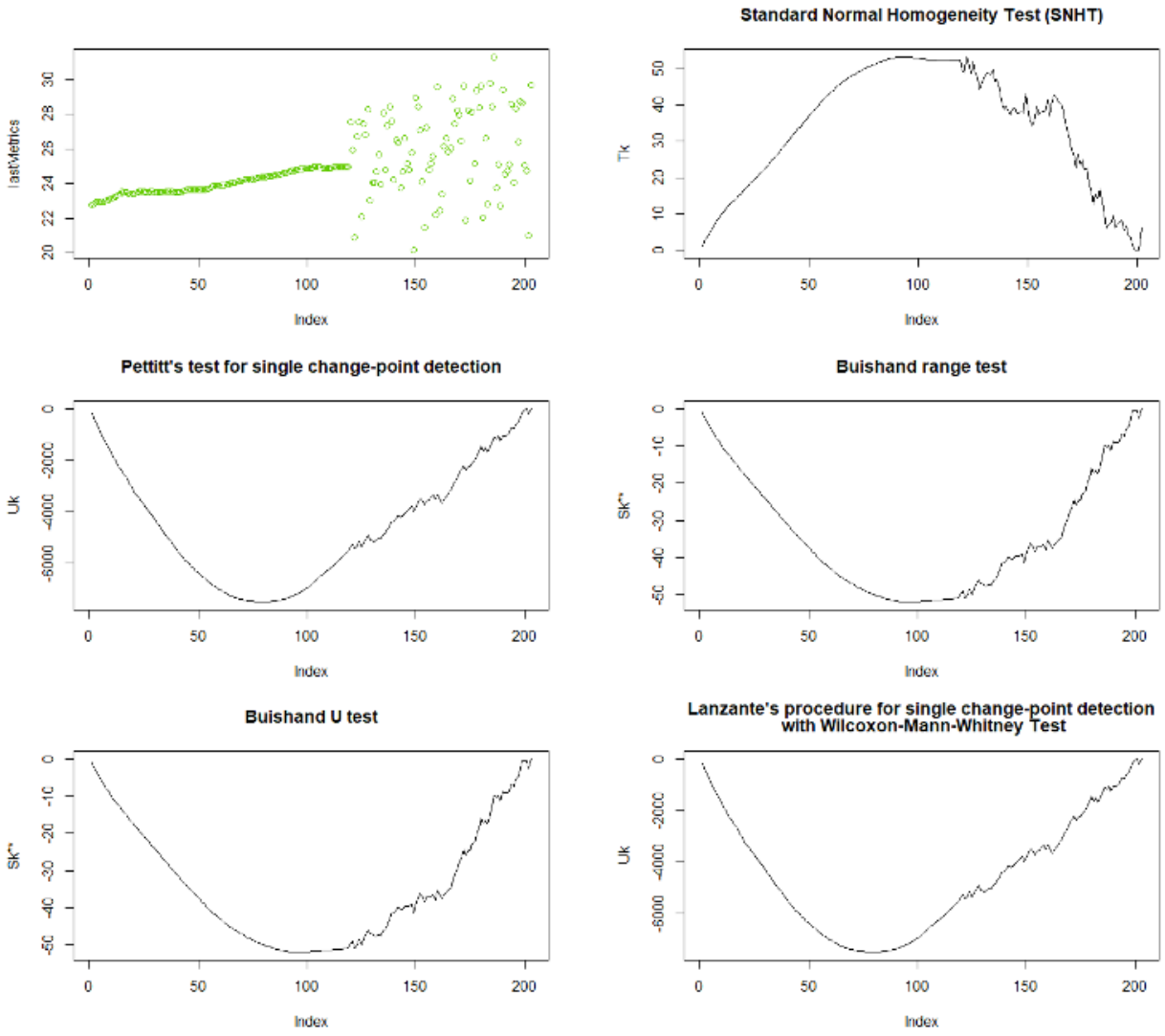

Figure 27 Results of the statistics calculated for homogeneity tests with malfunction error

If the size of the sample is smaller, so it only includes the beginning of the error, all the tests perform very well again and detect that there is some change in the time series, so it is not homogeneous. In an example, they all reported the change was in K = 110, which corresponded to the position where the error started, with the plots showing only one peak in the statistics in such location (so there was only one local maximum value).

The same analysis with the drift error for the long window gives very similar results. In a concrete example selected, all the tests detected the presence of a relevant change (with all p-value < 2.2e-16) and all indicated the problem in the same location (K = 764, this time

indicating the place where the error starts). Also, the plots showed two clear local maximum values (as clear peaks) indicating where the error started and ended. Since in the selected sample the error was mainly at the end of the time series, it is understandable that this point affects where the maximum value is generated by the statistic.

This time, when selecting the sample with short window (the one in the figure), the error started close to the end of the time series, which produced some differences in the outcomes. SNHT, Buishand Range and Buishand U tests provided very similar values, with p-value < 2.2e-16 and K = 164 (the right location of the error start). On the other hand, Lanzante, although indicated p-value < 2.2e-16, it reported the error location at K = 54, while Pettitt reported the same location, but a p-value = 3.829e-16. While the plots at SNHT and Buishand tests showed only one peak representing the maximum value of the statistic, both Lanzante and Pettitt showed two clear local maximum values with the one at the beginning being the highest one (U* = 7121 for Pettitt and W = 7583.5 for Lanzante). Interestingly, only the second local maximum (located at K = 164) shows a sharp change of tendency in the statistic, being the first one rounded. Therefore, we can infer that those sharp changes are more accurate for the type of change in the sample we are looking for.

The results are slightly different when taking a look at the result of analysing a dataset with the malfunction error injected. When using the long window, the homogeneity tests detect a change in the data sample, all of them reporting a p-vale < 2.2e-16, although the position they report is different depending on the sample. In a concrete example selected, SNHT was reporting a change at K = 492, Pettitt and Lanzante at K = 491, and Buishand range and U at K = 495. There is a change in such position (all of them could be considered valid), although it is rather natural in the dataset and the error is injected a bit later (in position K = 620). The plots of the corresponding statistics show the same information, with a clear minimum around position 492 for all the tests, although this time the peak is not so sharp, compared to the previous cases.

With a short window, the results are not so different from the numerical perspective. All the tests detect an important change (with p-value < 2.2e-16 in all cases) but, again, in the example showed in the figure, all of them report the position of the change too early (K = 93 for SNHT, K = 79 for Pettitt and Lanzante, K = 96 for Buishand range and U). On the other hand, the plots show a lot of local maximum and minimum values, like a mountain range, where the malfunction values appear. Therefore, these points really represent the values which are outliers in the sample.

Finally, when looking at the results of the dataset with random error injected, the results from the tests are quite similar to malfunction ones. With the long window, all the tests detect some clear change (with all reporting p-value < 2.2e-16 again). There are deviations detecting the position of the error, being even less accurate this time. As a concrete example, while the error started at position K = 484, the tests reported K = 816 (SNHT, Buishand range and Buishand U) and K = 705 (Pettitt and Lanzante). At least, SNHT and Buishand tests provided the position where the error finished, and the sample was back to normal. Once more, the plots show small peaks in those positions where the outliers are present, being quite sharp and clear for SNHT and Buishand tests, although not so clear for Pettitt and Lanzante.

If we look at the results with short windows, all the tests detect again that there is a change

(with p-values < 2.2 e-16, except for Buishand range, with abit higher values, like p-value = 0.000333 for the example selected). This time, there is less consensus about the position of the anomaly. While SHNT reports K = 173, Pettitt and Lanzante report K = 88. But the most accurate ones are Buishand tests, reporting (both U and range) K = 132, which is the correct position where the change starts. Additionally, all the plots show clear peaks in the position of the outliers, being especially clear in the case of SNHT and Buishand tests.

4.4. Comparison of Equivalent Sensors

As stated previously, sometimes it is very hard to distinguish a normal behaviour from an anomalous one, because of the nature of a sensor. If some large peaks are categorized as outliers in a dataset produced by a precipitation sensor, they might be right or they could be the result of some short malfunction of the sensor. In those cases, it is interesting to compare the measurements taken from equivalent sensors, so it would be possible to determine if there were significant differences that would lead to the identification of an issue. In principle, we could define equivalent sensors as sensors measuring the same metric (temperature, humidity, etc.) and located in the same area, in such a way they should behave in a similar way and, therefore, their measurements are expected to be very similar.

Although some IoT platforms allow for the identification of equivalent sensors (e.g., such as BETaaS and symbIoTe), it is not always possible to have such kind of sensors available. It may even happen that we assume that two sensors are expected to be equivalent (because of their location) but their behaviour is not so similar in the end.

Works such as (Aggarwal and Ranganathan, 2016) identified some general aspects on the use of correlation. Therefore, it is interesting to analyse to what extent some assumptions about the equivalence of sensors are true, and whether we can rely on some mechanisms to do some analysis that could be useful for the detection of anomalies.

4.4.1. Correlation in Sensors

As mentioned before, several solutions rely on the comparison of measurements between equivalent sensors in order to identify which ones are providing the expected data. Then, analysing the consensus of metrics, it is possible to determine how to clean the data, providing the adequate measurements to upper layers of software.

This might be certainly true in controlled and accurate environments, such as industrial ones, that have been designed to provide redundant metrics, placing the sensors in such a way there is a clear knowledge about the expected equivalences, and to be able to identify anomalies in the sensors or in the environment. But this is not the case for many other systems that only deploy sensors with the purpose of measuring the environment (because deploying a redundant system would be too expensive, or because the system key features are focused on other area and the sensor metrics are just complementary, for instance).

Therefore, it is necessary to analyse to what extent we can consider certain sensors equivalent and how correlation tests can determine if this is true. Still, it is important to be careful with the applicability of this aspect, since determining the consensus might be

complex, especially because perhaps it is not possible to determine easily which sensor (or group of sensors) is the one providing the anomalous measurements.

In order to do so, the experimentation was focused on two main aspects: the visual analysis (scatter plots) and the results provided by different statistical tests focused on correlation. Those statistical tests are Pearson (Benesty et al., 2009), Kendall (Kendall, 1945) and Spearman ("Spearman Rank Correlation Coefficient," 2008). While Pearson is known as a good solution for detecting linear correlation (as it uses covariance as part of its formula), Spearman can be more flexible, as it evaluates if the correlation is related to a monotonic relationship that could be linear or not. That means that it detects that, when one of the variables increases, the other one increases or decreases in a similar magnitude. On the other hand, Kendall may be a good option when some of the Pearson assumptions fail (continuous variables, homoscedasticity, linear relationship, lack of extreme outliers and normality of the variables) and when the samples are small (so Spearman could not be so effective).

Once the data samples are available, first of all, the normality of the datasets is analysed, since Pearson's test establishes that the data used should follow a normal distribution to work properly. This is done by using Q-Q plots as well as Anderson-Darling and Shapiro-Wilk tests. Although the Pearson correlation test has been used anyway in all cases, some misleading results could be explained because of the data distribution. In the case of Kendall and Spearman correlation tests, it is not necessary that the datasets follow a normal distribution (as they are non-parametric tests), so this is not an issue at all.

After this analysis the correlation tests are carried out, analysing the results of the tests. We must bear in mind that values close to 1 or -1 represent high correlation (1 means the correlation line increases, while for -1 it decreases) and values close to 0 represent very weak or no linear correlation. Also, we must observe the scatter plots generated (trying to determine if there is linear or non-linear correlation).

Since for some metrics we do not have data from similar sensors, we limited the analysis to those datasets we had in which we considered there could be some equivalency (similar sensors in close locations).

4.4.1.1. Wind Speed in Ports of Spain

Since there are several meteorological stations that produce data about wind speed in the port of Algeciras, we aim at analysing whether it is possible to say that the sensors in the area are equivalents, or if the currents influence the measurements in such a way that it is not possible to consider them similar.

Thanks to Ports of Spain, and the multiple meteorological stations that produce data, it is possible to analyse the equivalence in some types of sensors. One of the metrics to analyse is wind speed in the area of the port in Algeciras (at the South of Spain). Analysing the correlation, it is possible to determine whether the sensors in a certain area are equivalent, or if they cannot be considered similar, because of reasons like air currents that may affect the measurements.

The two datasets selected for such analysis are those produced by Dique Exento Norte and Dique Exento Sur weather stations, which are located in the same area of the port, only

separated by 1.12kms. Samples of one day and one week of data were extracted from the full datasets, selecting the initial date randomly, and comparing the metrics measured in the same dates, so they are comparable datasets, expecting to find similarities.

The normality has been checked by using the Anderson-Darling test with the full dataset. According to such test, the data is not following a normal distribution, since the p-value is really low (minor than 2.2e-16), very far from the threshold value 0.05. So, we conclude that, in general, wind speed does not follow a normal distribution.

In the first analysis, the samples selected randomly represented one week of data. According to the Shapiro-Wilk test, these samples were not following a normal distribution, since the p-values were very small (p-value < 2.2e-16 for Dique Exento Norte and p-value = 5.371e-15 for Dique Exento Sur). Additionally, the Q-Q plots of the samples also show that the measures do not follow a normal distribution, as mentioned in the previous section (the plot shows a skewed right distribution, as values concentrate at the right side of the mean).

As for the correlation analysis, the three tests were used (including Pearson) and all of them determined that the datasets are not correlated. The p-values were very close to 0 (0.00059 for Pearson, 0.00015 for Kendall and 9.3e-05 for Spearman) and the indexes are very close to 0 as well. In the case of Pearson, the correlation index was -0.1081096, so there is no linear correlation detected. Kendall tau value was -0.08060731, indicating very low correlation. Finally, Spearman rho took a value of -0.1227435, also indicating that there is no strong linear correlation between the wind speed measurements taken.

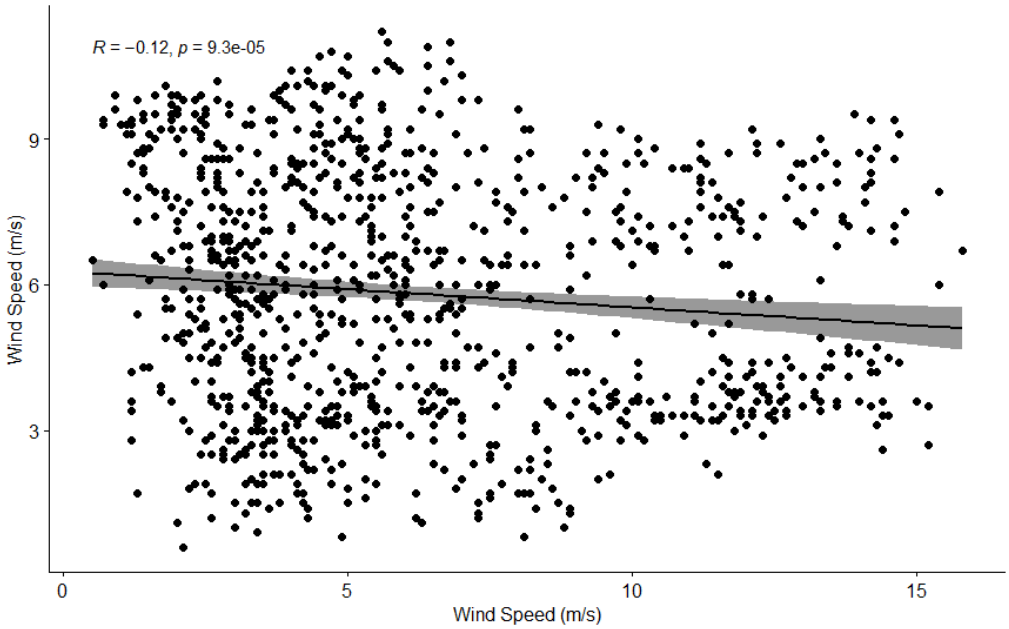

Figure 28 Scatter plot for wind speed correlation analysis (1 week)

In fact, when drawing the scatter plot with the two datasets together, we can observe that the values measured are very dispersed, not showing any linear or non-linear correlation and not showing homoscedasticity.

When using samples representing only one day of data, the results are not so different. In the samples selected randomly, the Shapiro-Wilk test indicated that there were already important differences with respect to the data distribution. While it reported a p-value of

4.764e-06 for the sample extracted from Dique Exento Norte (so the data is not following a normal distribution), it calculated a p-value of 0.05439 for Dique Exento Sur (more than 0.05 and, therefore, following a normal distribution. These results were confirmed looking at the Q-Q plots.

As for the calculation of the correlation tests, although the p-values of the tests are higher (0.449 for Pearson, 0.9972 for Kendall and 0.9048 for Spearman), the indexes calculated are too low for confirming correlation. In the case of Pearson test, the index value is 0.06357509 (very close to 0), so there is no evidence of linear correlation. For Kendall, its tau value is -0.0001984834, so correlation is too weak. Finally, Spearman rho is close to 0 as well (0.01005789) so there is not a strong correlation between the samples selected.

In general, we can conclude that wind speed is a metric in which the comparison between sensors in the same area will not provide good support to evaluate if one of the sensors is showing anomalies, assuming that air currents may have an important impact.

4.4.1.2. Temperature in Smart Santander

Another example analysed is extracted from the temperature sensors available in the Smart Santander datasets. Two of the sensors from the fault injection datasets were selected (with identifiers 181 and 193), located in an industrial area outside the city of Santander. Such sensors are located in the same area, separated by around 30 meters, and in locations open enough not to receive shadow from nearby buildings. Additionally, another sensor has been selected (with identifier 691), located 4,41kms away from the other two sensors, in the middle of the city.

Figure 29 Location of the selected sensors from Smart Santander

When analysing normality, the Anderson-Darling test determined that none of the datasets selected are following a normal distribution, when considering the full datasets (in all cases, it reports p-value < 2.2e-16). Two samples were selected from the full datasets, one representing one week of data and another one representing only one day of data. In cases, the Shapiro-Wilk test determined that none of the samples was following a normal distribution. For one week of data, the p-value of the test was reported as minor than 2.2e-16, while for one day of data in one case it reported a p-value < 2.2e-16, in another one

2.289e-07 and in the third one 1.286e-05.

Taking a look at the correlation information for one week of data, all the tests reported a strong correlation between device 181 and device 193 (the ones that are very close), as well as a quite strong correlation between device 181 and 691 as well. In the first comparison, Pearson had an index of 0.9164, Kendall a tau of 0.8414 and Spearman a rho of 0.9399. In the second one, Pearson had an index of 0.844, Kendall a tau of 0.744 and Spearman a rho of 0.899. In all cases, it happened that the p-value < 2.2e-16.

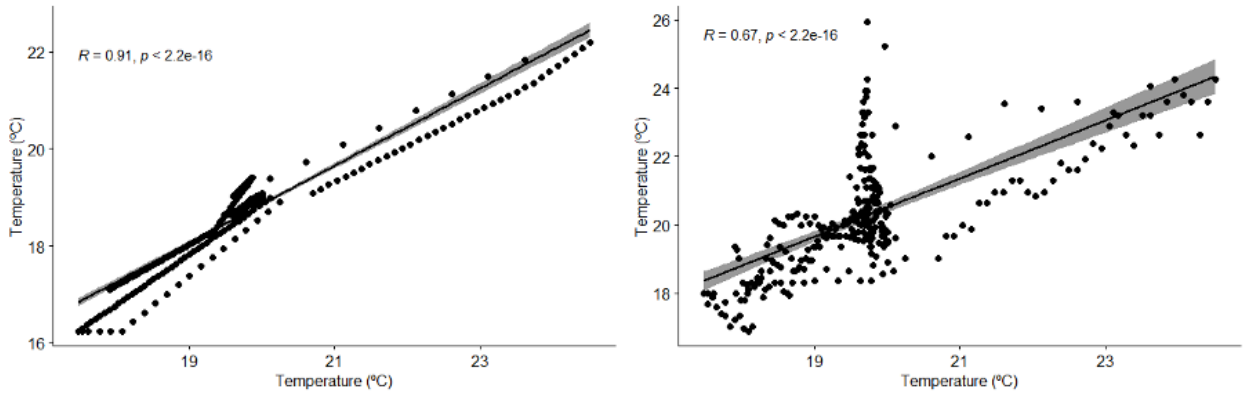

Figure 30 Spearman correlation in Sensors from Smart Santander, near vs far sensors (1 Day)

But when taking a look at the correlation for one day of data, while devices 181 and 193 remained showing a strong correlation, devices 181 and 691 showed correlation, but not so strong (as the indexes were close to 0.5). In this case, when correlating devices 181 and 193, Pearson provided an index of 0.9487, Kendall a tau of 0.7958 and Spearman a rho of 0.9124. On the other hand, when analysing devices 181 and 691 together, Pearson provided an index of 0.6944, Kendall a tau of 0.4985 (so correlation is too week) and Spearman a rho of 0.6699. Again, in all cases, it happened that the p-value < 2.2e-16.

Therefore, we could say that those sensors that are very close can be considered equivalent, but when there is more distance, this may change and, although there could be some correlation yet, this may be affected by factors that makes correlation to be too week to consider sensors equivalent (so they will provide different measurements too often).

4.4.1.3. Atmospheric Pressure in Ports of Spain

Ports of Spain provided several datasets in the area of the port of Algeciras that measure atmospheric pressure so, as in the case of wind speed, it is possible to analyse how they are related. In this concrete case, the purpose was to see how this natural event behaves in larger distances, as it seems to be quite stable in large areas. For doing so, two samples were selected from the Puerta Carnero buoy and the Golfo de Cádiz buoy (separated by 100kms), covering one month of data.

First of all, the normality test with Anderson-Darling (for the full datasets, with years of data) determined that the data is not following a normal distribution, as the p-value is very small (p-value < 2.2e-16) when it should be higher than 0.05.

When selecting the samples of one month of data, the Shapiro-Wilk test reported that they were not reporting a normal distribution neither (p-value = 2.547e-07 for Puerta Carnero and p-value = 9.931e-06 for Golfo de Cádiz), although Q-Q plots show that the one from

Golfo de Cádiz buoy could be close to normal.

Then, when looking at the correlation, all the tests determined that there is a strong linear correlation with very low p-values (p-value < 2.2e-16 in all cases) and indexes close to 1. The Pearson correlation index was 0.9276692 and Spearman rho was 0.9108291, showing such strong correlation. For Kendall, the tau calculated was 0.7908868, indicating a bit lower (but still strong) correlation.

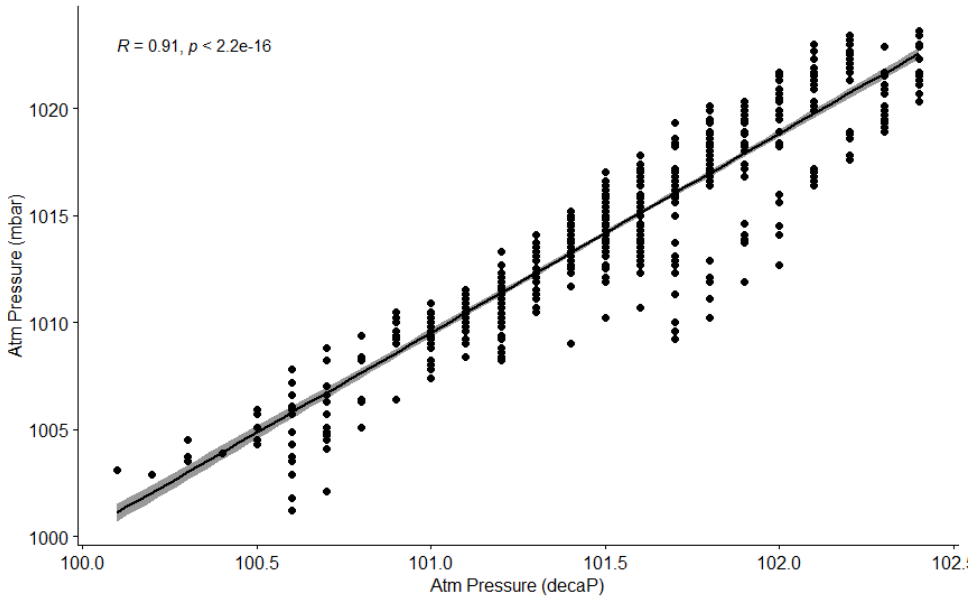

Figure 31 Scatter plot for atmospheric pressure correlation (1 month)

Therefore, we can say that atmospheric pressure is an event in which we can consider as equivalents sensors that are far away. Moreover, an interesting point in this case is that the sensors were providing the measurements in different units (while one uses mbar, the other one uses decaP), indicating that this aspect is not so important for the correlation in general (although perhaps it has some effect for the Kendall test).

4.4.2. Effect of the Outliers and Errors in the Correlation

Although for some tests it is already mentioned that the presence of outliers may be problematic (e.g., in the assumptions for the Pearson test), it is interesting to analyse how the presence of outliers may affect the outcomes of the tests, since it could be another indicator of potential issues, especially if we are able to compare historical results.

In order to analyse this aspect, it is interesting to use the datasets with faults injected, since it is possible to compare the results without errors and the results with different errors injected. Therefore, the base of this analysis is the two temperature sensors from Smart Santander analysed in the subsection 4.4.1.2 (with identifiers 181 and 193). As mentioned before, the datasets they produced do not follow a normal distribution but showed very strong correlation when using the three tests applied (Pearson, Kendall and Spearman).

This analysis was done using one week of data and, as before, using the same dates for the samples in both datasets. The same experiment was carried out for the different errors to facilitate a good comparison: malfunction, bias, drift and random. Only the data from the device 193 contained the errors injected, so the data from the device 181 was clean.

When looking at the result, it is clear that some correlation tests are more robust than others, although all of them are affected to some extent. In the case of malfunction error, all the tests performed well, showing still a strong correlation, with Kendall tau as the one scoring the lowest correlation value. Since this type of error only introduces some peaks in the data, the changes are not so big, although they could be detected.

Table 1 Correlation values with different errors in the data (one week sample)

Type of failure / Test	**Pearson**	**Kendall**	**Spearman**
No Error	0.916444	0.8414264	0.9399103
Malfunction	0.816932	0.752025	0.8621708
Bias	0.5387241	0.6189237	0.6764626
Drift	0.3560831	0.7168082	0.8339586
Random	0.3228478	0.7857923	0.8799186

In the case of bias errors, there is a clear jump in the data that lasts for a short period and such jump makes the correlation to drop. It is much lower in all cases and especially for Pearson. All of them are closer to 0.5 than to 1, so we can still consider that there is an important correlation, but it is not so strong and, therefore, it would be clear that something happened, when comparing with samples without errors.

With drift error, Pearson's test fails to show correlation, being closer to 0 than to 1, as it adds not only a jump in the data, but also some additional peaks. Therefore, we could interpret that there is no linear correlation between the two sensors. Still, in the case of Kendall and Spearman, they are more robust and manage to show that there is a strong correlation, like they did with malfunction.

The case of random is similar to drift, with Pearson failing to indicate correlation, while Kendall tau and Spearman rho show much better results. It is clear that Kendall and Spearman are more flexible for detecting correlation and do not get so much penalized by the errors. But, taking into account that this work aims at finding anomalies, Pearson test can show that something is wrong with important changes in its index. In any case, it is clear that, checking periodically these indexes we could obtain valuable information about one of the sensors producing anomalous measurements.

In fact, when looking at the scatter plots with the different errors, it becomes quite clear how some values are out of the line that represents the correlation. This is especially true for the drift and random errors, as we can see that some parallel lines appear far from the reference line.

When using samples of just one day of data, in some cases, it is even more evident that linear correlation is lost. In such case, it was not possible to build the same table of comparison, since the errors were not present in all the samples at the same dates, so it was necessary to select different starting dates and compare the correlation without error and with error.

As with one week of data, the Pearson index dropped from high values (between 0.948 and 0.99) to low values showing no correlation at all (between 0.3015 and 0.4942) in the case of bias, drift and random errors (for the malfunction error it reported 0.75939, compared to the 0.9728384 with no errors in the same sample). When looking at Kendall and Spearman tests, their indexes were lower than with the original data, but still they were always higher than 0.6. But, although they reported that still a strong correlation was there, those indexes dropped in 0.3 in many cases, making evident something was happening with the data.

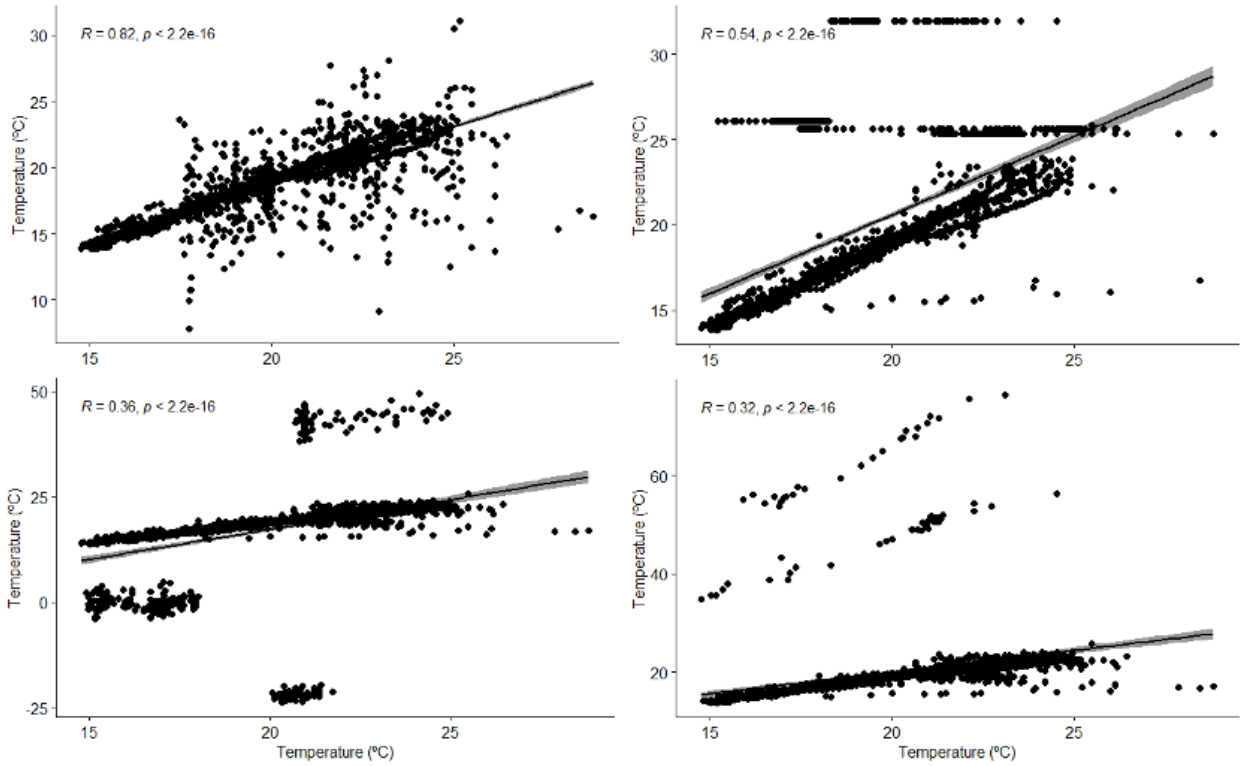

Figure 32 Correlation scatter plots with different errors in Smart Santander datasets

In general, the loss of correlation was more evident as the data samples were smaller, and this makes sense, since the errors dilute more in large samples than in the small ones, so errors have larger impact when the number of measurements is smaller.

4.5. Discussion

This work analysed different aspects of the datasets generated by sensors that may impact the application of certain solutions for data processing. For instance, some ML-based solutions (such as parametric classifiers) simplify the data processing, while doing certain assumptions (like the distribution of the data). Additionally, the presence of outliers has also an important impact on some solutions, altering the results. All these aspects should be taken into account when dealing with anomalies detection and when applying certain methods at early stages of the data preparation phase.

4.5.1. Particularities of Sensor Data

The type of natural element or property measured, the context of sensors, the measurement

units and the time window used all affect the data generated and the basic statistical features that are widely used (e.g., mean, variance). This is not new at all, but this work aims at looking at the particularities of different sensors, as it would be easier to understand the results obtained from some analyses (such as the outliers, as some "false positives" may not be wrong at all). Looking at the datasets, it was evident that providing generic solutions would be quite difficult, as a solution that may work for temperature would fail with water salinity.

Using unit-less solutions, such as the coefficient of variation, is useful but they are not applicable in all cases (because of the lack of absolute zero or because the application of the typical statistics requires complex processing). In such cases, it is useful to do a small transformation in the data, so all values will be positive. Still, even calculating such value, it seems it is hard to determine reference values that can be used for analysing data variability because of the different values it takes for different types of sensors, although it could be an area to explore.

IQR is also interesting because it may avoid the effects introduced by outliers, although it is not unit-less. There might be an interesting area of research applying these types of solutions instead of variance or standard deviation, to certain parametric models (both statistical methods and ML methods).

4.5.2. Distribution and Variability

A key aspect is the analysis of the probability distribution of the data. Although many works using large datasets with average values (using yearly or monthly means) reported (or assumed) that the data followed a normal distribution (e.g., in the case of temperature (Firat et al., 2012)), the observations show that raw data for near real-time processing, in many cases, does not follow normal distributions and, therefore, the assumptions of some models and solutions would not be right.

The selected sample sizes also have an impact on the distribution of the data. One of the observations done was that, even using the same time window, different samples taken from the same dataset also followed different distributions. Therefore, the methods selected to apply may appear to be suitable for part of the data, but not for all of the samples that might be selected. The work highlighted how several sensors follow certain distributions (lognormal, Weibull and gamma were the ones selected most of the time, aside from the normal distribution). Because of so much variation, the use of non-parametric solutions is recommended unless the selected distribution is the one fitting in a high percentage of cases (and this is quite unusual). Still, in the cases in which the distribution appears close to normal (although statistical tests provide a negative answer, Q-Q plots and histograms provide good support in this case), the tools and methods to apply might still be valid (although perhaps they will not provide the best accuracy). In cases where applicable, another option that could be explored would be to propose solutions that adapt themselves as the distribution changes.

In some cases, it is hard to find complete information about this kind of characteristic of the data when looking at the statistical and ML-based solutions listed in Section 2 (the surveys in (Teh et al., 2020) and (Erhan et al., 2021) mention potential issues with semi-supervised

learning techniques, but do not mention the distributions we may find), so it is not so clear that they are given enough importance (such as in (Hasan et al., 2019), (Maseda et al., 2021) and (Tanuska et al., 2021), to name a few). In some cases, authors apply some previous transformation to the data that may solve the problem, but we do not know about the original conditions of the data, as in (Oucheikh et al., 2020). Additionally, removing trends and seasonality is also shown as a solution (Ryan et al., 2020), but it could be still possible to obtain datasets with non-normal distributions (such as a uniform distribution).

Another aspect has to do with the identification of samples that represent random values or white noise. It is possible to identify such samples, but only if the random part of the data sample is big enough for the statistical tests to raise the alert. The experimentation showed that it was necessary to reduce the size of the samples, compared to the size required for other analyses, like for outliers identification.

Therefore, this analysis is useful because it is important for data scientists using sensors to address these aspects as soon as possible in their work, identifying appropriate strategies early in the process (the work in (Guh, 2002) notes some issues when using non-normal datasets with ANNs and data distribution that should be taken into account when normalizing the input values for the models).

4.5.3. Anomalous Values

Another aspect addressed is the detection of outliers and their role in other characteristics of the data. As mentioned in section 4.3, outliers may affect the data distribution by adding extreme values that can modify the original distribution. Therefore, a good way to proceed when preparing data would be to analyse the data distribution first (to understand if the application of certain methods for managing outliers might not be appropriate), then detect outliers (cleaning those that are real errors) and analyse the data distribution again as the last step.

In the case of outlier detection, the work proposes using certain statistical tests that can give a good idea regarding their presence (taking advantage of data transformation, especially in cases where trends and seasonality are present, or when dealing with certain types of errors). Although there are other solutions that might be more appropriate in each case (e.g., ML-based solutions can be more accurate in the concrete context they were designed for, but they can be affected by imbalanced datasets and require re-training and more analysis for new contexts), the statistical tests seemed to work fine, in general, in all the datasets used, and more complex models can be applied later, if required.

The work observed that solutions such as Dixon's Q are not adequate because of the issues with equal values being part of a sample. On the other hand, Grubbs and ESD worked fine when detecting small peaks. When multiple wrong values were together (drift and bias failures), homogeneity tests demonstrated better performance (whereas Grubbs and ESD were missing outliers). Therefore, it is necessary to apply a combination of both approaches to detect most of these failures, instead of using only one, as proposed in previous works. In fact, in the case of bias and drift errors, while homogeneity tests can detect some change in the sample, when applying the difference transformation, Grubbs and ESD can detect two clear outliers that represent the start and end of the anomaly in the sample, showing a

great complementarity.

Additionally, in the case of homogeneity tests, it was interesting to observe the presence of peaks in the position of the outliers when generating a plot with the calculated statistics (SNHT T, Buishand R and U, Pettitt U and Lanzante U). Even in cases in which they provided a different position for the outlier, the peak was in the same location in the plot, although other values, in the end, set the reference value for the statistic. Although it did not happen in all cases, in those in which the peaks appeared, it would be possible to improve the accuracy of the result, detecting such patterns and complementing the information provided by the tests. This would be a new approach to outlier detection.

4.5.4. The Role of Correlation

The last aspect addressed is related to correlation, a method used in several solutions for anomaly detection that exploits the comparison with other equivalent sensors (Zhang et al., 2016). As observed, correlation needs to be analysed case by case because it is not applicable to all types of sensors (so it cannot be generalized). While in some cases it will work even for sensors separated by kilometres (such as those monitoring atmospheric pressure), others do not show correlation even when they are close (such as wind speed).

Due to the robustness shown with outliers, Spearman is the best solution to apply (despite the potential issues mentioned by (Zhang et al., 2016)), but it is always interesting to also have Kendall and Pearson to compare results. Although they can be useful for comparing sensors, we must be aware that using Pearson when drift errors are present may cause the solution to fail. This may affect some solutions focused on principal component analysis (PCA) techniques.

It is also true that, if we are in a context in which we previously confirmed the equivalence of sensors, using these tests to check that there is a loss of correlation could be used as an alert that something wrong is going on.

Correlation also can be used when comparing sensors of different types, when we think they are related, in such a way that we can better understand the context of each sensor. This needs to be analysed case by case as well because not all types of sensors are correlated in every environment. It would be interesting to perform some correlation analysis for datasets with multiple sensors (all vs. all), in order to indicate which relationships should be analysed more carefully.

4.5.5. Research Limitations

This research is subject to a limitation related to the generalization of the results to other domains beyond smart cities, environmental monitoring, home automation or agriculture. There was no access to datasets from domains such as manufacturing or personalized medicine, in which sensors are also widely used today. Therefore, one way to improve this study is to look for more open datasets and industrial partners that would be able to share a few datasets that could complement the current work.

5. PROCESSES AND TRUST MODEL FOR SENSORS

Information is the oil of the 21st century, and analytics is the combustion engine.

Peter Sondergaard

Using the information produced and the analyses done in the previous section, it is possible to start determining an adequate way to find anomalies in sensors' data and to propose a way to automatically warn users when sensors seem to become untrustworthy.

As the hypothesis was proposing, the analysis done of the sensors so far covered aspects related to how their data vary in time and how to detect the presence of outliers in the data, as they might be a signal of potential issues in the sensors. Also, the correlation was analysed as an interesting solution when there are equivalent sensors in the same area.

The basic characteristics of the sensors, as well as the kind of distribution that the data follows, have shown to be important because they may limit the applicability of some solutions, or they may affect the results produced. Still, it was possible to confirm that some statistical tests and other calculations can provide interesting clues about the behaviour of the sensors only looking at the data, even if the type of sensor is not known.

The Coefficient of Variation, IQR and statistical tests for studying the randomness of the data can provide valuable information to identify some cases of potential failure, for instance (RQ1). On the other hand, homogeneity tests, as well as outliers-focused tests (especially Grubb's test) seem to be useful in most of the cases observed (RQ2, RQ3). An adequate combination of such solutions is one of the aspects that this section is addressing, defining how the outcomes of those techniques can be used together in order to maximize the identification of problems in sensors and to produce some result that can indicate whether a sensor should be trusted or not (RQ1, RQ2, RQ3, RQ6).

Additionally, the analysis of the outcomes produced showed that it would be possible to study other models that could improve the detection of outliers (RQ2, RQ3). There are circumstances in which the presence of trends in the data may be problematic and, although some simple data transformation may help, there could be other transformations (more oriented to highlighting the outliers) that perhaps could increase the performance of the statistical tests.

On the other hand, homogeneity tests produce some time series that seem to include information that could be used for a clearer identification of outliers. This section also analyses such topic in detail, proposing a complementary solution (RQ2, RQ3) that can be

used together with the rest of the tests and calculations.

All these approaches are combined together proposing, first of all, a set of procedures that can be used for data understanding and trust evaluation in a data analytics process, and secondly, a model that combines all the aspects to produce a measurement representing the trust evaluation of a sensor. Additionally, a parallel implementation of these models is explored, since it may be a starting point for future complex models to be executed in edge computing devices that require to optimize and exploit the computational resources available (RQ7).

5.1. Data Transformations

One of the problems when analysing the data with homogeneity tests is that, in the presence of trends and seasonality, they use to report the existence of changes in the time series. This is, mainly, because of the change of the mean as the time series decreases or increases. In the case of SNHT and Buishand's tests, they use to report a change in the middle of the dataset analysed, while in the case of Pettitt and Lanzante, this change is usually reported close to the beginning of the end of the dataset.

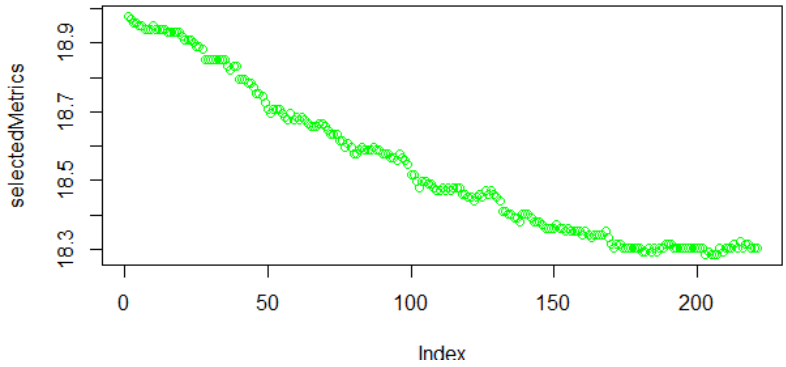

Figure 33 Right temperature measurements with decreasing trend

When using the homogeneity tests with this figure, all of them report changes in the time series, although looking for outliers with Grubbs' test reports that no outliers are reported (with a p-value=1). The problem is that, relying only on the Grubbs' test is normally misleading when looking at certain errors, like bias and drift, that use to appear in a context like failure with the network connection (so only a constant measure is provided, as no existence of metric).

Therefore, in order to avoid this issue (the reporting of false positives), the solution is to apply a transformation to the data, as suggested in Section 4. In such section, the difference transformation (with one step difference) was used in order to show some improvement, but there were still some situations in which it did not work very well (especially, when there was a clear trend together with some outliers).

As a way to reduce false positives and still enable a good detection of outliers, this work has analysed different transformations applied to the data, looking for one that could hide trends, while still highlighting outliers in the data. This achievement could improve the performance of the outlier detection model when combining the outcomes of the homogeneity tests with Grubbs.

When analysing data, data scientists use to apply some transformations, but these are focused on concrete purposes, like normalizing the data (change data distribution, so it will be close to follow a normal distribution), removing trends and removing seasonality. They are not focused on highlighting the outliers, as a way to ease their detection.

During the study of the most common transformations, the following focused on improving the normality of data were used with some samples: square-root transformation, log transformation and Yeo-Johnson transformation. In all cases, the dataset obtained improved with respect to its normality, but the trend was not removed, so these transformations are not useful in this context.

There was a similar result with the exponential transformation, as it has shown that it cannot remove trend totally (so the mentioned problems are still there). Therefore, this kind of transformation was discarded as well.

On the other hand, the difference transformation removes the trend and is able to highlight some values that could represent outliers, but when there are errors like malfunction, it introduces too much noise to the resulting dataset. This transformation has been used as the base of the study, as described in the following subsections.

5.1.1. Transformations Studied and Proposed

The objectives of the transformation to be done are the following:

- Remove the trend of the dataset;
- Avoid or reduce the noise in the data, smoothing those values that are not outliers;
- Highlight those values that represent outliers.

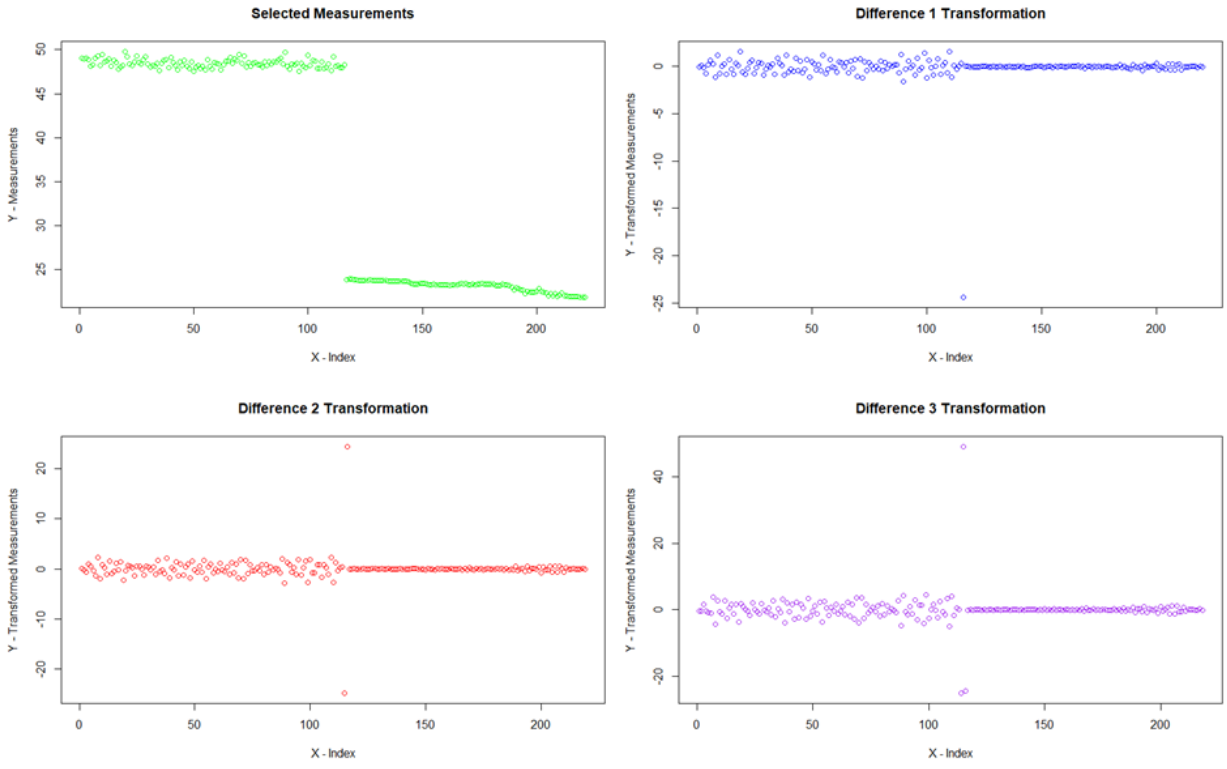

Figure 34 Difference transformations with drift error

During the analysis, three versions of the difference transformation were used, changing the number of steps to be used for computing the data (from one step to three steps). In general, the results were quite similar, except for the amplitude of the results, as it is possible to see in the following figure.

As a way to address the proposed objectives, this thesis has proposed a new modification of the difference transformation, by combining it with an exponential coefficient that decreases the difference when this is low and it increases it when this is larger. The coefficient is formulated as 1.1^{x-15}, so the equation for the transformation is the following:

$$\widehat{Y_t} = (y_t - y_{t-1}) * 1.1^{|y_t - y_{t-1}| - 15} \qquad (5\text{–}1)$$

The first part is the normal difference transformation with one step (lag=1). Then it is multiplied by a coefficient that has been determined based on the expectations of growth for the elements in the dataset. An exponential equation was selected, since it is close to 0 at the beginning, and it increases a lot as the values in the x axis grow.

Figure 35 Function of the coefficient for exponential difference transformation

Such exponential equations can be adapted in two ways. First, it is possible to move the values to the right part of the coordinates axis by removing values to the x in the exponent part. That is the reason because x-15 was selected in this case as the exponent, so values will start growing more when x=4.

Secondly, it is possible to configure the way in which the function grows, depending on the number selected as the base of the power function. In this case, a base of 1.1 provides a good balance for making the values to grow. The way to grow is smooth enough until x=20 and, still, after then, the function does not increase in a vertical way.

It is important to add here that, as the x is represented by the difference transformation, its absolute value is used, since a base with negative exponent provides very small values close to 0, and that is not the effect intended (as it is necessary to take negative values into account for outliers as well). Therefore, using the absolute value, the exponent reaches the adequate value to increase the weight of some elements in the time series.

Still, the calculated value of the difference is the one multiplied with the coefficient, without using its absolute value. This generates negative values that enable the grouping of points around the y=0 axis, keeping the average closer to the concentration of the majority of points.

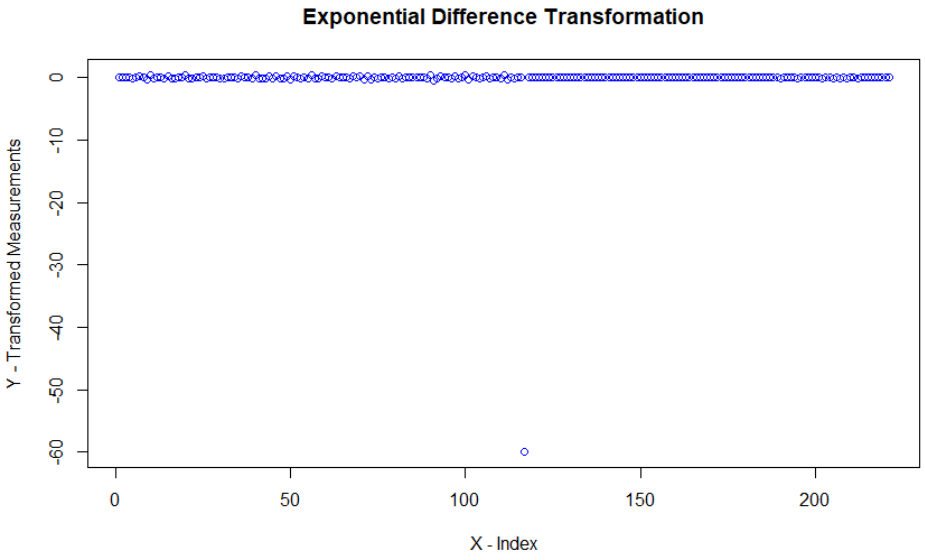

Figure 36 Exponential difference transformation with drift error

Another approach proposed in this work is the usage of polynomial regression in order to obtain a curve that fits well enough with the dataset. Since the regression has its limitation, it is expected that, when subtracting the value generated by the regression model from the original value of the dataset, the resulting time series will produce high values in the location of outliers, while reducing the trends in the data (although perhaps introducing others).

Several polynomial regression approaches were studied, generating models with different degrees, from 1 to 6. The purpose was to obtain a model good enough to avoid large error, while still not fitting so well that outliers would be represented with the model as well.

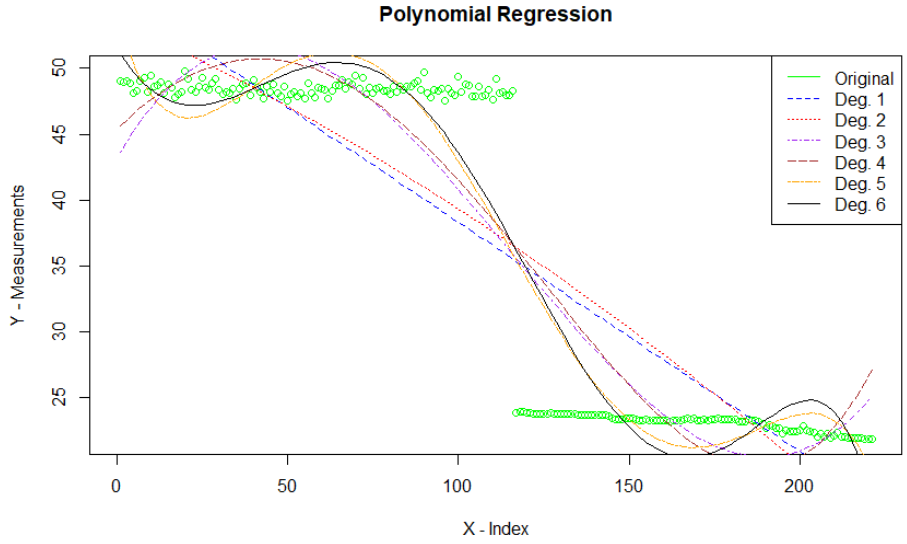

Figure 37 Polynomial regression models with drift error

As expected, the model that obtains the best fit is the polynomial regression with 6 degrees, although is very similar to the one with 5 degrees. In fact, in a few areas performs even a bit better, so there is a smaller risk of generating some high values during the transformation. As a result, bearing in mind that the approach is a bit simpler, the polynomial regression of degree 5 was used to test the solution.

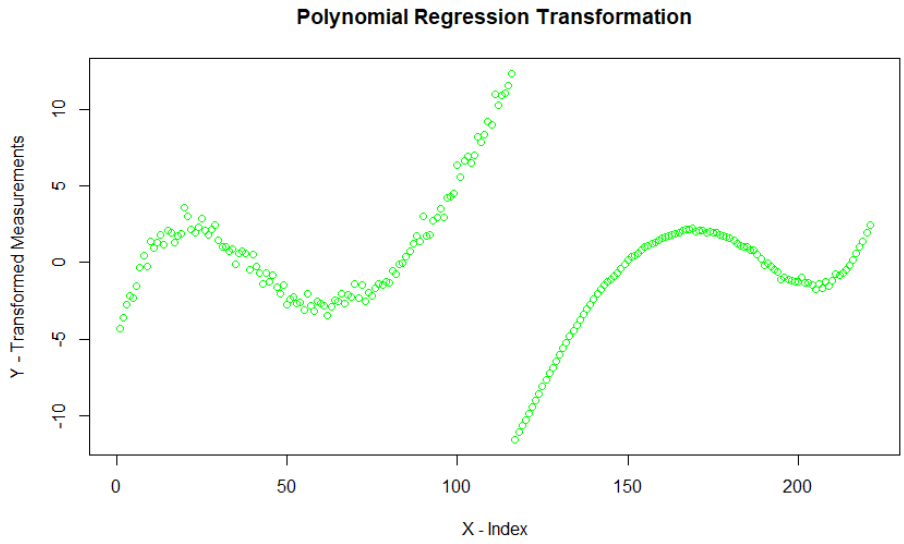

Figure 38 Polynomial regression transformation with drift error

In the case of drift errors, for instance, it seems this introduces some small trends, and there is a large jump in the area where the drift error ends. Therefore, that should facilitate the detection of the problem.

This transformation, together with the exponential difference one, has been applied to some datasets and the results they have generated have been compared against several situations and tests, described in the next subsection.

5.1.2. Results of Transformations Analysis

The way to check whether the transformations work is to use several types of samples and the calculation of several statistical tests. The transformation should show that they are able to remove trends and that the homogeneity tests produce the expected results (not reporting false positives in correct samples). Still, they should show that it is possible to detect some errors when they are present in the dataset.

The tests used are SNHT, Buishand R and Pettitt (from the group of homogeneity tests), as well as Grubbs (to check individual outliers). Since Lanzante is almost always like Pettitt and Buishand R is almost always like Buishand U, these tests have not been used as a way to simplify the analysis.

Since the exponential difference transformation is similar to the difference transformations analysed (and it seems to perform a bit better, improving the highlight of outliers and errors), only that transformation is tested. The polynomial regression transformation is the other transformation tested, as mentioned before.

In the first case, the selected sample contained a drift error (similar to a bias error) at the beginning of the dataset. When using the original dataset, all the homogeneity tests (SNHT, Buishand R and Pettitt) detected successfully that there was some important change in the data, with very low p-values. In the case of the polynomial regression transformation, all tests detected a change in the data, although for Pettitt the p-value was not as small as in the other cases (the p-value was 0.0283). On the other hand, the exponential difference transformation was not able to detect any change in the dataset, mainly because the jump that the drift error produces in the data is transformed into a clear outlier in the resulting sample, while the rest of the data remains stable.

When looking at the Grubbs test with the original dataset, no outliers were detected. This was expected, since the presence of the drift error impacts the test in such a way it cannot detect the change in the level of the time series. The sample generated with the polynomial regression dataset was not able to detect any outlier neither, although the p-value reported was 0.1296. On the other hand, the exponential difference transformation detected a clear outlier that corresponds to the position where the drift error ends.

Therefore, combining the result of the homogeneity tests against the original data (or transformed with the polynomial regression approach) with the Grubbs test obtained with the exponential difference transformation we may detect clearly drift and bias errors.

The second case analysed a malfunction error, that usually introduces complexities because of the noise it adds to the data. With the original data, the SNHT and Pettitt tests detect some change in the data (although the p-values are not very small), while Buishand does not report changes. The sample obtained from the polynomial regression transformation fails to find changes with SNHT and Buishand tests, but Pettitt test reports some change with a p-value = 0.01132. In the case of the exponential difference transformation, none of the tests reported changes in the data.

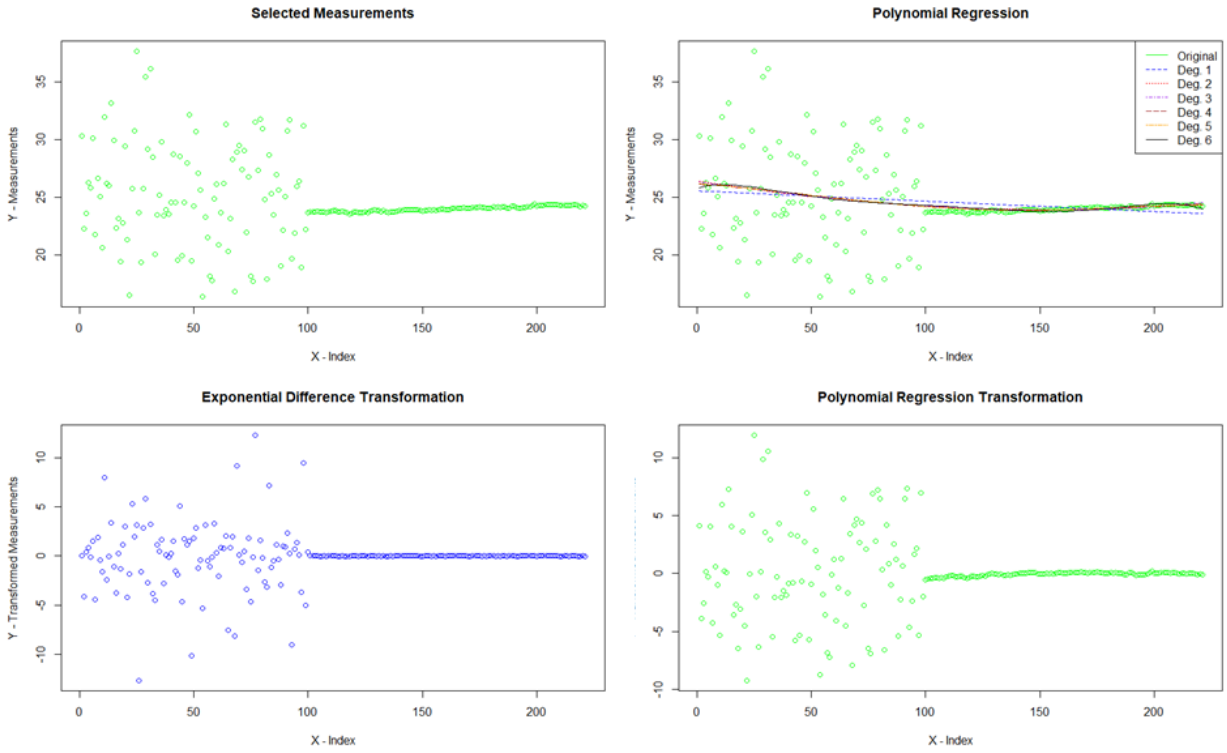

Figure 39 Transformations with malfunction error

When looking at the results with the Grubbs test, it reported the presence of outliers with all the samples used (the original one and the two ones obtained from transformations) so, in general, the problem was detected.

Finally, the last case studied is a sample with no errors, in which temperature is just decreasing normally. As expected, the three homogeneity tests report a change in the data even though it does not exist. In all cases the p-value is close to 0, reporting the change in the middle of the sample or nearby. When looking at the results produced with the dataset obtained with the polynomial regression transformation, they report changes in the dataset as well. The point in this case is that the p-values are close to the alpha value used (alpha=0.05): 0.02567 for SNHT, 0.0003333 for Buishand and 0.0211 for Pettitt. Changing the alpha value to 0.01 might mitigate the problem, gaining in accuracy. Finally, in the case of the exponential difference transformation, all the tests provided p-values higher than 0.05, so no change was detected.

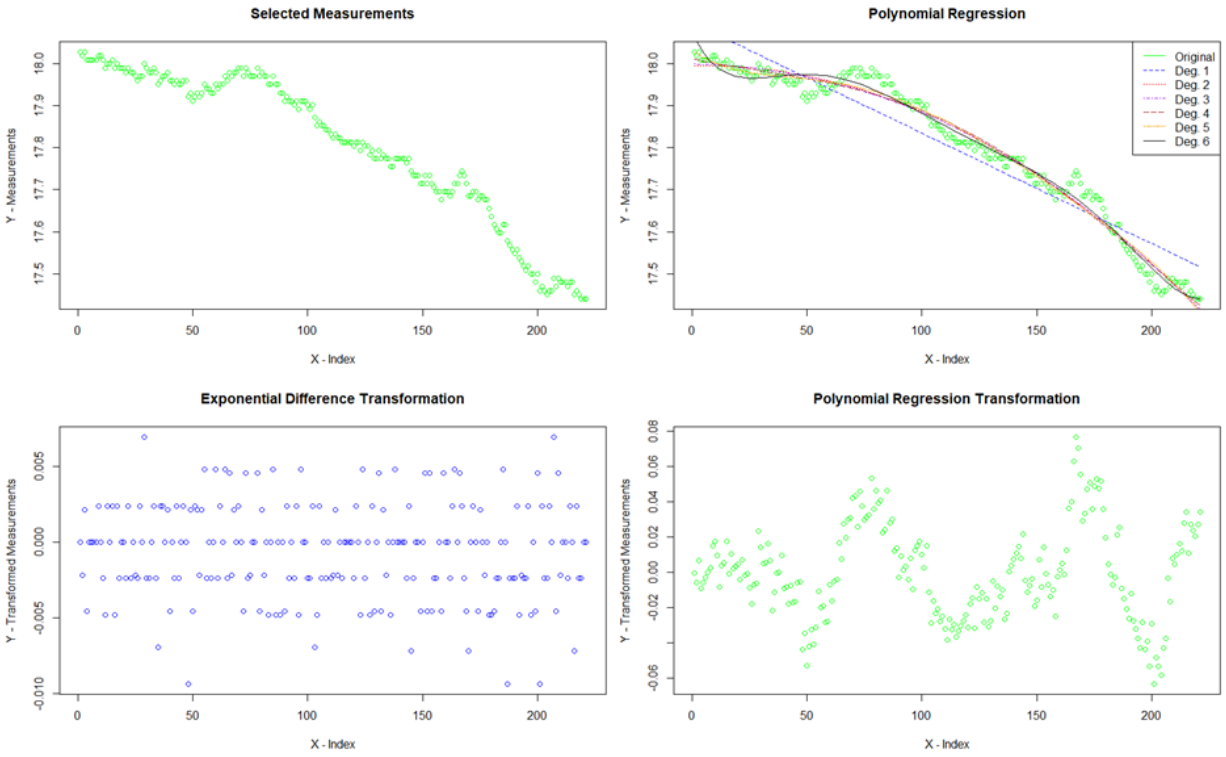

Figure 40 Transformations with no error

When looking at the outcomes of the Grubbs test, none of the samples reported any outlier. All the p-values were higher than 0.05, with the smallest one coming from the sample transformed with the polynomial regression (p-value = 0.3551).

It is worth to mention that other tests with more data showed similar results. Even in complex situations, like having only a few measurements right in the whole dataset, the tests reported problems with the right combination of the original data and transformations. Additionally, other cases, with a malfunction error in the middle of a decreasing trend, the exponential difference transformation performed even a bit better.

Therefore, it is interesting to integrate these approaches in the processes defined for data understanding and trust evaluation, as their complementarity become interesting for the model.

5.2. Homogeneity Statistics Analysis

One of the interesting findings mentioned in section 4.3 had to do with the plots created with the results of the statistical tests used for homogeneity. Using the corresponding equations, each test calculates a statistic in each point of the original time series, so it is possible to obtain another time series corresponding only to the outcomes of the statistical tests.

As shown in *Figure 25*, the statistics calculated produce some clear spikes in those locations in which there are important changes in the original time series (highlighted in red in the figure). That means that, whenever the original data has a clear outlier, the statistical tests also produce a characteristic figure when we plot the new time series representing the statistics calculated.

This is true for all the statistical tests used for analysing the homogeneity of the original dataset. Buishand U and Buishand R tests usually produce similar graphs, as well as Pettitt and Lanzante tests, while SNHT is like Buishand tests in some cases, and different to all the tests in other cases. But in all cases, they produce the mentioned 'peaks', depending on how data varies in the original data.

Somehow, this makes sense, since the statistical tests were designed in such a way that they would produce very high or low values when there is an important change in the dataset. In fact, the way in which the tests determine the position of the anomaly is based on detecting maximum and minimum values, that should correspond to those peaks produced by the equations.

The point is that it is not true that those peaks are always the maximum or minimum values of the data produced by the test, especially if the original data contains clear trend or seasonality. It is usual that trends and high variation in the data alter the results of the statistic calculated, so the are some 'peaks' that correspond to anomalies, but they are not absolute maximum and minimum values. For instance, looking back again to the mentioned figure, in the case of the Pettitt test, the two 'peaks' that are observed (and correspond to the location of outliers) are local maximum values, but they are surrounded by higher values.

When looking closely to the mentioned 'peaks', it is possible to observe the following:

- When there is an important change in the original data, there are peaks representing local maximum and minimum points;
- While errors like bias and drift show a low number of peaks, others like random and malfunction, that produce many variations in the data, produce many peaks in the statistics as well;
- The peaks seem to be produced in such a way that the angles between the previous points and the following points are close, compared to other parts of the time series, in which there are local maximum and minimum points with smooth slope.

Therefore, it seems it would be possible to identify outliers by analysing the dataset produced by the statistical tests, detecting the local maximum and minimum points and studying the angles generated in those points.

5.2.1. Outliers Detection Approach

The approach proposed for outliers detection using the statistics is not very complex, but it requires to follow several steps. Since the way to detect the outliers was rather visual and it was possible to identify them looking at the plots, the idea was to follow some tools from the area of geometry, that analyses curves and their properties.

The model proposed requires identifying the small peaks in the dataset generated by the statistics, determine the curves that represent that area of the plot and calculate the angle that is formed in the intersection of the curves. Then, depending on the angle values, it might be possible to determine whether it seems an outlier is present or not.

Therefore, according to this approach, the steps to follow are the following:

1. Calculate the homogeneity tests for the dataset under evaluation, storing the complete time series generated by each test;
2. Determine local maximum and minimum values in the new time series obtained;
3. Using nearby values as reference, determine the curves that form the 'peak' representing a local maximum/minimum;
4. Calculate the angle formed in the intersection of the calculated curves;
5. Check if the angles obtained for each test reach some threshold that indicates that there might be an outlier in the position of the local maximum/minimum analysed.

The first step has been already shown in Section 4. The proposed model calculates five statistical tests using the R *trend* package: SNHT, Pettitt, Buishand R, Buishand U and Lanzante. In the case of SNHT, the dataset generated with the T statistic is retrieved. Pettitt generates a dataset with the U statistic, while Buishand R generates the R / sqrt(n) statistic time series and Buishand U the U statistic time series. Lanzante generates another time series from its statistic called W.

The next step requires to find the local maximum and local minimum values in those time series obtained. In order to do so, the model utilizes the *ggpmisc* R package, which provides a function called *find_peaks*, able to detect those local spikes in which the trend of the data changes. It can be configured to determine how many values are used to identify the 'peaks'. As using only nearby values (this is, just three values of the dataset) many peaks were detected (and most of them were not relevant at all), the span parameter has been configured as 5, so the number of peaks identified is much lower and all of them have some relevance.

Additionally, since this library only detects peaks (representing local maximum values) but not 'valleys' (that would represent local minimum values), the function is invoked twice, but the second one all the input values are inverted, so the 'valleys' now become 'peaks' and the function can detect them as well.

The example in *Figure 41* shows a sample selected from a temperature dataset in which there is a drift error (almost at the end of the sample, around position 160). With the proposed configuration, when identifying local maximum and minimum values (turning points) in the Pettitt U statistic, it only identifies five values: three local maximum values (in red) and two local minimum values (in blue).

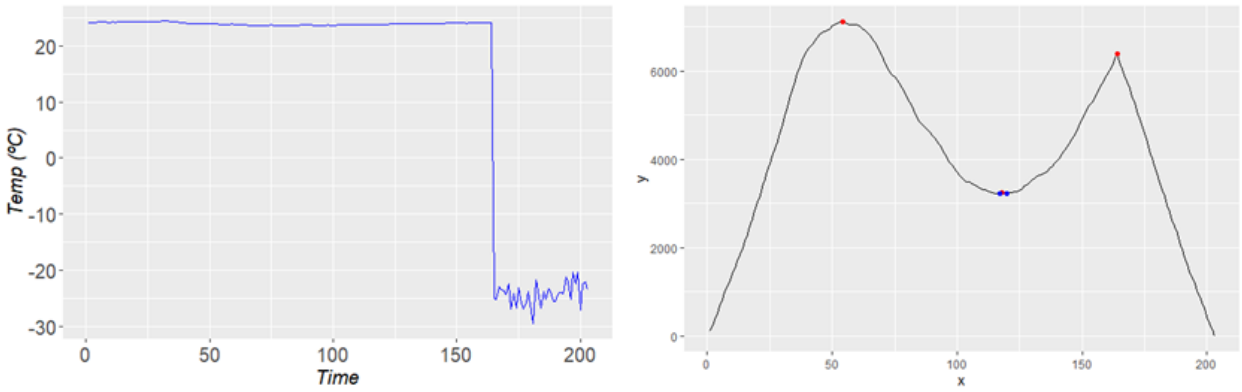

Figure 41 Local maximum and minimum values identified for the Pettitt test (temperature)

Once the turning points have been identified for each statistical test result, it is necessary to iterate through all those values, identifying the lines that form the change of trend in those areas. In order to do so, the model takes as reference the local maximum/minimum under study, together with its previous value and its next value.

Only two points are necessary to build the equation of a line, in such a way it is possible to represent it using the following equation:

$$y = m * x + n \tag{5–2}$$

In order to do so, it is necessary to calculate the slope (m) and the n parameter. This can be done using the coordinates of the points selected to draw the line using the following equations:

$$m = \frac{y_2 - y_1}{x_2 - x_1} \tag{5–3}$$

$$n = y_1 - \frac{(y_2 - y_1) * x_1}{x_2 - x_1} \tag{5–4}$$

The first line is determined by using the turning point with coordinates (x_2, y_2) and the previous point with coordinates (x_1, y_1). On the other hand, for the second line, the turning point is represented as (x_1, y_1) and the next value takes the coordinates (x_2, y_2). As a result, it is possible to plot the corresponding lines in the time series generated with the Pettitt statistic, as shown in *Figure 42*. Such figure only contains the examples for the first maximum, the first minimum and the third maximum since the others are very similar to the first minimum plotted.

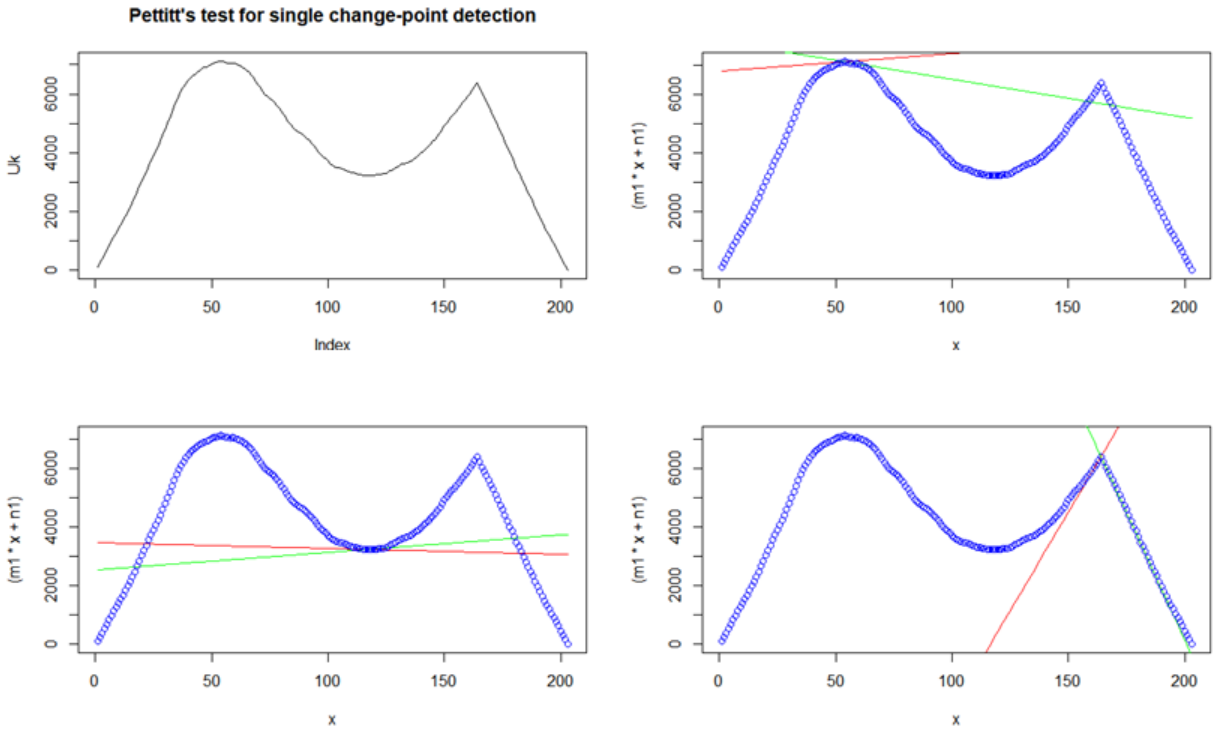

Figure 42 Lines generated for turning points identified with the Pettitt statistic (temperature)

Although drawing the lines is interesting and helpful, the value that is necessary to analyse is the angle with which the lines intersect. Assuming that with the previous equations two slopes were obtained (m_1 and m_2), the tangent of the angle would be calculated with the following equation:

$$\tan(\theta) = \left|\frac{m_2 - m_1}{1 + m_1 * m_2}\right| \qquad (5\text{–}5)$$

Therefore, the angle can be obtained by using the arctangent with the value obtained from the previous equation. Additionally, as the result is obtained in radians, it is easier to analyse if the result is transformed to degrees. The following equation shows how it should be implemented.

$$\theta = \tan^{-1}\left|\frac{m_2 - m_1}{1 + m_1 * m_2}\right| * \frac{180}{\pi} \qquad (5\text{–}6)$$

The problem with such equation is that, since the tangent value is the same for several angles, it is necessary to look for an implementation of the arctangent function that can differentiate the angle that generated such value with the calculated lines. For instance, the tangent value is 0 for the angles -2π, -π, π and 2π. Therefore, instead of using the *atan* function in R, the *atan2* function was used in the implementation. Such function, instead of using as input a single value (the one calculated with the 5-4 equation), takes as input the numerator and denominator of the equation in a separate way, being able to build internally the vectors that represent the lines and providing the angle value that is needed.

The angles obtained are the supplementary ones from the those that are to be used, so an additional transformation is necessary. Moreover, the angles must be transformed considering whether they are calculated with a local maximum or a local minimum as well.

Looking at the figures, the first line calculated is the red one, and the second line calculated is the green one. When calculating from local maximum values, the relevant angle is the one looking at the bottom, while in the case of local minimum values it is the one looking at the top (in fact, they are calculated as negative numbers). Therefore, they require different transformations, done with the following equations:

$$\acute{\theta} = (-180 - \theta) * -1 \quad (5–7)$$

$$\acute{\theta} = 180 - \theta \quad (5–8)$$

The first equation is the one used for those angles calculated from local maximum turning points, while the other one is used for the local minimum values.

All the angles are stored and analysed independently, with the purpose of identifying those peaks that could represent an important change in the original dataset.

5.2.2. Analysis of the Angles Produced

Following the proposed approach, it is possible to obtain the angles that are formed by the turning points in the data generated by the statistics, but it is necessary to study which angles may represent an outlier and which ones only represent the normal operation of a sensor.

The previous figures showing the time series generated by the tests are not at the right scale for looking at the angles (since they use to have a few values in the x axis, but the scale of values in the y axis is much higher). Therefore, it is necessary to take a close look at the results obtained.

The angles representing outliers seem to be close in general, but they are not the same, depending on the statistical test used. In the case of SNHT, the angles representing outliers are small and they seem to be around 30º. In the case of Buishand R and U tests, these angles are not so small and they seem to be around 85º. Finally, in the case of Pettitt and Lanzante, the angles representing outliers seem to be really small, in some cases smaller than 1º, although in others a bit higher (around 10º).

In all cases, when there are no outliers, the angles seem to be large, reaching around 160º for SNHT test, 175º for Buishand tests and 50º for Pettitt and Lanzante tests.

Therefore, in principle, taking into account a few experiments, it is possible to say that it should be feasible to differentiate those angles representing outliers and angles representing normal data, but a deeper analysis is necessary for confirming this aspect. In fact, further analysis enables the possibility to define clear thresholds for the angles generated by each test, so the model will be accurate enough.

The way to look for those thresholds was to do different executions of the solution proposed using several datasets available. The objective was to check how the approach behave with different types of errors and with different sensors.

The first experimentation was done with the benchmark datasets, selecting two temperature sensors from the Intel system and one temperature sensor from the Santander

system. For all of them, the datasets for the four types of errors injected were used (bias, drift, malfunction and random). In all cases, the samples extracted contained 202 values, so they were not very long, but representative enough to be able to detect anomalies.

In the case of the Intel temperature sensors, the selected samples were equivalent to 1 hour and 40 minutes of data. For the Santander temperature sensor, the samples represented almost 17 hours of data. The figures below show the results obtained for the different types of errors and all the types of errors injected. Additionally, Appendix B includes some tables with additional data about the outcomes of the tests done.

When using these datasets, seven samples were selected for each one, from which five contained errors and two were correct. This was applied to the four types of errors: bias, drift, malfunction and random.

SNHT shows a few turning points and good differentiation for bias and drift errors, but it is a bit more complicated with malfunction and random errors, with the amount of turning points increasing and bigger angles. Some of the angles in the case of error are closer to the minimum angles calculated, instead of to the maximum ones, but there is no problem to ignore the maximum values, since other angles in the presence of error will be smaller (and the error will be detected).

Figure 43 Box plots with angles obtained for SNHT in benchmark datasets

With Pettitt and Lanzante, maximum angles in bias errors use to be lower than 9.59 when there are errors. They used to be around 1.0, but they also reached higher values just a few times. With drift, they show some difficulties to detect some outliers, although it only happened once. In general, it is clear that malfunction and random errors generate more turning points as well, and the angles when errors are present are very close to 0 and there is some consistency when comparing different types of errors.

In this case, it seems that selecting a threshold lower than 1 could be valid for detecting most of the errors and for minimizing false positives.

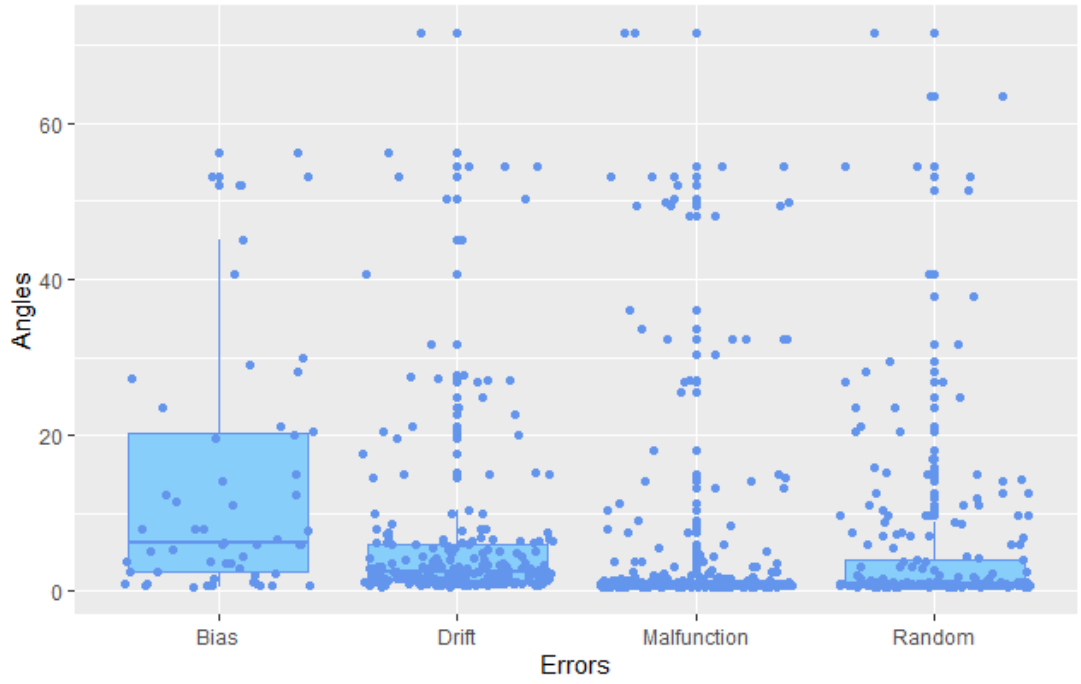

Figure 44 Box plots with angles obtained for Pettitt and Lanzante in benchmark datasets

The case of Buishand R and U tests is a bit different. There is a clear distinction, again, in the number of turning points when the data analysed contains malfunction and random errors but, in their case, it seems that detecting bias and drift errors is a bit complicated because angles are high in general. This makes a bit difficult to define the adequate thresholds because the small angle values for drift and bias are a bit high for random and malfunction.

Figure 45 Box plots with angles obtained for Buishand tests in benchmark dataset

In general, although there were some cases in which the proposed solution did not detect an error (producing large angles when small ones were expected), in most of the cases it was possible to observe a distinction between the datasets with and without errors. Although there may be some overlapping in the boxes depending on the type of error and statistical test, choosing small values as the threshold could be enough to cover most of the cases avoiding false positives, although a small number of errors would not be detected.

Another experiment was carried out with one of the Ports of Spain dataset (the one from Golfo de Cádiz), to check the results generated for the atmospheric pressure sensor. The window size corresponds to 6 days of data. This data contained a few wrong values (as outliers) and a small number of points representing a bias error (several values as -99.99, as data from the sensor was not received).

The outcomes showed that, when there are only a few points representing outliers, the tests and this approach are not able to detect the error (producing large angles). This only happened in one case (so it was not considered for building the table), behaving much better with the rest of cases.

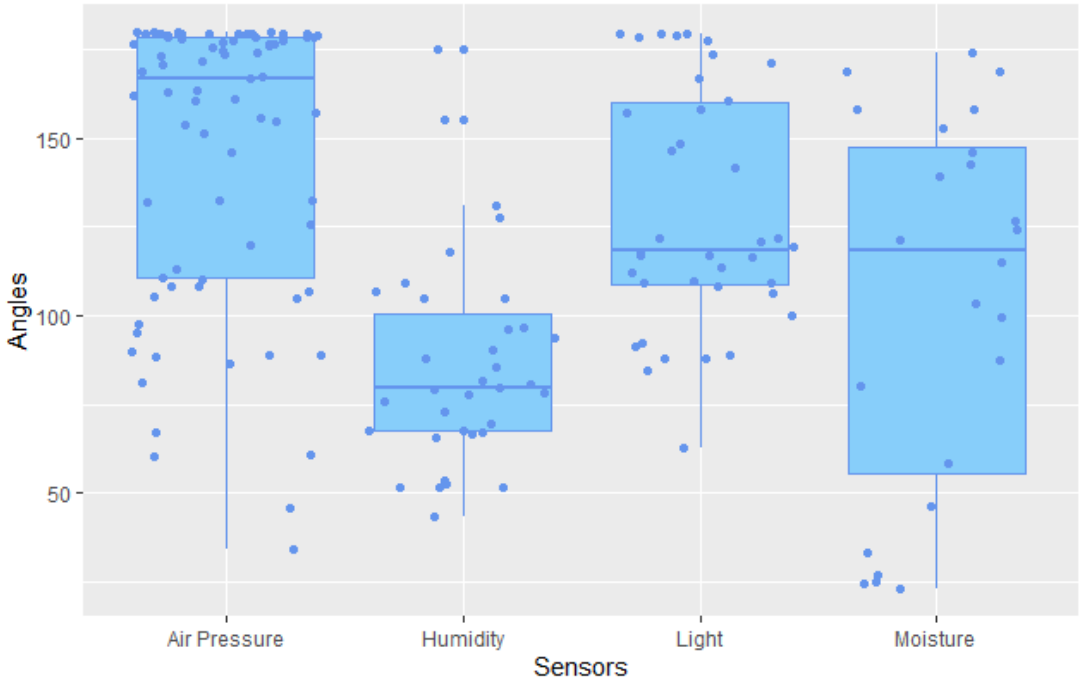

Figure 46 Box plots with angles obtained for SNHT tests in different types of sensors

Then, the Arduino dataset was used in order to analyse the results obtained by using the measurements collected with the moisture and air humidity sensors. In both cases, the window size selected corresponds to 1 hour and 40 minutes (50 values per sample).

It seems angles become smaller in the presence of some outliers, like a produced error and the moments of irrigation (these last ones are not errors, but they should be detected). The problem in this case is that the measurements provided by the sensor show some problems in different parts of the dataset, which make this solution to report small angles in samples that should not contain errors (that is the reason to have so small angles with no errors). These samples show some variation in the measurements that may indicate a problem with the sensor. In fact, after collecting the data, the sensor used broke up completely, so it is not so unexpected to find errors not tagged in the dataset.

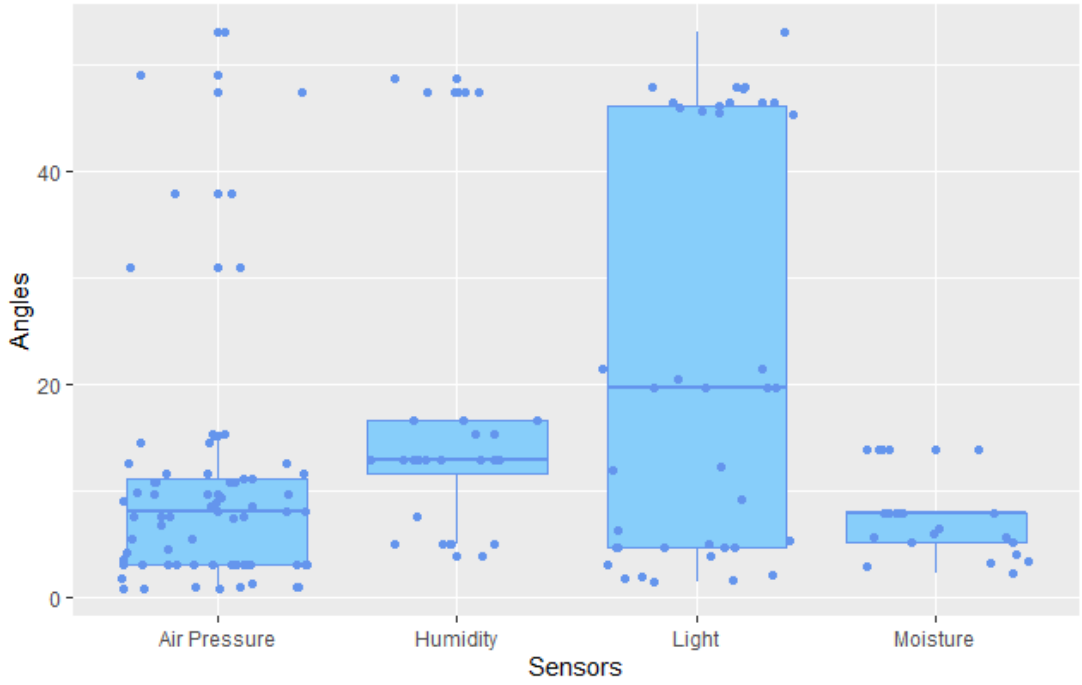

Figure 47 Box plots with angles obtained for Pettitt and Lanzante tests in different types of sensors

With this sensor, there is only one error in one of the samples. Although it seems there is a clear difference between correct data and the error, the angles generated for the error are not as small as in other cases observed before. This may be because the error is represented with very few values and, therefore, detecting it is very complex. It should be easier to detect it directly with the Grubbs' test.

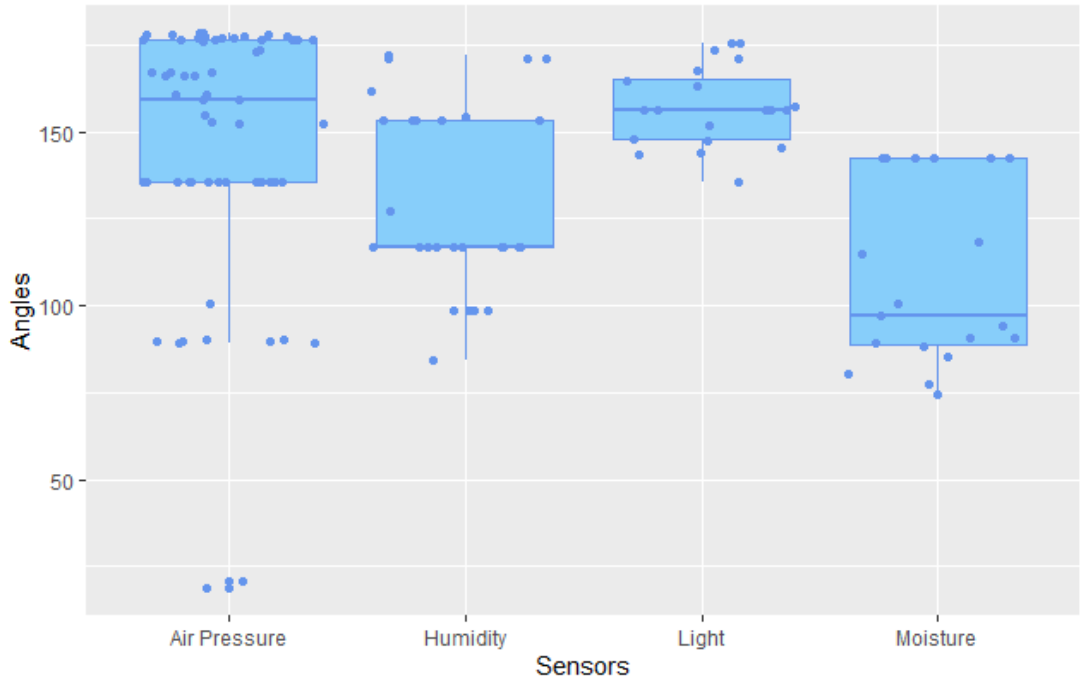

Figure 48 Box plots with angles obtained for Buishand tests in different types of sensors

Finally, one of the Toolbox datasets was used for looking at the results when applying this solution to a luminosity sensor located in the street. The windows size of the samples was

150 elements, corresponding to 30 minutes of data.

In this example, the angles calculated with SNHT and Pettit had issues only in one of the samples, but Buishand's tests were not able to detect any error. In the case of SNHT, the angles were higher than with other sensors, so it seems it is difficult to determine a set of threshold numbers that fit for all types of sensors.

It is interesting to mention that, in all the cases observed, Buishand R and Buishand U produced the same results, while Pettitt and Lanzante did the same. This has been observed in the great majority of cases in all the experiments done. Still, there are some cases in which they show different outcomes and, therefore, the decision is to keep using all of them.

Considering these observations, the thresholds defined for the different statistical tests are the following:

- *SNHT:* 45º
- *Pettitt and Lanzante:* 1.33º
- *Buishand R and Buishand U:* 88º

Since the Pettitt and Lanzante outcomes are so similar, they are aggregated with an OR operation. The same is applied to Buishand R and Buishand U outcomes. As a result, three outcomes are available (SNHT, Buishand R+U and Pettitt+Lanzante). Then, this approach proposes to report an error when, at least, two out of three should report an error. This way, it should be possible to reduce false positives, requiring some consensus between the different homogeneity tests used.

5.3. Processes for Data Understanding and Trust Evaluation

Taking into account the kinds of problems we may find in the sensors, according to the analysis performed, and their particularities, this work defined three decision processes in order to deal with sensors data during the data understanding phase (although it can also support in the data preparation phase, when data cleaning is carried out). Although the analysis covered several types of sensors and intends to be generic, it is necessary to assume these processes will have some limitations in terms of context of applicability.

These processes have been defined taking into account existing methodologies for data mining and machine learning activities. To be more specific, the base methodology we considered is the cross-industry standard process for data mining (CRISP-DM) (Wirth and Hipp, 2000) and one of its proposed extensions, much more focused on the engineering domain (Huber et al., 2019), that is relevant for IoT environments.

The data understanding phase is when there is a first analysis of the data that has been collected, so it is possible to understand the content of the data, the knowledge it represents and the problems that may be present with respect to the data quality (such as the presence of outliers). Thanks to the previous subsections, it has been possible to gain some deeper knowledge about the behaviour of sensors and the potential errors and problems that might be present in the IoT environment.

Additionally, the proposed approach is also useful as a way to identify anomalies in the sensors, determining whether we can trust in the values they produce for their integration in applications or for the creation of ML-based models that would use such data for training activities.

The way in which data varies, as well as the presence of outliers, are indicators of anomalies in the data most of the times, while the correlation can be a good instrument to confirm the existence of anomalies if there are equivalent sensors, complementing variation and outliers detection. Therefore, this work proposes the definition of three processes/algorithms for addressing the analysis of these aspects: variation analysis, outliers analysis and correlation analysis.

Looking at the different types of errors and their impact, the following table provides an insight of such impact and which kind of aspects could provide evidence of some anomaly. so, looking at the adequate combination of tests and transformations, it is possible to raise alerts about some device or sensor not working as expected.

Table 2 Mapping between types of errors and proposed processes

Type of Error	Characteristics	Variation	Outliers	Correlation
Malfunction	Small changes in homogeneity Local outliers Additional noise	x	X	X
Bias	Large changes in homogeneity Presence of outliers Constant values (lack of noise)	x	X	X
Drift	Large changes in homogeneity Presence of some outliers Additional noise in many cases	x	X	X
Random	Changes in homogeneity (moderate) Presence of outliers Additional noise (large)	X	X	X

A key point is the selection of the appropriate time window, depending on the problem to address. When analysing an existing dataset that will be used to build ML-based solutions, the full dataset should be analysed first, and then to select samples of different lengths, so it is possible to analyse what happened to the sensor in multiple contexts. Moreover, some issues become more evident with short samples, while others become more evident with large ones. On the other hand, when looking for outliers and analysing data streams, the usage of sliding windows is necessary, for a constant analysis of data and the adequate

detection of changes. In such case, the size of the sliding windows is not long, so it is applicable to real-time data streams and it enables a fast decision-making that would be applicable to real-time monitoring systems, like SCADA ones.

The following subsections provide details of the proposed algorithms and their interpretation, so data scientists and practitioners can determine the level of trust for a sensor and how to filter and clean the data.

5.3.1. Variation Analysis Process

The main purpose of this algorithm is to understand how the data is spread in time, so it is possible to determine if it is varying more than expected and, therefore, there could be some kind of error producing random data, or even constant values that we could link to a bias error. As we have seen in the previous subsections, we cannot rely on the normality of the data, so that should be taken into account, and variance is not always the best aspect to look at. Anyway, there are other mechanisms that can be used to detect anomalies, like the runs test (to determine whether values were created randomly), Ljung-Box test (for white noise detection), coefficient of variation and IQR (for determining if the variation of data is high with unitless approaches).

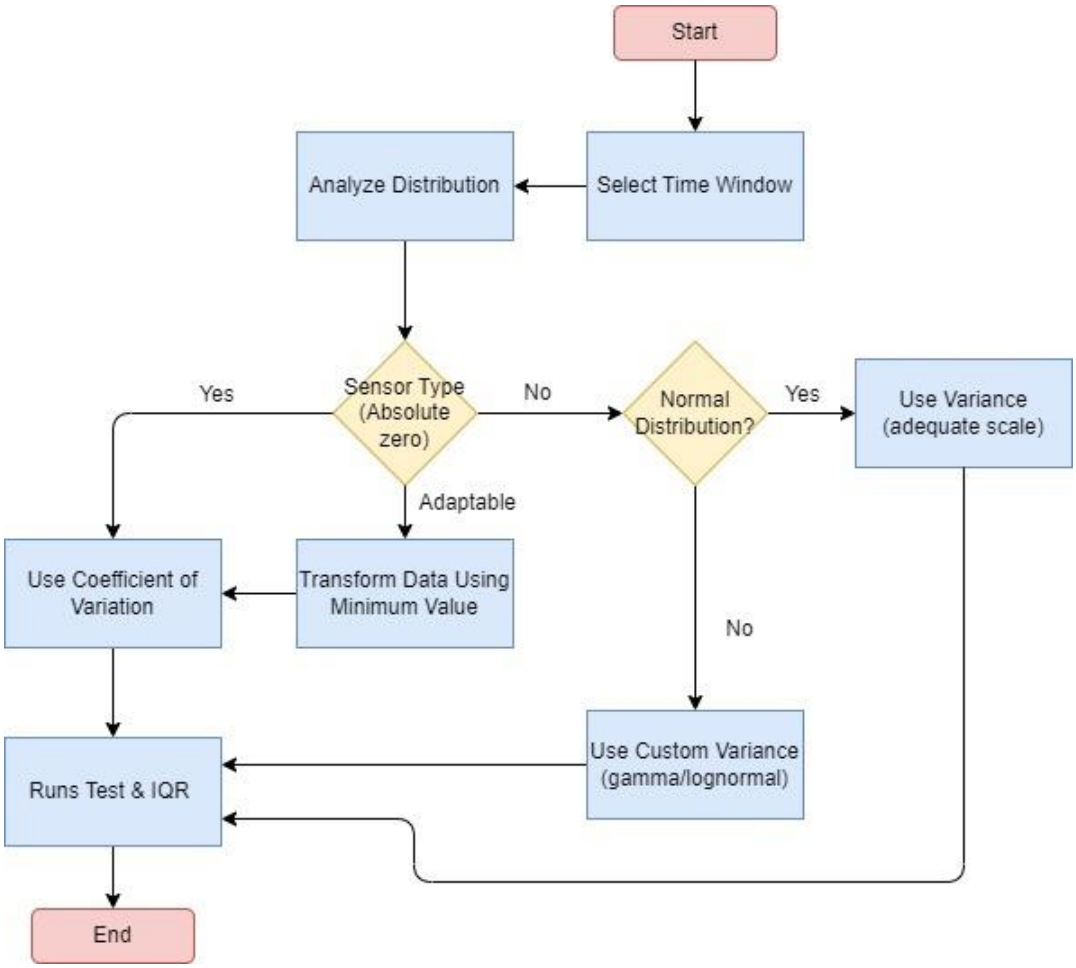

Figure 49 Process for analysing the variation in data

First of all, it is necessary to select the adequate time window for the samples and sliding windows. The proposed approach is to select long windows representing around the 6% of

the total size of the dataset, while short windows size is 1% of the total size of the dataset. For the sliding windows size, in the case of long windows, the proposed number of steps is 50, while for short windows this number is 15. These are numbers proposed for the implementation, but they could be changed depending on the dataset and the frequency with which the data is collected (for high frequencies, a larger number of steps could be necessary for the sliding windows).

The first condition to check is whether the sensor under analysis is measuring events with absolute zero, since this is important for calculating the coefficient of variation. This will depend on the sensor type and the way it measures (the units). For instance, temperature measured in ºC has no absolute zero (we may have negative values), but if temperature is stored in Kelvin, that kind of unit does have absolute zero (as it happens with light sensors as well).

In the case there is no absolute zero, it is necessary to transform the data. The way to do it is like when transforming ºC to Kelvin: add the number we need to have all the time series equal or higher than zero. It is necessary to determine the minimum value of the time series (that will be a negative number), multiply by -1 and add that number to every value in the sample. Then, the coefficient of variation can be calculated, although it is important to take into account that such transformation might not be valid. For instance, metrics like level of noise are measured in decibels, that follow a logarithmic scale and, therefore, adding 10 units to a temperature measurement has a very different effect that adding 10 units to a noise measurement. A coefficient of variation calculated following the proposed way with this kind of metric may not be very reliable and, therefore, it is not recommended.

Another condition has to do with the normality of the data. The way to calculate variance and the mean may be different depending on the distribution followed by the data. A first approach is to use Q-Q plots, Shapiro-Wilk and Anderson-Darling tests for checking out normality, but it is also necessary to fit the data to other distributions and to analyse the goodness of fit for them (Delignette-Muller and Dutang, 2015). Taking into account the observations done, the proposed distributions to check in the current implementation are gamma, Weibull, normal, uniform and lognormal distributions (although others like Paretto could be applicable as well).

Knowing the distribution, it is possible to calculate the adequate mean and variance, so they can be used in other calculations. It may be interesting to look at the variance with a Chi-square test if there is already some reference value, but it is not necessary at all. Then, this process proposes to calculate the runs test, the Ljung-Box test and the IQR.

In case that the runs test and Ljung-Box test produce a NaN (not a number) value, we may assume there are constant values and that is an indication of a potential bias error. If the runs test indicates that data seems to be random and the Ljung-Box test indicates that there is white noise, there could be other errors (high white noise could indicate the presence of random error, while low white noise or only runs issues could indicate the presence of malfunction or drift problems). A high IQR can be also related to some errors, especially to those impacting the homogeneity of the sample like bias, drift and random. In order to do so, it is interesting to compare previous values with the calculated IQR and determine if the current one is much larger than an average value of the previous ones. If such average value is larger, perhaps previous values are related to the presence of some error.

5.3.2. Outliers Analysis Process

The process for analysing outliers represents an algorithm focused on determining if there are changes in the homogeneity of the data and on detecting unexpected peaks. The test proposed by Grubbs is the reference for detecting the presence of outliers (as well as the ESD method, as its generalization if we expect multiple outliers). On the other hand, homogeneity can be checked out with several tests some of them parametric (SNHT, Buishand range and Buishand U) and others that are non-parametric (Pettitt and Lanzante).

The process starts with the selection of samples, in the same way as it has been proposed for the variation analysis process. As mentioned, it is up to the user to determine if other time windows may be more appropriate, since an automatic selection is not implemented.

Once the samples are available, the first step is to detect the presence of strong trends in the data. This is done by applying the Mann-Kendall test to each sample (other tests may be used as well). In the case that trend is detected, the current implementation transforms the data using the 'diff' function, that eliminates the temporal dependence (as it subtracts the previous observation from the current observation for all the values in the sample). Thanks to this transformation, it is possible to reduce the trend and some seasonality (although seasonality is not expected unless the time windows are very large, according to the experimentation reported in previous subsections).

After such step, it is important to transform the data in such a way that outliers will be highlighted. It is the adequate moment to use the transformations proposed at subsection 5.1 of this document. Both, the polynomial regression transformation and the exponential difference transformation may be useful.

When the data is transformed, it is necessary to determine whether the data is following a normal distribution or not. If it is following a normal distribution, all the parametric and non-parametric tests are calculated. This includes SNHT, Pettitt, Lanzante, Buishand U, Buishand R and Grubbs tests.

If the data is not following a normal distribution, the non-parametric tests are applied (Pettitt and Lanzante) as well as SNHT and Grubbs (since these last two, even with non-normal datasets, use to provide meaningful results). Buishand range and Buishand U may be calculated, but they should be ignored, as their results might be misleading.

In all cases, the statistics are calculated with two datasets: the original dataset (without trends and seasonality) and the dataset transformed for highlighting outliers.

It is important to look at the statistics results and, in the case of the homogeneity tests, also to the plot generated with the statistic they calculate, since such plot can also provide indications about the presence of outliers, even if the p-value does not report the presence of outliers. This can be done following the solution proposed in Section 5.2, that proposes a way to study the angles produced in the plots.

If the Grubbs test reports the presence of outliers, it is possible that the sample contains any of the four types of errors under analysis (malfunction, bias, drift or random). If the homogeneity tests report that there is a change in homogeneity, this is linked to the presence of bias, drift or random. It is interesting to compare the results of the tests

obtained when using the different datasets, as it may indicate additional information. For instance, if Grubbs is positive with the transformed dataset and homogeneity tests are positive with the original dataset, that indicates that there is a bias or drift error. Therefore, it is necessary to look at the outcomes together and obtain a consensus about the result.

This information is complementary to the one provided by the variation analysis, since combining such variation information could indicate the concrete type of error.

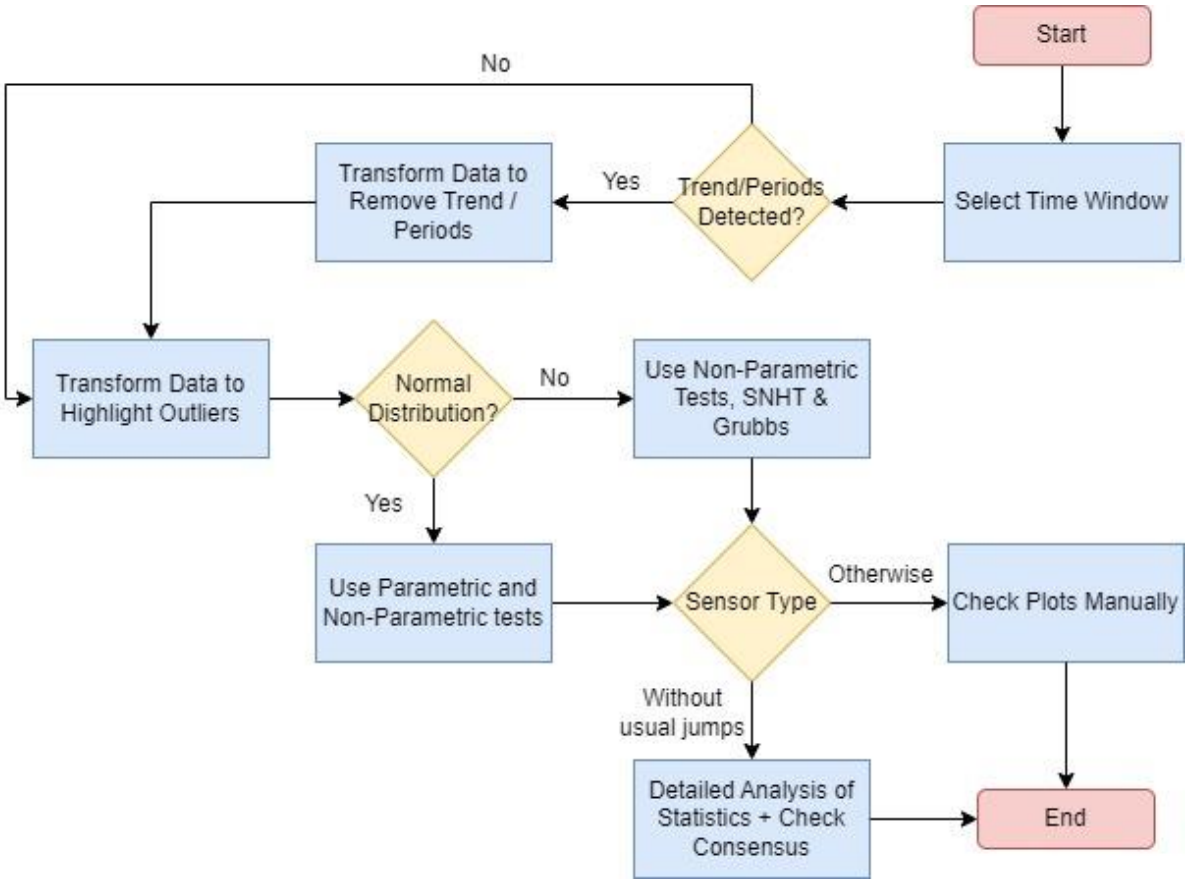

Figure 50 Process for analysing outliers in data

It is important to indicate that, depending on the type of sensor under analysis, it might be necessary to manually check plots of the data, since some types of sensors (such as salinity and luminosity) may generate misleading results in the tests because the natural event they measure generates changes in the homogeneity of the data in a normal way.

5.3.3. Correlation Analysis Process

The last process proposed is related to the analysis of equivalent sensors when they are available. This is done analysing the correlation between sensors that should have a similar behaviour, because they measure the same event and are located near enough to consider them equivalents (the distance may depend on the metric measured and the whole context of the sensors). The three tests used for analysing correlation are Pearson, Kendall and Spearman.

First of all, as in the case of the other processes proposed, we select a time window and extract several samples from the original dataset. Although correlation could show potential issues with short windows, it is also interesting to take a look at large time

windows as well, in order to get a general idea about the relationship between datasets. Obviously, if the sensor under analysis has no equivalent sensors defined, the process cannot be carried out.

A first step is to take a look at the distribution of data, since the Pearson test assumes that data is distributed normally and may provide misleading outcomes if this is not the case. Also, if it is already clear that there are outliers, Pearson will show low levels of correlation (much lower than without outliers).

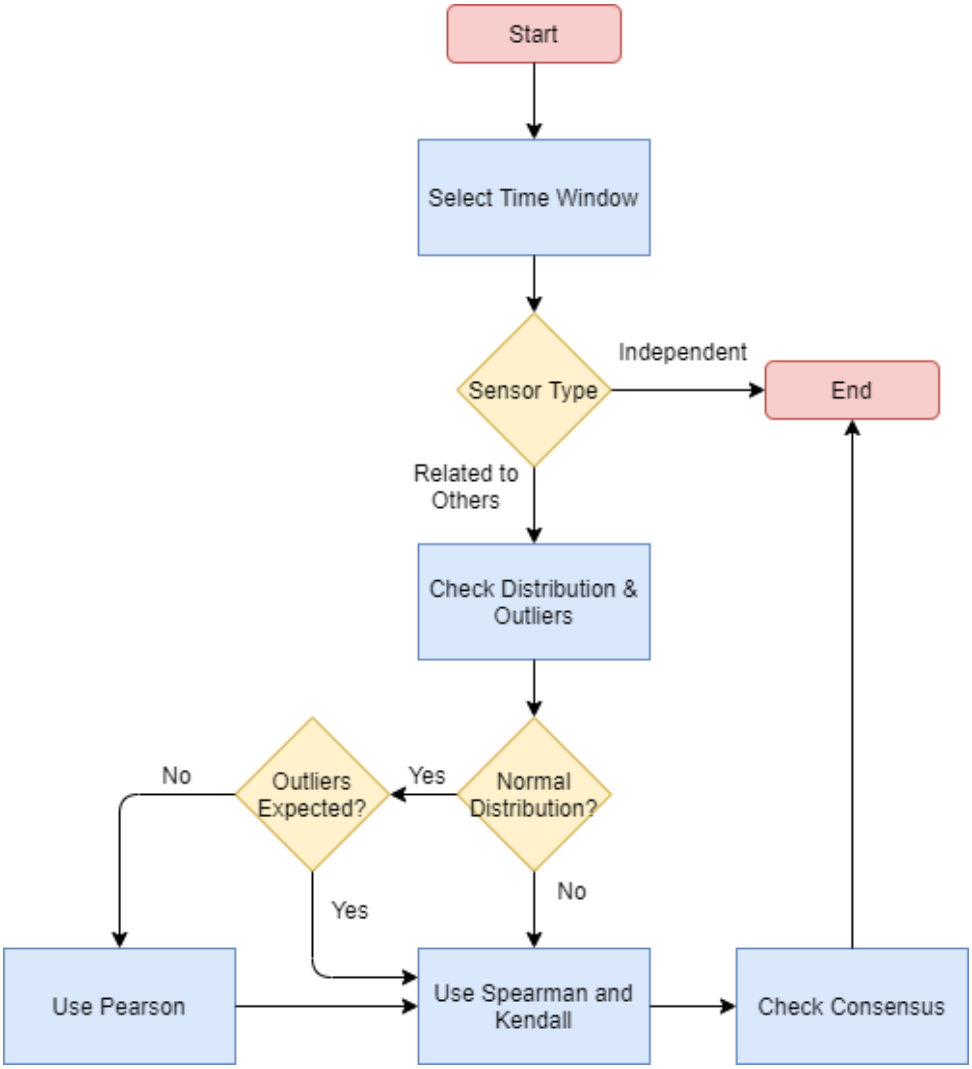

Figure 51 Process for analysing correlation in data

In any case, it is interesting to calculate Pearson test together with Kendall and Spearman, but giving less weight to the outcomes produced by Pearson in case we detect issues with the assumptions mentioned before. The objective is to reach a consensus that will determine if it seems that there is some correlation between the sensors (linear or non-linear), taking into account that values close to 1 indicate strong correlation, while values close to 0 indicate no correlation.

Additionally, it is important to compare the results obtained with different samples if there are doubts about the presence of errors. As shown in previous sections, the presence of errors like bias, drift and random has an important impact in the correlation results, while for malfunction the impact is lower. Therefore, the comparison with the outcomes from other samples could indicate if some error is present, even pointing to the type of error.

5.3.4. Trust Evaluation

The previous subsections propose a set of processes for analysing the data and finding anomalies. Still, it is up to the user to do an interpretation of which sensors can be trusted or not, as some of the results obtained may provide contradictory results. Therefore, in order to enable a measurable trust evaluation, this work proposes a way to aggregate the outcomes obtained with a type-1 fuzzy model, getting a specific trust value for each sensor evaluated.

This aggregation is based on the results observed when applying the proposed solutions to several heterogeneous datasets, considering the types of errors that sensors can produce. The analysis done allowed to find some rules that were consistent in most of cases when bias, drift, malfunction and random errors were present in the data.

Taking into account the classification of trust models suggested by (Alghofaili et al), the proposed trust model defines a clear Trust Composition (in fact, it is very focused on aspects that are related to QoS, as it evaluates the quality of the data produced). Trust Formation is multi-trust, since multiple aspects are taken into account. The Trust Aggregation property is mainly static, since the aggregation is done through a fuzzy model (based on a set of rules) with weights and rules that do not vary, as their importance is not expected to change (although the model could be reconfigured).

The proposed model does not deal with Trust Update, as this should be implemented as required by the user of the model and the context. It is recommended that the sensors' data (and the corresponding trust) is re-evaluated periodically. Depending on the criticality of the system, the period selected could be different. An IoT system controlling the production in an industrial environment should re-evaluate the sensors all the time, with rather short-medium windows of data, while other types of systems (like some applications in agriculture or open systems in smart cities) may evaluate the data once per day, or even less.

As for the Trust Propagation, this depends on the system including the model. It is feasible to have the model running in edge nodes or in a centralized Cloud environment that collects data from multiple sources. Therefore, it is feasible to use it in centralized and distributed environments.

Unlike other solutions in IoT, that are more focused on the trust evaluation of IoT devices and nodes, the proposed model is focused on the evaluation of single sensors. Therefore, it is possible to say that this is a ‘fine grain’ solution (working at a lower level) that could be used to evaluate sensors' data in the IoT nodes and, consequently, it is complementary to other solutions already described in Section 2.

The model requires a minor adaptation to be done to the data (transform null values to the extreme value -99.99). Then, the processes defined in the previous subsections are carried out, obtaining the outcomes of several statistics. With such results, it is possible to run the proposed fuzzy model, obtaining a specific value representing trust for the sensor.

5.3.4.1. Input Variables Definition

Thanks to the definition of the model for variability and the model for homogeneity tests aggregation, the number of inputs for the main model has been simplified.

The model for variability has three inputs, two of them modelled with a trapezoid membership function and one as gauss bell membership function:

- Runs test result (trapezoid)
- Number of runs (gauss bell)
- Ljung-Box test result (trapezoid)

In the case of the model for aggregating the homogeneity tests, the model has up to five input variables, one for each test used:

- SNHT (trapezoid)
- Pettitt test (trapezoid)
- Buishand R test (trapezoid)
- Buishand U test (trapezoid)
- Lanzante test (trapezoid)

There is a simplified version of this model, that only requires three input variables, which is used when the dataset analysed does not follow a normal distribution. In such case, only SNHT, Pettitt and Lanzante tests are used.

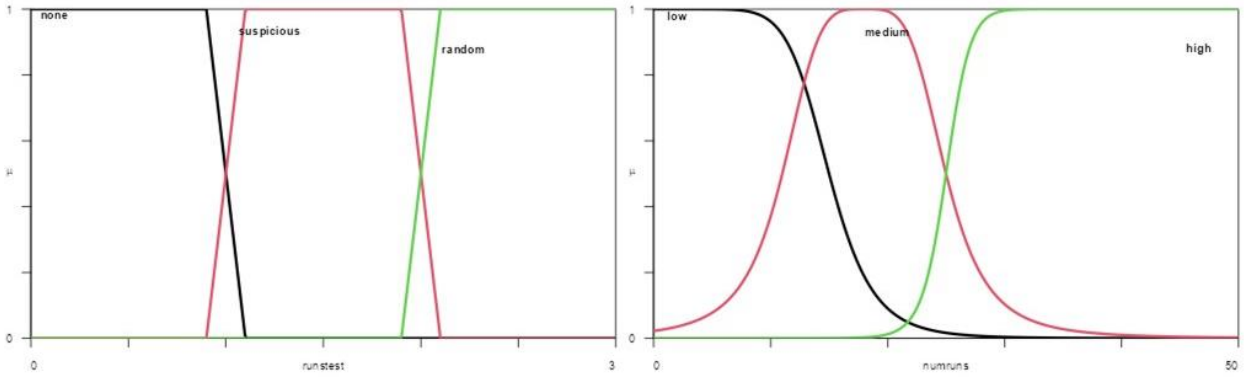

Figure 52 Examples of input variables for the variability fuzzy model (trapezoid and gauss bell)

The main membership function selected for the statistical tests is trapezoid because there are three clear options defined (none, suspicious, evidence), depending on the result of the p-value of the tests, and such p-value has a clear interpretation, being 0.05 usually the reference value to consider that there is evidence in the test result. In the case of the tests for this context, values between 0.025 and 0.049 are interpreted as suspicious situations, in which the test does not confirm the evidence, but the result is so close to it (as observed in the data analysis) that it is possible to consider that something might be going on with the data. Values between 0 and 0.024 are interpreted as lack of evidence (so the data is fine).

The functions are not represented in values from 0 to 1 because the membership functions are so unbalanced (the third one would go from 0.05 to 1) that the testing with such membership functions showed that the last one is always activated when fuzzifying the inputs, so the ruleset does not activate the expected outcomes. Therefore, they are defined from 0 to 3, with similar interval sizes, and a simple function calculates the proportional value for the input variable, depending on its corresponding membership function.

Finally, the main trust model has up to seven inputs, aggregating all the information

available from the previous models and additional tests:

- Variability result (trapezoid)
- Angles aggregation (gauss bell)
- Number of turning points (gauss bell)
- Grubbs test (trapezoid)
- Grubbs test with transformed data (trapezoid)
- Homogeneity tests aggregation (trapezoid)

The same concept as before is applied for representing the results of the statistical tests, so the ruleset will generate the adequate outputs. In the case of the variables represented with gauss bell membership functions, these are not very unbalanced for avoiding unexpected outcomes as well.

In all cases, there is some overlapping between the different membership functions defined for each variable, since otherwise the input variable may get a value of 0 (when fuzzifying) and that case has shown to produce problems in the output generation.

5.3.4.2. Outputs Generated

The proposed solution implements three models, and each model generates its own output variable (there is only one output variable per model). In all cases, trapezoid membership functions have been used, since they are easier to define and also to manage for obtaining the expected outcomes with the rules of the model.

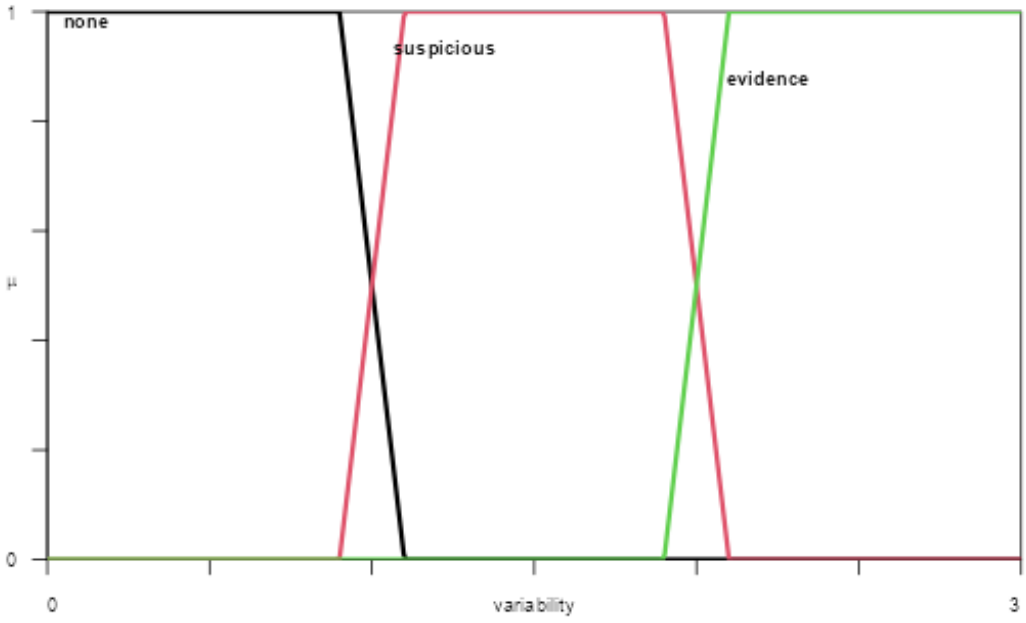

Figure 53 Output variable defined for the variability model

In the output definition for the variability model and the model for homogeneity consensus, three membership functions are defined, representing the options 'none', 'suspicious' and 'evidence'.

As in the case of the inputs definition, there is some overlapping between the membership functions as a way to avoid problems in the output generation (as the outcome of the

model may get a value of 0, which is problematic for deffuzifying).

On the other hand, the definition of the output variable for the main trust model has been designed with four membership functions, representing the values 'ok', 'suspicious', 'evidence' and 'severe'. The last one is for those cases in which there is a very evident issue with the dataset analysed.

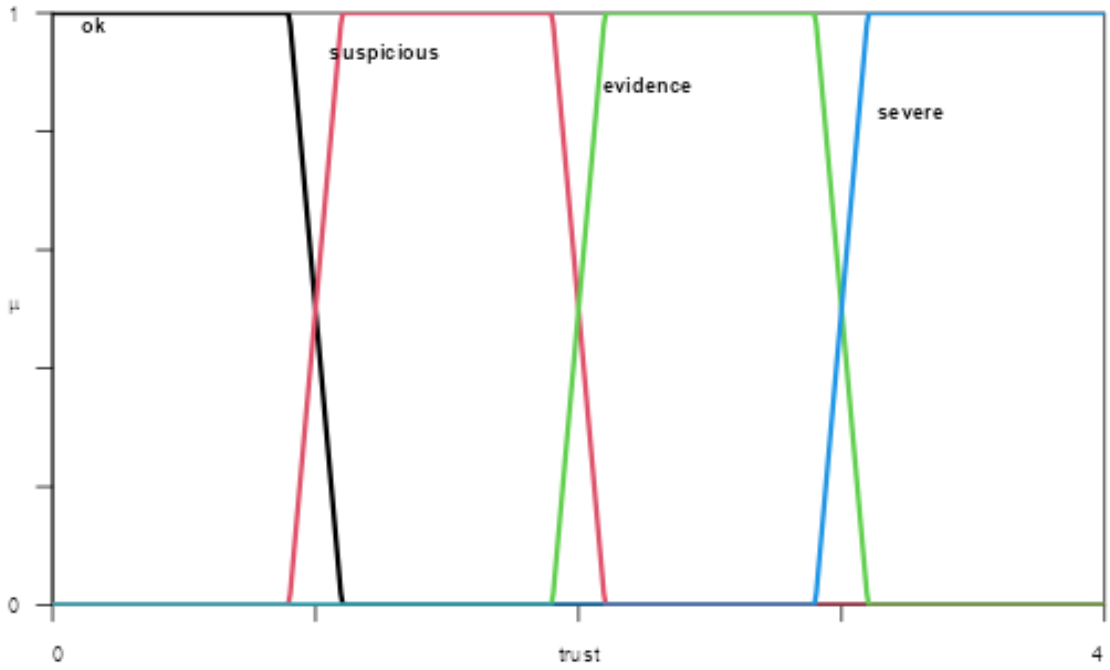

Figure 54 Output variable defined for the main trust model

Again, there is a small overlapping between membership functions to avoid issues. It is also important to note that the membership functions at the edges do not complete the trapezoid in a value of 0, but in 1, as this also avoid problems.

5.3.4.3. Rules of the Model

According to the observations done, a set of rules have been extracted to decide how to identify issues in the datasets by using the outcomes of the statistical tests and mechanisms used by the processes defined.

When defining the rules, different contexts have been taken into account:

- When the data follows a normal distribution, all the homogeneity tests can be used;
- When data does not follow a normal distribution, both Buishand R and Buishand U tests are not considered;
- When the sample size selected for analysis is small, the variability fuzzy model is considered;
- When the sample size selected for analysis is medium or large, the variability fuzzy model is not considered.

The variability model evaluates the result from those tests that provide information about the randomness of the data. Therefore, if the runs test detects random data (with many runs) and the Ljung-Box test detects noise, the model will indicate that there is evidence of an issue. On the other hand, if the tests do not detect randomness, it is assumed that the

data is fine. If only one of the tests detects an issue, this is considered evidence if the other one is close to detecting something. The following table lists the main rules defined.

Table 3 Ruleset for the variability fuzzy model

#	Rule	Output
1	runstest='none' ∧ (numruns='low' ∨ numruns='medium') ∧ ljungbox='none'	ok
2	runstest='suspicious' ∧ numruns='medium' ∧ ljungbox='suspicious'	suspicious
3	((runstest='none' ∧ ljungbox='suspicious') ∨ (runstest='suspicious' ∧ ljungbox='none')) ∧ numruns='medium'	suspicious
4	((runstest='evidence' ∧ ljungbox='suspicious') ∨ (runstest='suspicious' ∧ ljungbox='evidence')) ∧ numruns='medium'	evidence
5	runstest='evidence' ∧ (numruns='high' ∨ numruns='medium') ∧ ljungbox='evidence'	evidence

The model for the homogeneity tests is designed to find the consensus between the tests. It is similar to the consensus for variability, so when a certain number of tests detect an issue, the model considers that there is evidence. In the case all tests report that the dataset seems to be homogeneous, the model reports that there is no consensus. When some of the tests provide evidence of change in the dataset, the model reports suspicious behaviour. The table below shows the main rules defined for datasets not following a normal distribution.

Table 4 Ruleset for the homogeneity consensus fuzzy model (not normal distribution)

#	Rule	Output
1	SNHT='none' ∧ pettitt='none' ∧ lanzante='none'	none
2	SNHT='suspicious' ∧ pettitt='suspicious' ∧ lanzante='suspicious'	suspicious
3	((SNHT='none' ∧ pettitt='suspicious') ∨ (SNHT='suspicious' ∧ pettitt='none')) ∧ lanzante='suspicious'	suspicious
4	SNHT='suspicious' ∧ pettitt='suspicious' ∧ lanzante='none'	suspicious
5	((SNHT='evidence' ∧ pettitt='suspicious') ∨ (SNHT='suspicious' ∧ pettitt='evidence')) ∧ lanzante='evidence'	evidence
6	SNHT='evidence' ∧ pettitt='evidence' ∧ lanzante='suspicious'	evidence
7	SNHT='evidence' ∧ pettitt='evidence' ∧ lanzante='evidence'	evidence

This is valid for datasets not following a normal distribution because Buishand R and Buishand U tests are not considered, so the evidence can be achieved with two out of three positive tests. If the dataset follows a normal distribution, these two tests are added, and

the consensus requires three out of five positive tests.

The main model for trust evaluation defines some rules that detect the different types of errors. When those conditions are fully met, it is considered that there is strong evidence of an anomaly, so the trust is low. Otherwise, the trust is considered high. In case the conditions for strong evidence are not me, but still some of the input variables shows some evidence, the trust is considered medium, since it is not fully evident that there is an error, but there are reasonable indicators that it might be the case.

Table 5 Ruleset for the main trust model (long samples)

#	Rule	Output
1	angles='none' ∧ (num_points='low' ∨ num_points='medium') ∧ grubbsEDT='none' ∧ grubbs='none' ∧ homogeneity='none'	high
2	angles='low' ∧ (grubbsEDT='suspicious' ∨ grubbs='suspicious') ∧ homogeneity='suspicious'	medium
3	angles='med' ∧ (grubbsEDT='suspicious' ∨ grubbs='suspicious') ∧ homogeneity='suspicious' ∧ (num_points='medium' ∨ num_points='high')	medium
4	(angles='high' ∨ angles='med') ∧ num_points='medium' ∧ grubbsEDT='outlier' ∧ grubbs='outlier' ∧ (homogeneity='suspicious' ∨ homogeneity='evidence')	low (random)
5	(angles='high' ∨ angles='med') ∧ num_points='high' ∧ grubbsEDT='outlier' ∧ grubbs='outlier'	low (malfunct.)
6	(angles='high' ∨ angles='med') ∧ num_points='medium' ∧ grubbsEDT='outlier' ∧ homogeneity='evidence'	low (bias/drift)

In the case that the samples are short, the *variability* input variable is used, activating the 'low' outcome for the drift and random errors when it has the value 'evidence' and reinforcing the rules for 'medium' when its value is 'suspicious'.

Since the library used for the implementation only supports one operation between the variables, the code includes additional rules that implement the 'OR' operation of some of the rules, using only the 'AND' operator in most of cases. Additionally, those rules that take the highest and lowest values (such as the one having all inputs in 'evidence' and 'high'), have more weight than the others (they were configured with a weight of 3, while the rest of rules have a weight of 1), highlighting certain combinations of inputs.

5.3.5. Implementation of the Solution

All the solutions proposed have been implemented as R scripts:

- One script for the transformations,
- One script for the errors detection through angles,

- One script per process defined and one for the fuzzy models,
- One script integrating key points of the previous ones to facilitate trust evaluation.

These scripts read the data in CSV format and apply different statistical tests (using the libraries *nortest*, *trend*, *outliers*, *EnvStats*, *fitdistrplus*, *ggpmisc* and *randtests*), storing the results in a file with the statistics, angles and error detection information. The fuzzy model is defined and executed with the *FuzzyR* library, and it can use inputs already generated.

Since nowadays almost any device has multiple cores, the code was implemented in such a way that it can be executed by parallelizing the calculations in different cores, through the %dopar% feature (from the *foreach* and *doParallel* libraries). Taking into account the increasing number of sensors and data available, as well as the computational capabilities of new systems and devices, it is important to understand whether it is possible to reduce execution times, thus facilitating the inclusion of more complex pre-processing features, especially for edge devices (allowing them to apply some data cleansing and filtering before sending data to complex analytical tasks).

The implementation of the three processes defined (variation, outliers and correlation) selects several main indexes in a random way, as the point to start extracting data samples. Then, from those indexes, it generates a sliding window with the number of steps selected, having as a result a nested iteration that analyses all the window steps as if they were "screenshots" of the sensor data (in the same way a complex event processing engine would do).

The parallelization has been applied to the outer iteration, so each core analyses one complete sliding window of a main index. This allows the CPUs to take advantage of their cache memory, while giving each core enough computational complexity to benefit from the parallelization (otherwise, the overhead for managing parallelization would be too high compared to the calculation time). Spreading each "screenshot" of each sliding window among the different cores might require that the CPUs access the main memory many times because the data is not already stored in their L1 or L2 caches, whereas handling the process in the same core takes advantage of the existence of all the data values already loaded into cache memory, except one.

Once the implementation was completed and tested, the code has been organized in an R package. The package includes the functions for the transformations in one file, the function for calculating angles from the homogeneity tests outcomes in another file and the implementation of the processes in another file. All the functions are properly documented. The package also includes the Arduino station dataset in *'.rda'* format, so the users can run examples with some annotated data.

All the R scripts and the package have been published in a GitHub repository that can be found in the following URL: https://github.com/fjaviernieto/Sensor-Trust-Tools. The README file contains information about the structure of the repository and other details like licensing, references and a description on how to use the code.

The package can be easily retrieved and installed from the GitHub repository through a few simple R lines. It requires to have the *devtools* library already installed. Once it is installed, such library provides the *install_github* function, that retrieves and installs the

package directly from GitHub:

```
devtools::install_github("fjaviernieto/Sensor-Trust-
Tools/iotanomdet")
```

Then, the new package can be used as usual. A simple example using the included dataset could be executed with the following code:

```
library(iotanomdet)

temp_data <- arduino_station$tempC
result <- anomalies_analysis(temp_data, 200, 4, 4)
```

Such code will analyse four samples of the Arduino dataset selected randomly, with a length of 200 elements. And it will do the analysis using four cores of the CPU.

5.3.6. Performance Analysis

The way to check whether the parallel implementation worked as expected or not was through a performance analysis done in line with the guidelines defined by (Hoefler and Belli, 2015), that describe how to collect and report the information, and the kind of metrics that can be used to compare the observations done.

The three scripts corresponding to the processes defined (variation, outliers and correlation) were executed in a laptop equipped with an Intel® Core i5-8350U vPro processor (featuring four physical cores at 1.70 GHz and 3.60 GHz in turbo mode, 6 MB of smart cache and a bus with a speed of 4 GT/s) and a RAM memory of 8 GB DDR4 2400 MHz. The main software installed was a Windows 10 Enterprise (compilation 19041.1052) operating system with R version 3.6.2.

As it may happen that some executions perform better than others under the same conditions (there are some factors that cannot be controlled, that may affect the cache memory and the CPU utilization), the scripts were executed 10 times using as input the same dataset.

The time of execution of different parts of the code was measured using the "*system.time*" function of R (obtaining the elapsed time property in seconds). The initial load of the CSV files was omitted from the measurement, since in real-time systems data would be expected to be available in memory (through some data stream). There were five blocks of time measurement, with two blocks doing parallel operations and three doing serial operations (like writing results in file or setting up the cluster of cores).

Looking at the initial results, it is clear that adding more cores does not mean that the performance will improve. The maximum number of cores that was possible to use was seven. The main reason is that system used has four physical cores and allows to use eight virtual threads. In practice, when using more than five cores, the CPU was too overloaded because of the operating system and the R environment fighting for resources. In fact, in some cases, the R engine was failing eventually because such overload of the system, with the system reporting issues in the memory allocation.

Table 6 Summary of the performance analysis of the parallelized R scripts (seconds)

Script	Metric	Serial	2 Cores	3 Cores	4 Cores	5 Cores	6 Cores	7 Cores
Variation Process	Mean	94.72	48.42	34.37	27.73	34.77	23.58	31.99
	Best	57.48	45.23	24.05	21.39	29.52	21.08	26.87
	Worse	138.88	52.53	38.32	34.12	47.17	26.13	34.84
	Coef. Var.	0.28	0.04	0.13	0.15	0.12	0.06	0.06
Outlier Process	Mean	471.42	247.58	212.99	208.1	154.95	160.91	186.21
	Best	432.64	235.89	199.65	178.57	143.91	144.05	169.06
	Worse	535.75	255.68	218.5	230.13	159.91	178.54	202.19
	Coef. Var.	0.08	0.02	0.02	0.09	0.03	0.06	0.04
Correlation Process	Mean	4.36	2.82	2.47	2.45	2.63	2.75	2.83
	Best	4.09	2.68	2.36	2.39	2.55	2.53	2.77
	Worse	4.69	3.25	2.89	2.53	2.75	2.97	2.92
	Coef. Var.	0.05	0.07	0.05	0.01	0.01	0.04	0.01

It is also worth mentioning that the coefficient of variation is low, which means that the time it took to carry out all the executions was stable.

When looking at the plots obtained with the scripts, it is also possible to see that the performance was not as good as expected in some cases, although all the scripts show some improvement thanks to the parallelization.

The plots in the first line correspond to the variation process, those in the second line correspond to the outliers process and the last row corresponds to the correlation process. Plots at the left show the speedup obtained compared to the ideal (linear) situation and to the serial overhead ideal situation, that was determined using the Amdahl's equation (Amdahl, 1967). On the other hand, the plots at the right side show the execution time measured (in seconds), also compared with the linear scalability (ideal) and with the serial overhead ideal situation (using the Amdahl's equation again, but this time using the serial execution time as the base of the equation).

The ideal linear scalability assumes that all the code could be parallelized and therefore, the load can be split perfectly between all the cores (1/number of cores). But, in the case of the Amdahl's equation (5-9), it is possible to differentiate the code that must run in a serial way from the code that can run in a parallel way, so it represents a more realistic boundary. In the equation, *pctPar* represents the estimated percentage of parallel code, while *p* is the number of cores used.

$$Speedup = \frac{1}{(1 - pctPar) + \frac{pctPar}{p}} \quad (5–9)$$

The percentage of parallel code was estimated by measuring the blocks of code established with the "*system.time*" function. Two parallel blocks and three serial blocks were defined and measured, running 10 times the code in a serial way and obtaining the average execution time of each block.

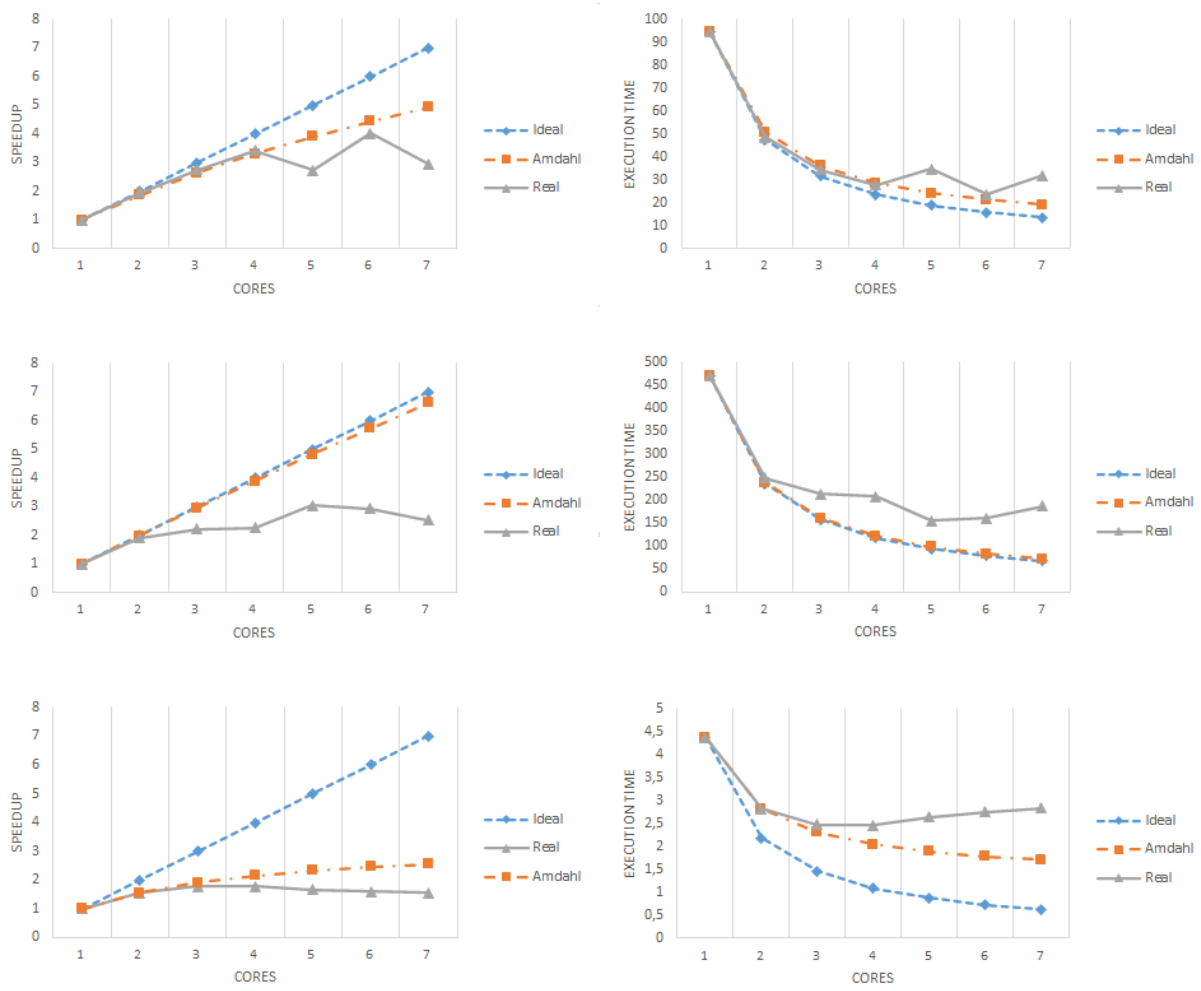

Figure 55 Plots showing the performance of the parallel version of the scripts

According to the plots, using two cores there is an important improvement, and three cores still show good results in scalability. When using more than three cores, only the variation process is able to scale well, but the rest is not able to do so. In fact, in some of the cases, the scalability is even worse than with three cores. Since the CPU used only had four physical cores, it is not very clear whether it would be possible to improve the scalability if the cores used are physical, and not virtual.

The case of the correlation process is a bit special, since around 71% of the code is estimated to be parallel, while in the case of the outlier script 99% of the code was parallel and in the case of variation it was 93%. That means that the scalability in the correlation process is more limited and that using more than three cores was not beneficial (due to the 29% of serial code).

Another point to observe is that it seems that the real execution measurements of the variation script were a bit better than the Amdahl's boundary in some parts of the plot. This improvement is very low (0.1), and it may happen because the parallel execution does not print messages (while the serial execution does) and because it may be benefitting from memory and hard drive caching mechanisms.

The main positive point is that applying parallelization reduces the execution time, and this is very relevant for real-time environments like edge computing, where it is important to be efficient and have the capability to process more data as it arrives (e.g., detecting problems, applying filters, cleaning data, etc.). Therefore, it makes sense to invest effort in this kind of parallelization, especially taking into account that it is not so difficult to implement (compared to parallelization with multiple nodes, that requires much more specific knowledge).

5.4. Discussion

The work presented in the previous subsections has addressed the improvement of the way to detect anomalies in datasets exploiting the statistical tests and other calculations studied in Section 4. One of the main areas selected for such improvement was the way to transform the data in such a way it would facilitate the analysis with statistical tests. On the other hand, as found when analysing the outcomes of the homogeneity tests in Section 4, it seems that the time series generated by the tests might be useful to identify the presence of outliers as well, complementing the results of the tests.

All these improvements have shown that they can facilitate the identification of anomalies, but it was necessary to formalize somehow the way to proceed to do so. Therefore, this section also defined three models that are implemented as processes to carry out data understanding and to determine whether a sensor can be trusted depending on the data it generates. Thanks to such processes, the data processing activities should be easier for data scientists, since they are useful for guiding their work.

5.4.1. Data Transformations

When studying the way to transform the data in such a way that it would be easier to find outliers, one of the main steps was to look at the transformations that are usually applied in the area of data analytics and ML. Surprisingly, most of them are focused on improving the normality of the data and only a few of them are oriented to a modification of the data for eliminating trends and seasonality. Some of them support the scale of the metrics and, although one was mentioned to be useful for removing outliers, it was not in line with the proposed solutions.

Therefore, two different types of transformations were proposed in this work. One of them based on the difference transformation adapted with an exponential factor, that was giving decent results as is but had margin for improvement. The other one was based on the usage of polynomial regression so, when subtracted to the original data, it highlights certain values from the dataset.

The first one has shown to be interesting for avoiding false positives with homogeneity

tests, especially when there is no error and there is a clear tendency in the data. Additionally, it reduces drift and bias errors to clear outliers easily detected with the Grubbs test. Still, with errors like malfunction, it does not perform so well with homogeneity tests (although it still works fine with Grubbs).

The second one is not as effective as the first one in general, but it has shown to be a very good complement, especially when using the homogeneity tests, since it manages to highlight jumps in the data a bit better, making the tests to work fine when facing errors like malfunction and random.

An additional fine tuning of the transformations, especially for the one combining the difference transformation with an exponential factor, could improve the results for sure. Automating some parameter analysis with Monte-Carlo simulations could optimize their configuration and increase their efficiency.

Another area to explore is the combination of transformations that normalize the data with an adaptation of the exponential difference transformation. The current definition of this last transformation highlights values when the difference between nearby values reaches a certain threshold that may not be the most adequate one for all types of sensors (since they are ruled by different metrics and range of values). Applying a normalization transformation (like Yeo-Johnson), the result facilitates that the range of values for any sensor will be the same. In such circumstance, it is possible to fine tune the exponential difference transformation coefficients to highlight outliers with the same intensity for any sensor. For instance, if the range of normalized time series is between 0 and 1, the exponential difference transformation may start highlighting values strongly when the difference is higher than 0.5.

In any case, the proposed transformations are very useful, and the key for exploiting their full potential is to find the adequate combination of the execution of tests with and without transformations.

5.4.2. Using the Data from Homogeneity Tests

Thanks to the analysis done in Section 4, it was possible to identify another way to exploit the results produced by the homogeneity tests. It was evident that the time series produced with their statistics were plotting some characteristics figures in the locations where outliers and errors were present, in the form of spikes with close angles.

The solution proposed is based on finding the turning points in the time series of the statistics and to calculate the angle formed in such turning points, assuming small angles would represent the outliers and errors to be reported.

It has been possible to implement a model for doing the identification of these points, but determining the thresholds to be applied to the angles was very complicated. They are different for each of the statistical tests considered and, additionally, although in most of the situations the range of valid angles is rather clear, there are also cases that do not fit very well with those thresholds.

In some cases, this is produced because the selection of points for building the slopes used for the angle calculation are only the two points around the turning point. This causes

sometimes to miss other nearby points that reduce the angle of the peak from a higher-level perspective. An interesting solution to explore would be to use two or three points per slope, building a small regression model with them, or calculating an average position to be used as reference.

On the other hand, the angles have been studied in a manual way and this makes complicated the adequate selection of thresholds. This seems to be a good example of a context in which ML would help to find an adequate classification model that, using the information available about the number of turning points and all the angles (together with annotated outliers), could provide a service that could determine whether a sample contains some problem or not.

5.4.3. The Models for Data Understanding and Trust

This section proposed following three processes to gain insights into sensor data before applying cleansing mechanisms or using the data. These processes addressed data variation (including data distribution), outliers in the data and correlation between similar sensors. Such processes can be seen as a way to specify and implement certain steps that are part of the CRISP-DM methodology. They fit very well with the data understanding phase, and the implementation done can automate part of it, reducing the time spent by data scientists working with sensor data, thus enabling them to focus on their models.

Moreover, the results that are generated with the processes are used with a set of rules (of a fuzzy model) to obtain a trust evaluation, that can aggregate all the calculated values in a meaningful way. This provides a result that facilitates the interpretation of the evaluation done, in such a way that a user does not need to have a deep knowledge of the calculations done and can just focus on deciding if some action should be taken. Other aggregation mechanisms could be studied, such as a solution based on ML models. ML methods focused on classification could be an interesting solution although more annotated data and evaluations are needed to obtain good results (so the models can be trained and evaluated adequately).

The best way to implement the processes and the trust aggregation was through R scripts because of the strong support for the statistical methods available in this language (as multiple libraries are available). It would be interesting to compare implementations with other languages such as Python (also with good support for data processing), to analyse their performance and potential integration with other existing systems.

Additionally, a parallel implementation of the processes was proposed to improve the performance of the solution. As more and more devices become available, more data streams will be available, and the computational resource requirements will be higher. Exploiting parallel processing is a step towards higher efficiency in the use of resources, already available in edge devices, as well as Cloud and high-performance computing (HPC) environments. Although other works focused only on the solution itself, we considered this to be an important point to address for enabling new capabilities.

As a way to facilitate data processing in Cloud environments, for example, it would be possible to exploit multiple cores available in edge devices in order to perform a real-time

preliminary analysis of data, annotating the data that arrives to the processing environment and facilitating complex data analysis. It could even raise alerts under certain circumstances (e.g., when detecting a large number of outliers).

The parallelization strategy was based on the iterations performed when a sample was selected (one core assigned to each sample). Each core calculated all the statistics for sliding windows belonging to the same sample (not parallelized), taking advantage of memory caching mechanisms. Otherwise, sending sliding windows to different processors would result in more calls to main memory, as pages in cache would fail.

The parallel implementation showed its utility, especially when using up to three cores, and it is a topic to explore for more solutions. The code was not scaling as well as it could, but this might have been due to the machine and operating system we used. We believe that the performance and scalability could improve with a Linux-based system, and it would be deployable in multiple environments.

6. EVALUATION

The computing scientist's main challenge is not to get confused by the complexities of his own making.

Edsger W. Dijkstra

After the analysis of the applicable approaches and the definition of the solutions for analysing sensors' data and detect anomalies, it is necessary to evaluate whether such proposed solutions really work as expected in different environments from those used to define the base models.

Section 5 already analysed how the proposed algorithms and solutions perform when applying parallelization in order to exploit the computational resources (linked with RQ7, evaluating the performance of the implementation). Therefore, this section is only focused on the performance of models, but from the perspective of the quality of the outcomes generated.

This section describes the approach followed to evaluate how the main proposed solutions perform, clarifying the main criteria and metrics to be used and explaining the main reasons to do so. Thanks to this evaluation it is possible to determine how the solutions perform in different circumstances (as expected from RQ5 perspective).

As the solutions under evaluation include the data understanding processes and the trust model, it is also possible to determine whether they are a valid solution, including new domains not analysed in the previous sections (in line with RQ6), as a way to proof that these approaches may be generic enough to be applied for new types of sensors (although this does not mean that the models will perform fine with any kind of sensor).

6.1. Evaluation Methodology

The proposed methodology is in line with the approach followed when analysing data analytics and machine learning models (Hamarashid et al., 2022), which implies the usage of certain evaluation metrics once a model has been trained. In this concrete case, there is not a training, but the algorithms and thresholds of some solutions have been defined based on the experience, so these metrics are interesting to check the correctness of the models, as well as the generalization that might be achieved with unseen data.

The evaluation is applied to three main outcomes from this thesis (defined in Section 5), taking into account their importance and their utility:

- The approach for automated outliers detection based on the homogeneity statistics;
- The processes proposed for data understanding and trust evaluation;

- The trust model that aggregates automatically the processes outcomes.

First of all, it is important to guarantee that the data used for the evaluation is not the same as the one used for the models definition. In the case of machine learning models, it is typical to divide the original dataset in two parts, using one for the training and another one for the evaluation. Cross-validation is usually applied as well, since it allows re-sampling the data used for training and evaluation, so a better perspective is obtained.

In the case of this thesis, since it was possible to collect data from different sources, instead of splitting the original datasets, the decision made was to use most of the datasets for analysing data and defining models, and reserve a few of them for doing the evaluation. This approach does not require a technique like cross-validation because the datasets are totally independent and because with such approach it is possible to cover two objectives of the evaluation:

- The correctness evaluation by using datasets that measure a metric already analysed (temperature), but using different sensors from another system not used before, and located in a different place;
- The generalization evaluation by using a new type of sensor not used during the model definition phase, so it will be totally new for the models.

Using a technique like cross-validation would not facilitate the achievement of the second objective, since it would imply to use always the same types of sensors both for training and for evaluation.

The models were evaluated against several metrics, so it is possible to analyse their outcomes from different perspectives, with especial emphasis in evaluating whether it is possible to avoid false negatives, since this is a point that might reduce the utility of the proposed models because of their nature (reporting false positives in trust models will make users to ignore alerts if this happens too often).

6.1.1. Datasets Used for the Evaluation

Taking into account that it is necessary to used annotated datasets for the evaluation of the solutions, it was important to reserve two types of datasets that would facilitate this task, even if they are not especially large.

Since the models should be able to detect anomalies in short periods of time (it is important to detect any problem as fast as possible, not in days, so a countermeasure may be applied), the usage of datasets with data from a few days is more than enough.

In the case of the dataset that represents an already used metric with known failures, the choice was to use some files from the benchmark datasets (de Bruijn et al., 2016) generated with the SensorScope system. These files contain temperature measurements, with different errors injected (bias, drift, malfunction and random). This required to select one concrete mote (mote 15 in this case) and to use one file per each error injected. The datasets include information about the presence of the error, facilitating the construction of confusion matrices and the calculation of the evaluation metrics.

When using a new type of sensor, the choice was to use the air quality datasets from the

city of Gyor. Although the models were built taking into account sensors like temperature, wind speed, atmospheric pressure and other continuous metrics with similar variation, air quality metrics were not considered, so they are a very good choice to check how the models behave with new types of sensors.

Those datasets represent a few days of data, taking several metrics (temperature average, air pressure, NO_2, O_3, PM10, PM2P5). The NO_2 metric has been selected, although in the case of these datasets there is no annotation about the errors. Therefore, it required some manual processing to annotate those clear errors in the data, easing the evaluation and its automation.

6.1.2. Metrics Used for the Evaluation

There are several metrics that can be used for evaluating models. Works like (Hamarashid et al., 2022) and (Hossin and M.N, 2015) propose some common metrics, giving a clear definition. In the case of this thesis, the metrics used are accuracy, precision, recall and F-score, for several reasons. First of all, they provide relevant evaluation information for the concrete models and, secondly, it was possible to compute them easily by using a R library (named MLmetrics) that was integrated in the code.

First of all, the outcomes of the models are categorized in a binary way: they predict that there is an anomaly or not. Then, they are organized in a confusion matrix that includes:

- *True positives:* the model reports that the data shows anomalies and the anomalies are really there;
- *True negatives:* the model reports that the data is fine and, the data is correct (there are no anomalies);
- *False positives:* the model reports that there is some anomaly in the data, but the data is correct;
- *False negatives:* the model reports that there are no anomalies, but the data contains errors.

Such outcomes are used as inputs for the different metrics that are used during the evaluation.

The first metric to measure is **accuracy**, that provides information about the capacity of the models to do right predictions, taking into account all the predictions done. In this case, we may expect that, most of the time, the data will be right and no error will be present, so there is no a balance between the amount of data that show anomalies and the amount of data that is right. Still, it is interesting to have a high-level view of the results produced by the models. It is calculated as shown in the following equation.

$$accuracy = \frac{correct\ predictions}{all\ predictions} \qquad (6\text{–}1)$$

Precision and recall are other two interesting metrics, since they are focused on understanding how the model predicts with respect to certain types of errors. Moreover, the fact that the cases in the evaluation data show class imbalance has no negative impact

in the evaluation result.

In the case of **precision**, it determines the proportion of positive predictions compared to the right cases and false positives, so it is possible to see to what extent false positives impacted the result. It is calculated using the formula below.

$$precision = \frac{true\ positives}{true\ positives + false\ positives} \quad (6–2)$$

It is important to highlight that this metric is especially interesting in the case of these models under evaluation, since it takes into account false positives, which are the main type of error to avoid in solutions that alert about anomalies. The impact of missing some false negative could be assumed (if something is wrong in a sensor, the error will show up again and it will be, in the end, detected), but too many false positives may have a negative impact in how much the users trust the results of the models.

The **recall** metric, instead, is focused on understanding the weight of positive cases that were tagged as negative. This means positive cases that were missed by the model when predicting. The following equation shows how to calculate this metric.

$$recall = \frac{true\ positives}{true\ positives + false\ negatives} \quad (6–3)$$

As mentioned before, the fact that the models miss some anomalies is not considered so relevant, since it is assumed that they will appear sooner or later again, and the models will have more opportunities to detect them.

Using the previous metrics, it is possible to calculate the **F-score**, which provides a combined view of precision and recall, giving a numerical result that can tell us how good models are from a high-level perspective. It can be calculated using the following equation:

$$F_\beta = (1 + \beta^2)\ \frac{precision * recall}{(\beta^2 * precision) + recall} \quad (6–4)$$

The parameter β can be used to determine the weight to be given to the precision metric. A value of $\beta > 1$ gives more importance to the recall, while a value of $\beta < 1$ gives more importance to the precision. As in the case of this thesis the precision is considered more important than the recall, the proposed value for β is 0.5, which is the typical value used in this context.

Another complementary metric is the **Area Under the Receiver Operating Characteristic Curve (ROC AUC)**. It measures the separability between the classes, meaning that it evaluates whether the model is able to perfectly separate the two classes used (existence of anomaly or non-existence of anomaly). If there are no false positives and false negatives, AUC would be equal to 1. As false positives and negatives appear, there is more overlapping in the classification, so AUC will be lower. If AUC = 0.5, it means that a model has not the capability to discriminate and identify the classes.

This metric requires to calculate the Receiver Operating Characteristic (ROC) curve, by using the recall and the False Positive Rate (FPR), which is calculated using the following formula:

$$FPR = 1 - Specificity = \frac{false\ positives}{true\ negatives + false\ positives} \quad (6–5)$$

The AUC is the area that is under the ROC curve calculated and as mentioned, it is expected to be between 0.5 and 1 (values below 0.5 may mean that classes are not defined in the same way and are flipped).

Since the models under evaluation are focused on a classification problem (they determine whether there are anomalies or not), other metrics that are for evaluation in models providing continuous outcomes, like RMSE (Root Mean Square Error) and RAE (Relative Absolute Error), have not been used.

6.2. Evaluation Scenarios and Results

As mentioned before, two scenarios have been defined for doing the evaluation of some of the solutions proposed in Section 5. The first scenario uses the benchmark datasets to evaluate the solutions against known types of errors. The second one, uses a new dataset to evaluate whether the solutions can be generalized to a new type of sensor.

The scenarios are executed to evaluate two concrete solutions: the detection of outliers through the angles of the homogeneity tests statistics and the analysis of datasets through the three processes defined in Section 5 for data understanding and trust.

6.2.1. Temperature in Benchmark Datasets

This scenario consists in using the datasets with injected errors that were created for benchmarking. While the datasets provided with the Intel and Santander platforms have been considered when designing the solutions, the datasets provided using the SensorScope platform were reserved for the evaluation of results.

Therefore, this scenario uses four datasets from the SensorScope platform, each one representing one type of error: bias, drift, malfunction and random. In each case, 20 samples are selected randomly from each dataset, extracting the annotated failures as well. This is done twice: one with small samples and one with larger samples.

Once the data is processed, the failures retrieved from the dataset are compared to those predicted by the proposed solutions, calculating the evaluation metrics already selected.

6.2.1.1. Outliers Detection with the Homogeneity Statistics

In this scenario, although it is possible to calculate the evaluation metrics separately for each homogeneity test, it was decided to only look at the final result, according to the definition of the solution. Therefore, the aggregated result is the one that has been used for evaluating against the error annotations already provided by the datasets.

When using the dataset with bias errors, the proposed solution was able to detect four errors correctly, while another two were false positives and there were no false negatives. The overall accuracy of the model was 0.9, while precision was 0.66 and recall was 1. The weighted F-score was 0.714 and the AUC calculated was 0.937. In general, the result was

good, although a higher precision was expected.

With the drift error, the results were a bit worse, as the proposed solution was too sensitive and produced five false positives, although it classified correctly the rest of cases (with six errors correctly detected). The overall accuracy calculated was 0.75, with a precision of 0.54 and a recall of 1. The weighted F-score calculated was 0.6 and the AUC was 0.82. The high number of false positives showed that perhaps the thresholds for the angles should be lower than in the current configuration.

Table 7 Summary of evaluation metrics with injected errors for outliers detection

Type of Error	Accuracy	Precision	Recall	F0.5	AUC
Bias	0.9	0.66	1	0.714	0.937
Drift	0.75	0.54	1	0.6	0.82
Malfunction	0.85	0.72	1	0.769	0.875
Random	0.85	0.9	0.818	0.88	0.853

The case of malfunction was better than the previous ones. The proposed solution was able to detect eight errors correctly, producing only three false positives. The accuracy obtained was 0.85, while precision was 0.72 and recall was 1. In this case, the weighted F-score was 0.769 and the AUC was 0.875. These are good results, especially considering that malfunction is a type of error difficult to detect and that homogeneity tests perform better for errors like bias and drift, because of the clear changes in the samples.

Finally, when evaluating against the dataset with random errors, the outcome was good as well. The proposed approach managed to detect nine errors correctly, while only one false positive was reported and two false negatives. Thanks to this, the accuracy was calculated as 0.85, while precision was 0.9 and recall 0.818. The result of the weighted F-score was 0.88 and the AUC was 0.853.

Taking into account that sliding windows were not used (only independent samples were selected), the results were quite good, especially for malfunction and random errors, which are usually hard to identify. Additionally, according to the AUC, it seems the proposed model has a good capability to classify, not having a large overlapping with false positives and false negatives.

Therefore, the proposed approach seems to be a good complement for the statistical tests, although the precision obtained for bias and drift suggests a careful usage. Another selection of thresholds could, perhaps, reduce false positives, improving the precision, although that may cause the production of more false negatives.

6.2.1.2. Errors Detection with the Data Understanding and Trust Models

This scenario makes use of a R script that integrates parts of the three processes defined in Section 5.3, together with the other solutions described in Sections 5.1 and Section 5.2. Although the outcomes of such script include information about each statistical test and solution, this evaluation is done against the global result, since the combination of the

results is one of the key points to be able to detect problems in sensors.

In the case of this script, it selects two groups of samples: one with long window size (6% of the dataset) and one with short window size (1% of the dataset). 10 long samples and 15 short samples are selected, and in all cases sliding windows are used (30 steps for long samples and 15 steps for short ones).

When using the dataset with bias error, in the case of short samples, the proposed solution managed to detect correctly five errors, with only two false positives and one false negative. Therefore, the accuracy reached is 0.8, while the precision is 0.71 and the recall 0.83. The F0.5-score calculated is 0.735 and the AUC is 0.80. The result seems to be good and, in fact, one of the false positives was in the limit of the consensus, so it would be feasible to improve with more fine tuning of the aggregation.

Table 8 Summary of evaluation metrics with injected errors for outliers process (short samples)

Type of Error	Accuracy	Precision	Recall	F0.5	AUC
Bias	0.8	0.714	0.833	0.735	0.80
Drift	0.8	0.66	0.5	0.625	0.70
Malfunction	0.8	0.75	0.6	0.71	0.75
Random	1	1	1	1	1

With the drift error, results are not bad at all. It detected two errors correctly, reporting only one false positive and two false negatives. The accuracy reaches 0.8, while the precision is 0.66 and the recall is 0.5. The F0.5-score is 0.625 and the AUC is 0.7. It manages to avoid false positives, but it fails to find some of the errors. It is also true that the false positive detected had some changes, but they were not tagged as error.

When evaluating against malfunction error, short samples are managed quite well. Three errors were correctly found, while only one false positive was reported, as well as two false negatives. Therefore, the accuracy was 0.8, the precision was 0.75 and the recall was 0.6. In general, the F0.5-score was 0.71 and the AUC was 0.75. In those cases in which the Grubbs test (using the original data) was not able to detect the error, the consensus of angles and the Grubbs test with the transformed data were the ones reporting the problem.

The best result is obtained when evaluating the detection of random errors. The classification done by the model was perfect, detecting all the errors and not producing false negatives. Therefore, the accuracy, precision, recall F0.5-score and AUC were 1. The Grubbs test, with the original data and with the transformed data, discriminated very well most of the cases.

In the case of long samples, it seems the results are not as good as for short samples. The following table summarizes the outcomes collected.

In the case of the bias error, the results were not so good in general. The proposed solution tends to report errors with long samples, generating too many false positives. It managed to detect four errors correctly, but it also reported four false positives. Therefore, the

accuracy was 0.6 and the recall 1, but the precision dropped to 0.5. The F0.5-score reached only 0.55 and the AUC was 0.66. When looking at the plots of the false positives, it is not so unexpected to find problems, since the samples represent more than three days of data and the temperature was showing some variations those days with a clear change of the average, that is detected by the homogeneity tests. Shorter samples would not have the same issue.

Table 9 Summary of evaluation metrics with injected errors for outliers process (long samples)

Type of Error	Accuracy	Precision	Recall	F0.5	AUC
Bias	0.6	0.5	1	0.55	0.66
Drift	1	1	1	1	1
Malfunction	0.7	1	0.5	0.83	0.75
Random	1	1	1	1	1

In the case of drift, the results improved a lot, with a perfect classification of the samples. All the errors were correctly classified. Accuracy, precision, recall F0.5-score and AUC reached 1. Still, it might be too optimistic, since the models seem to have a small bias towards reporting errors, and in this case, the great majority of the samples contained errors.

The outcomes of the malfunction case are rather good in general. Three errors were detected correctly, and no false positives were reported, although three false negatives were discovered. The accuracy reached 0.7, while precision was 1 and recall 0.5. Consequently, F0.5-score was 0.83 and AUC was 0.75. In this case, as it happened with short samples, the Grubbs with the transformed data and the angles analysis were the ones leading the detection of the errors.

Finally, the result with the random error dataset was very good. As in the case of the short samples, the detection of random errors with long samples worked perfectly, detecting correctly all the errors and not producing false positives. Again, in such case, accuracy, precision, recall, F0.5-score and AUC were 1.

In general, the evaluation shows that the proposed approach works well in general, although the detection of bias and drift errors seem to be a bit more problematic than the detection of malfunction and random errors, with better performance when using short samples.

6.2.2. Air Quality from Gyor Datasets

This scenario uses the sensor data from the city of Gyor (Hungary) to extract the NO_2 measurements and evaluate how the models perform with a new kind of sensor. This dataset did not contain annotations about the errors, so it has been annotated manually through a visual inspection of the data. Such inspection shows some clear outliers, with too large values, or very small values minor than 0 (when this metric should have positive

values). In fact, the dataset looks like there are several errors, especially random (with some individual outliers) and malfunction (with set of values becoming too sparse and including clear outliers as well). No errors were introduced in this dataset artificially.

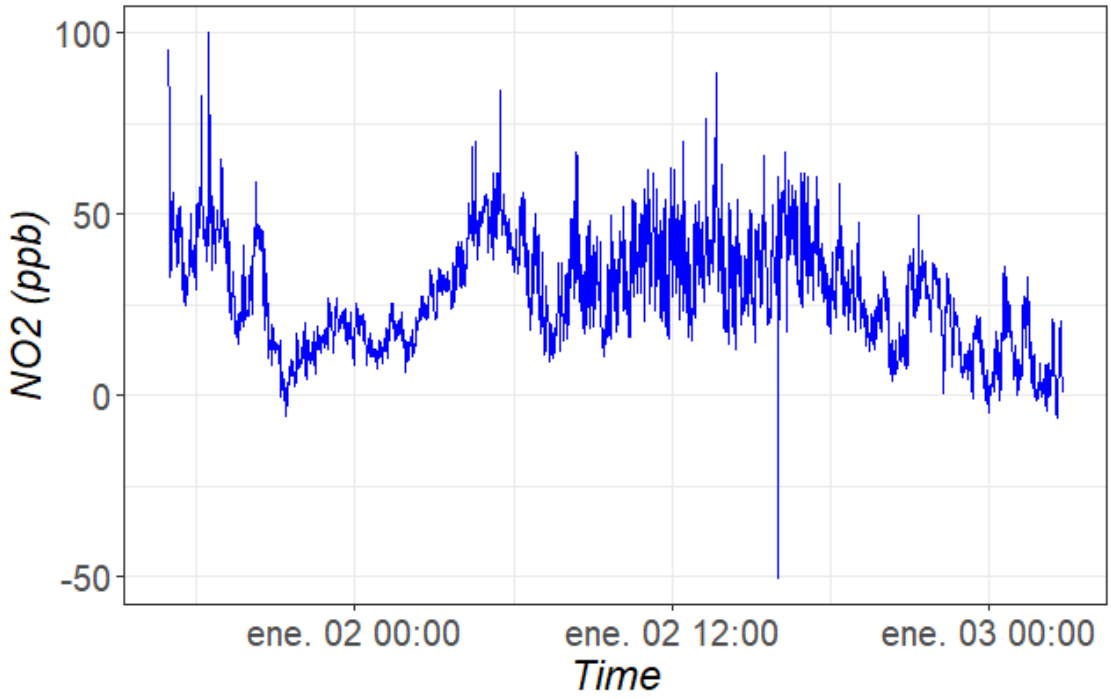

Figure 56 NO_2 measurements from the city of Gyor

As this dataset contains data only for one week, the window size selected is rather small, corresponding to two hours of data (120 observations per sample). As with the injected faults evaluation, 20 samples are selected randomly for the analysis. That means that, in the end, 2400 observations are used (out of the 9951 available) for the evaluation, which represent more than 25% of the dataset. The device selected in the dataset is the one with identifier '359072065634004'.

Thanks to the annotations included, it is possible to use such information to compare it easily with the outcomes generated by the evaluated models.

6.2.2.1. Outliers Detection with the Homogeneity Statistics

As in the previous scenario, although it is possible to provide an evaluation per homogeneity test, the evaluation is done against the final result produced, since the way to aggregate the results is important as well.

The evaluation performed showed that the model was able to detect most of the errors present in the selected samples. It managed to detect correctly 8 out of 15 errors. It only produced one false positive, although it produced 7 false negatives.

Therefore, the overall accuracy of the model was 0.6, while the precision was 0.888 and the recall calculated was 0.533. On the other hand, the F0.5-score reported a result of 0.784 and the AUC was 0.666.

Although the accuracy is not so good, the precision is high, which was already defined as the priority when the metrics were presented. In the same line, F0.5-score is high as well, meaning that the proposed solution is able to detect errors and it does not report many

false positives. On the other hand, the recall demonstrates that the model fails to detect some of the errors available.

The dataset used is a quite complicated one, mainly because it has multiple parts with sparse points and there are also many individual outliers (random error), that are not very well detected with homogeneity tests. A less conservative set of thresholds could improve the number of errors detected, but it is difficult to get the optimal ones, as it may happen that the number of false positives is increased when using data from other sensors.

The Grubbs test is more appropriate to find such type of errors, so an adequate combination of both solutions (together with other tests) may for sure improve the performance in the detection of the errors in the data. This is particularly positive, since it seems this kind of solution is better detecting malfunction errors (one of the weak points of Grubbs), so there is a good complementarity.

6.2.2.2. Errors Detection with the Data Understanding and Trust Models

As in the previous scenario, this evaluation makes use of the R script that integrates parts of the three processes defined in Section 5.3, together with the other solutions described in Sections 5.1 and Section 5.2. Again, the evaluation is done against the final result (that aggregates the different statistical tests and solutions) and there is no individual analysis of each test.

When using short samples, the proposed solution detects correctly seven errors, although it produces five false negatives (errors not detected). Because of this, the accuracy is 0.66 and the recall is 0.583, but the precision is 1 and the F0.5-score is 0.875, which are very good results, since errors are detected and, although some of them are missed, the model does not produce false alarms. Additionally, it is worth mentioning that those errors not detected were very close to be detected so it is expected that this approach would be able to detect them with a few more values. AUC takes a value of 0.79, so the model can classify quite well.

In the case of long samples, as the original dataset contains a lot of errors, all the samples selected contained errors. The proposed solution was able to detect all the errors, although some of the errors were identified as drift errors (although all the errors of the dataset are random or malfunction errors). Therefore, the accuracy, precision, recall and F0.5-score are 1, while AUC could not be calculated (because of the lack of samples without errors).

6.3. Discussion

This section has presented the evaluation that has been carried out to check how the proposed solutions work. The proposed methodology was clarified, with some clear objectives. First of all, check how the models perform using datasets that had no link with the design and implementation of the models themselves. That was achieved by using an annotated dataset of a known sensor (temperature), but not used before, and a dataset produced with the data generated with types of sensors that were not studied before (from a system for monitoring air quality). In such case, the NO_2 observations were selected. This is an important aspect, as it is in line with the objective of proposing solutions that are as much generic as possible, and the usage of a new type of sensor demonstrates the flexibility

of the mechanisms implemented.

The metrics proposed for the evaluation were focused on understanding how well the model can classify between data with errors and right data. That is the reason because metrics like precision, recall, accuracy, and F-score were important. In fact, the weight of precision was considered more important, as the presence of false positives may have a negative effect from the user side.

The evaluation of the solution proposed for calculating angles based on the homogeneity tests statistics performed fine in general. When using the annotated temperature datasets, the accuracy, recall and AUC were very good in general for all types of errors. In the case of precision, it shows that the proposed solution deals very well with malfunction and random errors, but it is a bit more problematic for it to detect bias and drift errors (although precision values are not bad at all). That made the F0.5-score to be a bit lower than the others as well. Therefore, it is clear that this approach should be complementary to other aspects when detecting bias and drift errors.

The application of the air quality dataset to this approach was interesting, in the sense that the precision was very good, although the recall was not obtaining as good results as with temperature. It is a kind of event that is a bit different from temperature, so different results were expected, but in general the result was still good, and the presence of false negatives is not as much harmful as false positives (especially considering that it would be normal that a faulty sensor produces more faulty measurements that may be detected a bit later).

Again, complementing this model with others is the best way to have a system that can minimize wrong classifications, although it might also be improved with more fine tuning in the thresholds defined for the angles.

With respect to the evaluation of the processes implemented, an integrated script was used to check the capabilities to detect problems in a dataset by combining several statistical tests, the new transformation proposed and the angles detection approach.

The result obtained when applying this solution to the temperature datasets (testing against bias, drift, malfunction and random errors) was good in general, especially in the case of random errors.

With short samples, all the types of errors showed a very good accuracy. Precision and recall are good in general (and F0.5-score and AUC as well), although they are not very high for the drift error. It seems that the combination of statistics, angles calculation and specific transformations work well for these errors. The values are a bit lower with long samples. Except in the case of random errors, that were classified perfectly for both size of samples, the fact that the long samples represent several days of data made the models to report some false positives. This is because the temperature varied quite a lot in some days, with the difference of temperature between day and night producing changes in the average value of the time series.

When using the air quality dataset for evaluating this part, the long samples showed that the model was quite robust detecting all of them. On the other hand, with short samples, it missed some errors (so the accuracy is a bit lower, although the precision is very high), although it almost detected them, so it would be expected that such errors could be

detected as new values are added.

Therefore, although the proposed solutions perform quite well, it is better to use short samples, as it is easier to avoid additional observations that may alter the result of the statistical tests. In fact, in real-time environments, it would make a lot of sense not to use samples of three days of data and use a few hours of data instead (focusing the analysis on the latest measurements done).

7. CONCLUSIONS AND FUTURE WORK

Data is not information, information is not knowledge, knowledge is not understanding, understanding is not wisdom.

Clifford Stoll

Once this dissertation has presented and described in detail the problem addressed and the solutions proposed, this section aims at looking from a high level perspective at all the conclusions, lessons learned and limitations found, and proposes some ideas with respect to the future work related to the domain addressed.

This chapter looks at the initial research questions and the proposed hypothesis to determine whether all the expected topics were addressed and whether it was possible to confirm the hypothesis according to the work done and the research findings obtained, as discussed in the previous sections.

The dissertation has detailed the way to carry out the analysis of data produced by different types of sensors and, since several new approaches have been proposed, this section describes the main contributions done through this work as well as more conclusions extracted from the discussion about the solutions proposed.

Finally, the future work is focused on those areas addressed specifically by the dissertation, although it also proposes other ideas related to the work done, since some of the results could inspire more solutions in the area of the Internet of Things.

7.1. Summary of Work and Conclusions

This dissertation started presenting one of the problems that has arisen as the devices available on the Internet of Things were growing. Trusting in the sensors and the data they produce is important in this context. Hence, this dissertation has tackled how to process the sensors' data in early phases of the data usage for determining whether there are anomalies in the datasets and, therefore, the sensors used are trustworthy. As one of the objectives was to be as much generic as possible (especially in open environments like smart cities), the work was focused on providing solutions based on statistical calculations from the beginning.

Once a good set of datasets, with data coming from different types of sensors and systems, was available, this work has provided a deep analysis of the data collected. Part of such analysis was to understand how the basic statistics of the data (like mean, variance, mode, etc.) varied from sensor to sensor, also taking into account the size of the sample selected (in line with RQ1 and RQ2). This part of the analysis showed that sensors may have very

heterogeneous behaviours and that metrics like variance and mean are affected by the size of the window, by the units used (using ºC and ºF results in different outcomes from the statistics) and by the presence of certain values that might seem outliers but are not necessarily abnormal. On the other hand, other metrics like Coefficient of Variation and IQR showed that they are more reliable when analysing the variation of the data (answering RQ1).

Other aspects, like the distribution followed by the data and the randomness, were analysed to know a bit more about the behaviour of sensors. In most of cases, the probability distribution of the data is not normal, and, in fact, the distribution is unstable and may change depending on the sample selected (clarifying more the answer to RQ1). With respect to the randomness of the data, there are some types of errors (especially in the case of malfunction and random errors) that produce 'noise' in the time series and some statistical tests (Ljung-Box test and Runs test) showed that it is possible to detect such noise, although it has some limitations (they cannot detect the error if it is just a small part of the dataset). This part of the research answered RQ2 and contributed a bit more to RQ1.

The analysis of the data continued focusing on the detection of outliers in the data samples (in line with RQ2 and RQ3), as errors in sensors imply the existence of these abnormal values. Because of how the errors are reflected in the data, the idea was to use two types of statistical tests: tests to detect outliers (as abnormal values that show up from time to time) and tests to detect important changes in the data (to determine if the data is homogeneous).

In the case of outliers, the Grubbs test (and its generalization called ESD) showed good capacity to detect errors in the data, although with some limitations, like the number of anomalous values present (when there are too many, it starts to fail). Therefore, this seems to be a good approach to detect errors like malfunction and random, and even bias and drift, but only when they are starting to appear in the sample. On the other hand, the Dixon test was discarded because of its multiple problems: it can be used only with very small samples and if there are two or more outliers with the same value, it fails. These findings answered partially RQ2 and RQ3.

As for the homogeneity, five typical tests were studied to complete the answer to RQ2 and RQ3: SNHT, Pettitt, Buishand R, Buishand U and Lanzante. These tests showed good results when detecting bias and drift errors, especially when the error is represented by more than 10 values. Still, they have issues with malfunction and random errors sometimes, and they use to fail when only a few individual outliers are present in the data. Additionally, these tests produce false positives when there is trend in the data, as they interpret there is a relevant difference in the mean and the standard deviation. Therefore, it is necessary to apply transformations to the data, that can improve the results.

Another aspect analysed was correlation between sensors, in line with RQ4 and RQ5. When equivalent sensors are available to compare, it is possible to detect issues if the measurements of equivalent sensors do not match (as also addressed by one of the solutions revised in the state of the art). Three tests for correlation were used to analyse how errors in the data would affect them: Pearson, Kendall and Spearman. The loss of correlation was made clear when looking at the results, especially in the cases of bias, drift and random errors. This work concluded that, if the correlation between sensors was previously confirmed, these tests can be used to identify that there is an unexpected

behaviour. But if this is not the case, the identification of correlation may be misleading. These findings answered RQ4 and RQ5 as expected.

Taking into account all the knowledge generated with the mentioned analyses, the work continued towards the proposition of a set of workflows to facilitate data understanding and to identify anomalies that would define how much the users can trust sensors, in line with RQ6. First of all, the dissertation proposed another type of transformation to be done in the data, as a way to improve the usage of homogeneity tests. After looking at several options, this work proposed two new types of transformations: one modifying the difference transformation with an exponential factor and another one using polynomial regression. These transformations, especially the first one, showed that they can remove the problems when analysing samples with trend, while also improving the outcomes of the outliers and homogeneity tests (contributing to answer RQ2, RQ3 and RQ6). Still, they are not perfect and an adequate combination of tests with untouched and transformed data is the best option to detect problems.

As the analysis of homogeneity tests had shown some particularities in the statistics generated when there were errors in the data, this work also studied whether the analysis of those particularities could improve the identification of outliers and errors in the data, instead of relying only on the p-value from final results of the statistical tests. The proposed approach extracts the data produced by the statistical test, finds the turning points in the data (representing local maximum and minimum points), determines the lines that represent the intersection from which the turning point is produced and calculates the angle of the intersection. After testing the proposed approach using several datasets (representing the different situations we may find with different sensors), some thresholds were defined for each homogeneity test, so when the angles are smaller than the threshold, the dataset is considered to have an error. Although this solution seems to be useful and it is interesting to combine it with the previous ones, it is not always effective, mainly because the thresholds defined are rather static, and may not be the optimal ones for all types of sensors and errors. Anyway, this approach was able to detect some issues that were not detected by the homogeneity tests themselves (especially, when the number of values representing the error is low), and it contributed to answer RQ2, RQ3 and RQ6.

Using all these analyses and approaches as input, the next step was to define a set of processes/algorithms to carry out the analysis of sensors' data during data understanding phases, in such a way that it would be possible to identify faulty sensors in early stages, in line with RQ6. Three processes were defined: one for analysing the variation of the data, one for analysing the existence of outliers and other errors, and another one for analysing correlation, whenever possible.

In each one, these data cleansing guideline workflows indicate how to select samples, transform them (whenever necessary) and run different tests, depending on some characteristics of the data. Such definitions also indicate how the information generated by the tests can be interpreted, even combining the outcomes from different tests and processes. These processes are not complex to follow and have been implemented using R (and several libraries).

As observed when analysing the data, not all the errors can be detected correctly until they reach a certain number of values affected. Therefore, the proposed processes use sliding

windows, selecting an initial sample and then moving the origin of the sample to select more chunks of data to analyse. Doing so, it is possible to improve the detection of the anomalies, because the error is moving in the sample analysed. In fact, it is similar to what happens when collecting the data in real time through streams, as new measurements arrive continuously, so old measurements are moved out of the sample observed (something that is the way in which Complex Event Processing engines work).

Since it seems changes in the variation of data and the presence of errors do not detect anomalies in the same moment (it may take more time to the variation process to identify them), the sliding windows are also useful and allow for a combination of results at different times.

In order to facilitate the evaluation of the level of trust based on the tests used, a fuzzy model was proposed, acting as an aggregation mechanism. As a way to complete the answer to RQ6, this model represents a set of rules that use as inputs the outcomes of the statistical tests and produces an output value indicating high, medium or low trust on the sensor that produced the data analysed.

After all this work answered RQ6 and, in order to address RQ7, a parallel implementation of the processes was carried out, showing that exploiting multiple cores improves performance and may reduce a lot the execution time, although not in all cases the speedup reached is the optimal one (mainly, because of the limitations of the hardware system where the proposed solutions were executed).

Finally, the dissertation reports about the performance of the proposed solutions by doing an evaluation with datasets not used before. In one of the cases, with a type of sensor already used (temperature), and in the second case, using a dataset with a new metric not analysed before. Such evaluation demonstrates that the solutions proposed are useful to identify problems in the sensors with enough precision, answering successfully RQ6.

As a conclusion, it is fair to consider that the *hypothesis was validated*, since this research work was able to demonstrate that a set of statistical-based solutions can be used to analyse the variation of the data produced by sensors and to identify anomalous values (outliers), and those aspects are indicative of problems in sensors that should not be trusted. The evaluation of the proposed solutions demonstrated that it is possible to reach high levels of accuracy in such classification and that this information can be used to guide the task of data understanding in data analytics pipelines.

7.2. Contributions

This work has contributed to further progress on the Internet of Things area, especially in those areas related to the data analysis and the identification of potential issues in sensors. Therefore, this subsection describes each of the contributions done, organized by chapters.

Chapter 2 presents the state of the art related to the area under study with the purpose of understanding the current approaches:

- It studied the different kinds of errors that have been analysed in the data generated by sensors, as well as kinds of attacks that the IoT systems can suffer, although its main focus is on those solutions that look for anomalies in the data. In that sense, the

work analysed proposed solutions and explained why they are not addressing the objectives proposed in this dissertation.

Chapter 3 describes the main data sources used and their particularities, with a brief description of the systems that generated the data:

- It showed the heterogeneity of the IoT systems and how they manage data in different ways, as well as other aspects to take into account. From the format to the units used and the location of the sensors, this chapter has shown how the context of the sensors is important and require not only to properly pre-process the information, but also to bear in mind the environment of the sensors for the analysis.

Chapter 4 carries out a deep observation of sensors' data by analysing the kind of values produced, their distribution and how certain statistical tests perform for detecting issues in the data (in line with the analysis of variation of data and presence of outliers):

- It performed an analysis of basic statistics (average, variance, mode, etc.) using data collected from multiple sources and sensor types (up to 12 types of sensors). It showed that there is a huge heterogeneity in the behaviour of sensors, not only related to their type. It demonstrated that, measuring the same aspect (e.g., atmospheric pressure) with different units affects the basic statistics, so they cannot be considered reliable. Still, other statistics like Coefficient of Variance and IQR showed that they may have more utility to analyse how data varies.
- It carried out an analysis of the probability distribution of the data, demonstrating that, in most of cases, the data does not follow a normal distribution, especially when using samples of a small size, as the ones to be used for real-time data analysis. It showed that distributions like lognormal, Weibull and gamma are also usual in the data, so perhaps the formulas used for calculating average values and variance should be adapted to such reality. Moreover, the lack of normality in the data may limit the statistical tests to use and their performance, unless there is some specific transformation of the data.
- It analysed the results from statistical tests focused on the detection of random data and white noise, showing that these can be used to detect some anomalies in the data, but only with small samples, as they do not perform well with big ones, unless the random part is evident.
- It analysed the results of using specific statistical tests focused on the detection of outliers in the data, to check if they can detect errors. While one of the tests showed to work quite well with most of the error types (especially malfunction and random) in many cases (Grubbs) another one showed that it had to be discarded (Dixon), as it was problematic for the datasets used (because of its formulation).
- It analysed whether homogeneity tests were useful to identify some anomalies and outliers in the data as well. The result showed that they are very useful for identifying some errors (bias and drift), although others are a bit more problematic. It also showed some of the issues with these tests, like the presence of trends in the data, that produces misleading results, requiring data to be transformed.

- It carried out an analysis of the role of correlation for identifying anomalies. It showed that this is not always an adequate solution, as it depends heavily on the type of sensor and the location. It demonstrated how the presence of errors and outliers affect the statistical tests for correlation (Pearson, Spearman and Kendall), so it could be useful if the context is adequate.

Chapter 5 goes further and proposed some solutions to look for anomalies in another way and to improve the way to detect such anomalies with the previous tests, as well as a set of processes/algorithms to understand how the sensors behave, identify issues and determine how much a sensor can be trusted:

- It proposed a different way to transform the data, beyond those transformations that are focused on improving the normality of the data or the elimination of trend. One of the proposed transformations is based on polynomial regression and showed some improvement with errors like malfunction, while maintained a good performance for bias and drift. On the other hand, the proposed exponential difference transformation was able to perform better than the original data, especially avoiding false positives when there are trends in the data.
- It analysed another way to exploit homogeneity tests by using the time series produced by the statistics, looking for particular patterns in the data ('peaks' that was possible to map with errors when observing the outcomes). This chapter proposed a new solution by identifying the turning points and calculating the angles produced, in such a way that the small angles would indicate potential issues in the data.
- It proposed three processes/algorithms for analysing the sensors' data, in such a way it is possible to understand the data much better and to identify problems with the datasets, which would indicate that the sensors used should not be trusted at all. These processes analyse the variation of the data, the presence of known errors and outliers, and the correlation with other known sensors. Such processes indicate how to deal with data transformations and with the statistical tests in such a way that, combining them adequately, it is possible to obtain an accurate picture of the behaviour of the sensors.
- It defined a fuzzy model as a way to aggregate the outcomes produced by the processes that analyse the sensors' data. Such fuzzy model defines the rules that combine those outcomes and provides a category of trust as result, indicating if the trust level of a sensor is high, medium or low.
- It described how to take advantage of the capabilities of parallelization to improve the performance of the calculations done. It showed how the performance can be increased when making use of the computational power of devices, an aspect that is becoming very relevant as devices at edge computing gain in computational power.

Chapter 6 is the one describing how the evaluation of the proposed solutions was done. It presented the methodology followed, as well as the results of the evaluation:

- It selected the set of relevant metrics for measuring the performance of the proposed solutions and it described the evaluation done, using a type of sensor already

studied (but from a dataset not used for the theoretical definition) and using a new type of sensor not considered before.

Technically speaking, the thesis has contributed with a set of R scripts that have been implemented in line with the theoretical designs and have been published openly, as they were released with an Apache v2 license. These R scripts are the implementations for doing the transformations proposed, the detection of outliers using the approach presented in Section 5, the three processes defined for the analysis of sensors' data (variation, presence of outliers and correlation) and the fuzzy aggregation model to provide a trust score. All these scripts are available in a GitHub repository (https://github.com/fjaviernieto/Sensor-Trust-Tools).

7.3. Relevant Publications

Some of the analysis and solutions proposed have been already published and presented to the scientific community, both in conference and journal papers.

In particular, in the case of the analysis of sensors' data and the processes proposed, the following article addressed those topics.

- Nieto, F.J., Aguilera, U., López-de-Ipiña, D., 2021. Analyzing Particularities of Sensor Datasets for Supporting Data Understanding and Preparation. Sensors 21, 6063. https://doi.org/10.3390/s21186063

Additionally, the following article addressed the initial ideas about a trust model for IoT (that would include a first approach to the analysis of data) and how such model would fit in an IoT environment:

- Vallati, C., Mingozzi, E., Tanganelli, G., Buonaccorsi, N., Valdambrini, N., Zonidis, N., Martínez, B., Mamelli, A., Sommacampagna, D., Anggorojati, B., Kyriazakos, S., Prasad, N., Nieto, F.J., Rodriguez, O.B., 2016. BETaaS: A Platform for Development and Execution of Machine-to-Machine Applications in the Internet of Things. Wireless Pers Commun 87, 1071–1091. https://doi.org/10.1007/s11277-015-2639-0

Finally, this conference paper addressed the exploitation of resources at edge devices in order to perform data analysis:

- Nieto De Santos, F.J., Villalonga, S.G., 2015. Exploiting Local Clouds in the Internet of Everything Environment, in: 2015 23rd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing. Presented at the 2015 23rd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, pp. 296–300. https://doi.org/10.1109/PDP.2015.117

Prior to those works, these conference papers addressed the different scenarios that should be considered in IoT environments and the architecture that could support them:

- Nieto, F.J., 2013. An architecture for a platform providing things as a service, in: 2013 16th International Symposium on Wireless Personal Multimedia Communications (WPMC). Presented at the 2013 16th International Symposium on Wireless Personal Multimedia Communications (WPMC), pp. 1–4.

- Kyriazakos, S., Nieto, F.J., 2013. Scenarios and applications in a Things as a service environment, in: 2013 16th International Symposium on Wireless Personal Multimedia Communications (WPMC). Presented at the 2013 16th International Symposium on Wireless Personal Multimedia Communications (WPMC), pp. 1–5.

7.4. Future Work

This dissertation has presented several contributions and has discussed about it in the corresponding chapters, indicating several improvements that could be done in future work. Additionally, there are other aspects that could be addressed in the future as well, based on the work presented.

7.4.1. Improvement of the Identification of Errors with the Homogeneity Statistics

As mentioned in Section 5, the current solution depends on a set of threshold values, selected depending on the study done with the available data. The study already showed that, depending on the type of sensor and the kind of errors in the data, the selection of the threshold was not easy, since there were many overlapping between minimum angles for correct samples and maximum angles of faulty samples.

The selection was done taking into account the possibility to reduce false positives, but without a total guarantee that those thresholds were the most accurate ones, assuming some errors might not be detected. In fact, even if the thresholds were low in general, the evaluation showed that false positives are still present, and the proposed model cannot be as stable as expected.

It should be possible to improve the outcome of the proposed solution by using a ML-based classification model that could use the information obtained from the analysis of the homogeneity tests as its main variables, like the number of turning points detected and the angles calculated.

There are multiple ML models that could be trained in order to determine which one performs best for this concrete kind of problem. Such training would require to generate a new dataset as result of multiple analysis of known datasets that are already tagged, so the resulting dataset would include, not only the angles and turning points, but also a variable indicating the type of error contained in the data.

Even if such model is not perfect, considering the type of classification problem, it seems it could be promising to optimize the classification of errors.

7.4.2. Automatic Identification of Sensors

Thanks to the observation and analysis done in Section 4, so much statistical data could be used to identify the type of sensor automatically. Given a set of measurements, a classification model could determine the type of sensor that fits best when calculating certain statistics.

The observations made clear that there are sensors that behave similar, while others are

very different (e.g., water salinity). Even when looking at the variance, the measurements are different, because of the expected thresholds of values, the magnitude of the units and many more factors.

Using aspects like the basic statistics (average, variance, mode, maximum value, minimum value) together with other information (like IQR), the type of distribution of the data (measured with statistical tests), presence of trends, etc… it would be possible to generate a dataset that could be used to train a ML model, able to classify the types of sensors.

Such functionality would be very useful for IoT systems. First of all, it could be used to annotate semantically sensors that are not known in open systems. Second, knowing the concrete type of sensor may facilitate the analysis of anomalies. Such information might be used to apply analysis models that fit much more the type of sensor analysed, so the results could be more accurate.

7.4.3. Additional Improvements to Detect Outliers

Besides the ideas proposed in Section 5 of the dissertation, there were other ideas that were going to be explored, but there was not time to do so.

The first one had to do with the adaptation of the Grubbs test calculation, since it shows some issues depending on the error of the sample. When the number of faulty observations is high, the impact in the mean makes the test to start failing. Therefore, one solution to explore would be to use another mean based on the IQR, in such a way not all the elements are used, assuming that there will be some outliers. The filtering using the IQR would remove some extreme values that could affect the calculation of the mean, giving more robustness to the statistical test.

In theory, when the sample is free of faulty values, the values produced by a sensor should not be too much dispersed and, therefore, the removal of some of the values should not be so relevant, although some risks could be considered. Depending on how the values are located, the calculation of such mean may imply that some values are considered extreme when they are not. Therefore, a deep analysis should be carried out.

Another idea has to do with the concept of supports and resistances in the area of economics. Such indexes are specific levels calculated depending on the previous behaviour of stock actions, so they act as a threshold that, when surpassed, indicate that something important may change in the time series in the short term. In the case of IoT, this principle could be applied, calculating supports and resistances with the latest metrics, and interpreting that something could go wrong if the next values start surpassing these thresholds.

It would require a specific study, not only to determine if it can be really useful, but also to design the adequate way to calculate the resistances and supports, as there may be several ways to do so.

Additionally, the proposition to combine normalization transformations of data with an adapted version of the exponential difference transformation may be interesting to facilitate the detection of outliers using the Grubbs test, as the outliers might be further highlighted. It is an idea that could be explored, looking for the adequate combination of

coefficients for the equation of the transformation.

7.4.4. Trust Management System based on Statistical Inputs

In the same way that it would be possible to use some statistical information to map samples and types of sensors, it would be interesting to explore how a ML model could support the detection of anomalies using statistical information.

The current solutions that use ML models to detect anomalies are based on the data produced by the sensors. They usually propose some transformations to the data (sometimes this is optional) and they classify the data to determine whether there is an error or not. That produces solutions that are very specific and dependent on the training datasets.

On the other hand, using the results of the statistical tests (Grubbs, SNHT, Buishand…), together with other aspects like coefficient of variation, the analysis of angles proposed in section 5, etc… could be the base of a model that is not so much tied to the data produced by concrete types of sensors, but to have a model that is focused on what happens in the mathematical part of the information.

The current model proposed in this dissertation combines these sources of information by defining some rules that support the detection of problems according to the observations done, but it is not perfect. A ML-based model would be another way to aggregate and combine the information, with a great potential to be more generic and flexible, as well as accurate.

7.4.5. Integrated Real-Time Trust Management System

The current dissertation has been focused on the definition of theoretical solutions with an implementation that is able to take CSV files as input and to process the information. The current implementation is not integrated or linked to any IoT system that could be collecting data in real-time.

Further work could be done in the implementation of the proposed solutions in such a way that they could be integrated with existing IoT systems. One possibility could be to re-implement the models in another language that provides the libraries required (e.g., Python has very good support for data analytics). Another solution could be to embed the invocation to the R scripts in a small piece of software that would expose some APIs that can be used by external systems that require some evaluation.

When thinking in real-time systems, it could be possible to embed the calculations proposed in a Complex Event Processing (CEP) engine, that could define the window size preferred and would be evaluating the trust of a sensor from time to time, as the new values arrive.

In any case, it requires some additional design, thinking in the adequate interfaces that would allow easy collection of the data and invocation of the data processing.

7.4.6. Generalize Data Analysis and Trust Evaluation

Although this work has used more than 12 types of sensors to design and evaluate a set of solutions for data understanding and trust evaluation, these types of sensors are limited in terms of domains of applicability. As previously mentioned, there is a limitation in the domains covered, as the datasets used were mainly related to agriculture, maritime surveillance, building automation and smart cities.

There are other domains that make intensive use of IoT related technologies, like the industrial environment or vehicles (autonomous or not), that were not analysed because of the lack of data. They produce huge amounts of data when operating and they may have their own particularities.

Therefore, the work presented in this dissertation could be extended and improved by using additional datasets that could, perhaps, require to adjust some of the solutions proposed, or to combine the inputs used in bit different way.

7.5. Final Remarks

With this dissertation, we aimed at providing some contributions in the area of data understanding and trust evaluation for IoT environments. It is the result of several years of work, also in trust models for other environments. Although during the last years some solutions were published, we deem that this dissertation was able to research further. We hope that other researchers will benefit from the contributions proposed, as well as from the future lines of research highlighted, since it is a topic that still has room for improvement, especially when addressing generic solutions, and considering that more and more systems will make use of data produced by sensors in the near future.

REFERENCES

Adams, S., Beling, P.A., Greenspan, S., Velez-Rojas, M., Mankovski, S., 2018. Model-Based Trust Assessment for Internet of Things Networks, in: 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE). Presented at the 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), pp. 1838–1843. https://doi.org/10.1109/TrustCom/BigDataSE.2018.00278

Aggarwal, R., Ranganathan, P., 2016. Common pitfalls in statistical analysis: The use of correlation techniques. Perspect. Clin. Res. 7, 187–190. https://doi.org/10.4103/2229-3485.192046

Alexandersson, H., 1986. A homogeneity test applied to precipitation data. J. Climatol. 6, 661–675. https://doi.org/10.1002/joc.3370060607

Alghofaili, Y., Rassam, M.A., 2022. A Trust Management Model for IoT Devices and Services Based on the Multi-Criteria Decision-Making Approach and Deep Long Short-Term Memory Technique. Sensors 22, 634. https://doi.org/10.3390/s22020634

Al-Rakhami, M.S., Al-Mashari, M., 2021. A Blockchain-Based Trust Model for the Internet of Things Supply Chain Management. Sensors 21, 1759. https://doi.org/10.3390/s21051759

Amdahl, G.M., 1967. Validity of the single processor approach to achieving large scale computing capabilities, in: Proceedings of the April 18-20, 1967, Spring Joint Computer Conference, AFIPS '67 (Spring). Association for Computing Machinery, New York, NY, USA, pp. 483–485. https://doi.org/10.1145/1465482.1465560

Anaconda, 2020. State of Data Science 2020 [WWW Document]. URL https://www.anaconda.com/state-of-data-science-2020 (accessed 9.21.22).

Anderson, T.W., Darling, D.A., 1954. A Test of Goodness of Fit. J. Am. Stat. Assoc. 49, 765–769. https://doi.org/10.2307/2281537

Baljak, V., Tei, K., Honiden, S., 2013. Fault classification and model learning from sensory Readings — Framework for fault tolerance in wireless sensor networks, in: 2013 IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing. Presented at the 2013 IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing, pp. 408–413. https://doi.org/10.1109/ISSNIP.2013.6529825

Bao, F., Chen, I.-R., 2012. Trust management for the internet of things and its application to

service composition, in: 2012 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM). Presented at the 2012 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), pp. 1–6. https://doi.org/10.1109/WoWMoM.2012.6263792

Benesty, J., Chen, J., Huang, Y., Cohen, I., 2009. Pearson Correlation Coefficient, in: Cohen, I., Huang, Y., Chen, J., Benesty, J. (Eds.), Noise Reduction in Speech Processing, Springer Topics in Signal Processing. Springer, Berlin, Heidelberg, pp. 1–4. https://doi.org/10.1007/978-3-642-00296-0_5

Box, G.E.P., Cox, D.R., 1964. An Analysis of Transformations. J. R. Stat. Soc. Ser. B Methodol. 26, 211–252.

Buishand, T.A., 1984. Tests for detecting a shift in the mean of hydrological time series. J. Hydrol. 73, 51–69. https://doi.org/10.1016/0022-1694(84)90032-5

Buishand, T.A., 1982. Some methods for testing the homogeneity of rainfall records. J. Hydrol. 58, 11–27. https://doi.org/10.1016/0022-1694(82)90066-X

Che Ros, F., Tosaka, H., Sidek, L.M., Basri, H., 2016. Homogeneity and trends in long-term rainfall data, Kelantan River Basin, Malaysia. Int. J. River Basin Manag. 14, 151–163. https://doi.org/10.1080/15715124.2015.1105233

Chen, D., Chang, G., Sun, D., Li, J., Jia, J., Wang, X., 2011. TRM-IoT: A trust management model based on fuzzy reputation for internet of things. Comput. Sci. Inf. Syst. 8, 1207–1228.

CrowdFlower, 2017. 2017 Data Scientist Report. CrowdFlower.

de Bruijn, B., Nguyen, T.A., Bucur, D., Tei, K., 2016. Benchmark Datasets for Fault Detection and Classification in Sensor Data, in: Proceedings of the 5th International Confererence on Sensor Networks, SENSORNETS 2016. SCITEPRESS - Science and Technology Publications, Lda, Setubal, PRT, pp. 185–195. https://doi.org/10.5220/0005637901850195

Delignette-Muller, M.L., Dutang, C., 2015. fitdistrplus: An R Package for Fitting Distributions. J. Stat. Softw. 64, 1–34. https://doi.org/10.18637/jss.v064.i04

Dixon, W.J., 1953. Processing Data for Outliers. Biometrics 9, 74–89. https://doi.org/10.2307/3001634

Erhan, L., Ndubuaku, M., Di Mauro, M., Song, W., Chen, M., Fortino, G., Bagdasar, O., Liotta, A., 2021. Smart anomaly detection in sensor systems: A multi-perspective review. Inf. Fusion 67, 64–79. https://doi.org/10.1016/j.inffus.2020.10.001

Firat, M., Dikbas, F., Koc, A.C., Gungor, M., 2012. Analysis of temperature series: estimation of missing data and homogeneity test. Meteorol. Appl. 19, 397–406. https://doi.org/10.1002/met.271

Grubbs, F.E., 1969. Procedures for Detecting Outlying Observations in Samples. Technometrics 11, 1–21. https://doi.org/10.2307/1266761

Gu, L., Wang, J., Sun, B., 2014. Trust management mechanism for Internet of Things. China Commun. 11, 148–156. https://doi.org/10.1109/CC.2014.6821746

Guh, R.-S., 2002. Effects of Non-Normality on Artificial Neural Network Based Control Chart Pattern Recognizer. J. Chin. Inst. Ind. Eng. 19, 13–22. https://doi.org/10.1080/10170660209509363

Hamarashid, H.K., Qader, S.M., Saeed, S.A., Hassan, B.A., Ali, N.A., 2022. Machine Learning Algorithms Evaluation Methods by Utilizing R. UKH J. Sci. Eng. 6, 1–11. https://doi.org/10.25079/ukhjse.v6n1y2022.pp1-11

Hasan, M., 2022. Number of connected IoT devices growing 18% to 14.4 billion globally [WWW Document]. URL https://iot-analytics.com/number-connected-iot-devices/ (accessed 9.21.22).

Hasan, M., Islam, Md.M., Zarif, M.I.I., Hashem, M.M.A., 2019. Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. Internet Things 7, 100059. https://doi.org/10.1016/j.iot.2019.100059

Hevner, A.R., March, S.T., Park, J., Ram, S., 2004. Design Science in Information Systems Research. MIS Q. 28, 75–105. https://doi.org/10.2307/25148625

Hoefler, T., Belli, R., 2015. Scientific benchmarking of parallel computing systems: twelve ways to tell the masses when reporting performance results, in: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC '15. Association for Computing Machinery, New York, NY, USA, pp. 1–12. https://doi.org/10.1145/2807591.2807644

Hossin, M., M.N, S., 2015. A Review on Evaluation Metrics for Data Classification Evaluations. Int. J. Data Min. Knowl. Manag. Process 5, 01–11. https://doi.org/10.5121/ijdkp.2015.5201

Huber, S., Wiemer, H., Schneider, D., Ihlenfeldt, S., 2019. DMME: Data mining methodology for engineering applications – a holistic extension to the CRISP-DM model. Procedia CIRP, 12th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 18-20 July 2018, Gulf of Naples, Italy 79, 403–408. https://doi.org/10.1016/j.procir.2019.02.106

Ingelrest, F., Barrenetxea, G., Schaefer, G., Vetterli, M., Couach, O., Parlange, M., 2010. SensorScope: Application-specific sensor network for environmental monitoring. ACM Trans. Sens. Netw. 6, 17:1-17:32. https://doi.org/10.1145/1689239.1689247

Javed, N., Wolf, T., 2012. Automated Sensor Verification Using Outlier Detection in the Internet of Things, in: 2012 32nd International Conference on Distributed Computing Systems Workshops. Presented at the 2012 32nd International Conference on Distributed Computing Systems Workshops, pp. 291–296. https://doi.org/10.1109/ICDCSW.2012.78

Jeffery, S.R., Alonso, G., Franklin, M.J., Hong, W., Widom, J., 2006. Declarative Support for Sensor Data Cleaning, in: Fishkin, K.P., Schiele, B., Nixon, P., Quigley, A. (Eds.), Pervasive Computing, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 83–100. https://doi.org/10.1007/11748625_6

Jøsang, A., Ismail, R., Boyd, C., 2007. A survey of trust and reputation systems for online service provision. Decis. Support Syst., Emerging Issues in Collaborative Commerce

43, 618–644. https://doi.org/10.1016/j.dss.2005.05.019

Kenda, K., Mladenić, D., 2018. Autonomous Sensor Data Cleaning in Stream Mining Setting. Bus. Syst. Res. J. 9, 69–79. https://doi.org/10.2478/bsrj-2018-0020

Kendall, M.G., 1945. The treatment of ties in ranking problems. Biometrika 33, 239–251. https://doi.org/10.1093/biomet/33.3.239

Lanzante, J.R., 1996. Resistant, Robust and Non-Parametric Techniques for the Analysis of Climate Data: Theory and Examples, Including Applications to Historical Radiosonde Station Data. Int. J. Climatol. 16, 1197–1226. https://doi.org/10.1002/(SICI)1097-0088(199611)16:11<1197::AID-JOC89>3.0.CO;2-L

Leys, C., Ley, C., Klein, O., Bernard, P., Licata, L., 2013. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. J. Exp. Soc. Psychol. 49, 764–766. https://doi.org/10.1016/j.jesp.2013.03.013

Liu, J., Adams, S., Beling, P.A., 2020. An Ensemble Trust Scoring Method for Internet of Things Sensor Networks, in: 2020 IEEE 6th World Forum on Internet of Things (WF-IoT). Presented at the 2020 IEEE 6th World Forum on Internet of Things (WF-IoT), pp. 1–6. https://doi.org/10.1109/WF-IoT48130.2020.9221203

Liu, J., Deng, H., 2013. Outlier detection on uncertain data based on local information. Knowl.-Based Syst. 51, 60–71. https://doi.org/10.1016/j.knosys.2013.07.005

Magán-Carrión, R., Camacho, J., García-Teodoro, P., 2015. Multivariate Statistical Approach for Anomaly Detection and Lost Data Recovery in Wireless Sensor Networks. Int. J. Distrib. Sens. Netw. 11, 672124. https://doi.org/10.1155/2015/672124

Mann, H.B., 1945. Nonparametric Tests Against Trend. Econometrica 13, 245–259. https://doi.org/10.2307/1907187

Martí, L., Sanchez-Pi, N., Molina, J.M., Garcia, A.C.B., 2015. Anomaly Detection Based on Sensor Data in Petroleum Industry Applications. Sensors 15, 2774–2797. https://doi.org/10.3390/s150202774

Martins, H., Palma, L., Cardoso, A., Gil, P., 2015. A support vector machine based technique for online detection of outliers in transient time series, in: 2015 10th Asian Control Conference (ASCC). Presented at the 2015 10th Asian Control Conference (ASCC), pp. 1–6. https://doi.org/10.1109/ASCC.2015.7244794

Maseda, F.J., López, I., Martija, I., Alkorta, P., Garrido, A.J., Garrido, I., 2021. Sensors Data Analysis in Supervisory Control and Data Acquisition (SCADA) Systems to Foresee Failures with an Undetermined Origin. Sensors 21, 2762. https://doi.org/10.3390/s21082762

Mooney, P., 2018. 2018 Kaggle Machine Learning & Data Science Survey [WWW Document]. URL https://www.kaggle.com/code/paultimothymooney/2018-kaggle-machine-learning-data-science-survey/notebook (accessed 9.21.22).

Ni, K., Ramanathan, N., Chehade, M.N.H., Balzano, L., Nair, S., Zahedi, S., Kohler, E., Pottie, G., Hansen, M., Srivastava, M., 2009. Sensor network data fault types. ACM Trans. Sens. Netw. 5, 25:1-25:29. https://doi.org/10.1145/1525856.1525863

Noor, T.H., Sheng, Q.Z., Alfazi, A., 2013. Reputation Attacks Detection for Effective Trust Assessment among Cloud Services, in: 2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications. Presented at the 2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, pp. 469–476. https://doi.org/10.1109/TrustCom.2013.59

Oucheikh, R., Fri, M., Fedouaki, F., Hain, M., 2020. Deep Real-Time Anomaly Detection for Connected Autonomous Vehicles. Procedia Comput. Sci., The 11th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2020) / The 10th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH 2020) / Affiliated Workshops 177, 456–461. https://doi.org/10.1016/j.procs.2020.10.062

Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S., 2007. A Design Science Research Methodology for Information Systems Research. J. Manag. Inf. Syst. 24, 45–77. https://doi.org/10.2753/MIS0742-1222240302

Pettitt, A.N., 1979. A Non-Parametric Approach to the Change-Point Problem. J. R. Stat. Soc. Ser. C Appl. Stat. 28, 126–135. https://doi.org/10.2307/2346729

Presser, M., Krco, S., 2011. Initial report on IoT applications of strategic interest (No. D2.1). The Internet of Things Initiative Project.

Ramotsoela, D., Abu-Mahfouz, A., Hancke, G., 2018. A Survey of Anomaly Detection in Industrial Wireless Sensor Networks with Critical Water System Infrastructure as a Case Study. Sensors 18, 2491. https://doi.org/10.3390/s18082491

Reed, G.F., Lynn, F., Meade, B.D., 2002. Use of coefficient of variation in assessing variability of quantitative assays. Clin. Diagn. Lab. Immunol. 9, 1235–1239. https://doi.org/10.1128/cdli.9.6.1235-1239.2002

Reinsel, D., 2019. The Global DataSphere & Its Enterprise Impact | IDC Blog [WWW Document]. URL https://blogs.idc.com/2019/11/04/how-you-contribute-to-todays-growing-datasphere-and-its-enterprise-impact/ (accessed 9.21.22).

Rosner, B., 1983. Percentage Points for a Generalized ESD Many-Outlier Procedure. Technometrics 25, 165–172. https://doi.org/10.1080/00401706.1983.10487848

Ryan, C.M., Parnell, A., Mahoney, C., 2020. Real-Time Anomaly Detection for Advanced Manufacturing: Improving on Twitter's State of the Art. https://doi.org/10.48550/arXiv.1911.05376

Schulz, T., Leister, W., 2012. Ideas for a Trust Indicator in the Internet of Things. Presented at the SMART 2012 : The First International Conference on Smart Systems, Devices and Technologies.

SHAPIRO, S.S., WILK, M.B., 1965. An analysis of variance test for normality (complete samples)†. Biometrika 52, 591–611. https://doi.org/10.1093/biomet/52.3-4.591

Shen, W., Han, G., Cheng, M., Zhu, C., Hu, G., 2010. Energy prediction based trust manangement in hierarchical sensor networks, in: 2010 International Conference on Computer Application and System Modeling (ICCASM 2010). Presented at the 2010

International Conference on Computer Application and System Modeling (ICCASM 2010), pp. V12-453-V12-456. https://doi.org/10.1109/ICCASM.2010.5622353

Spearman Rank Correlation Coefficient, 2008. , in: The Concise Encyclopedia of Statistics. Springer, New York, NY, pp. 502–505. https://doi.org/10.1007/978-0-387-32833-1_379

Sun, Y.L., Han, Z., Yu, W., Ray Liu, K.J., 2006. Attacks on Trust Evaluation in Distributed Networks, in: 2006 40th Annual Conference on Information Sciences and Systems. Presented at the 2006 40th Annual Conference on Information Sciences and Systems, pp. 1461–1466. https://doi.org/10.1109/CISS.2006.286695

Tanuska, P., Spendla, L., Kebisek, M., Duris, R., Stremy, M., 2021. Smart Anomaly Detection and Prediction for Assembly Process Maintenance in Compliance with Industry 4.0. Sensors 21, 2376. https://doi.org/10.3390/s21072376

Teh, H.Y., Kempa-Liehr, A.W., Wang, K.I.-K., 2020. Sensor data quality: a systematic review. J. Big Data 7, 11. https://doi.org/10.1186/s40537-020-0285-1

Tietjen, G.L., Moore, R.H., 1972. Some Grubbs-Type Statistics for the Detection of Several Outliers. Technometrics 14, 583–597. https://doi.org/10.1080/00401706.1972.10488948

Vailshery, L.S., 2022. IoT connected devices worldwide 2019-2030 | Statista [WWW Document]. URL https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/ (accessed 9.21.22).

Wegner, P., 2022. Global IoT market size grew 22% in 2021 [WWW Document]. URL https://iot-analytics.com/iot-market-size/ (accessed 9.21.22).

Wirth, R., Hipp, J., 2000. CRISP-DM: Towards a standard process model for data mining. Proc. 4th Int. Conf. Pract. Appl. Knowl. Discov. Data Min. 29–39.

Yeo, I.-K., Johnson, R.A., 2000. A New Family of Power Transformations to Improve Normality or Symmetry. Biometrika 87, 954–959.

Zhang, Y., Szabo, C., Sheng, Q.Z., 2016. Reduce or remove: Individual sensor reliability profiling and data cleaning. Intell. Data Anal. 20, 979–995. https://doi.org/10.3233/IDA-160853

APPENDIX A

This appendix contains tables that show the basic statistical data of the different sensors analysed in Section 4. It is part of the study that was carried out in the context of subsection 4.1.

Air humidity measures were taken from the Arduino customs system located indoors, in a room with air conditioning. The following table shows the basic statistics for this kind of sensor.

Table 10 Basic analysis of air humidity from the Arduino system

Period	Average	Variance	Max. Value	Min. Value	Mode(s)
All Data	63.96	21.9894	90	56	62
Last 3 Days	62.61	9.0744	76	58	61
Last 1 Day	62.47	8.2047	76	59	62
Last 12 Hours	61.42	5.5884	69	59	59
Last 6 Hours	59.62	0.6483	62	59	59
Last 3 Hours	59.91	0.8588	62	59	59
Last 2 Hours	60.18	0.95	62	59	60
Last 1 Hour	60.94	0.4623	62	60	61
Last 30 Minutes	61.12	0.5166	62	60	61
Last 10 Minutes	61.67	0.2666	62	61	62

Temperature is a very common metric taken in many systems. Therefore, it was possible to collect temperature metrics from several systems. As the data sources are heterogeneous (in location, sensor models and frequency), each one has been analysed separately.

In the case of Ports of Spain, there are two datasets providing information about air temperature: REDEXT Golfo de Cádiz and REDCOSM Puerta Carnero.

Table 11 Basic analysis of temperature from REDEXT Golfo de Cádiz

Period	Average	Variance	Max. Value	Min. Value	Mode(s)
All Data	18.3645	11.5625	30.7	5.0	16.1
Last Year	18.6057	14.6622	30.6	8.8	15.2

Last 6 Months	15.2687	4.1171	20.6	8.8	15.2
Last Month	15.9480	2.5046	20.0	11.9	17.2
Last 2 Weeks	16.3391	2.2276	19.1	12.2	17.2
Last Week	16.8142	1.8376	19.1	13.4	17.2
Last 3 Days	17.8191	0.2860	19.1	17.2	17.5
Last Day	17.416	0.0772	18.4	17.2	17.2
Last 12 Hours	17.4769	0.1285	18.4	17.2	17.2, 17.3
Last 6 Hours	17.6714	0.1590	18.4	17.2	17.8
Last 3 Hours	17.925	0.1024	18.4	17.7	17.8
Last 2 Hours	18.0	0.1199	18.4	17.8	17.8
Last Hour	18.1	0.1799	18.4	17.8	17.8, 18.4
Last 30 Minutes	18.4	0.0	18.4	18.4	18.4

Table 12 Basic analysis of temperature from REDCOSM Puerta Carnero

Period	**Average**	**Variance**	**Max. Value**	**Min. Value**	**Mode(s)**
All Data	17.6947	8.4207	27.1	9.3	18.7
Last Year	14.4908	1.6742	18.0	10.1	14.9
Last 6 Months	14.4908	1.6742	18.0	10.1	14.9
Last Month	14.7641	1.5899	18.0	11.6	14.9
Last 2 Weeks	15.1598	1.3180	18.0	12.0	15.6, 15.7, 16.0
Last Week	15.4712	0.8279	17.3	12.7	15.7
Last 3 Days	16.0593	0.4040	17.3	14.2	16.0
Last Day	16.5043	0.2195	17.3	15.9	16.0
Last 12 Hours	16.5909	0.2049	17.3	16.0	16.8
Last 6 Hours	16.8571	0.1095	17.3	16.5	16.8
Last 3 Hours	17.05	0.0833	17.3	16.8	16.8, 17.3

Last 2 Hours	17.1333	0.0833	17.3	16.8	17.3
Last Hour	17.3	0.0	17.3	17.3	17.3
Last 30 Minutes	17.3	0.0	17.3	17.3	17.3

In the case of the benchmark datasets, we selected one indoor sensor (from the Intel platform) and two outdoor sensors (from the Smart Santander and the SensorScope platforms). In all cases, we include a table with the clean data and another one with one of the errors injected for the same sensor and using the same samples (enabling a fair comparison).

Table 13 Basic analysis of indoor air temperature (ºC) from the benchmarking dataset (Intel)

Period	Average	Variance	Max. Value	Min. Value	Mode(s)
All Data	21.12	4.1363	25.37	17.62	21.81
Last 3 Days	20.43	4.2483	25.21	17.62	17.95
Last 1 Day	19.20	0.8182	20.83	17.66	19.01
Last 12 Hours	18.69	0.6294	20.35	17.66	17.95
Last 6 Hours	18.52	0.2506	19.50	17.71	17.95
Last 3 Hours	18.94	0.1059	19.50	18.41	19.43
Last 2 Hours	19.13	0.0534	19.50	18.75	19.43
Last 1 Hour	19.33	0.0127	19.50	19.11	19.43
Last 30 Minutes	19.42	0.0019	19.50	19.34	19.43
Last 10 Minutes	19.47	0.0003	19.50	19.44	19.45

Table 14 Basic analysis of indoor air temperature (ºC) from the benchmarking dataset (Intel) with drift error injected

Period	Average	Variance	Max. Value	Min. Value	Mode(s)
All Data	23.12	194.339	71.34	-19.80	18.73
Last 3 Days	24.03	276.6082	71.34	-19.32	17.95
Last 1 Day	19.20	0.8182	20.83	17.66	19.01
Last 12 Hours	18.69	0.6294	20.35	17.66	17.95

Last 6 Hours	18.52	0.2506	19.50	17.71	17.95
Last 3 Hours	18.94	0.1059	19.50	18.41	19.43
Last 2 Hours	19.13	0.0534	19.50	18.75	19.43
Last 1 Hour	19.33	0.0127	19.50	19.11	19.43
Last 30 Minutes	19.42	0.0019	19.50	19.34	19.43
Last 10 Minutes	19.47	0.0003	19.50	19.44	19.45

Table 15 Basic analysis of outdoor air temperature (ºC) from the benchmarking dataset (Smart Santander)

Period	Average	Variance	Max. Value	Min. Value	Mode(s)
All Data	19.86	7.5579	28.83	14.32	21.61
Last Week	19.74	7.0705	28.83	14.77	20.96
Last 3 Days	21.09	4.9016	27.41	17.48	20.96
Last 1 Day	21.16	5.8634	27.41	17.54	18.25
Last 12 Hours	20.92	7.4822	27.41	18.00	20.96
Last 6 Hours	23.05	5.5311	27.41	20.12	20.96
Last 3 Hours	25.08	2.4245	27.41	22.49	27.00
Last 2 Hours	25.97	1.0743	27.41	24.27	27.00
Last 1 Hour	26.86	0.2392	27.41	26.05	27.00
Last 30 Minutes	27.21	0.0232	27.41	27.00	27.00
Last 10 Minutes	27.09	0.0072	27.17	27.00	27.00

Table 16 Basic analysis of outdoor air temperature (ºC) from the benchmarking dataset (Smart Santander) with bias error injected

Period	Average	Variance	Max. Value	Min. Value	Mode(s)
All Data	22.32	34.7670	41.80	14.51	29.22
Last Week	22.93	45.2683	41.80	14.77	29.22

Last 3 Days	21.65	7.2336	27.41	17.48	26.68
Last 1 Day	21.16	5.8634	27.41	17.54	18.25
Last 12 Hours	20.92	7.4822	27.41	18.00	20.96
Last 6 Hours	23.05	5.5311	27.41	20.12	20.96
Last 3 Hours	25.08	2.4245	27.41	22.49	27.00
Last 2 Hours	25.97	1.0743	27.41	24.27	27.00
Last 1 Hour	26.86	0.2392	27.41	26.05	27.00
Last 30 Minutes	27.21	0.0232	27.41	27.00	27.00
Last 10 Minutes	27.09	0.0072	27.17	27.00	27.00

Table 17 Basic analysis of outdoor air temperature (ºC) from the benchmarking dataset (SensorScope)

Period	Average	Variance	Max. Value	Min. Value	Mode(s)
All Data	5.194	21.1983	15.66	-4.84	1.36
Last Week	4.3633	21.9965	13.59	-3.42	7.21
Last 3 Days	7.528	6.8349	13.59	2.94	7.21
Last 1 Day	6.979	12.4719	13.59	2.94	3.7
Last 12 Hours	3.819	0.2018	5.7	2.94	3.7
Last 6 Hours	3.77	0.078	4.5	2.94	4.08
Last 3 Hours	3.852	0.1046	4.5	2.94	4.08
Last 2 Hours	3.732	0.0901	4.3	2.94	3.36
Last 1 Hour	3.588	0.0946	4.1	2.94	3.36
Last 30 Minutes	3.468	0.1024	4.01	2.94	3.36
Last 10 Minutes	3.254	0.0378	3.46	2.98	3.46

Table 18 Basic analysis of outdoor air temperature (ºC) from the benchmarking dataset (SensorScope) with malfunction error injected

Period	Average	Variance	Max. Value	Min. Value	Mode(s)
All Data	5.212	21.9635	21.754	-6.227	7.21
Last Week	4.364	22.1334	13.59	-4.682	7.21
Last 3 Days	7.528	6.8349	13.59	2.94	7.21
Last 1 Day	6.979	12.4719	13.59	2.94	3.7
Last 12 Hours	3.819	0.2018	5.7	2.94	3.7
Last 6 Hours	3.77	0.078	4.5	2.94	4.08
Last 3 Hours	3.852	0.1046	4.5	2.94	4.08
Last 2 Hours	3.732	0.0901	4.3	2.94	3.36
Last 1 Hour	3.588	0.0946	4.1	2.94	3.36
Last 30 Minutes	3.468	0.1024	4.01	2.94	3.36
Last 10 Minutes	3.254	0.0378	3.46	2.98	3.46

The Arduino system has stored the metrics in two different units: ºC and ºF. The following tables show the basic statistics for both cases, so it is possible to analyse how the unit of the metric affects the analysis of the data.

Table 19 Basic analysis of air temperature (ºC) from the Arduino system

Period	Average	Variance	Max. Value	Min. Value	Mode(s)
All Data	26.69	0.6497	31.90	25.40	26.5
Last 3 Days	26.47	0.3058	27.80	25.40	26.5
Last 1 Day	26.34	0.1946	27.60	25.60	26.5
Last 12 Hours	26.39	0.1235	27.30	25.70	26.5
Last 6 Hours	26.55	0.0097	26.80	26.30	26.5
Last 3 Hours	26.56	0.0109	26.80	26.30	26.5
Last 2 Hours	26.54	0.0113	26.70	26.30	26.5
Last 1 Hour	26.52	0.0131	26.70	26.30	26.5

Last 30 Minutes	26.53	0.0156	26.70	26.30	26.5
Last 10 Minutes	26.6	0.012	26.7	26.5	26.5

Table 20 Basic analysis of air temperature (ºF) from the Arduino system

Period	Average	Variance	Max. Value	Min. Value	Mode(s)
All Data	80.05	2.1051	89.42	77.72	79.7
Last 3 Days	79.64	0.9911	82.04	77.72	79.7
Last 1 Day	79.41	0.6305	81.68	78.08	79.7
Last 12 Hours	79.51	0.4001	81.14	78.26	79.7
Last 6 Hours	79.79	0.0315	80.24	79.34	79.7
Last 3 Hours	79.80	0.0353	80.24	79.34	79.7
Last 2 Hours	79.77	0.0368	80.06	79.34	79.7
Last 1 Hour	79.74	0.0425	80.06	79.34	79.7
Last 30 Minutes	79.76	0.0506	80.06	79.34	79.7
Last 10 Minutes	79.88	0.0388	80.06	79.70	79.7

The atmospheric pressure is a meteorological aspect that seems to be quite stable in time and in large areas. In this case, all the datasets available were part of the Ports of Spain datasets.

Table 21 Basic analysis of atmospheric pressure from REMPOR Dique Abrigo

Period	Average	Variance	Max. Value	Min. Value	Mode(s)
All Data	1014.7939	13.3567	1027.2	999.9	1015.4
Last Year	1014.7939	13.3567	1027.2	999.9	1015.4
Last 6 Months	1015.2992	11.5479	1027.2	1000.8	1015.4
Last Month	1015.7387	43.1070	1027.2	1000.8	1020.4
Last 2 Weeks	1021.9064	5.9373	1027.2	1017.6	1022.1
Last Week	1023.2334	0.6157	1025.5	1021.6	1023.6
Last 3 Days	1023.2334	0.6157	1025.5	1021.6	1023.6

Last Day	1022.8827	0.3390	1023.9	1021.6	1023.6
Last 12 Hours	1022.9880	0.2387	1023.9	1022.2	1023.6
Last 6 Hours	1022.9187	0.2274	1023.9	1022.3	1022.4
Last 3 Hours	1023.3	0.1057	1023.9	1022.8	1023.0
Last 2 Hours	1023.4627	0.0758	1023.9	1022.8	1023.6
Last Hour	1023.6612	0.0091	1023.9	1023.5	1023.6
Last 30 Minutes	1023.6437	0.0039	1023.7	1023.5	1023.7

Table 22 Basic analysis of atmospheric pressure from REDEXT Golfo de Cádiz

Period	Average	Variance	Max. Value	Min. Value	Mode(s)
All Data	1017.3426	33.2493	1038.3	982.0	1016.0
Last Year	1017.5171	39.3756	1038.3	987.3	1014.1
Last 6 Months	1018.6806	65.9958	1038.3	987.3	1018.9
Last Month	1014.9055	16.9953	1023.6	1001.2	1014.1
Last 2 Weeks	1015.4694	4.0931	1019.5	1010.9	1017.2
Last Week	1014.5118	4.0125	1019.5	1010.9	1012.9
Last 3 Days	1013.7520	1.1230	1015.6	1011.3	1012.9
Last Day	1013.712	0.9269	1015.2	1012.1	1012.9, 1014.1
Last 12 Hours	1012.9769	0.5235	1014.5	1012.1	1012.9
Last 6 Hours	1012.7	0.1999	1013.3	1012.1	1012.9
Last 3 Hours	1012.95	0.0633	1013.3	1012.7	1012.9
Last 2 Hours	1012.9666	0.0933	1013.3	1012.7	1012.7, 1012.9, 1013.3
Last Hour	1012.8	0.0199	1012.9	1012.7	1012.7, 1012.9
Last 30 Minutes	1012.7	0.0	1012.7	1012.7	1012.7

Table 23 Basic analysis of atmospheric pressure from REDCOSM Puerta Carnero

Period	Average	Variance	Max. Value	Min. Value	Mode(s)
All Data	101.7920	0.4333	104.2	95.4	101.6
Last Year	101.5848	0.1917	102.7	100.1	101.5
Last 6 Months	101.5848	0.1917	102.7	100.1	101.5
Last Month	101.5559	0.1800	102.4	100.1	101.5
Last 2 Weeks	101.5621	0.0370	102.0	101.1	101.6
Last Week	101.4738	0.0335	101.8	101.1	101.6
Last 3 Days	101.4390	0.0236	101.7	101.2	101.5
Last Day	101.2695	0.0067	101.4	101.2	101.2
Last 12 Hours	101.2727	0.0061	101.4	101.2	101.2
Last 6 Hours	101.2857	0.0080	101.4	101.2	101.2
Last 3 Hours	101.35	0.0033	101.4	101.3	101.3, 101.4
Last 2 Hours	101.3666	0.0033	101.4	101.3	101.4
Last Hour	101.4	0.0	101.4	101.4	101.4
Last 30 Minutes	101.4	0.0	101.4	101.4	101.4

In the context of the benchmark datasets, the platform used for indoor sensors data collection (Intel) provided measures for light intensity. We have taken one of the sensors as an example, creating one table with the clean measurements and an additional one with one of the errors injected. The same samples (in time) are selected, so it is possible to compare how the statistics vary.

Table 24 Basic analysis of indoor light intensity from the benchmarking dataset (Intel)

Period	Average	Variance	Max. Value	Min. Value	Mode(s)
All Data	382.85	173501.9	1435.20	1.38	114.08
Last 3 Days	334.23	138273.7	1258.56	1.38	114.08
Last 1 Day	378.3	130578.8	1199.7	108.6	114.08
Last 12 Hours	406.1	157419.9	1140.8	108.6	114.08
Last 6 Hours	111.5	7.4377	114.08	108.6	114.08

Last 3 Hours	113.9	0.9653	114.08	108.6	114.08
Last 2 Hours	114.08	0	114.08	114.08	114.08
Last 1 Hour	114.08	0	114.08	114.08	114.08
Last 30 Minutes	114.08	0	114.08	114.08	114.08
Last 10 Minutes	114.08	0	114.08	114.08	114.08

Table 25 Basic analysis of indoor light intensity from the benchmarking dataset (Intel) with random error injected

Period	Average	Variance	Max. Value	Min. Value	Mode(s)
All Data	410.46	248059.6	4611.04	1.38	114.08
Last 3 Days	346.72	160110.1	3374.56	1.38	114.08
Last 1 Day	378.3	130578.8	1199.7	108.6	114.08
Last 12 Hours	406.1	157419.9	1140.8	108.6	114.08
Last 6 Hours	111.5	7.4377	114.08	108.6	114.08
Last 3 Hours	113.9	0.9653	114.08	108.6	114.08
Last 2 Hours	114.08	0	114.08	114.08	114.08
Last 1 Hour	114.08	0	114.08	114.08	114.08
Last 30 Minutes	114.08	0	114.08	114.08	114.08
Last 10 Minutes	114.08	0	114.08	114.08	114.08

PM10 measures the inhalable particles that are floating in the air, representing air pollution. This metric has been obtained from the Gyor dataset.

Table 26 Basic analysis of PM10 in a Gyor Station

Period	Average	Variance	Max. Value	Min. Value	Mode(s)
All Data	27.09	391.1305	534.60	3.07	17.23
Last 3 Days	20.72	122.1424	149.10	5.88	17.23
Last 1 Day	21.59	165.7923	149.10	5.88	19.03
Last 12 Hours	17.71	129.1766	126.71	5.88	16.03

Last 6 Hours	14.91	108.4779	77.09	5.88	7.0
Last 3 Hours	10.690	38.6210	62.730	6.110	13.95
Last 2 Hours	11.820	28.3537	54.290	7.480	13.95
Last 1 Hour	13.38	9.1598	28.36	9.76	13.95
Last 30 Minutes	15.32	9.2535	28.36	12.25	15.94

Precipitation measures how much water has been collected in a concrete area, because of the rains.

Table 27 Basic analysis of precipitation from a meteorological station

Period	Average	Variance	Max. Value	Min. Value	Mode(s)
All Data	0.032	0.0376	8.8	0.0	0.0
Last 6 Months	0.0313	0.0295	6.4	0.0	0.0
Last Month	0.0423	0.0829	6.4	0.0	0.0
Last 2 Weeks	0.049	0.0928	5.4	0.0	0.0
Last Week	0.051	0.1123	5.4	0.0	0.0
Last 3 Days	0.0027	0.0013	0.6	0.0	0.0
Last 1 Day	0.0	0.0	0.0	0.0	0.0
Last 12 Hours	0.0	0.0	0.0	0.0	0.0
Last 6 Hours	0.0	0.0	0.0	0.0	0.0
Last 3 Hours	0.0	0.0	0.0	0.0	0.0
Last 2 Hours	0.0	0.0	0.0	0.0	0.0
Last 1 Hour	0.0	0.0	0.0	0.0	0.0
Last 30 Minutes	0.0	0.0	0.0	0.0	0.0

Ports of Spain has provided several datasets that include information about sea level. In principle, this kind of metric is expected to show a clear periodicity, linked to the tides.

Table 28 Basic analysis of sea level from REDMAR Algeciras

Period	Average	Variance	Max. Value	Min. Value	Mode(s)
All Data	68.7072	744.5783	164.0	-12.0	50.0

Last Year	69.2927	754.6904	153.0	-12.0	52.0
Last 6 Months	66.0490	738.1573	153.0	-12.0	52.0
Last Month	63.5775	628.6282	114.0	9.0	84.0
Last 2 Weeks	63.3149	652.3095	114.0	11.0	84.0, 86.0
Last Week	63.6569	454.6123	102.0	11.0	54.0, 86.0
Last 3 Days	68.3606	209.8743	94.0	43.0	54.0
Last Day	67.3356	171.0432	89.0	48.0	52.0
Last 12 Hours	67.0827	202.2153	89.0	48.0	50.0
Last 6 Hours	79.8630	44.3420	89.0	66.0	87.0, 88.0
Last 3 Hours	78.6216	48.6306	89.0	66.0	68.0, 76.0, 81.0, 84.0, 86.0, 87.0
Last 2 Hours	74.88	26.8600	84.0	66.0	68.0, 76.0, 81.0
Last Hour	70.6923	8.2307	76.0	66.0	68.0
Last 30 Minutes	68.5714	2.6190	71.0	66.0	68.0

Table 29 Basic analysis of sea level from REDMAR Huelva

Period	Average	Variance	Max. Value	Min. Value	Mode(s)
All Data	202.9581	6702.9120	427.0	-12.0	140.0
Last Year	205.3395	6932.4485	418.0	5.0	136.0
Last 6 Months	201.3112	6907.8280	418.0	5.0	136.0
Last Month	202.2161	6756.5275	358.0	32.0	154.0, 156.0, 268.0
Last 2 Weeks	200.6250	6642.7349	347.0	44.0	166.0
Last Week	205.0029	4284.2043	330.0	62.0	166.0
Last 3 Days	209.4658	1858.2815	287.0	127.0	166.0
Last Day	207.9515	1564.1087	274.0	146.0	251.0
Last 12 Hours	215.0827	1614.2569	274.0	161.0	170.0

Last 6 Hours	251.3013	303.0468	274.0	218.0	269.0, 272.0
Last 3 Hours	249.3783	342.8528	274.0	218.0	273.0
Last 2 Hours	239.4	188.9166	263.0	218.0	232.0
Last Hour	228.3846	50.5897	240.0	218.0	232.0
Last 30 Minutes	223.0	19.3333	230.0	218.0	218.0, 219.0, 220.0, 223.0, 224.0, 227.0, 230.0

Table 30 Basic analysis of sea level from REDMAR Bonanza

Period	Average	Variance	Max. Value	Min. Value	Mode(s)
All Data	172.6480	5216.1991	386.0	-3.0	108.0
Last Year	176.0339	5009.4763	385.0	25.0	134.0
Last 6 Months	173.5978	4907.6023	385.0	25.0	104.0
Last Month	175.5082	4624.9184	313.0	49.0	118.0, 130.0
Last 2 Weeks	172.8184	4542.2855	308.0	55.0	130.0
Last Week	173.1432	3044.4591	286.0	65.0	118.0
Last 3 Days	173.6971	1510.5632	246.0	96.0	138.0
Last Day	172.2941	1244.7222	236.0	118.0	210.0
Last 12 Hours	179.2965	1303.3906	236.0	128.0	136.0
Last 6 Hours	212.0684	247.6480	236.0	183.0	223.0
Last 3 Hours	212.8378	282.8618	236.0	184.0	231.0, 232.0
Last 2 Hours	203.64	150.3233	226.0	184.0	199.0, 201.0, 202.0
Last Hour	194.4615	41.4358	202.0	184.0	199.0, 201.0
Last 30 Minutes	189.4285	17.6190	196.0	184.0	184.0, 186.0, 187.0, 189.0, 191.0, 193.0, 196.0

Another of the datasets, the one collected with the Arduino, includes information about soil moisture, that is important because it provides information about the humidity in the soil,

which may be very problematic if it is too low or too high. The following table shows its basic statistics.

Table 31 Basic analysis of moisture from the Arduino system

Period	Average	Variance	Max. Value	Min. Value	Mode(s)
All Data	42.29	385.692	95.00	1.0	24
Last 3 Days	46.87	252.259	78.0	17.0	60
Last 1 Day	60.72	25.402	69.00	47.00	66
Last 12 Hours	65.05	2.397	69.00	61.00	66
Last 6 Hours	66.19	0.675	69.00	65.00	66
Last 3 Hours	66.55	0.561	69.00	66.00	66
Last 2 Hours	66.7	0.611	69.00	66.00	66
Last 1 Hour	66.71	0.746	68.00	66.00	66
Last 30 Minutes	67.12	0.783	68.00	66.00	68
Last 10 Minutes	67.67	0.266	68.00	67.00	68

Water current speed is a rather uncommon metric, measured only in a few systems dedicated to sea monitoring. Therefore, only two datasets were analysed, provided by Ports of Spain.

Table 32 Basic analysis of water current speed from REDEXT Golfo de Cádiz

Period	Average	Variance	Max. Value	Min. Value	Mode(s)
All Data	16.5090	126.8770	97.2	0.0	7.0
Last Year	15.7343	157.7060	89.8	0.4	7.4
Last 6 Months	18.9193	224.2414	89.8	0.4	1.6
Last Month	5.4406	36.6922	40.6	0.4	1.2
Last 2 Weeks	3.3819	9.4973	20.7	0.4	1.2
Last Week	3.1113	8.2705	19.1	0.4	1.2
Last 3 Days	3.1901	9.4094	19.1	0.4	1.2
Last Day	2.088	1.3761	5.1	0.8	1.2

Last 12 Hours	1.9923	1.1707	4.3	0.8	1.2
Last 6 Hours	2.5571	1.3928	4.3	0.8	2.3
Last 3 Hours	2.8	0.3600	3.5	2.3	2.3
Last 2 Hours	2.9666	0.3733	3.5	2.3	2.3, 3.1, 3.5
Last Hour	3.3	0.0799	3.5	3.1	3.1, 3.5
Last 30 Minutes	3.5	0.0	3.5	3.5	3.5

Table 33 Basic analysis of water current speed from REDCOSM Puerta Carnero

Period	Average	Variance	Max. Value	Min. Value	Mode(s)
All Data	33.3021	405.0354	122.8	0.3	19.3
Last Year	31.8919	355.5000	121.2	0.3	22.7, 24.6, 25.7
Last 6 Months	26.9920	260.1734	88.4	0.3	13.2, 14.6
Last Month	26.9932	250.8083	83.3	0.3	13.2, 14.4
Last 2 Weeks	30.7505	332.1167	83.3	0.6	30.4
Last Week	27.4621	212.3901	71.1	0.6	29.0
Last 3 Days	30.5288	201.4284	61.5	4.3	45 values (very varied)!
Last Day	37.5318	157.8375	61.5	15.7	22 values (very varied)!
Last 12 Hours	43.7666	129.2096	61.5	21.7	12 values (very varied)!
Last 6 Hours	45.8571	24.6361	51.9	38.5	38.5, 39.6, 46.5, 47.6, 47.7, 49.2, 51.9
Last 3 Hours	45.75	23.8966	49.2	38.5	38.5, 47.6, 47.7, 49.2
Last 2 Hours	48.1666	0.8033	49.2	47.6	47.6, 47.7, 49.2
Last Hour	48.45	1.125	49.2	47.7	47.7, 49.2
Last 30 Minutes	49.2	0.0	49.2	49.2	49.2

As in the case of water current speed, water salinity is measured only in a few systems

focused on sea monitoring. Therefore, we only analysed one dataset provided by Ports of Spain.

Table 34 Basic analysis of water salinity from REDEXT Golfo de Cádiz

Period	Average	Variance	Max. Value	Min. Value	Mode(s)
All Data	36.4358	0.0411	37.4	34.8	36.5
Last Year	36.4319	0.0232	36.7	35.2	36.3
Last 6 Months	36.3879	0.0266	36.7	35.2	36.3
Last Month	36.2387	0.0041	36.3	36.0	36.3
Last 2 Weeks	36.2459	0.0045	36.3	36.1	36.3
Last Week	36.2165	0.0054	36.3	36.1	36.2
Last 3 Days	36.2630	0.0023	36.3	36.2	36.3
Last Day	36.3	0.0	36.3	36.3	36.3
Last 12 Hours	36.3	0.0	36.3	36.3	36.3
Last 6 Hours	36.3	0.0	36.3	36.3	36.3
Last 3 Hours	36.3	0.0	36.3	36.3	36.3
Last 2 Hours	36.3	0.0	36.3	36.3	36.3
Last Hour	36.3	0.0	36.3	36.3	36.3
Last 30 Minutes	36.3	0.0	36.3	36.3	36.3

All the datasets analysed providing information about water temperature are from the datasets provided by Ports of Spain. Since the water acts as a kind of buffer for temperature, we expect the metrics to be more stable than those from air temperature.

Table 35 Basic analysis of water temperature from REDEXT Golfo de Cádiz

Period	Average	Variance	Max. Value	Min. Value	Mode(s)
All Data	19.1080	6.3406	25.7	14.4	16.4
Last Year	19.3746	7.0837	24.5	15.0	15.8
Last 6 Months	17.0508	2.2087	20.6	15.0	15.8
Last Month	16.5299	0.5019	18.2	15.0	15.6

Last 2 Weeks	17.0697	0.1442	18.2	16.4	16.9
Last Week	17.0408	0.1392	17.9	16.4	17.0
Last 3 Days	17.3273	0.0947	17.9	16.8	17.0, 17.4
Last Day	17.656	0.0267	17.9	17.4	17.5
Last 12 Hours	17.7846	0.0114	17.9	17.6	17.7, 17.9
Last 6 Hours	17.7571	0.0128	17.9	17.6	17.7
Last 3 Hours	17.675	0.0024	17.7	17.6	17.7
Last 2 Hours	17.6666	0.0033	17.7	17.6	17.7
Last Hour	17.7	0.0	17.7	17.7	17.7
Last 30 Minutes	17.7	0.0	17.7	17.7	17.7

Table 36 Basic analysis of water temperature from REDCOSM Puerta Carnero

Period	Average	Variance	Max. Value	Min. Value	Mode(s)
All Data	16.0370	5.2164	25.2	11.4	15.0
Last Year	17.4193	6.7398	25.2	11.9	15.0
Last 6 Months	15.3379	1.5121	18.9	11.9	15.0
Last Month	14.2776	0.2263	15.6	13.0	14.1
Last 2 Weeks	14.5519	0.1517	15.6	13.8	14.5
Last Week	14.7052	0.1952	15.6	13.8	15.0
Last 3 Days	14.9906	0.0326	15.5	14.6	15.0
Last Day	14.8521	0.0226	15.1	14.6	14.8
Last 12 Hours	14.7727	0.0081	14.9	14.6	14.8
Last 6 Hours	14.8	0.0033	14.9	14.7	14.8
Last 3 Hours	14.8	0.0	14.8	14.8	14.8
Last 2 Hours	14.8	0.0	14.8	14.8	14.8
Last Hour	14.8	0.0	14.8	14.8	14.8

Last 30 Minutes	14.8	0.0	14.8	14.8	14.8

All the datasets analysed providing information about wind speed come from Ports of Spain. The datasets providing information about this kind of sensor are: REMPOR Dique ExSUR, REMPOR Dique ExNORTE, REMPOR Endesa, REMPOR Dique Abrigo, REMPOR Campamento, REDEXT Golfo de Cádiz and REDCOSM Puerta Carnero.

Table 37 Basic analysis of wind speed from REMPOR Dique ExSUR

Period	Average	Variance	Max. Value	Min. Value	Mode(s)
All Data	4.7380	9.3996	23.6	0.1	1.9
Last Year	4.7456	9.5836	23.6	0.2	2.0
Last 6 Months	5.2763	12.4507	23.6	0.3	2.3
Last Month	5.3247	11.1503	23.1	0.3	1.6, 2.1
Last 2 Weeks	4.5791	7.6450	12.9	0.3	1.7
Last Week	4.6561	7.1708	12.9	0.3	4.7
Last 3 Days	4.7840	9.2459	12.9	0.4	1.4, 4.7
Last Day	6.4368	14.0963	12.9	0.6	2.8
Last 12 Hours	9.1027	9.4053	12.9	2.5	11.1, 12.6
Last 6 Hours	11.7583	0.6830	12.9	10.2	11.1, 12.6
Last 3 Hours	11.8666	0.7317	12.9	10.3	11.1
Last 2 Hours	11.9583	0.7371	12.9	10.3	12.5
Last Hour	12.35	0.3109	12.9	11.3	11.3, 12.3, 12.4, 2.5, 12.7, 12.9
Last 30 Minutes	12.0666	0.4433	12.5	11.3	11.3, 12.4, 12.5

Table 38 Basic analysis of wind speed from REMPOR Dique ExNORTE

Period	Average	Variance	Max. Value	Min. Value	Mode(s)
All Data	4.8767	8.6807	22.0	0.1	3.1
Last Year	4.7418	7.5515	22.0	0.1	3.0

Last 6 Months	5.2718	8.5605	22.0	0.2	3.4
Last Month	5.4689	9.0441	17.1	0.3	2.4
Last 2 Weeks	4.4898	6.6885	12.4	0.3	2.5
Last Week	4.5528	6.0790	12.4	0.3	2.0
Last 3 Days	4.9657	5.5855	9.7	0.3	6.0, 6.4
Last Day	2.5138	2.0981	7.0	0.3	1.8, 2.0, 2.4
Last 12 Hours	2.9791	2.9258	7.0	0.4	2.0, 2.3, 2.4, 2.5
Last 6 Hours	3.4722	5.1220	7.0	0.4	0.4, 1.0, 1.2, 2.5, 4.3, 4.9, 5.3, 6.0, 6.4
Last 3 Hours	5.5777	0.6230	7.0	4.3	4.3, 4.9, 5.3, 6.0, 6.4
Last 2 Hours	5.7833	0.5851	7.0	4.3	6.0
Last Hour	6.35	0.1549	7.0	5.9	5.9, 6.0, 6.3, 6.4, 6.5, 7.0
Last 30 Minutes	6.2333	0.0933	6.5	5.9	5.9, 6.3, 6.5

Table 39 Basic analysis of wind speed from REMPOR Endesa

Period	Average	Variance	Max. Value	Min. Value	Mode(s)
All Data	4.6958	7.3192	21.1	0.1	2.5
Last Year	4.6583	7.3389	21.1	0.2	2.7
Last 6 Months	5.0827	9.3344	21.1	0.2	2.5
Last Month	5.2550	7.4689	16.8	0.3	5.4
Last 2 Weeks	4.7697	5.6111	12.0	0.3	4.6
Last Week	4.8739	4.7699	11.2	0.3	4.6
Last 3 Days	4.8298	4.5546	9.7	0.4	4.6
Last Day	5.6680	1.9981	9.5	2.3	3.8, 4.1, 4.5, 5.5, 6.1, 6.6
Last 12 Hours	5.2819	2.5037	9.5	2.3	3.8

Last 6 Hours	6.0611	2.8104	9.5	2.3	5.8
Last 3 Hours	7.3611	0.9966	9.5	5.7	7.3
Last 2 Hours	7.7416	0.7626	9.5	6.3	7.3
Last Hour	7.8166	1.4536	9.5	6.3	6.3, 6.6, 7.9, 8.0, 8.6, 9.5
Last 30 Minutes	8.6666	0.6433	9.5	7.9	7.9, 8.6, 9.5

Table 40 Basic analysis of wind speed from REMPOR Dique Abrigo

Period	Average	Variance	Max. Value	Min. Value	Mode(s)
All Data	4.1376	7.0786	21.8	0.0	2.3
Last Year	4.6663	9.2344	21.8	0.0	2.6
Last 6 Months	4.5473	7.5791	21.8	0.0	2.6
Last Month	5.5052	8.2310	15.6	0.1	5.3
Last 2 Weeks	5.6529	9.9476	15.6	0.1	6.3
Last Week	5.8385	11.3773	15.4	0.1	7.7
Last 3 Days	3.0673	2.3517	7.5	0.1	2.7
Last Day	2.5	0.9539	5.5	0.2	3.4
Last 12 Hours	2.6291	0.9029	5.5	0.7	1.7
Last 6 Hours	2.3	1.1365	5.5	0.7	1.7
Last 3 Hours	2.8833	1.2014	5.5	1.7	1.9, 2.3, 2.4
Last 2 Hours	2.8916	1.4553	5.5	1.7	2.3, 2.4
Last Hour	2.55	0.147	3.3	2.3	2.3, 2.4
Last 30 Minutes	2.7333	0.2633	3.3	2.3	2.3, 2.6, 3.3

Table 41 Basic analysis of wind speed from REMPOR Campamento

Period	Average	Variance	Max. Value	Min. Value	Mode(s)
All Data	4.9106	7.3937	22.7	0.1	4.5
Last Year	4.9899	7.9090	22.7	0.1	5.0
Last 6 Months	5.5077	10.0523	22.7	0.2	2.9
Last Month	5.4079	8.5105	17.8	0.2	6.9, 7.3
Last 2 Weeks	5.0323	6.0723	12.1	0.2	5.5
Last Week	4.9570	5.1271	10.9	0.5	5.5
Last 3 Days	4.9196	5.5917	10.2	0.5	6.5
Last Day	6.7979	1.1943	10.2	4.4	6.5
Last 12 Hours	6.4583	0.8940	8.9	4.4	6.0, 6.2
Last 6 Hours	6.625	1.1813	8.9	4.4	6.0, 6.2, 7.4
Last 3 Hours	7.1666	1.0305	8.9	4.9	6.2, 6.9, 7.4, 7.5
Last 2 Hours	7.1833	1.4306	8.9	4.9	7.4, 7.5
Last Hour	7.6	2.1199	8.9	4.9	4.9, 7.4, 7.5, 8.2, 8.7, 8.9
Last 30 Minutes	8.6	0.1300	8.9	8.2	8.2, 8.7, 8.9

Table 42 Basic analysis of wind speed from REDEXT Golfo de Cádiz

Period	Average	Variance	Max. Value	Min. Value	Mode(s)
All Data	5.2931	7.8901	19.0	0.0	4.2
Last Year	5.5503	8.2787	16.4	0.2	5.9
Last 6 Months	6.2944	8.9184	16.4	0.2	6.6
Last Month	5.6570	9.6291	14.8	0.2	4.2
Last 2 Weeks	4.6874	5.9256	11.0	0.2	5.4
Last Week	4.5664	4.8377	10.6	0.2	5.4

Last 3 Days	4.9619	5.5649	10.6	0.2	5.4
Last Day	3.5478	2.5544	5.9	1.4	1.6
Last 12 Hours	4.4923	1.7741	5.9	2.1	5.9
Last 6 Hours	3.9000	2.2100	5.9	2.1	4.9
Last 3 Hours	3.025	1.6491	4.9	2.1	2.1, 2.3, 2.8, 4.9
Last 2 Hours	2.4	0.1299	2.8	2.1	2.1, 2.3, 2.8
Last Hour	2.45	0.2449	2.8	2.1	2.1, 2.8
Last 30 Minutes	2.8	0.0	2.8	2.8	2.8

Table 43 Basic analysis of wind speed from REDCOSM Puerta Carnero

Period	Average	Variance	Max. Value	Min. Value	Mode(s)
All Data	5.5735	8.1181	19.7	0.2	3.5, 3.7
Last Year	6.5868	10.3340	18.4	0.5	6.5
Last 6 Months	6.5868	10.3340	18.4	0.5	6.5
Last Month	6.4041	10.9481	18.4	0.5	3.5, 8.7
Last 2 Weeks	6.2273	9.7749	13.0	0.5	8.7
Last Week	6.4104	12.4419	13.0	0.5	1.0
Last 3 Days	6.6140	16.6631	13.0	0.5	1.0
Last Day	11.2173	1.0524	13.0	9.5	11.5, 12.4
Last 12 Hours	11.9727	0.5801	13.0	10.4	12.4
Last 6 Hours	12.4571	0.0795	13.0	12.1	12.4
Last 3 Hours	12.6	0.0799	13.0	12.4	12.4
Last 2 Hours	12.6666	0.0933	13.0	12.4	12.4, 12.6, 13.0
Last Hour	12.7	0.1799	13.0	12.4	12.4, 13.0
Last 30 Minutes	12.4	0.0	12.4	12.4	12.4

APPENDIX B

This appendix contains tables that show some information about the angles calculated in the context of the work described in subsection 5.1, that analyse the outcomes from the homogeneity tests.

The first analysis was carried out with the benchmark datasets, using the data collected from the Intel temperature sensors and the Santander temperature sensors. The first table provides the outcomes from calculating the turning points when bias errors are injected in the data. The number of turning points detected is not high and the size of the angles vary a lot depending on the type of homogeneity test studied.

Table 44 Angles analysis for temperature sensor with bias error

Type of Test	Num. Turning Points	Min. Angle Error	Max. Angle Error	Min. Angle No Error	Max. Angle No Error
SNHT	1-17	6.49	39.44	94.27	179.25
Pettitt	0-19	0.60	9.59	2.97	53.13
Buishand R	1-14	74.63	103.44	123.49	179.42
Buishand U	1-14	74.63	103.44	123.49	179.42
Lanzante	0-19	0.60	9.59	2.97	53.13

The case of drift error is very similar, with not so many turning points and quite different angles depending on the test. When comparing the maximum and minimum angles for the same type of test, they are quite similar.

Table 45 Angles analysis for temperature sensor with drift error

Type of Test	Num. Turning Points	Min. Angle Error	Max. Angle Error	Min. Angle No Error	Max. Angle No Error
SNHT	1-17	10.30	61.69	108.43	178.93
Pettitt	0-47	0.60	5.68	14.88	71.56
Buishand R	1-14	79.52	106.00	123.49	179.42
Buishand U	1-14	79.52	106.00	123.49	179.42
Lanzante	0-47	0.60	5.68	14.88	71.56

The case of malfunction error has been also analysed, showing this time a higher number of turning points but still keeping some similarities in the angles when we compare the

outcomes for the same type of test.

Table 46 Angles analysis for temperature sensor with malfunction error

Type of Test	Num. Turning Points	Min. Angle Error	Max. Angle Error	Min. Angle No Error	Max. Angle No Error
SNHT	3-55	19.00	155.13	106.52	179.99
Pettitt	1-52	0.57	7.91	8.43	71.56
Buishand R	1-52	41.72	152.73	68.68	179.74
Buishand U	1-52	41.72	152.73	68.68	179.74
Lanzante	1-52	0.57	7.91	8.43	71.56

The last analysis with the benchmark datasets was done with the data with random errors injected. In this case, the number of turning points is a bit smaller than in malfunction errors, but higher than with bias and drift errors. And again, when comparing the angles for the same type of test, they are similar.

Table 47 Angles analysis for temperature sensor with random error

Type of Test	Num. Turning Points	Min. Angle Error	Max. Angle Error	Min. Angle No Error	Max. Angle No Error
SNHT	1-40	22.45	161.94	75.56	179.22
Pettitt	1-37	0.57	20.37	7.09	71.56
Buishand R	1-38	58.63	105.54	172.37	179.42
Buishand U	1-38	58.63	105.54	172.37	179.42
Lanzante	1-37	0.57	20.37	7.09	71.56

As it was necessary to complement the results for temperature sensors with the outcomes of analysing other types of sensors, the angles were calculated using one of the datasets provided by Ports of Spain (to be more concrete, from the Golfo de Cádiz dataset). The following tables show the results when analysing the outcomes from SNHT, Pettitt, Buishand R, Buishand U and Lanzante homogeneity tests.

The first type of sensor analysed was atmospheric pressure. It contained only a few points representing outliers and, therefore, the number of turning points was low. Now, when comparing the angles, even when using the same homogeneity test, it is possible to see that the values are not so similar with respect to the ones calculated with temperature.

Table 48 Angles analysis for atmospheric pressure sensor with a few outliers

Type of Test	Num. Turning Points	Min. Angle Error	Max. Angle Error	Min. Angle No Error	Max. Angle No Error
SNHT	3-20	33.97	105.45	60.10	179.59
Pettitt	6-16	0.79	7.48	2.97	37.87
Buishand R	4-15	20.96	90.31	135.46	178.37
Buishand U	4-15	20.96	90.31	135.46	178.37
Lanzante	6-16	0.79	7.48	2.97	37.87

The Arduino dataset (using moisture and humidity) was also used to analyse the angles obtained. In this case, the datasets had a few small errors (outliers).

Table 49 Angles analysis for moisture sensor with a few outliers

Type of Test	Num. Turning Points	Min. Angle Error	Max. Angle Error	Min. Angle No Error	Max. Angle No Error
SNHT	1-9	24.56	58.20	23.03	174.14
Pettitt	1-10	2.29	6.36	2.86	13.84
Buishand R	1-10	77.30	85.07	74.28	142.22
Buishand U	1-10	77.30	85.07	74.28	142.22
Lanzante	1-10	2.29	6.36	2.86	13.84

In the case of the moisture sensor, there are very few turning points and it showed some values that are different from all the sensors analysed before.

Table 50 Angles analysis for humidity sensor with a few outliers

Type of Test	Num. Turning Points	Min. Angle Error	Max. Angle Error	Min. Angle No Error	Max. Angle No Error
SNHT	1-17	43.50	43.50	51.43	175.17
Pettitt	0-10	3.79	3.79	5.02	48.81
Buishand R	1-10	84.48	84.48	98.59	171.84
Buishand U	1-10	84.48	84.48	98.59	171.84
Lanzante	0-10	3.79	3.79	5.02	48.81

When looking at the results from the humidity sensor, it also showed a small number of turning points and it also had differences with the previous cases, with the moisture sensor being the most similar one.

Finally, the luminosity sensor data obtained with the Toolbox application was used to analyse the angles produced. It contains a few outliers and its results show a small number of turning points with angles that are, in general, higher than the ones produced by other sensors when errors are present.

Table 51 Angles analysis for light sensor with a few outliers

Type of Test	Num. Turning Points	Min. Angle Error	Max. Angle Error	Min. Angle No Error	Max. Angle No Error
SNHT	1-10	62.87	99.88	87.75	179.53
Pettitt	3-9	1.36	2.12	1.96	48.01
Buishand R	1-7	143.91	152.01	143.56	167.49
Buishand U	1-7	143.91	152.01	143.56	167.49
Lanzante	3-9	1.36	2.12	1.96	48.01