# APACHE Decision Support System: Sensor Data Intelligence for Cultural Heritage Preventive Conservation

Ca' Foscari University of Venice
Department of Environmental Sciences, Informatics and Statistics

**Ph.D. Degree**
in Computer Science Year 2023

**Graduand** Giacomo Chiarot
**Supervisor** Claudio Silvestri

# Acknowledgments

This research has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 814496.

# Abstract

The majority of the cultural objects in museums stay in storage areas, placed inside different types of open or closed containers. A common problem, especially for small and middle-size museums, is to understand if objects are well conserved, or if they are affected by some agents of deterioration such as, for instance, incorrect temperature or relative humidity. This thesis presents the framework and implementation of the APACHE Decision Support System (DSS) developed under the "Active  intelligent PAckaging materials and display cases as a tool for preventive conservation of Cultural HEritage" (APACHE) project, funded by the Horizon 2020 European program. The APACHE DSS is developed to help cultural institutions to understand the possible threats that may affect a collection and guide them to the selection of the best preventive measure to apply. To achieve this goal, the APACHE DSS records environmental data that come as time series from the sensors installed in object's containers. This data is processed, and if some threats are identified, the APACHE DSS suggests a ranked list of preventive measures that address that threats.

The use of compression techniques can improve the efficiency of sensor data management, streaming, and storage. For this reason, the second part of the thesis focuses on methods for the compression of time series. In addition to analysing the relevant literature in this field, this thesis also presents a novel lossy compression method based on the use of neural networks to learn how to compress time series in a lossy way while preserving the accuracy of classification based on compressed data.

# Contents

# Chapter 1

# Introduction

Collecting objects of the past is important for many reasons, ranging from curiosity to the need of understanding the world around us [10]. In this perspective, whether paintings or craft tools, the role of cultural heritage objects is to be a testimony of past cultures and to make the reconstruction of our history possible. When these objects enter a museum's collection, they should be preserved in the same conditions as they arrive to be displayed, appreciated, and studied. To do so, museums have to reduce all the potential threats that can damage cultural objects both in the short and long term. Reducing this threat doesn't involve direct interaction with single objects: it is an activity that considers all the collection together and, since the museum curators act before objects are damaged, this activity is called preventive conservation [11].

To understand the actual needs of institutions and to take a snapshot of the current situation, a survey was conducted at the beginning of 2019. This survey was posted on the website of a large number of associations and professional organizations related to the preservation of cultural heritage, including UNESCO[1], ICOM CC[2], AIC[3], IIC[4], IGIIC[5], AICRAB[6], FFCR[7], and AICCM[8]. This survey collected 756 responses from archives, galleries, museums, historical houses, and libraries located in 75 countries, mainly Hungary, Italy, Spain, and Brazil, as shown in Figure 1.1.

Analysing the responses to the survey, it can be noticed that especially small and medium-sized museums with few or no resources fail to store objects properly: only 15% of the institutions have a dedicated staff position for preventive conservation. Another interesting result is that it is a common practice to display only a small proportion of their collections at any one time, with the remaining majority of objects kept in storage. Even if objects placed in the display area are well-preserved, the majority of the collection, which is placed in the storage area, experience unsuitable conditions which can cause significant deterioration [8, 14, 64].

The Canadian Conservation Institute[9] proposes a classification of the possible

[1]https://www.unesco.org
[2]https://www.icom-cc.org
[3]https://www.culturalheritage.org
[4]https://www.iiconservation.org
[5]https://www.igiic.org
[6]https://www.aicrab.org
[7]https://www.ffcr.fr
[8]https://aiccm.org.au
[9]https://www.canada.ca/en/conservation-institute.html

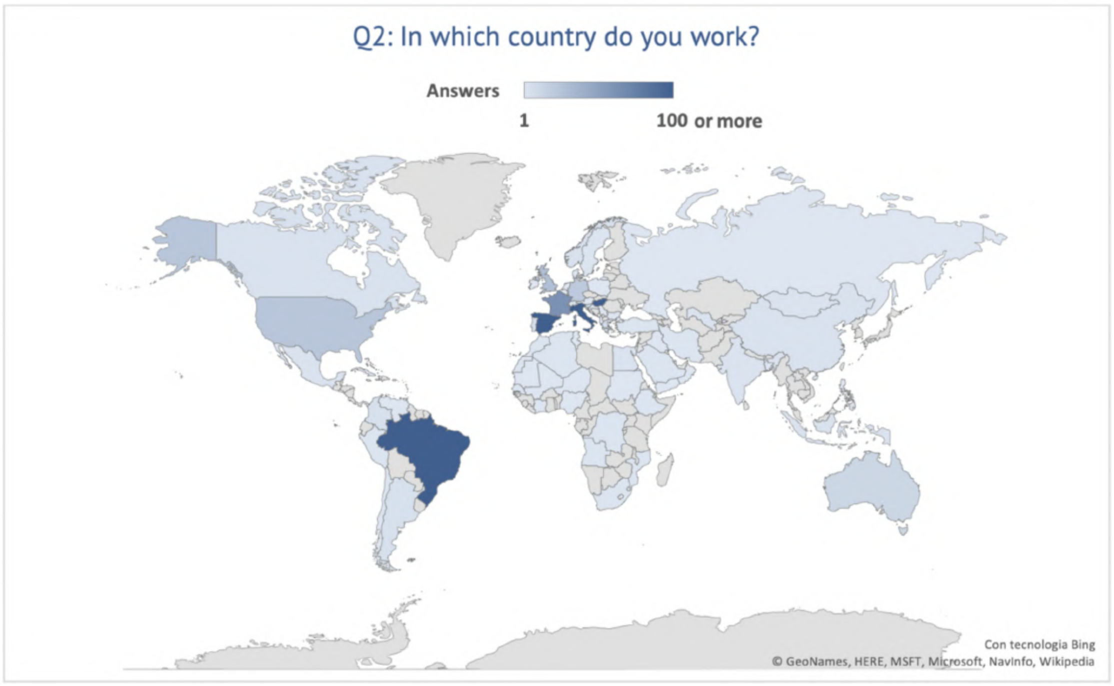

Q2: In which country do you work?
Answers
1
100 or more
Con tecnologia Bing
© GeoNames, HERE, MSFT, Microsoft, NavInfo, Wikipedia

Figure 1.1: Distribution of responses for different countries

threats that can affect cultural heritage objects, identifying 10 different classes or agents of deterioration, which are: physical forces, thieves and vandals, fire, water, pests, pollutants, light, ultraviolet and infrared, incorrect temperature, incorrect relative humidity, and dissociation.

Among all the aforementioned agents of deterioration, the environmental conditions to which objects are exposed are of utter importance in preventive conservation, and, since objects are retained in containers, containers' microclimate is a key factor that needs to be considered [20]. So, temperature, relative humidity, and pollutants have to be monitored, and then some actions have to be taken if values go outside safety ranges.

For this reason, it is a standard practice nowadays to place sensors inside storage or display containers to deeply understand objects' actual conditions. The survey revealed that 62% of the institutions installed sensors to measure mainly temperature, relative humidity, and light. Since the microclimate inside a container is mediated by the container itself, it is also important to study its capacity to reduce internal temperature and relative humidity fluctuations. Besides this, containers can be equipped with materials that are capable to stabilize relative humidity and reduce pollutants. Some of these materials are developed by the institutions involved in the European APACHE project, as they developed conservation techniques that are specific for cultural objects, covering long retention periods. Such techniques are divided into absorbents for pollutants and humidity, and temperature controllers. In the field of generic object conservation, some techniques involving the use of active and intelligent packaging materials have recently been developed, for instance, in the food industry field [12, 67].

Current methods to record exposure conditions, especially for pollutants, encompass Passive Sampling Devices (PSDs). These techniques are expensive, hard to calibrate, and do not deliver analysis in real-time, since data must be collected and processed manually. This was also found by the survey's responses, as only

31% of the institutions use remotely connected sensors to collect data. These downsides make such systems difficult to use and resource-consuming, and consequently, they are not widely adopted. These issues do not only affect storage areas, but also exhibition areas. To make monitoring systems more accessible and affordable for small cultural heritage institutions, the process of collecting and analysing data should be automatic and managed by a proper decisional tool [12]. The results of the survey showed that only 7% of the institution use automatic tools to analyse data, while 45% of them store the collected data without performing any analysis.

This thesis presents a framework and its implementation of the APACHE DSS aiding decision-makers to automatically collect and process monitoring data in real-time, using new generation sensors, and select the best preventive conservation actions to take in case of a threatening situation occurs. The preventive measures proposed by the DSS are meant to be effective and affordable, therefore suitable even for small and medium-sized heritage institutions. Besides this, the system also provides guidelines and best practices for museums without sensors infrastructure. The implementation of the framework into the APACHE Decision Support System is already online and publicly accessible at apachedss.dais.unive.it.

The preventive conservation needs of cultural heritage objects are strongly determined by their specific material composition and environmental context – the nature of which is highly varied if not unique. For this reason, the APACHE DSS cannot be generic enough to cover all possible scenarios and is not intended to be a tool able to perform a complete risk assessment [7, 51, 69]. The proposed solution considers only a selection of specific materials, deterioration agents, and preventive measures to cover as many types of objects as possible with the highest affordable level of detail. Moreover, it is scaled to the levels of information provided by museum curators, ranging from a more general level, which considers collections as a whole, to a more detailed one considering single containers.

Since in real application, scenarios collections have many sensors, and they produce new value regularly with small intervals between measurements, the amount of data to be stored is relevant. Considering the case studies presented in Chapter 4, sensors produce an average of 20 KB for each month and each sensor, considering only temperature and relative humidity measures. The institutions involved in the survey collect on average between 10,000 and 100,000 objects. This means that if institutions want to monitor all the objects inside their collection, sensors would produce an average of 12Gb per year. For this reason, it is necessary to consider compression techniques to store and then retrieve this data efficiently.

In addition to the compression task, compressed data have to be classified to understand if the environmental condition of a collection is dangerous or not. Current methods require the compressed data to be first reconstructed to be then classified. The downside of this method is speed: in the first place, reconstructing a long series of measurements requires time that could be avoided, then performing classification on a decompressed representation is slower than performing the classification directly on the compressed representation. So, the compression method has not only to achieve good reconstructions, but also to be used as input for a classifier.

## 1.1 Research problems

### 1.1.1 The APACHE framework

To achieve good results in the monitoring of the collection's environments, a framework and methodology have to be developed to enable strategies for effective preventive conservation of cultural heritage. The framework has to address both institutions which have and do not have environmental monitoring systems or professional conservators. To address institutions without an active monitoring system, the framework has to be divided into static and dynamic parts:

- **static assessment**: gives general guidelines and useful information for curators to handle objects in a collection;
- **dynamic assessment**: connects damage functions, sensors data, and preventive measures, returning alerts in case of treating situations are detected and a ranked list of the best preventive measures to apply.

### 1.1.2 Time series compression

The APACHE DSS collects environmental data from different sensors. The values measured by sensors are bound with time since each measurement is taken with a specific timestamp. This type of data is called time series.

Besides the APACHE DSS, time series are relevant in several contexts, and the Internet of Things (IoT) ecosystem is among the most pervasive ones. IoT devices, indeed, can be found in different applications, ranging from health care (smart wearables) to industrial ones (smart grids) [4], producing a large amount of time-series data. For instance, a single Boeing 787 fly can produce about half a terabyte of data from sensors [62]. In those scenarios, characterized by high data rates and volumes, time series compression techniques are a sensible choice to increase the efficiency of collection, storage, and analysis of such data. In particular, the need to include in the analysis both information related to the recent and the history of the data stream leads to considering data compression as a solution to optimize space without losing the most important information.

To understand the best techniques to apply, a survey of the state-of-the-art is conducted [13]. The algorithms that are considered can deal with the continuous growth of time series over time and are suitable for generic domains (as in the different applications in the IoT). Furthermore, these algorithms take advantage of the peculiarities of time series produced by sensors, such as:

- **Redundancy**: some segments of a time series can frequently appear inside the same or other related time series;
- **Approximability**: sensors in some cases produce time series that can be approximated by functions;
- **Predictability**: some time series can be predictable, for example using deep neural network techniques.

In the case of the APACHE DSS, the time series produced by the measurement of environmental data follow all the above-mentioned features.

### 1.1.3 Time series classification

Besides the compression problem, it is important to valorize the collected data. To do this, the compressed data have to be analysed and classified to extract new information, behind the pure graphical visualization. Classification algorithms are generally trained on uncompressed training sets or a dimensional reduction derived from the original data. This means that, once time series data are compressed, they must be decompressed to be passed to a classification model. This operation has a very high computational cost, in particular for time series that grow in time. In this case, new measurements update the compressed representation and, since they may change the classification of the whole time series, to update the classification result, the time series have to be decompressed at every newly added measure.

To solve this problem, a compression model has to find a compressed representation that can be used both to train a classifier and to reconstruct the original time series. Existing time series compression models don't fit this requisite: despite they achieve good results in the compression task, the compressed representation is not suitable to train a classification algorithm.

# Chapter 2

# The APACHE framework

The APACHE framework aims to help decision-makers to automatically collect and process monitoring data in real-time, using new-generation sensors developed by the APACHE project, and select the best preventive conservation actions to take in case of a threatening situation. Both the framework and methodology are implemented within the APACHE DSS.

The preventive conservation needs of cultural heritage objects are strongly determined by their specific material composition and environmental context – the nature of which is highly varied, if not unique. For this reason, the APACHE DSS framework cannot be generic enough to cover all possible scenarios and is not intended to be a tool able to perform a complete risk assessment [7, 51, 68]. The project considers only a selection of specific materials, deterioration agents, and preventive measures to cover as many types of objects as possible with the highest affordable level of detail. Moreover, the project is scaled to the levels of information provided by museum curators, ranging from a more general level, which considers collections as a whole, to a more detailed one considering single containers.

The framework is divided into two main parts: the preventive measures open repository and the DSS. The repository collects the preventive measures created within the APACHE projects, in addition to the ones selected from the literature, with their characterizing information. This repository is the source of information of the DSS. The DSS is then divided into two tiers, giving general information and suggestions tailored for a specific collection about the most suitable preventive measures to apply.

The proposed framework and methodology described in this chapter resulted from the collaboration among UNIVE[1], Antonio Mirabile[2], ICCROM[3], and the user group of the APACHE project, composed of the Italian Ministry of Culture (MiC)[4], Centre Pompidou[5], National Museum of Slovenia[6], Hungarian National Museum[7], Fondazione Scienza e Tecnica Museum[8], Peggy Guggenheim Collection Venice[9].

[1]https://www.unive.it
[2]http://antoniomirabile.com
[3]https://www.iccrom.org
[4]https://sabapchpe.beniculturali.it
[5]https://www.centrepompidou.fr/en
[6]https://www.nms.si/si/
[7]https://mnm.hu/en
[8]http://www.fstfirenze.it/?lang=en
[9]http://www.fstfirenze.it/?lang=en

## 2.1 Preventive measures open repository

The preventive measures open repository encompasses preventive conservation techniques developed and studied in the APACHE project and in other international projects, as well as selected from the literature. Preventive measures in the repository are collected with their characterizing information and properties, including costs, durability, effectiveness, and expertise needed to apply that preventive measure.

This repository is the source of information of the APACHE DSS for the selection of the most suitable preventive measures for a specific collection and environmental conditions.

The repository is open, so all the preventive measures are freely accessible. Moreover, users can contribute by adding new comments, suggestions, and proposals of new preventive measures to be added. All the suggestions and proposals are moderated by a selected group of experts before becoming publicly visible. The repository is versioned: after many proposals are accepted, moderators can upgrade the version of the repository to include all the newly accepted proposals. The repository keeps track of the changes, and users can navigate through versions.

Users can interact with a graphical interface. In addition to this, the repository also encompasses a REST API. The API can be used without access tokens and can be accessed by other software tools in read mode to get preventive measures information.

The repository is designed to be flexible and extendable to additional knowledge bases besides the preventive measures, as materials and agents of deterioration. This is accomplished by moderated editing, where users can propose new elements, and designated moderators can accept or decline such additions.

### 2.1.1 Logical organization

Preventive measures are organized using a tree structure as represented in Figure 2.1. The root of the tree is the project level which represents a container for all information related to a topic, in this case, the APACHE project. Projects are composed of a list of knowledgebases that represent categories of elements inside the topic, the only knowledgebase currently in the repository is the one related to preventive measures. Preventive measures' elements contained in the knowledgebase are grouped into 8 categories: lightning systems, shielding, exposition, containers, absorbents, monitoring, environmental control, and storage.

This structure is designed to be extended and modified, new knowledgebases, categories, and elements can be added and edited.

### 2.1.2 Preventive measures characterization

Preventive measures are characterized by the following parameters:

- **Effectiveness**: categorical field that describes the effectiveness of the preventive measure with respect to the other preventive measures covering the same agent of deterioration using three classes (Low, Medium, High);
- **Comments on effectiveness**: adds some information about the effectiveness;

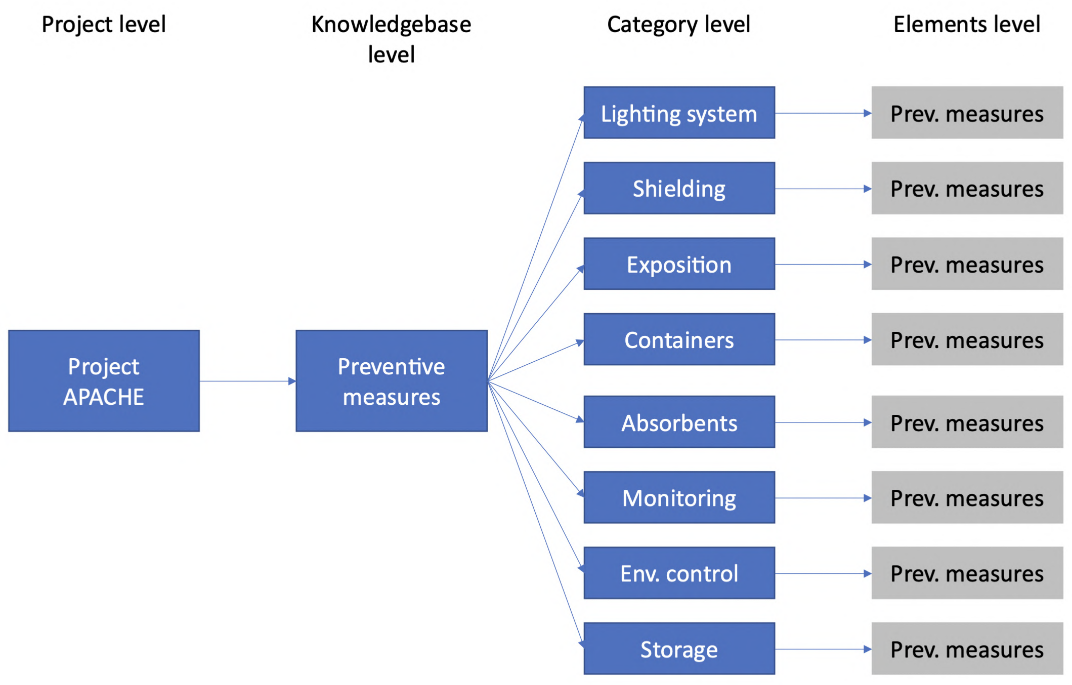


Figure 2.1: Open repository scheman

- **Durability**: categorical field that describes the durability of the preventive measure with respect to the other preventive measures covering the same agent of deterioration using three classes (Low, Medium, High)
- **Comments on durability**: adds some information about the durability
- **Average cost**: categorical field describing the cost of the preventive measure using three classes (Low, Medium, and High). The average cost must be indented in relation to other preventive measures that aim to solve the same problem.
- **Comments on cost**: adds some information about the average cost.
- **Level of expertise**: categorical field describing the level of expertise needed to apply the preventive measure using three classes (Low, Medium, High). Low is assigned to a preventive measure that can be installed with no or low specific training (e.g., covering windows with UV filters), medium if some training or help from professionals (e.g., electrician) is needed, high if only experts can apply the preventive measure (i.e., conservator professionals are needed).
- **Comments on expertise**: adds some information about the expertise level.

## 2.2 The APACHE DSS

The APACHE DSS is designed for both institutions with professional conservation staff and interconnected environmental monitoring infrastructures, as well as institutions that do not have dedicated conservation staff or appropriate sensors. This distinction is made because the solutions suggested by the framework are technical and can be applied only by conservation professionals. Also, the active monitoring and alert system need a system of interconnected sensors to obtain ongoing information regarding the microclimate inside containers. To this end, the APACHE DSS framework is developed into two tiers, one dedicated to institutions that are not equipped with a monitoring system, which provides general advice, and one dedicated to institutions equipped with a monitoring system and conservation staff, which provides more detailed information.

It is worth pointing out that the materials, tools, and solutions developed within the APACHE project cover the preservation of materials and physical structures of collection items that are tangible, movable, and indoor. To maintain consistency with the APACHE project proposal, out of the primary ten agents of deterioration1 the APACHE DSS took into consideration only four: Temperature, Relative Humidity, Light, and Pollutants.

### 2.2.1 Tier 1

Tier 1 is developed for museums that do not have dedicated conservation staff or a connected sensor infrastructure. It aims to give general conservation information to the curators based solely on the objects' materials and agents of deterioration that may affect a collection.

Access to this information is public and there is no need for registration. Nevertheless, conservators can log in with an account and insert information about their collections to find tailored information quickly.

#### Structure

This tier is structured as a wiki in which guidelines, and information about materials and agents of deterioration can be found. The wiki is organized into two sections: materials and agents of deterioration.

Information is related only to materials and agents of deterioration that are covered by the APACHE project. Concerning the material section, they are: paper, oil painting, acrylic paint, glass, ceramics, basketry, ivory, herbaria, stone and plaster, skin and taxidermy, wood, photographic materials, textiles, and metals. For the agents of deterioration section: temperature, light, ultraviolet and infrared, pollutants, and relative humidity

The materials section shows the list of the materials that are covered by the project and, for each material, the related page follows the same structure:

- **list of objects**: shows the most common cultural heritage objects that are composed mainly of that material.

- **relative humidity**: describes the damages caused by incorrect relative humidity. In addition to the textual description, the section presents a table showing optimal and acceptable ranges, and maximum fluctuations for 24h. Values are expressed in percentages.

- **temperature**: describes the damages caused by incorrect temperature and presents a table showing optimal and acceptable ranges and maximum fluctuations for 24h. Values are expressed in Celsius degrees.

- **Light, ultraviolet and infrared**: describes the damages caused by light and presents a table showing the maximum values for visible light and ultraviolet light expressed in lux.
- **pollutants**: describes the damages caused by pollutants in contact with the object's materials. Different materials have different pollutants involved in the degradation process. This section also presents a table showing how pollutants affect different subcategories of the same material.
- **references**: lists the references of the over-mentioned information, in addition to more resources from which users can learn more detailed information.

Similarly, the agents of deterioration section shows the list of agents of deterioration covered by the project, and for each of them, the related page follows the same structure:

- **general introduction**: gives a general introduction about the possible threats caused by the agent of deterioration.
- **guidelines**: presents a table showing general optimal and acceptable ranges, and maximum fluctuations that could be taken into consideration for most of the object's materials. Moreover, it adds some comments explaining the motivations behind these limits.
- **monitoring**: explains the best technique to measure the environmental conditions related to one agent of deterioration.
- **controlling**: explains the best technique to control the environmental conditions related to one agent of deterioration.
- **references**: lists the references of the over-mentioned information, in addition to more resources from which users can learn more detailed information.

A glossary is also included for users unfamiliar with specific technical terms, and to provide consistent definitions alongside the tool.

### 2.2.2 Tier 2

Tier 2 has been developed for institutions with dedicated professional conservators and a connected infrastructure of monitoring sensors. The access to sensor data and collection information is limited only to data owners through an authentication system. To implement a higher level of security, museums might decide to install the proposed system on a private intranet server.

Tier 2 provides dynamic monitoring of the objects' environment together with suggestions regarding preventive measures taking into consideration the environmental conditions, objects' composition, and possible threats. To achieve this level of detail, which is strongly dependent on the material composition of a particular collection and its conditions of exposure, users must provide information about their institution and collections, such as locations, containers, and materials. In addition to this, museums must be using environmental sensor systems that regularly send the requisite data to the APACHE DSS.

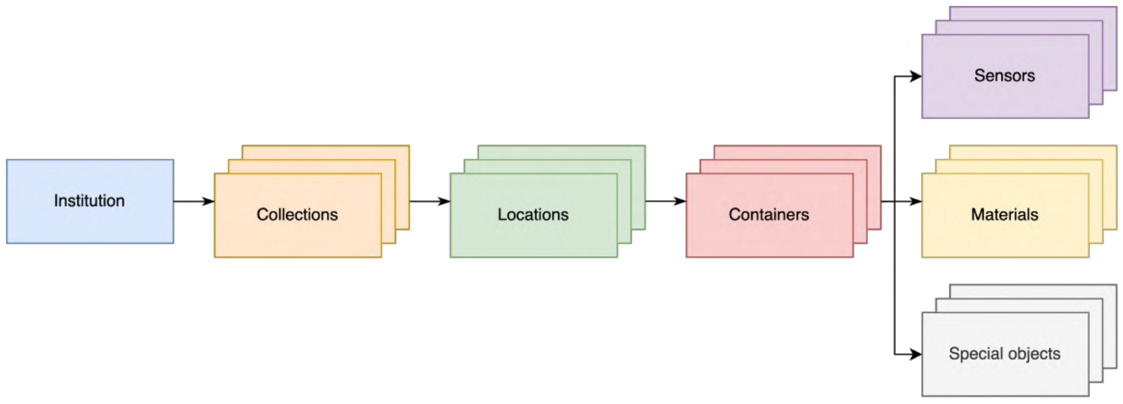


Figure 2.2: Schema representing how information related to institutions are structured

### Institution's information

Institution's information is organized with a tree representation composed of 5 layers, as shown in Figure 2.2:

- **Institution**: it's the institution registered to the APACHE DSS. They can be any type of organization, such as museums, foundations, or individuals.
- **Collections**: each institution can create many collections, which are generally independent one from the others;
- **Locations**: each collection can have many locations. Locations can be storages, storages with a consultation room (archives), or exhibition spaces.
- **Containers**: each location can have many containers in which objects are placed. If objects are not placed inside a container, the room itself is considered the container of that group of objects. The types of containers considered in the APACHE project include metal drawers, wooden boxes, metal bookshelves, cardboard boxes, large plexiglass display cases, plexiglass boxes, wood creates, historical cabinets, and rooms.
- **Materials**: inside one container, it's possible to find many objects, composed of different materials. So, in each container, the APACHE DSS stores a list of materials. The material composition is determined at the container level and considers a preset list of materials, including acrylics, ceramics, glass, herbaria, leather, metal, paint, paper, photographic paper, plastics, PVC, stones, textile, wood, plaster, and oil painting.
- **Sensors**: having the possibility to install different sensors inside the same container (for instance, for measuring different environmental properties), the APACHE DSS can associate many sensors to the same containers.
- **Special objects**: inside one container, it is possible to add some special objects. This information is used by damage functions presented in the next subsection, as they need details that are specific to one object (e.g., the damage function addressing paper needs the pH information of one particular object).

The main aim of the APACHE DSS Tier 2 framework is to classify the status of each container in an ordered scale of alert conditions, ranging from "no threat" to "conservation measures to be considered". The selection of the appropriate alert level is based on sensor data and models, as well as characterization information related to the materials stored in the assessed container. If threatening conditions of any level are identified, the framework suggests a ranking of the most suitable preventive measures for consideration to address the issue. The framework is implemented in a single software solution, and modelling data are an integral part of the APACHE DSS.

### Tier 2 composition

Tier 2 is divided into two parts, one based on a static characterization of the containers, and the other on sensor measurements. These two parts give two different ranked lists of preventive measures: the first suggests which preventive conservation measures could be applied in a specific context, while the other suggests how to maintain the specified conditions, given the applied conservation measures.

Regardless of its parts, the logical APACHE DSS framework can be modelled by several entities which fall into the following categories:

- **dynamic data sources**: e.g. sensors installed in containers that produce real-time raw data by measuring environmental parameters inside containers;
- **static data sources**: e.g. databases containing preventive techniques data, damage functions, and information on materials composing e.g. objects and containers;
- **computational components**: mathematical functions used to analyse raw data.

The logical framework of the APACHE DSS is presented in Figure 2.3. In this schema the different categories are represented with different colours (blue for repositories, green for computational components, yellow for results, and orange for relationships maintainers), while links are represented with arrows: the output of a component becomes the input of others that it connects to.

### Static conservation planning

The static conservation planning part of the framework is aimed at scheduling preventive conservation actions regardless of real-time sensor readings. The decisional logic of this part of the framework is mainly based on containers' characterization data, determined by the materials which are contained therein. The material information module stores and manages information about materials stored in the assessed container as specified by the museum's curator. To correctly structure such information, the module makes use of a material ontology[10] developed in collaboration with Fraunhofer[11], which is based on the Elementary Multiperspective Material Ontology (EMMO)[12], a structured knowledge base that collects information on material properties. This ontology is then used to assess materials and their characteristics. The full schema is presented in Figure 2.4.

[10]https://github.com/emmo-repo/domain-cultural-heritage-preservation/tree/master
[11]https://www.fraunhofer.de
[12]https://github.com/emmo-repo

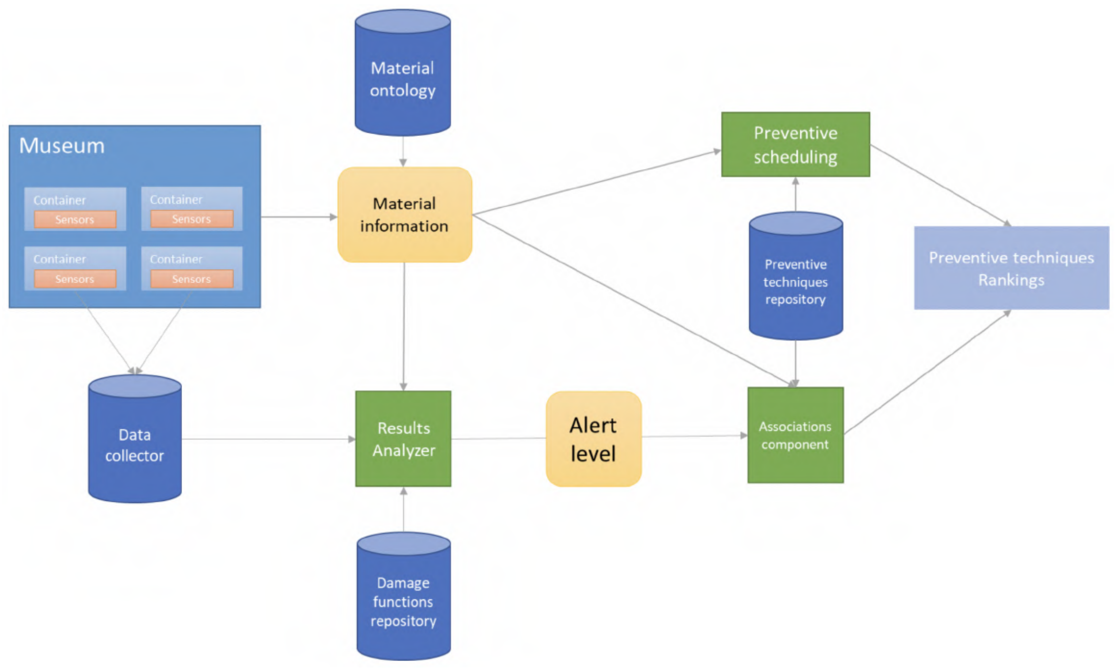


Figure 2.3: Full representation schema for the APACHE DSS Tier 2 framework

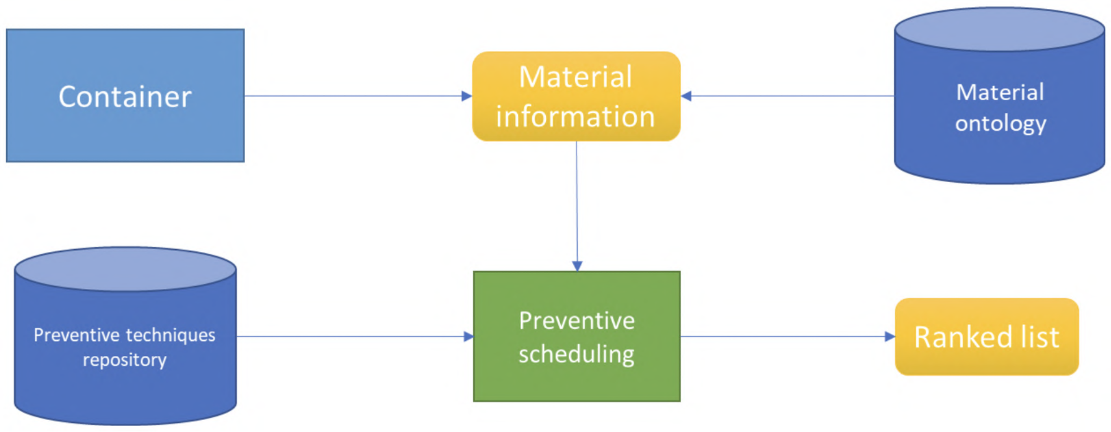


Figure 2.4: Static conservation scheduling schema

The preventive scheduling component gathers information from the materials' information module and the preventive techniques' repository to find the best preventive conservation technique concerning all materials in a particular container. The preventive techniques' repository stores, for each known preventive conservation technique, characterizing parameters such as cost, duration, and suitability score for each material, which are integrated into the so-called fitting index in Equation 2.1.

$$Idx = E + D - C - X + 4 \tag{2.1}$$

Where E is the effectiveness value, D, is the durability value, C the cost value, and X is the expertise value. Since these values are categorical, 0 is associated with Low, 1 with Medium, and 2 with High. A constant is added for having the lowest score starting from 0.

This information gives as an output a ranked list of preventive conservation techniques. The user can then select the preferred ones, taking into account how objects are already stored/displayed.

As a container might contain several materials (indeed this is likely to be the general case), the preventive scheduling component must consider the characterization parameters of all materials present. For this reason, Multi-Criteria Decision Analysis (MCDA) [22] techniques are used. MCDA is a branch of operational research that explicitly evaluates multiple conflicting criteria in decision-making.

The proposed methodology works at the container level and considers:

- All the different materials contained;
- All possible threats;
- All available preventive measures;

Preventive measures candidates are selected by initially filtering out those not suitable for any of the involved materials, then for each threat, the list of remaining candidates is prioritized by taking into account both the list of materials inside a container and the characteristics of each candidate (effectiveness and durability for a certain material), this is obtained by the application of the Equation 2.2:

$$I_{x,c} = \frac{\left(\sum_{m \in M_c} E_{x,m} \cdot D_{x,m}\right) \cdot (1 - C_x - X_x)}{|M_c|} \tag{2.2}$$

Where:

- $I_{x,c}$ is the fitting index of candidate preventive measure $x$ for container $c$;
- $M_c$ is the list of materials inside container $c$, in which elements are materials $m$;
- $E_{x,m}$ is the effectiveness of candidate preventive measure $x$ with respect to the material $m$;
- $D_{x,m}$ is the durability of the candidate preventive measure $x$ with respect to the material $m$;
- $C_x$ is the cost of the candidate preventive measure $x$;
- $X_x$ is the expertise level of the candidate preventive measure $x$.

All the utilized data (i.e. $E$, $D$, $C$, and $X$) are in the [0,1] domain as well as the resulting fitting index $I$, used to rank preventive measures (higher $I$ means higher priority in the ranking).

As anticipated, the result of the preventive scheduling component is a ranked list of suitable preventive techniques. Since the information on materials remains the same over time (unless changes are made to the container contents), the result of this component is fixed and can be precomputed.

## Active damage detection

This part of the framework gives a ranked list of preventive conservation techniques based on materials information and damage alerts. Damage alerts are based on the integration of data related to materials inside the container, degradation parameters measured by sensors, and damage functions.

The estimated potential threat level alert of the materials inside a container derives from the measure of environmental conditions by specific sensors. Each container is equipped with a set of sensors that continuously measure degradation parameters, like temperature, humidity, and pollutants. The full schema is reported in Figure 2.5.

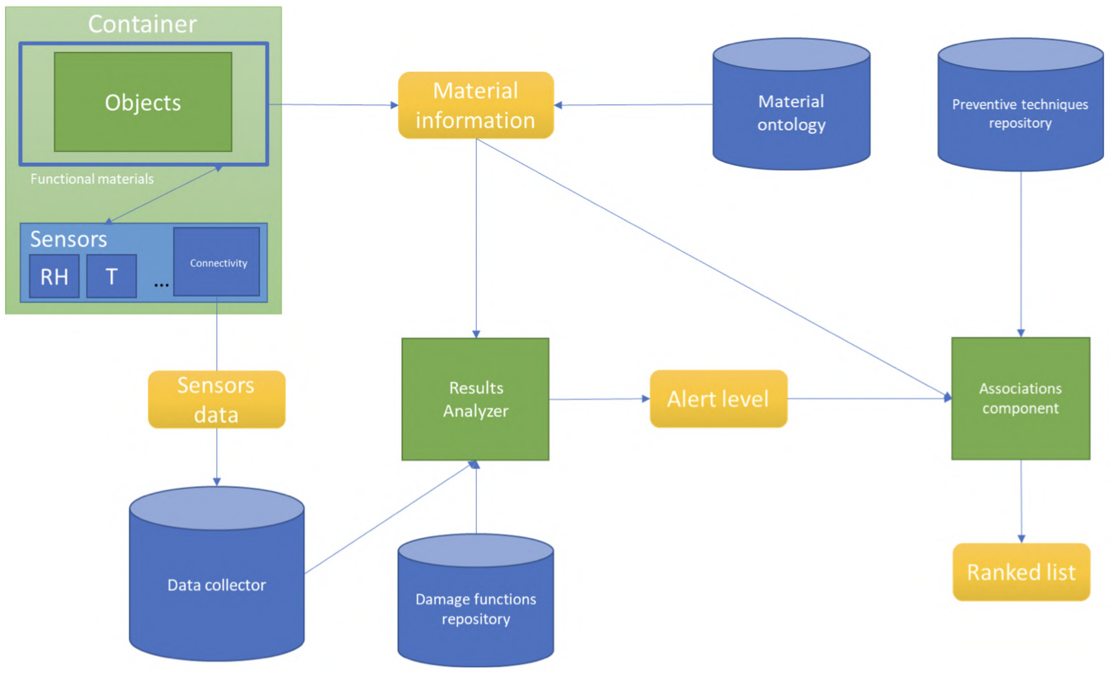


Figure 2.5: Active damage detection schema

The APACHE DSS sensor data integration is a generic process that does not require institutions to purchase specific sensors to be installed in containers, therefore there is no need to install new sensors if the existing ones already provide the required measurements and are already installed. However, new sensors are developed by the APACHE partners, for which the data acquisition process is fully integrated into the framework, instead of uploading data manually.

To assess the container's objects' potential threat level, a repository of damage functions is used. This repository stores mathematical functions fitted to degradation parameter data to enable predictions regarding the analysed container's status. These functions typically determine the effects of degradation parameters (e.g. temperature, relative humidity, and pollutants) on a selected material and determine thresholds that, in the long term, can be considered safe for the conservation of the objects inside a specific container. The APACHE DSS framework can handle two different types of thresholds to inform whether it is sufficient to apply a preventive measure to ensure the objects' ongoing conservation or if the damage risk to objects is such that following proper assessment by a trained conservator remedial actions might also be needed, as shown in Figure 2.6. The framework, however, does not provide any recommendations regarding remedial conservation actions, since such actions require proper assessment of the object's condition and context by a trained conservator.

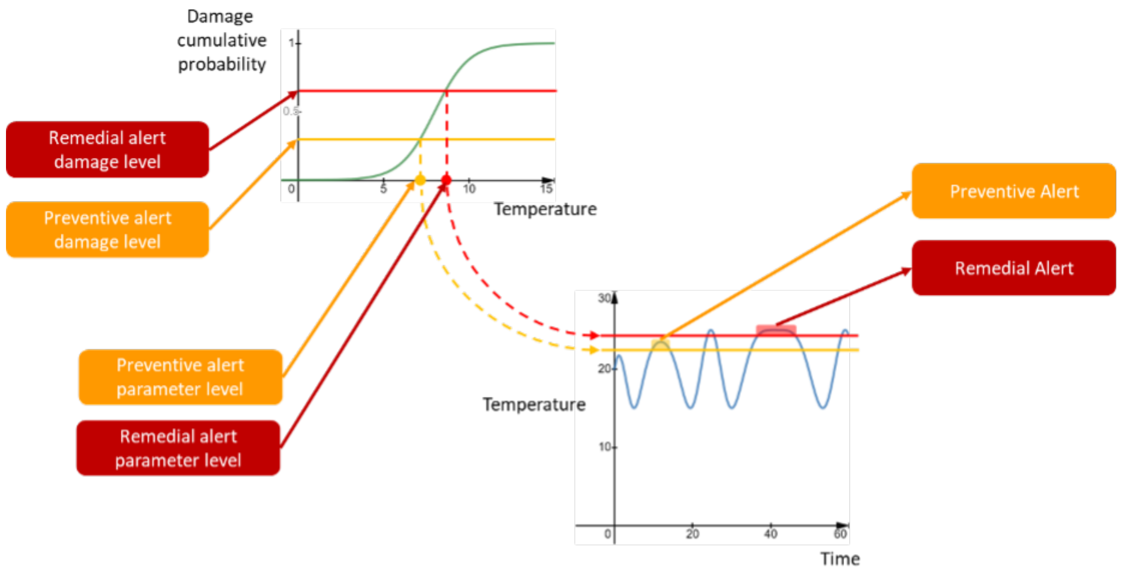

Damage cumulative probability
Remedial alert damage level
Preventive alert damage level
Temperature
Preventive Alert
Remedial Alert
Preventive alert parameter level
Remedial alert parameter level
Temperature
Time

Figure 2.6: Example of damage function (top-left expressed as cumulative density function) and alert detection (bottom-right expressed as time-dependent function) for temperature

The damage functions repository stores the damage functions addressing some materials from the ones covered by the project. These functions are already published and were provided by the project partners: paper [78], acrylics [90], and PVC [57]. These functions can be applied only to special objects defined inside one container. If containers have no special objects, the thresholds for environmental conditions defined for each material in Tier 1 as guidelines are applied.

The appropriate damage function is selected from the damage functions repository based on the material information. The output of these functions is a real value in the interval [0, 1] indicating the damage probability of the objects in relation to the level of the specific degradation parameter. Each damage function should include at least one threshold related to the "preventive conservation measure to be considered" alert, but might also include more thresholds, e.g. "remedial conservation measure to be considered". The aforementioned damage threshold corresponds to a value that is used to fire an alert whenever it is exceeded by the degradation parameter.

The alert status, alongside materials' information, is used to find the most suitable conservation measures from the preventive conservation technique's repository[13]. These measures are sorted from the most suitable to the less.

The logic that gives the ranked list of suitable conservation techniques is contained in the associations component, which computes the fitting index, described in eq. 2. This component is connected with the materials information, alert level, and preventive conservation techniques' repository; the resulting preventive measures ranking is obtained by integrating this information through the use of Multi-Criteria Decision Analysis (MCDA). MCDA is used to solve decision and planning problems that involve multiple criteria, which are, in this case, materials information and alert level. The function to calculate the fitting index of each

[13]https://apacherepository.dais.unive.it

preventive measure is shown in Equation 2.3

$$I_{x,c,e} = \frac{\left(\sum_{m \in \{m \in M_c : E_{x,m,e} > A_{m,e}\}} E_{x,m,e} \cdot D_{x,m,e}\right) \cdot (1 - C_x - X_x)}{|M_c|} \tag{2.3}$$

Where:

- $I_{x,c,e}$ is the fitting index of candidate preventive measure $x$ for container $c$ being threatened by environmental condition $e$;
- $M_c$ is the list of materials inside container $c$, which elements are materials $m$;
- $A_{m,e}$ is the alert level returned by the damage function related to material $m$ threatened by environmental condition $e$;
- $E_{x,m,e}$ is the effectiveness of candidate preventive measure $x$ in respect to the material $m$ and threatening environmental condition $e$;
- $D_{x,m,e}$ is the durability of candidate preventive measure $x$ in respect to the material $m$ and threatening environmental condition $e$;
- $C_x$ is the cost of candidate preventive measure $x$;
- $X_x$ is the expertise level of candidate preventive measure $x$;

#### Preventive measures ranking

The Static conservation planning and Active damage detection parts, despite giving different results, are developed in the same unique APACHE DSS framework. This choice is made both because they share some elements, like the materials information module, and because results will be displayed to the museum's decision-makers, as complementary information in the same user interface.

The ranking of suggested preventive techniques, both in the case of static conservation planning and active damage detection, is based on the following criteria:

- **market price**: evaluates how much applying a certain technique may cost.
- **expected durability**: evaluates how long a certain preventive conservation technique will last. This value is dependent on the container type (e.g., an absorbent material last more time if placed inside an insulated container like a closed box, than on an open shelf).
- **effectiveness**: evaluates how recommended it is to apply a certain preventive measure to address a specific threatening environmental condition for the material, taking into account the information provided by experts.
- **expertise level**: indicated the level of expertise needed to apply a certain preventive measure.

## 2.3 Conclusions

This chapter presented the framework and methodology to enable strategies for effective preventive conservation decision-making for cultural heritage. The framework is divided into two tiers. The first tier addresses institutions that do not have environmental monitoring systems or professional conservators, and offers general guidelines for different materials and environmental threats. The second tier presents two functionalities: static conservation planning and active damage threat detection. The former functionality helps professional conservators to develop a preventive conservation plan even if their institution lacks sensors, the latter functionality suggests a dynamic selection of preventive measures for consideration based on input environmental measurements, damage functions, alarms, and collection material composition.

Since preventive measures and damage functions are developed specifically for a certain material and threat, the list of materials supported by the framework is limited. Despite this, the APACHE DSS framework has been developed to give the possibility for future inclusion of new supported materials and preventive measures to be useful for a wider range of cultural heritage collections. To reach this goal, the preventive measure and damage functions repositories are kept open, so they can be updated over time, allowing the addition of new supported materials.

# Chapter 3

# The APACHE DSS implementation

The APACHE DSS framework is implemented into a single software solution with a user-friendly interface (i.e., a web application). The web application allows users to access all the composing sections of the APACHE DSS described in Chapter 2, as the characterization of their collections, the status of objects inside containers, and get suggestions on the most appropriate preventive measure to apply to the different collection's containers. The web application is a reactive MEAN stack-like application, which could run in a closed intranet as well as in the open internet.

Besides the graphical interface design, the implementation of the APACHE DSS includes the backend services:

- **sensor's data collector**: the APACHE DSS needs to receive environmental measures from the wireless sensors developed by project partners, as well as commercial sensors already used by museums. The collector integrates both wireless and offline sensors.

- **damage functions repository**: the damage functions are implemented as models and stored in the damage functions repository.

- **APIs**: some services, as the preventive measures open repository, use APIs to allow communication between the APACHE DSS and other software programs.

The development of the decision support tool was guided through consultation and feedback from the cultural institutions included as partners within the project. Besides the feedbacks received from the partners, the web application was presented during 3 public dissemination events held in Paris, Turin, and Budapest. The participants were mostly curators and appreciated the simplicity and usefulness of the web application. Their comments particularly appreciated the possibility of monitoring the entire collection, analysing environmental data, and consulting conservation guidelines in one place with little effort.

This chapter describes the strategies adopted to implement the APACHE DSS framework and the software features of the resulting software.

## 3.1 The preventive measures open repository

The preventive measures open repository is implemented as a new project inside the Terminology Harmonizer tool[1]. This tool ensures the repository to be open, as all the preventive measures are freely accessible and users can contribute by adding new comments, suggestions, and proposals of new preventive measures to be added. All the suggestions and proposals are moderated by a selected group of experts before becoming publicly visible. In addition, this tool offers a user-friendly graphical user interface and a rest API which can be accessed by other software tools in read mode. Moreover, the Terminology Harmonizer tool allows future extension of possible knowledgebases, in addition to the preventive measures one.

The open repository is currently online at apacherepository.dais.unive.it

### 3.1.1 Login and account

Even if the repository is open and users can access it without an account, registration is needed to interact adding comments or proposals. Figure 3.1 shows the interface for the login page.

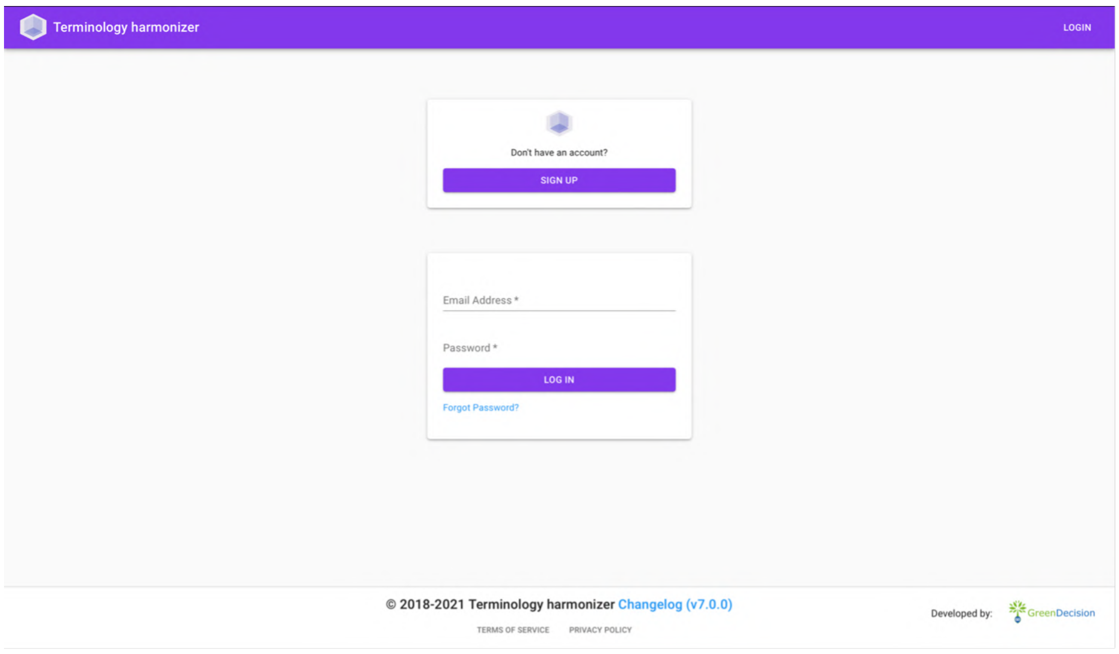


Figure 3.1: Open repository login page

To avoid malicious contents to be added, accounts can be assigned the following roles:

- **Editor**: this is the standard role that is assigned to new users. Editors can view all the preventive measures collected in the repository, propose new preventive measures, categories, and add comments to existing preventive measures;

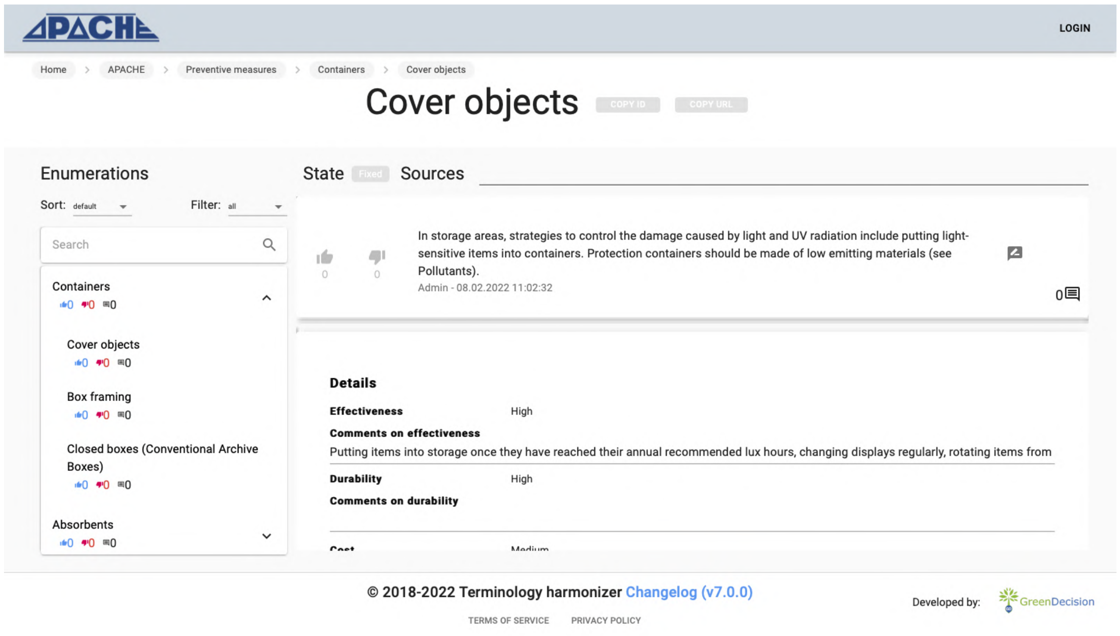
[1]https://terminology-harmonizer.greendecision.eu

- **Moderator**: this role is assigned to conservation experts charged to review the suggestions proposed by editors. They have the same permissions as editors, and, in addition, they can accept or discard the proposals;

- **Admin**: this role is assigned to users charged to manage users and roles. They have the same permissions as the editors, but they cannot moderate proposals.

### 3.1.2 Preventive measures exploration

Figure 3.2 shows how the preventive measures exploration page is designed: in the left bar, users can scroll between categories. Each category can be expanded, revealing the preventive measures inside that category. Once the user selects one preventive measure, in the content area, all the details are revealed.



In storage areas, strategies to control the damage caused by light and UV radiation include putting light-sensitive items into containers. Protection containers should be made of low emitting materials (see Pollutants).
Admin - 08.02.2022 11:02:32

| | |
|---|---|
| **Effectiveness** | High |
| **Comments on effectiveness** | |
| Putting items into storage once they have reached their annual recommended lux hours, changing displays regularly, rotating items from | |
| **Durability** | High |
| **Comments on durability** | |
| Cost | Medium |

© 2018-2022 Terminology harmonizer Changelog (v7.0.0)
TERMS OF SERVICE PRIVACY POLICY
Developed by: GreenDecision

Figure 3.2: Preventive measures open repository

#### Preventive measures

By expanding a category, users can access the preventive measures of that category and show the information related to a specific preventive measure (Figure 3.2). The title of the page is the name of the preventive measure, while the first box shows a brief description. All the additional details are shown in the secondary box, including:

- **Effectiveness**: categorical field that describes the effectiveness of the preventive measure with respect to the other preventive measures covering the same agent of deterioration using three classes (Low, Medium, High).

- **Comments on effectiveness**: adds some information about the effectiveness.

- **Durability**: categorical field that describes the durability of the preventive measure with respect to the other preventive measures covering the same agent of deterioration using three classes (Low, Medium, High).
- **Comments on durability**: adds some information about the durability.
- **Average cost**: categorical field describing the cost of the preventive measure using three classes (Low, Medium, and High). The average cost must be indented in relation to other preventive measures that aim to solve the same problem.
- **Comments on cost**: adds some information about the average cost.
- **Level of expertise**: categorical field describing the level of expertise needed to apply the preventive measure using three classes (Low, Medium, High). Low is given to a preventive measure that can be installed with no or low specific training (e.g., covering windows with UV filters), medium if some training or help from professionals (e.g., electrician) is needed, high if only experts can apply the preventive measure (i.e., conservator professionals are needed).
- **Comments on expertise**: adds some information about the expertise level.

### 3.1.3 Adding new proposals

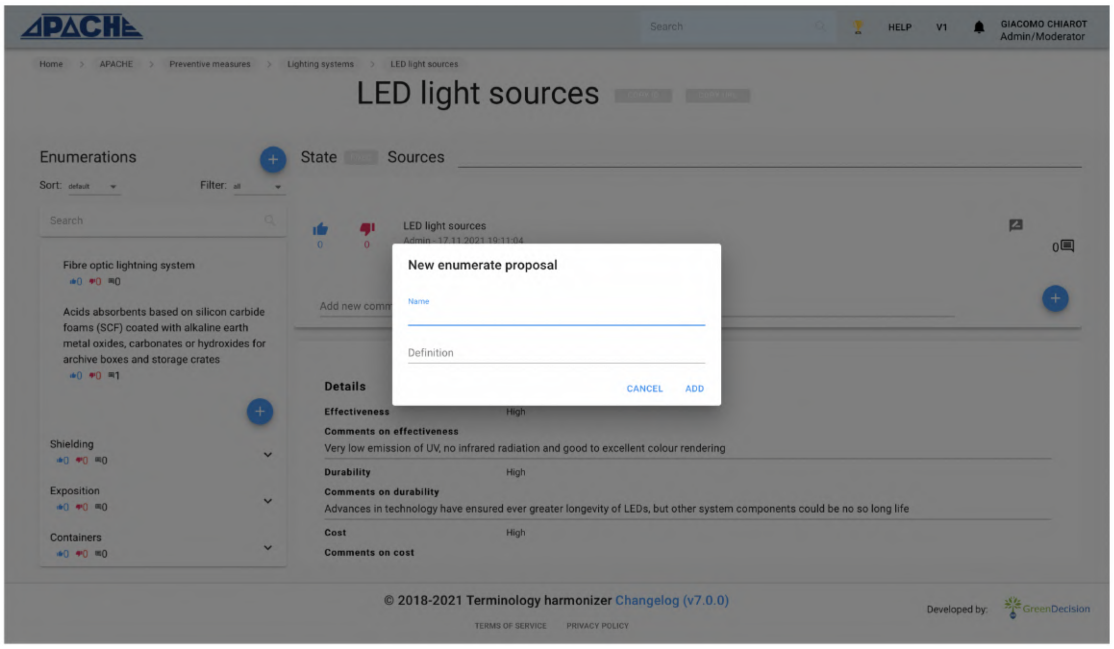
Editors can propose new categories and preventive measures by clicking the plus button. This button can be found at the top of the category list to add a new category or at the end of the preventive measures list of a specified category to add a new element. The button opens a dialog as shown in Figure 3.3, asking for the name of the new category or preventive measure and a brief description. The additional details for a preventive measure can be inserted later, after the proposal is created.

Such new items are marked as “proposals” (or prop). Proposals are visible to everyone for commenting, but only moderators can promote them to the state of “candidates” or discard them to the state of “discarded”. Candidates will then be included in new versions of the knowledgebase and be marked as “new”, while discarded proposals will not make their way to the new version. The full list of proposal states includes:

- **Fixed**: elements already there since the previous version;
- **New**: elements added in the current version of the repository;
- **Updated**: elements modified in the current version of the repository;
- **Proposal**: elements proposed by editors and still to be moderated;
- **Candidate**: proposals accepted by moderators, which will make their way into the next version of the repository.
- **Candidate with modifications**: proposals accepted by moderators which will make their way into the next version of the repository, but with some modifications (name, definition, or category);

- **Discarded**: proposals discarded by moderators.

The different states are highlighted with coloured tags, as shown in Figure 3.4.



Fibre optic lightning system

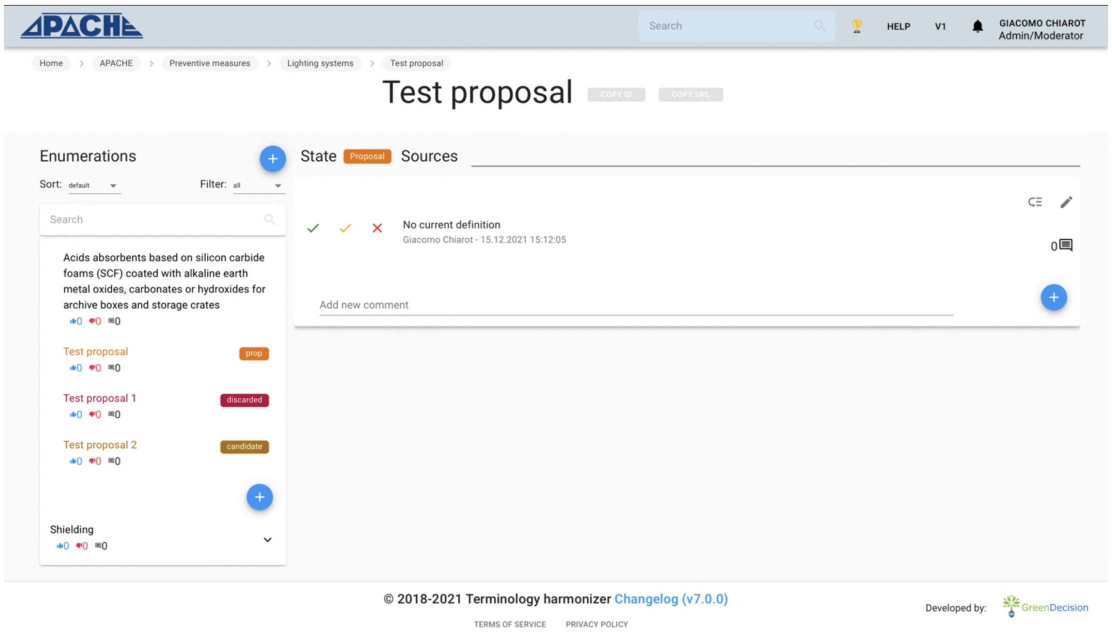
Acids absorbents based on silicon carbide foams (SCF) coated with alkaline earth metal oxides, carbonates or hydroxides for archive boxes and storage crates

LED light sources
Admin - 17.11.2021 19:11:04

Very low emission of UV, no infrared radiation and good to excellent colour rendering

Advances in technology have ensured ever greater longevity of LEDs, but other system components could be no so long life

Figure 3.3: New proposal dialog



Acids absorbents based on silicon carbide foams (SCF) coated with alkaline earth metal oxides, carbonates or hydroxides for archive boxes and storage crates

No current definition
Giacomo Chiarot - 15.12.2021 15:12:05

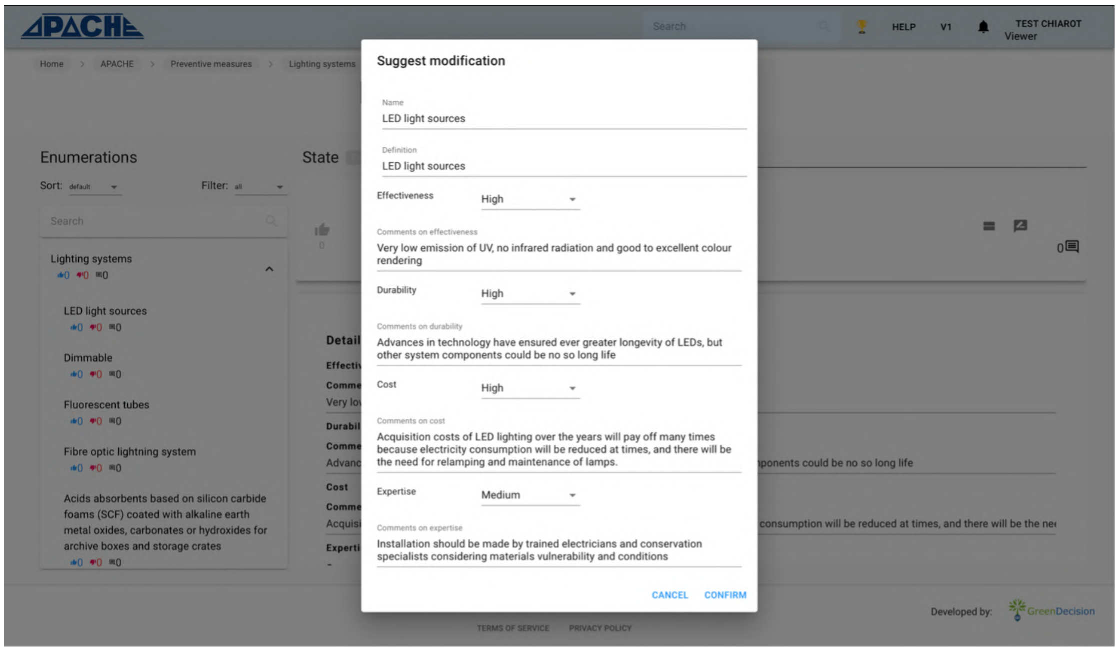
Figure 3.4: Example of different labels for proposals states

### 3.1.4 Interact with the preventive measures

#### Editor interactions

Editors can interact with preventive measures by:

- **Adding a new comment**: after selecting a preventive measure, editors can add a new comment by clicking the plus button inside the definition box. Once a comment is added, the author can delete or edit it.
- **Upvoting or downvoting**: after selecting a preventive measure, editors can upvote or downvote (blue and red thumbs) it by clicking the buttons inside the definition box. This is used mainly for proposals, to vote if they should be accepted or not.
- **Add suggestions**: editors can suggest a modification to a preventive measure definition or details. By clicking the suggest modification button, a modal is opened asking for the proposals. An example is shown in Figure 3.5. After a modification is suggested, moderators can view the suggestion and accept or reject it.


Suggest modification

Name: LED light sources

Definition: LED light sources

Effectiveness: High

Comments on effectiveness: Very low emission of UV, no infrared radiation and good to excellent colour rendering

Durability: High

Comments on durability: Advances in technology have ensured ever greater longevity of LEDs, but other system components could be no so long life

Cost: High

Comments on cost: Acquisition costs of LED lighting over the years will pay off many times because electricity consumption will be reduced at times, and there will be the need for relamping and maintenance of lamps.

Expertise: Medium

Comments on expertise: Installation should be made by trained electricians and conservation specialists considering materials vulnerability and conditions

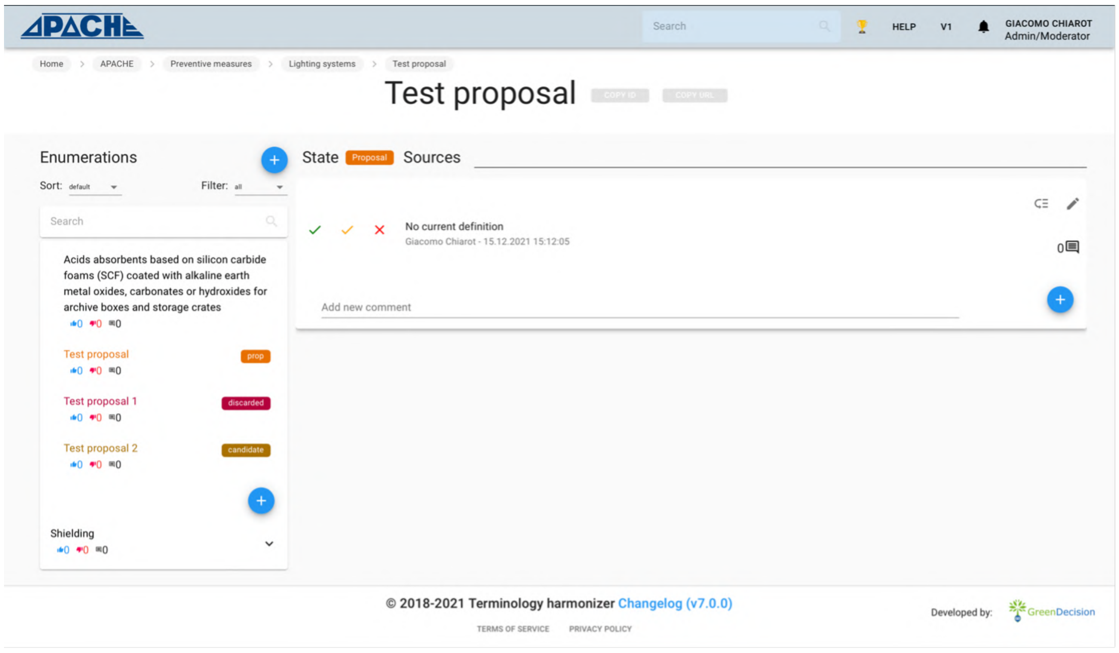
CANCEL
CONFIRM
HELP
V1
TEST CHIAROT
Viewer
Enumerations
Lighting systems
LED light sources
Dimmable
Fluorescent tubes
Fibre optic lightning system
Acids absorbents based on silicon carbide foams (SCF) coated with alkaline earth metal oxides, carbonates or hydroxides for archive boxes and storage crates
Developed by: GreenDecision
TERMS OF SERVICE
PRIVACY POLICY

Figure 3.5: Example of a suggest modification dialog

#### Moderator interactions

Moderators can interact with proposals, suggestions, and versions. Figure 3.6 shows the view of a moderator for a proposal. From the definition box, moderators can accept, accept with modifications, and discard a proposal. If the proposal is accepted, it becomes a candidate for the next version, otherwise, it is tagged as discarded.

To moderate suggestions, moderators can open the suggestion list using the suggestions list button, as shown in Figure 3.7. From this list, moderators can accept or discard suggestions. If they are accepted, they become visible to everyone.



Test proposal

No current definition
Giacomo Chiarot - 15.12.2021 15:12:05

Acids absorbents based on silicon carbide foams (SCF) coated with alkaline earth metal oxides, carbonates or hydroxides for archive boxes and storage crates

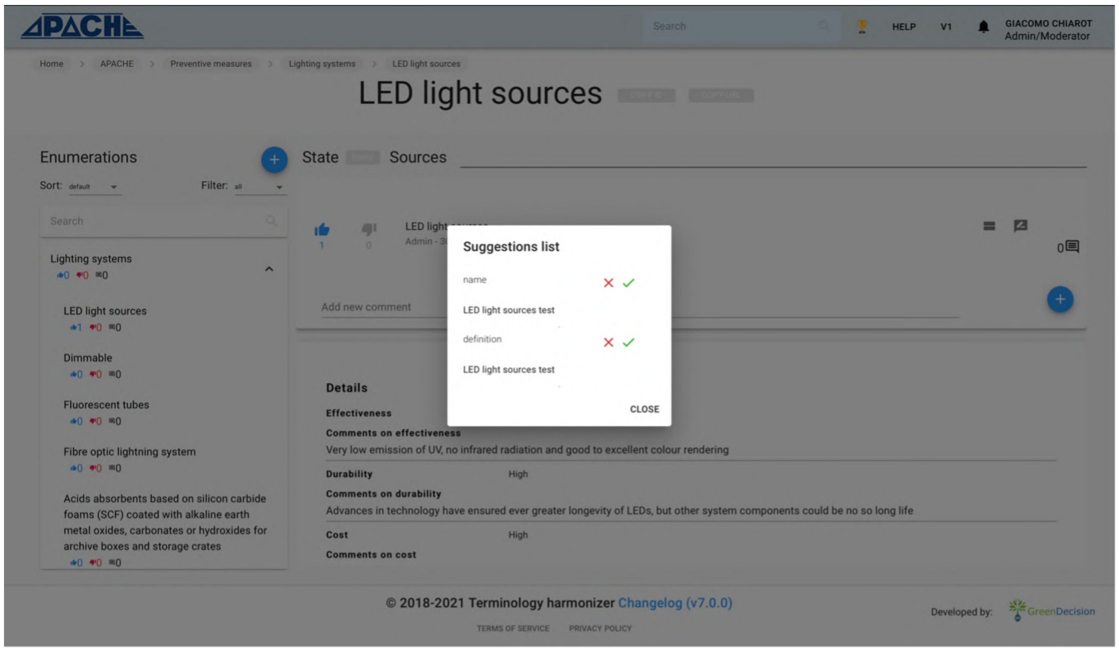
Test proposal

Test proposal 1

Test proposal 2

Shielding

Figure 3.6: Example of a proposal to be moderated



LED light sources

Lighting systems

LED light sources

Dimmable

Fluorescent tubes

Fibre optic lightning system

Acids absorbents based on silicon carbide foams (SCF) coated with alkaline earth metal oxides, carbonates or hydroxides for archive boxes and storage crates

Suggestions list

name

LED light sources test

definition

LED light sources test

Details

Effectiveness

Comments on effectiveness

Very low emission of UV, no infrared radiation and good to excellent colour rendering

Durability High

Comments on durability

Advances in technology have ensured ever greater longevity of LEDs, but other system components could be no so long life

Cost High

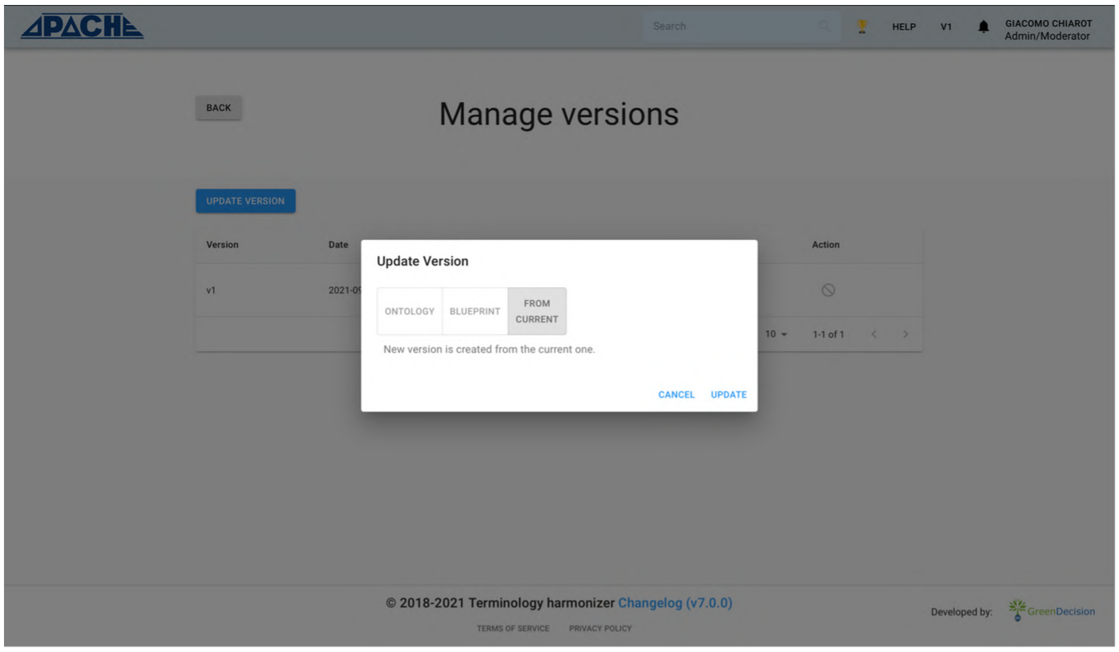
Comments on cost

Figure 3.7: Example of a list of suggestions to be moderated

From the update version page, Figure 3.8, moderators can update the current version. By doing this, candidates are marked as new, while new preventive measures are marked as fixed.

### 3.1.5 Open repository rest API

The open repository rest API is used to access the full list of preventive measures without needing an account and without the graphical user interface. The API has one endpoint, used for retrieving the full list of preventive measures, the preven-



New version is created from the current one.

© 2018-2021 Terminology harmonizer Changelog (v7.0.0)

Figure 3.8: Update open repository version dialog

tive measures contained in a particular category, or the information of a specific preventive measure. The response is a JSON object. The API takes as input the following parameters in the request body:

- **Category**: the preventive measures' category (e.g., Absorbents);
- **Name**: the name of the selected preventive measure (e.g., Graphene-based aerogels). If this parameter is specified, also the category has to be specified.

If no input parameter is passed in the body, the API returns the full list of preventive measures, otherwise the list of preventive measures in the specified category, or only the information related to the specified preventive measure.

The endpoint is terminology-harmonizer.greendecision.eu/api/APACHE/preventive-measures. The response is a JSON object containing an array of objects. Each object in the list has all the attributes described in Section 2.1, and the name of the category of that preventive measure. If the name and the category are passed as input, the output is still a list with only one element. If the category or the name doesn't exist, the response is an empty list.

## 3.2 The APACHE DSS

The APACHE DSS is developed as one single software solution accessible with a web application at apachedss.dais.unive.it.

### 3.2.1 Backend and frontend

The framework used to develop the web application is Meteor[2]. It's an open-source platform developed on top of Node and React.

[2]https://www.meteor.com

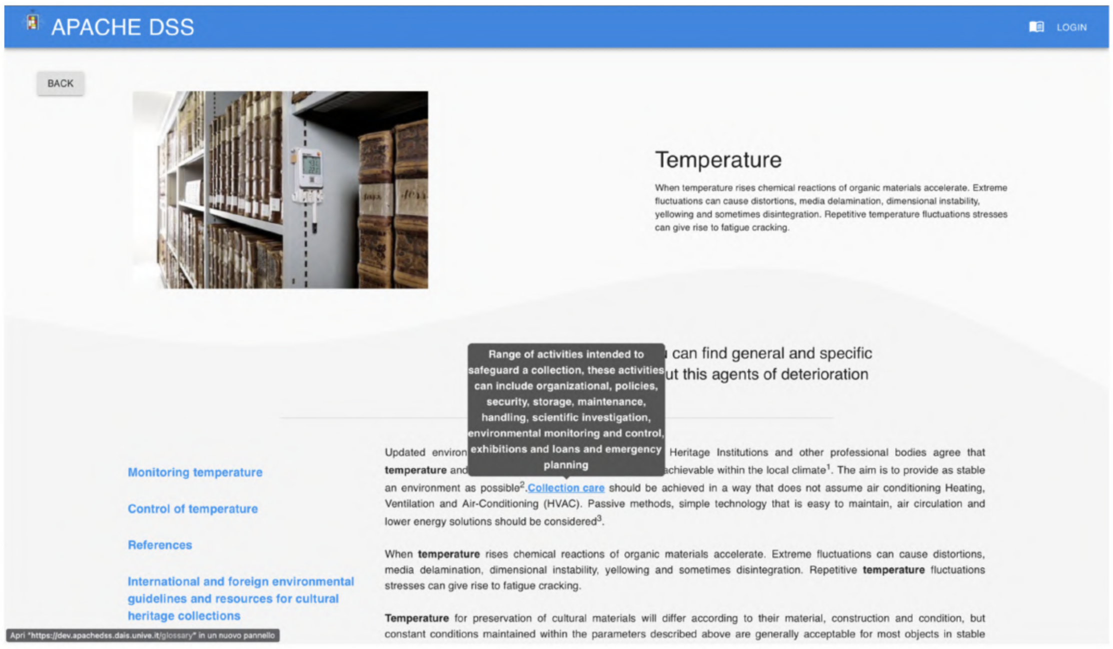
The backend is run by Node and empowered by Meteor. It is composed of a NoSQL database, MongoDB[3], users management, and by an environment used to run functions in the background.

React manages the front-end graphical interface of the web application. It's connected with the backend through socket connections to be updated in real-time if some changes happen to the database.

The software is encapsulated with a Docker container[4] and hosted by UNIVE.

### 3.2.2 Tier 1

Tier 1 is structured as a wiki with static content, and it's accessible without registering an account. It is divided into two sections dedicated to materials and agents of deterioration guidelines. The textual descriptions are integrated with a glossary: words that can be found in the glossary are marked in bold and placing the mouse over them shows their definition, as shown in Figure 3.9. The full glossary can be accessed by clicking the glossary "book" button in the navigation bar.



APACHE DSS
LOGIN
BACK
Monitoring temperature
Control of temperature
References
International and foreign environmental guidelines and resources for cultural heritage collections
Apri "https://dev.apachedss.dais.unive.it/glossary" in un nuovo pannello

Temperature

When temperature rises chemical reactions of organic materials accelerate. Extreme fluctuations can cause distortions, media delamination, dimensional instability, yellowing and sometimes disintegration. Repetitive temperature fluctuations stresses can give rise to fatigue cracking.

Range of activities intended to safeguard a collection, these activities can include organizational, policies, security, storage, maintenance, handling, scientific investigation, environmental monitoring and control, exhibitions and loans and emergency planning

can find general and specific ut this agents of deterioration

Updated environ Heritage Institutions and other professional bodies agree that **temperature** and achievable within the local climate[1]. The aim is to provide as stable an environment as possible[2].Collection care should be achieved in a way that does not assume air conditioning Heating, Ventilation and Air-Conditioning (HVAC). Passive methods, simple technology that is easy to maintain, air circulation and lower energy solutions should be considered[3].

When **temperature** rises chemical reactions of organic materials accelerate. Extreme fluctuations can cause distortions, media delamination, dimensional instability, yellowing and sometimes disintegration. Repetitive **temperature** fluctuations stresses can give rise to fatigue cracking.

**Temperature** for preservation of cultural materials will differ according to their material, construction and condition, but constant conditions maintained within the parameters described above are generally acceptable for most objects in stable

Figure 3.9: Example of the temperature guideline with an opened definition

### 3.2.3 Tier 2

Tier 2 allows registered users to set up their collection, monitor the environmental conditions, and consult the preventive measures that are suggested for their collections.

It is organized into different sections starting from the list of user collections. Once a collection is selected, a dashboard is presented while places, containers, and sensors included in the collection are also accessible. If some alerts are fired, users

[3]https://www.mongodb.com
[4]https://www.docker.com

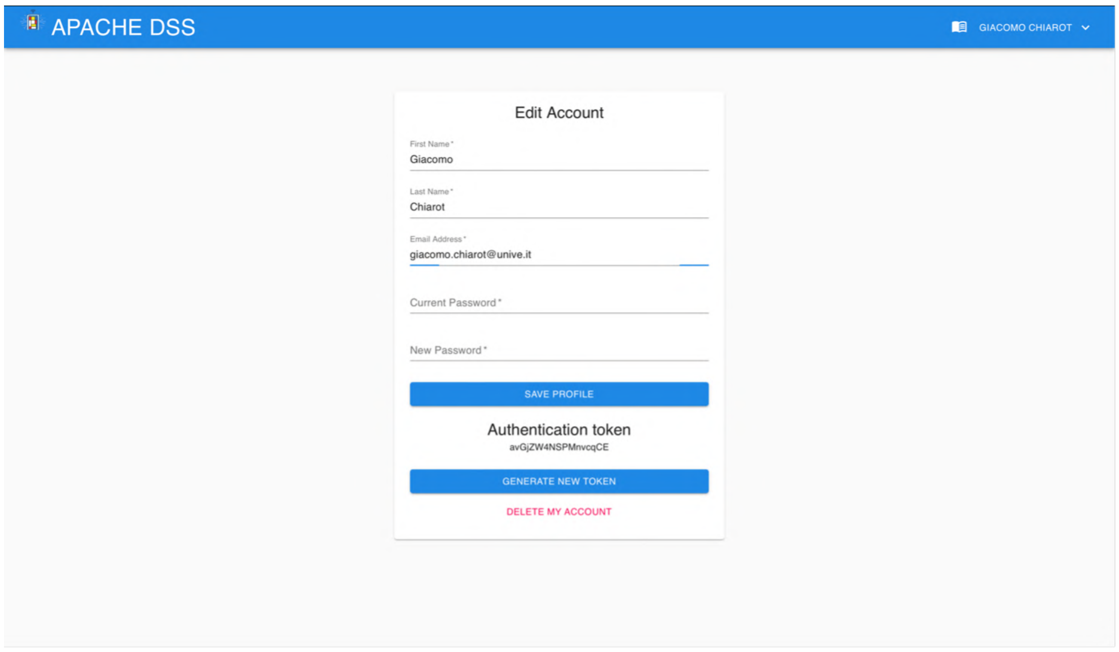
can jump directly to the preventive measures section to see the related suggestions. From the sensors section, users can monitor the environmental conditions.

The following subsections provide a more detailed description of all the sections composing the APACHE DSS, with example screenshots.

### Accounts

To access Tier 2 users must log in with their account or create a new one. The only information required to create a new account is first and last name, a valid email, and a password.

After logging in, users can view and edit the account information, and update their password. From the same section, users can view their authentication token and generate a new one. This token is requested to access collection information and send sensor data through the APACHE DSS rest APIs. An example of the account page is shown in Figure 3.10.



Edit Account

First Name *
Giacomo

Last Name *
Chiarot

Email Address *
giacomo.chiarot@unive.it

Current Password *

New Password *

Authentication token
avGjZW4NSPMnvcqCE

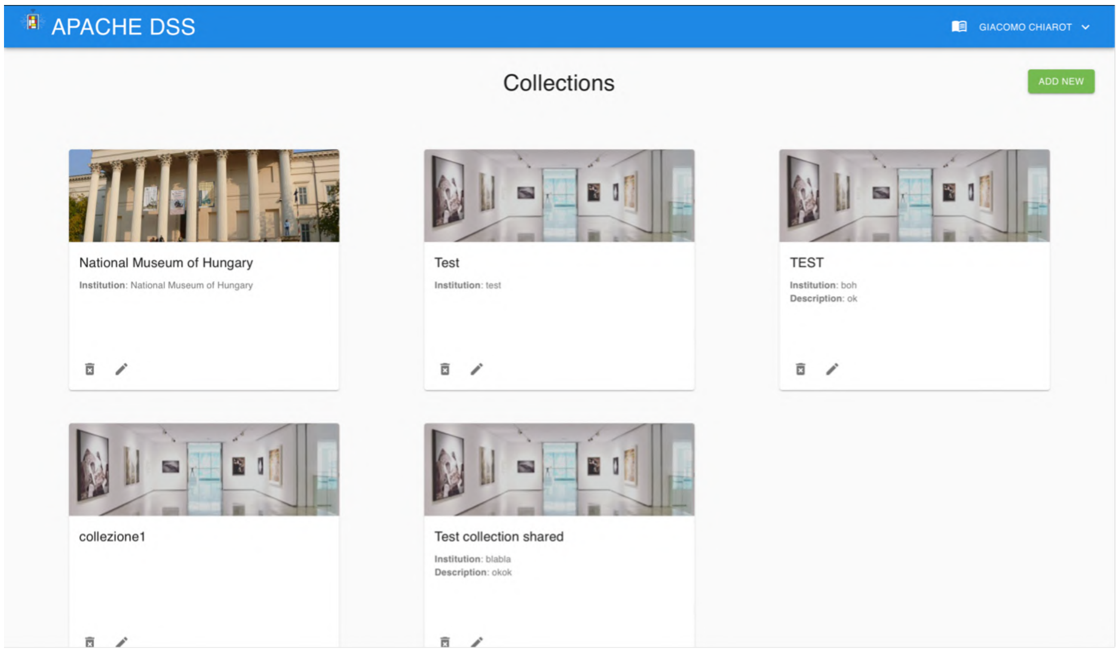
Figure 3.10: Example of the account page

### Collections

The collections section is the first page shown after entering Tier 2. It shows all the collections created or shared with the logged user. For each collection, users can set the name of the collection, the name of the institution, a cover image, and a brief description. Collections can also be removed and updated. Figure 3.11 shows an example of the collections page.

### Dashboard

The dashboard gives quick information about a selected collection. Alerts related to environmental conditions which might be fired by the APACHE DSS are shown at the top. Each alert shows the threat detected by the APACHE DSS, the container in which the threat is detected (on the left), and the name of the sensor that



National Museum of Hungary
Institution: National Museum of Hungary

Test
Institution: test

TEST
Institution: boh
Description: ok

collezione1

Test collection shared
Institution: blabla
Description: okok

Figure 3.11: Example of the collections page

measured dangerous environmental conditions (on the right). By clicking on the container name, the APACHE DSS opens the suggested preventive measures for that container and particular threat, while clicking the sensor's name opens the sensor's page, showing the data measured by that sensor. Alerts are automatically removed when the measured environmental conditions return to safe values.

The sharing box on the left side of the page shows the list of users the collection is shared with. Users can share the collection with new users or remove the existing ones. After a collection is shared with some users, they have full access to the collection.

The comment box on the right side is a tool for sharing information among users and adding useful notes related to the collection.

On the top-left, the sensors' status shows if the connection with the interconnected sensors' network is working fine. If this is the case, the colour of the status is green, otherwise red. In this second case, placing the mouse over the status icon shows which sensors are having a connection issue. This status appears only if interconnected sensors are set.

Figure 3.12 shows an example of the dashboard of a testing collection.

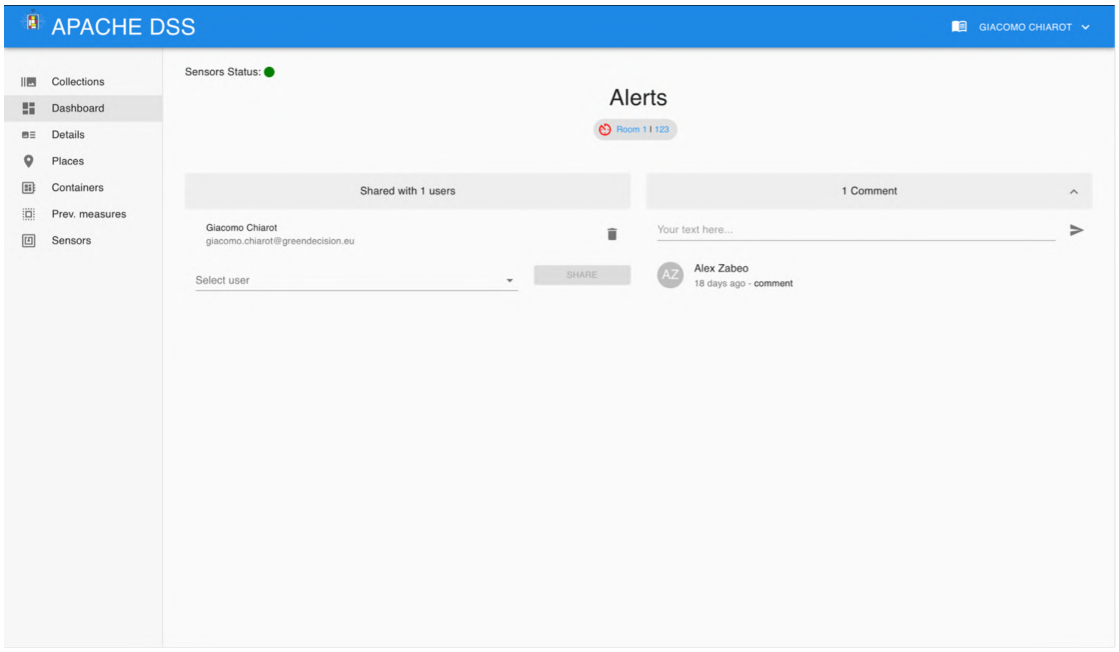
### Collection's details

The general information of a collection can be accessed and modified from the Details section. Such information includes the name of the collection, the institution, and a brief description. The cards shown in the collections list are updated if any modifications occur.

Figure 3.13 shows an example of this section.

### Collection's places

Collections are characterized by a list of places. Places are the physical locations composing the selected collection. They can be different buildings or areas of the same building with different functions (i.e., storage and exhibition).



Sensors Status: ●

Alerts

Room 1 | 123

Shared with 1 users

Giacomo Chiarot
giacomo.chiarot@greendecision.eu

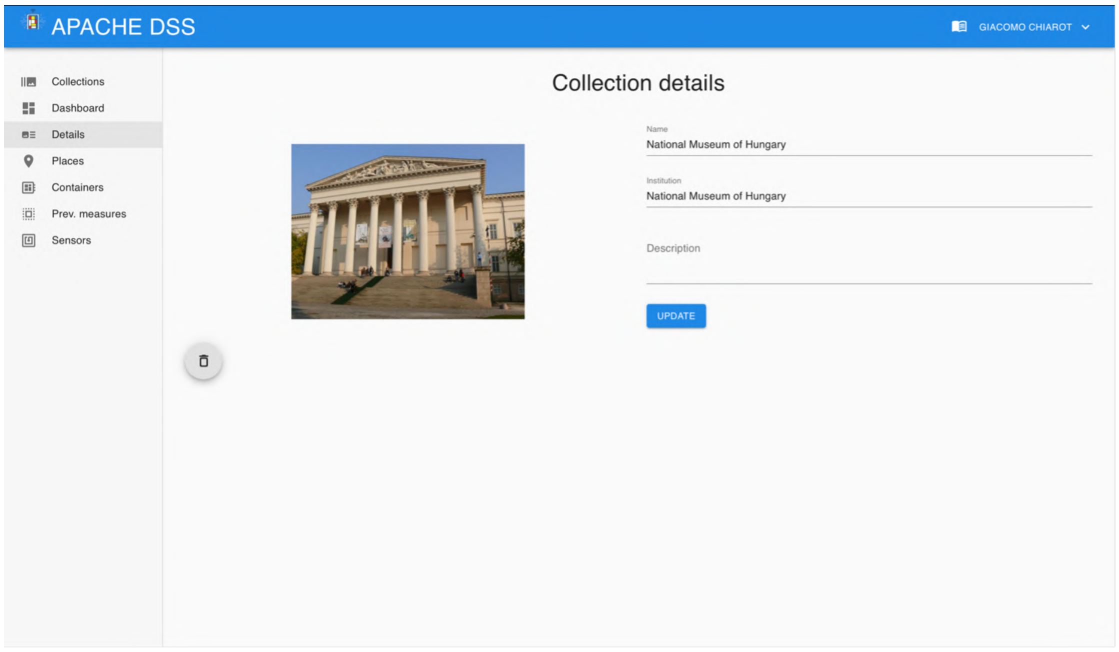
Select user

1 Comment

Your text here...

Alex Zabeo
18 days ago - comment

Figure 3.12: Example of the collection dashboard section



Collection details

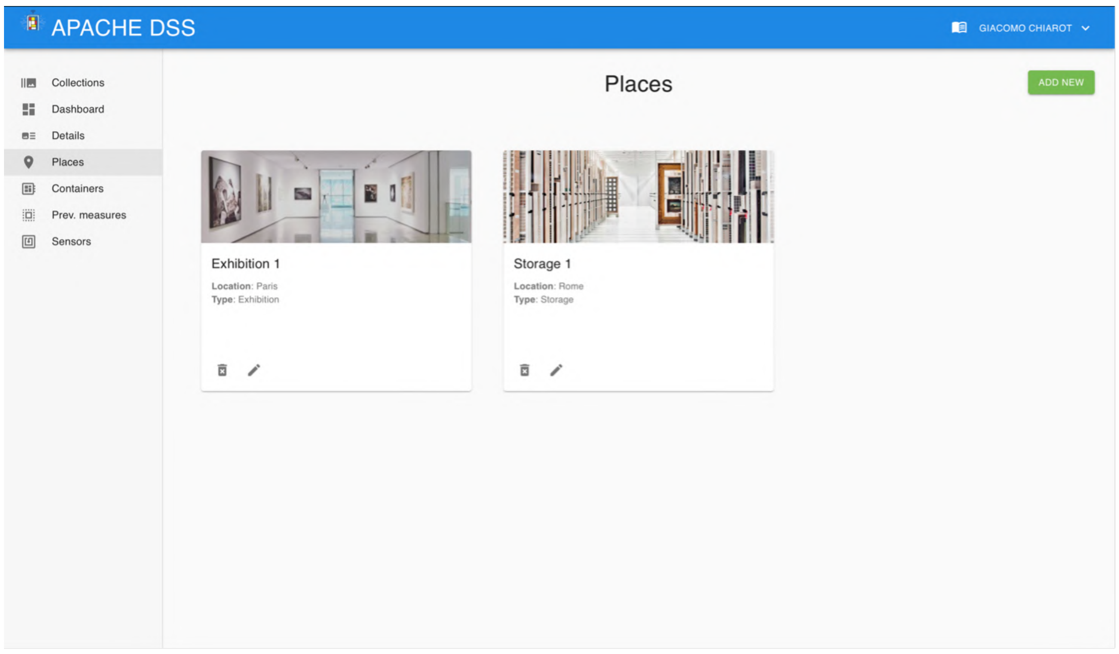
Name
National Museum of Hungary

Institution
National Museum of Hungary

Description

Figure 3.13: Example of the collection details section

To each place, users can specify the name of the place, its location (could be either an address or general indications), and the type of the place, choosing from archive, exhibition, and storage. All the information inserted can be modified and places can be removed. Figure 3.14 shows an example of a list of places.



Exhibition 1
Location: Paris
Type: Exhibition

Storage 1
Location: Rome
Type: Storage

Figure 3.14: Example of places list

## Place's containers

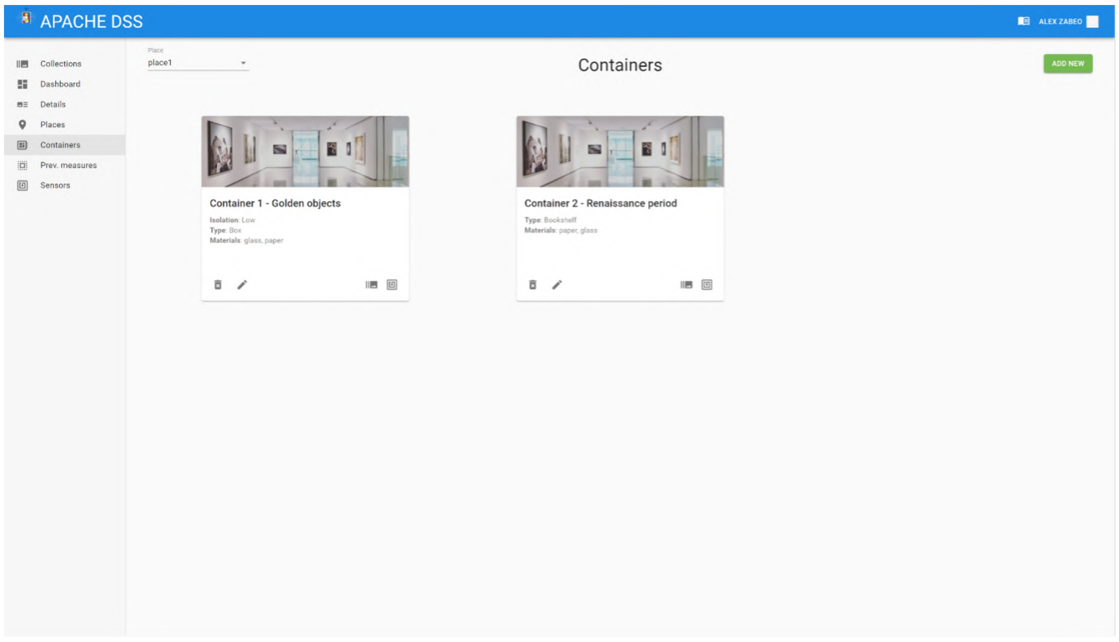
Containers are the closest level to cultural objects. They are placed in a particular location and contain cultural objects. To simplify the user interaction, the DSS only requires inputting objects' materials rather than listing all objects inside a container. Containers can also have sensors appointed to them which measure environmental conditions inside the container. Finally, it is possible to specify a list of special objects placed in the container to monitor them with more specific damage functions, which require more detailed object information rather than just its materials. Rooms are also considered containers for cultural objects freely exposed, as can happen with statues.

The information required for adding a new container includes:

- The name of the container.
- The isolation level, intended as how much the environmental condition inside the container is affected by external changes. This value can be set to Low, Medium, and High.
- The type of the container, chosen from a list of pre-determined values (Bookshelf, Metal Bookshelf, Box, Wood Box, Plexiglas Box, Cardboard Box, Glass Display Case, Large Plexiglass Display Case, Display Case, Wood Crate, Metal Drawers, Historical Cabinet, Room).
- The list of materials of the objects placed inside the container.
- A brief description of the container and the objects inside it.

After a container is added, users can specify which sensors are placed in the container, by setting their name and which parameters they measure. Similarly, users can add special objects by providing a name and the specific data required by the damage function, such as the painting technique and the thickness of the paint.

Figure 3.15 shows an example of a list of containers inside the exhibition place of the previous example.



Place: place1

Container 1 - Golden objects
Isolation: Low
Type: Box
Materials: glass, paper

Container 2 - Renaissance period
Type: Bookshelf
Materials: paper, glass

Figure 3.15: Example of containers list

### Preventive measures

The preventive measures section displays three lists of preventive measures for each container: already applied, suggested, and complete list. The items in the lists are retrieved from the preventive measures open repository rest API described in Section 3.1.5.

The preventive measures are characterized by textual description and categorical values related to four properties: effectiveness, durability, cost, and expertise. These values are directly taken from the Apache preventive measures' open repository (https://apacherepository.dais.unive.it) where they've been initially input by experts in the Apache project and are now openly modifiable in the repository through a moderated crowdsourcing protocol where users can propose changes and moderators decide whether to accept them. Properties values are used to compute the score of a preventive measure, defined in Equation 3.1.

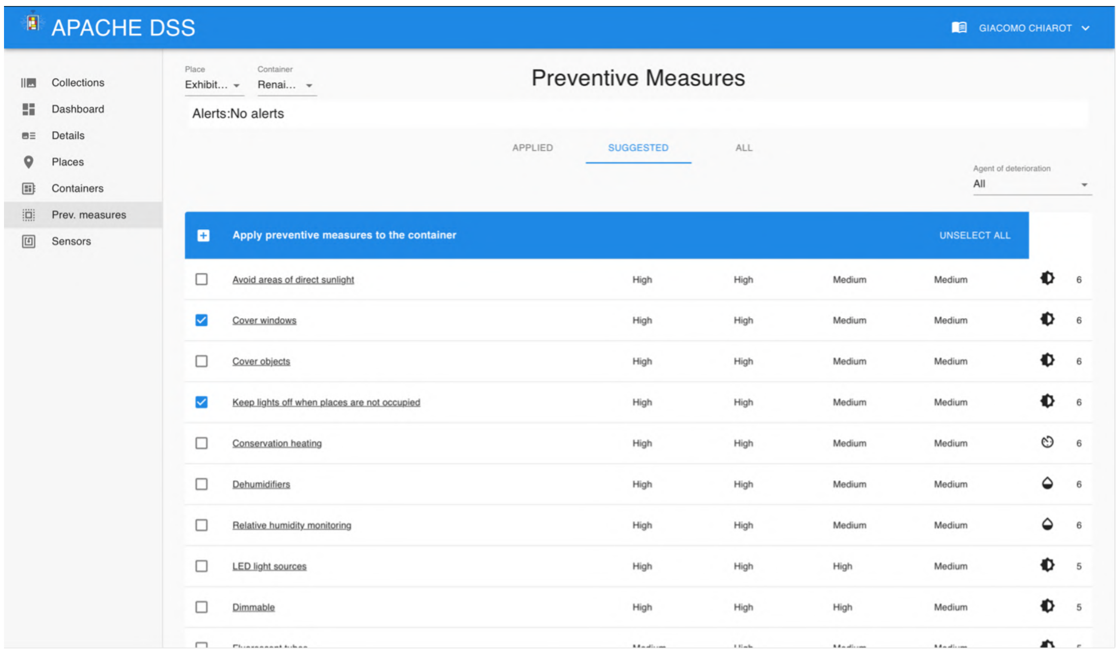
$$score = E + D - C - X + 4 \tag{3.1}$$

Where $E$ is the effectiveness value, $D$ is the durability value, $C$ is the cost value, and $X$ is the expertise value. Since these values are categorical, 0 is associated with Low, 1 with Medium, and 2 with High. A constant is added for having the lowest score starting from 0.

This score is used to sort the preventive measures from the most relevant to the less relevant. Moreover, a filtering widget can be used to limit the list of preventive measures to a specific agent of deterioration.

It's possible to define some selected preventive measures as already applied if this is the case in the modelled real-world container. Such applied preventive measures can also be removed from the corresponding list.

Figure 3.16 shows an example of how users can select the preventive measures that are already applied to a container among the suggested ones.



Place: Exhibit...
Container: Renai...

Alerts:No alerts

Agent of deterioration: All

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ☐ | Avoid areas of direct sunlight | High | High | Medium | Medium | ◐ | 6 |
| ☑ | Cover windows | High | High | Medium | Medium | ◐ | 6 |
| ☐ | Cover objects | High | High | Medium | Medium | ◐ | 6 |
| ☑ | Keep lights off when places are not occupied | High | High | Medium | Medium | ◐ | 6 |
| ☐ | Conservation heating | High | High | Medium | Medium | ◷ | 6 |
| ☐ | Dehumidifiers | High | High | Medium | Medium | ◒ | 6 |
| ☐ | Relative humidity monitoring | High | High | Medium | Medium | ◒ | 6 |
| ☐ | LED light sources | High | High | High | Medium | ◐ | 5 |
| ☐ | Dimmable | High | High | High | Medium | ◐ | 5 |

Figure 3.16: Example of the selection of preventive measures

## Sensors

The sensors section monitors the environmental conditions inside a particular container. Sensor's data can be either uploaded with an Excel file, read by a smartphone through the specific APACHE Android app introduced in Section 3.2.4, and by connecting to a wireless sensors' network or rest API.

The file option is the simplest one: users can download an Excel template and fill it with data measured by offline sensors. The updated file can then be uploaded using the Upload Sensor Data button. The second option needs containers to be provided with the NFC sensors produced by Tyndall. Using the Android reading app, users can log in with their account, read the value by approaching the smartphone near the sensor, and subsequently send data whenever an internet connection is available. The third option is related to the Bluetooth sensors network designed by Ericsson. Users can configure their connection to such sensors' network by inserting the configuration data, which can be found by clicking the Setup button.

The alerts box on top shows the lists of alerts fired by the APACHE DSS. If users click on an alert, the data corresponding to that alert is shown.

Data are visualized both with a graph and a table. Opening the graph visualization, users see data of the last month drawn in a graph with, in addition, the thresholds that are commonly used for that parameter. Green lines represent the optimal range in which that parameter should be, while red lines identify the safe area. Alerts are fired when the mean of the last 10 days is outside the safe area. The historical data visualization shows all the row data.

Figure 3.17 shows an example of the sensors section with the graph visualization.

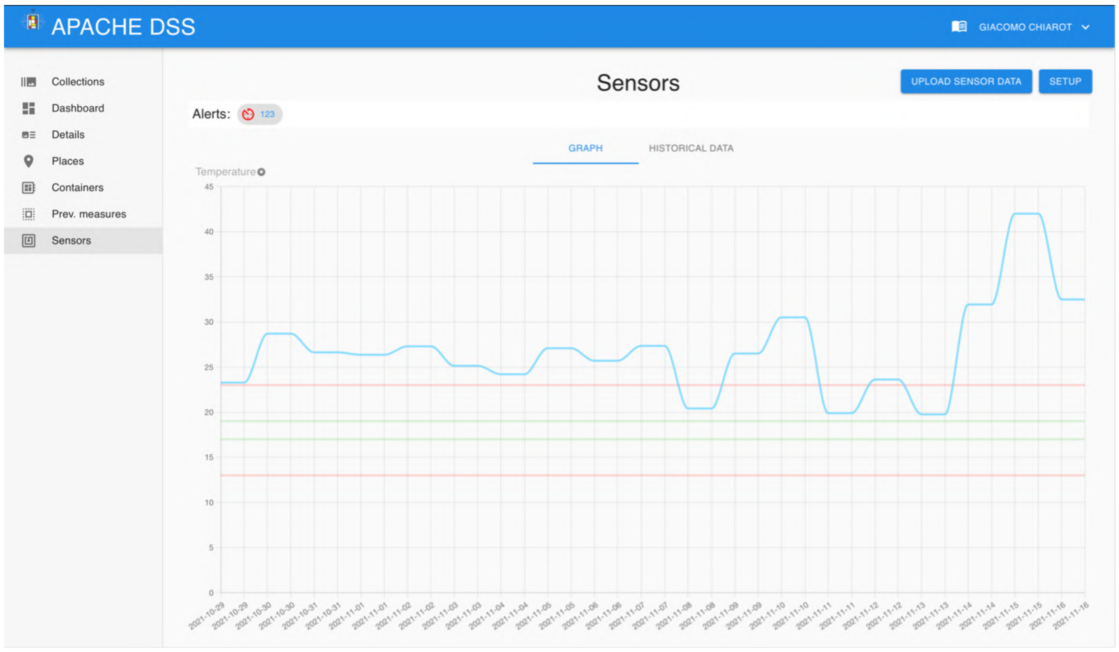


Figure 3.17: Example of the sensor section

### 3.2.4 Sensors integration

This section presents the implementation of the APACHE Leshan server, Android application, LoRaWAN network, and APIs to collect the sensor's data.

The sensors that are included in the implementation are characterized by different types of communication means:

- **Near Field Communication (NFC)**: NFC sensors are powerless and harvest energy from a nearby external device to read the environmental data and send them to the APACHE DSS server.
- **Wi-Fi**: They are connected to local routers with a LoRaWAN network.
- **Bluetooth**: They are connected to local routers with a Bluetooth mesh network developed by Ericsson.

The solutions described in this section are developed in collaboration with Greendecision[5], Ericsson[6], and Tyndal[7].

#### Powerless tags (NFC)

The NFC sensor developed by Tyndall is a card responder with an RFID antenna, as shown in Figure 3.18. This sensor, when powered by a reader, measures temperature and relative humidity. The result of the reading is plain text with comma-separated values.

[5]https://www.greendecision.eu
[6]https://www.ericsson.com/en
[7]https://www.tyndall.ie

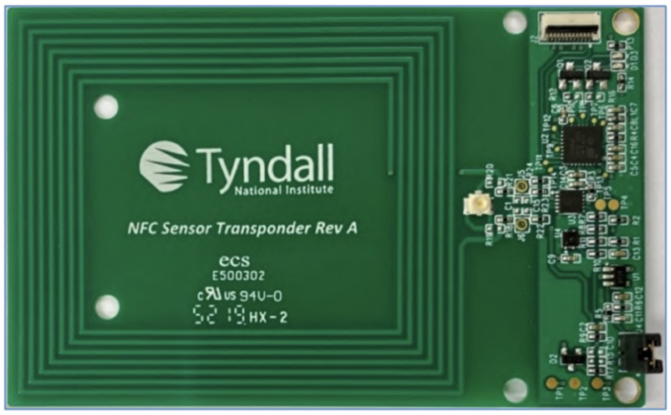


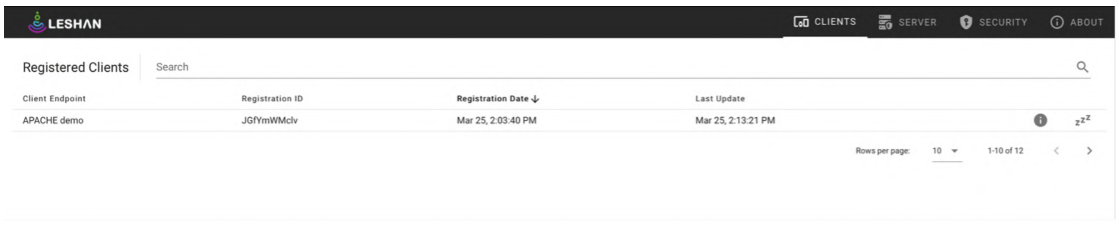
Figure 3.18: Tyndall NFC sensor

Sensor's readings are delivered to the APACHE DSS through the Android smartphone application described in Section 3.3. To perform a reading, users must log in, using the same credentials used for the APACHE DSS. This allows them to have access to their collection and authorize the application to insert new data.

### Bluetooth sensors' application server

To store data coming from Bluetooth sensors' mesh developed by Ericsson, a customized version of the Leshan LWM2M server[8] is developed. The Leshan server is a web server implementing the LWM2M communication protocol, which is designed for remote device management and telemetry within the Internet of Things. Customization involved updating the IPSO objects for the server backend to receive the sensor's measurements correctly.

Data acquired by the Leshan server are then sent to the APACHE DSS's database and displayed on the page dedicated to sensors' readings. This Leshan server itself is equipped with a basic graphical interface used to check the status of the server and the connected sensors, as shown in Figure 3.19



| Client Endpoint | Registration ID | Registration Date ↓ | Last Update |
|---|---|---|---|
| APACHE demo | JGfYmWMclv | Mar 25, 2:03:40 PM | Mar 25, 2:13:21 PM |

Rows per page: 10 1-10 of 12

Figure 3.19: Customized Leshan server interface

### Wi-Fi sensors' application server

To store data coming from the Wi-Fi LoRa sensors developed by Tyndall, the communication is managed with a LoRaWan protocol [34]. This protocol is implemented by integrating the storage integration service provided by The Thing Industries[9]. This storage integration watches for upstream traffic and stores data

[8]https://github.com/eclipse/leshan
[9]thethingsindustries.com

for up to 24 hours. To this end, the community edition is used with its free tier. The APACHE DSS uses the storage integration API to retrieve data before they are deleted.

#### APIs

The API is implemented with the Meteor backend. It is composed of 3 endpoints used to retrieve user tokens, collect information, and send data.

**User token** The user token is used to authenticate the reading of the collection's information and send data. It can also be found on the user's profile page inside the APACHE DSS. If the token is compromised, users can create a new one. The endpoint used to get the token is "apachedss.dais.unive.it/api/users" accessed in a POST request, giving in input the credential used to log in the APACHE DSS: email and password. The response is either the token if the provided credentials are correct, or an error message.

**Collection's information** To retrieve the collection's information, the endpoint is "apachedss.dais.unive.it/api/collections" accessed in POST and requires as input only the user's token. Each token is unique, so unambiguously associated with one user. The response is either a JSON object with all the collection's information (list of collections, places, containers, and sensors) or an error if the token is not correct. Since sensor data are related to one specific sensor, this endpoint is useful to get the full list of sensors with their IDs and position.

**Send data** To send data to the APACHE DSS specifying the destination sensor, the endpoint is "apachedss.dais.unive.it/api/sensors" accessed in POST. It requires as input the user token, sensor ID, measure type (temperature, humidity, acetic acid, formic acid, formaldehyde, acetaldehyde, SO2, NO2, O3, H2S), collection ID, container ID, date, time, and value. The response is either a success message or the error if the user token is not correct or the input is malformed.

## 3.3 Android App

As anticipated in Section 3.2.4, the Android App is developed to read the environmental values measured by the powerless sensors developed by Tyndall and send the read values to the APACHE DSS.

The application's structure is linear and composed of 4 screens put in sequence: login, home page and data read, sensor's list, and data confirmation.

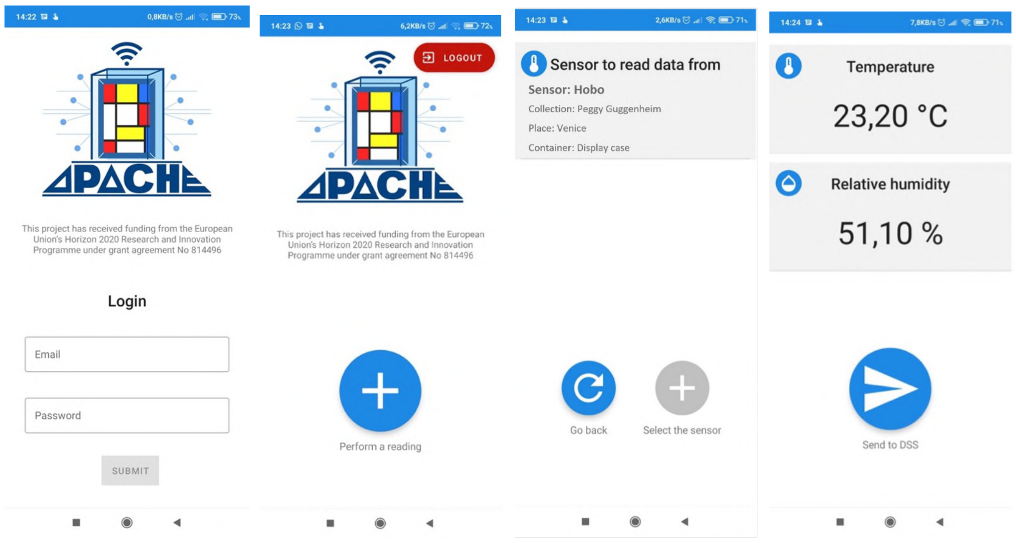
The login page shown allows users to log in using their credentials. If the credentials typed by the user are correct, the APACHE server returns a user token described in Section 3.2.4 that is safely stored by the application.

After logging in, the main screen lets the user read the sensor's values. To perform a new read, the user needs to bring the smartphone close to the sensor and wait a few moments to get the environmental measurements.

After measures are read, the application shows the list of sensors that are present in the APACHE DSS showing their name assigned by the user and the container and location in which they are placed. This list can be modified only from the APACHE DSS by adding or removing sensors.

After selecting one of these sensors, the read measurements are displayed before being submitted. Once readings are delivered, they are received by the APACHE DSS's online database through a web API described in Section 3.2.4 which stores the data in the correct user's collection and displays them in the DSS's page dedicated to sensors' readings.

Figure 3.20 shows the screenshots taken from the Android application of the 4 screens previously described. The application's apk file is available to be downloaded inside the APACHE DSS at apachedss.dais.unive.it/app.



APACHE
This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No 814496
Login
Email
Password
SUBMIT
LOGOUT
Perform a reading
Go back
Select the sensor
Send to DSS

Sensor to read data from
Sensor: Hobo
Collection: Peggy Guggenheim
Place: Venice
Container: Display case

Temperature
23,20 °C

Relative humidity
51,10 %

a. Login screen
b. Read data screen
c. Sensor's list screen
d. Send data screen

Figure 3.20: Screenshots of the different screens of the Android App

## 3.4 Conclusions

The APACHE Decision Support System framework presented in Chapter 2 is implemented in all its parts and released at: apachedss.dais.unive.it. This chapter describes the implementation presenting some screenshots from the web application.

The DSS allows curators to organize their collections' data into places and containers by specifying their materials and environmental conditions. By integrating this information with sensors' readings, the DSS can propose effective preventive measures and fire alarms when any threat is likely to harm objects.

Different growing versions of the DSS were presented and tested throughout the three APACHE public workshops which took place in Paris, Turin, and Budapest, as well as during internal project workshops as specifically organized in Venice in November 2021. As shown in the next chapter, the DSS is tested also by the application to case studies.

The general response was positive, and the involved users were interested in exploring the capabilities of the software and putting it in place in their collections.

# Chapter 4

# Results of the APACHE DSS

This chapter presents the results obtained for the APACHE DSS. It is divided into two main sections, related to the preventive measures open repository and to the application of the APACHE DSS to real case studies.

These results are achieved thanks to the collaboration of the user group of the APACHE project, composed by the Italian Ministry of Culture (MiC)[1], Centre Pompidou[2], National Museum of Slovenia[3], Hungarian National Museum[4], Fondazione Scienza e Tecnica Museum[5], Peggy Guggenheim Collection Venice[6].

## 4.1 Open Repository

The initial set of preventive measures included in the repository is collected through a questionnaire filled in by experts, case studies representatives, and preventive measures' producers among the partners of the APACHE project.

The open repository currently collects 50 preventive measures, divided into 8 categories:

- **lighting systems**: suggestions on the best lighting system to apply.
- **shielding**: preventive measures addressing how to protect objects from direct light (natural and artificial).
- **exposition**: suggestions on how to place objects in the display areas.
- **containers**: preventive measures related to storing objects inside containers.
- **absorbents**: list of absorbents materials against humidity and pollutants.
- **monitoring**: suggestions on how to monitor environmental conditions.
- **environmental control**: suggestions on how to control environmental conditions.
- **storage**: suggestions on how to place objects in the storage areas.

[1]https://sabapchpe.beniculturali.it
[2]https://www.centrepompidou.fr/en
[3]https://www.nms.si/si/
[4]https://mnm.hu/en
[5]http://www.fstfirenze.it/?lang=en
[6]http://www.fstfirenze.it/?lang=en

The distribution of the preventive measures on such categories are shown in Figure 4.1.

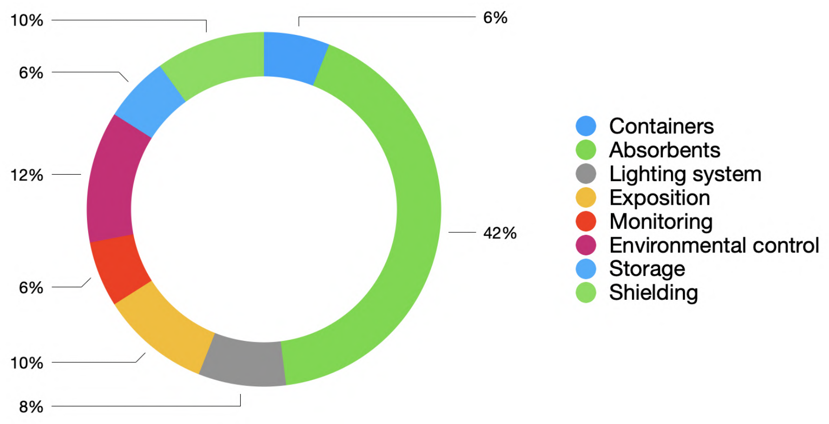


Figure 4.1: Preventive measures distribution

## 4.2 Case Studies

This section presents the application of the APACHE DSS to six case studies. For each institution, it is described the type of cultural heritage contained, the number and typology of places and containers, and the number and name of the rooms. Further information is given about the sensors, such as the number and the type. For each container the environmental data measured by sensors are reported in graphs while the suggested preventive measures are reported in tables. As far as unit of measurements are concerned, values are reported in degrees centigrade for temperature, as a percentage for relative humidity and in g/m3 for Volatile Organic Compounds pollutants (VOCs).

### 4.2.1 Peggy Guggenheim Collection

The Peggy Guggenheim Collection (in Figure 4.2) is one of the most important museums for artworks of the 20th century. The art objects considered for this case study are located in one exhibition place named Palazzo Venier dei Leoni in Venice. Four containers were selected, two rooms inside the museum: Barchessa and Sala del Cubismo, and two display cases: Noland-Birth and Boccioni-Dinamismo di un cavallo in corsa. The four selected containers were chosen as being the most representative having enough sensor's data.

The internal temperature and relative humidity conditions were often critical for the two room containers. Both inside and outside the two display cases conditions were often critical for both temperature, relative humidity and VOCs. Of the five VOCs analyzed, one reached a level that resulted in the critical condition being outside the two display cases and two or more VOCs reached a level that resulted in the critical condition within the display cases.

#### Barchessa's room

The Barchessa room contains materials such as oil painting, acrylic paint, stone and plaster, and it has a low isolation. Inside the room, humidity, and temperature sensors are present.

Figure 4.2: Peggy Guggenheim institution

Figure 4.3 shows the temperature data obtained from the sensor inside the Barchessa container from the period of January 2020 to mid-February 2021, during which time conditions inside the container have been satisfactory.

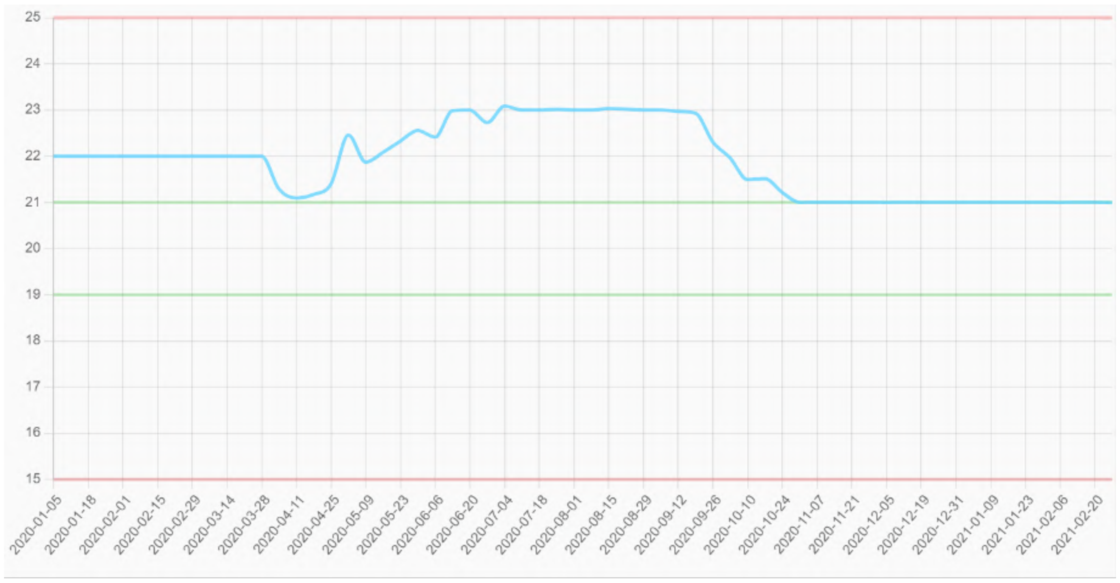


Figure 4.3: Temperature data from the sensor inside Barchessa container

Figure **??** shows the relative humidity data obtained from the sensor inside Barchessa container from the period of January 2020 to the end of February 2021. It can be observed that the conditions inside the container have been good for most of the time, except on some days in the period between the end of April 2020 and mid-August 2020 when relative humidity has exceeded 60% and the condition became critical.

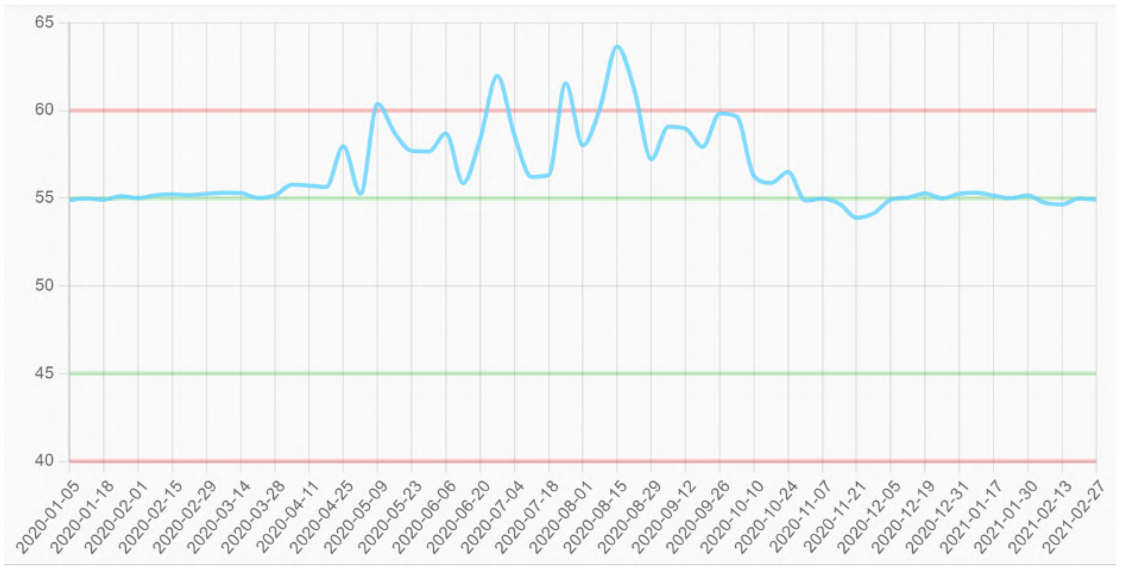


Figure 4.4: Relative humidity data from the sensor inside Barchessa container

Table 4.1 shows the suggested preventive measures for Barchessa container.

Table 4.1: Suggested preventive measures for Barchessa container by agent of deterioration

| Category | Name | Score |
|---|---|---|
| Light | Dimmerable | 5 |
| | Fluorescent tubes | 5 |
| | Display facsimiles | 5 |
| Temperature | Conservation heating | 6 |
| | Remove objects from external walls | 5 |
| | Cold storage (below 0°C) for unsuitable materials | 5 |
| Relative humidity | Dehumidifiers | 6 |
| | Insulate from walls | 5 |
| | Compensation of humidity fluctuations using graphene-based coatings | 4 |
| Pollutants | Layered double hydroxides (LDH) dispersed in heptane for acid sorption | 4 |
| | Layered double hydroxides (LDH) for acid sorption | 4 |
| | Layered double hydroxides (LDH) in PVA membrane for acid sorption | 4 |

### Sala del cubismo room

Sala del cubismo contains materials such as paper, glass, wood, oil painting, stone and plaster, photographic materials and metals and has a low isolation. Inside the room humidity and temperature sensors are present.

Figure 4.5 shows the temperature data obtained from the sensor inside Sala del cubismo container from the period of January 2020 to mid-February 2021. It can be observed that the conditions have been good for most of the time, except on some days in the period between the beginning of June 2020 and the end of September 2020 when the temperature rose above 23 degrees and the condition became critical.

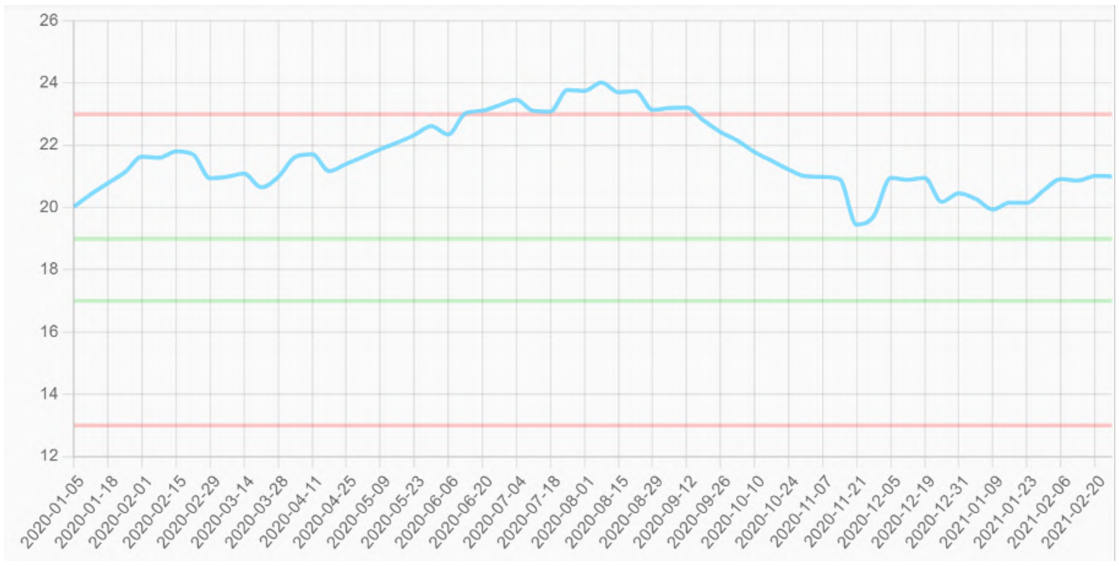


Figure 4.5: Temperature data obtained from the sensor inside Sala del cubismo container

Figure 4.6 shows the relative humidity data obtained from the sensor inside Sala del cubismo container from the period of January 2020 to the end of February 2021. It can be observed that the conditions inside the container have been good for most of the time, except on some days in the period between the end of July 2020 and mid-August 2020 when humidity has exceeded 60% and the condition was critical.

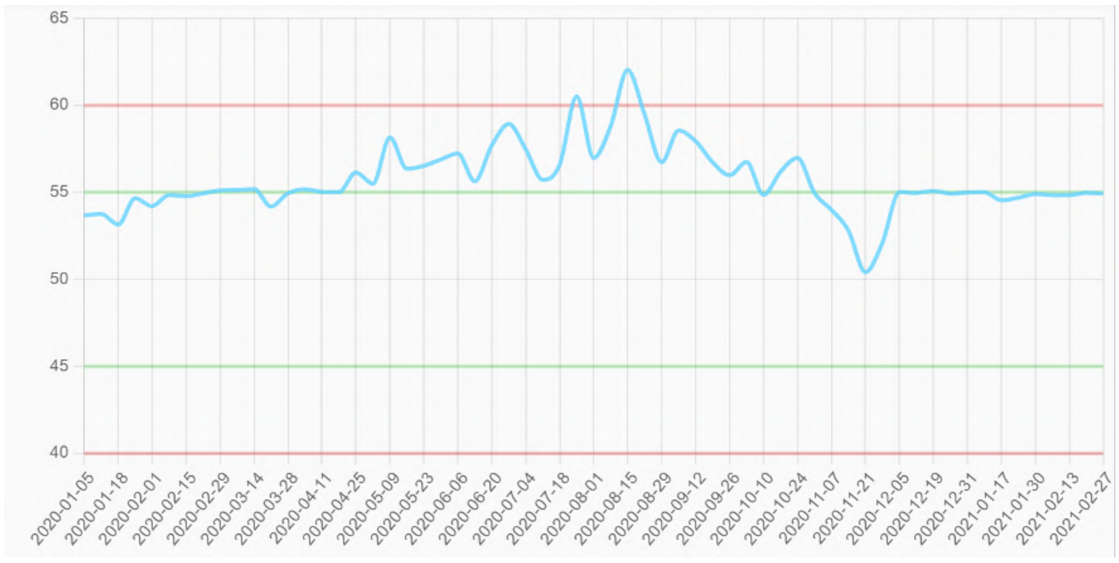


Figure 4.6: Relative humidity data obtained from the sensor inside Sala del cubismo container

Table 4.2 shows the suggested preventive measures for Sala del cubismo container.

Table 4.2: Suggested preventive measures for Sala del cubismo container by agent of deterioration

| Category | Name | Score |
|---|---|---|
| Light | Dimmerable | 5 |
| | Fluorescent tubes | 5 |
| | Display facsimiles | 5 |
| Temperature | Conservation heating | 6 |
| | Remove objects from external walls | 5 |
| | Cold storage (below 0°C) for unsuitable materials | 5 |
| Relative humidity | Dehumidifiers | 6 |
| | Insulate from walls | 5 |
| | Compensation of humidity fluctuations using graphene-based coatings | 4 |
| Pollutants | Layered double hydroxides (LDH) dispersed in heptane for acid sorption | 4 |
| | Layered double hydroxides (LDH) for acid sorption | 4 |
| | Layered double hydroxides (LDH) in PVA membrane for acid sorption | 4 |

## Noland-Birth

Noland-Birth has high isolation and contains acrylic paint only. Humidity, temperature, and VOCs (Volatile Organic Compounds) sensors both inside and outside the display case are present.

Figure 4.7 shows the temperature data obtained from the sensor inside Noland-Birth container in the period from mid-July 2020 to the end of October 2020. From the image it can be observed that the conditions inside the container have been good for most of the time, except on some days in August 2020 when the temperature rose above 25 degrees and the condition became critical.

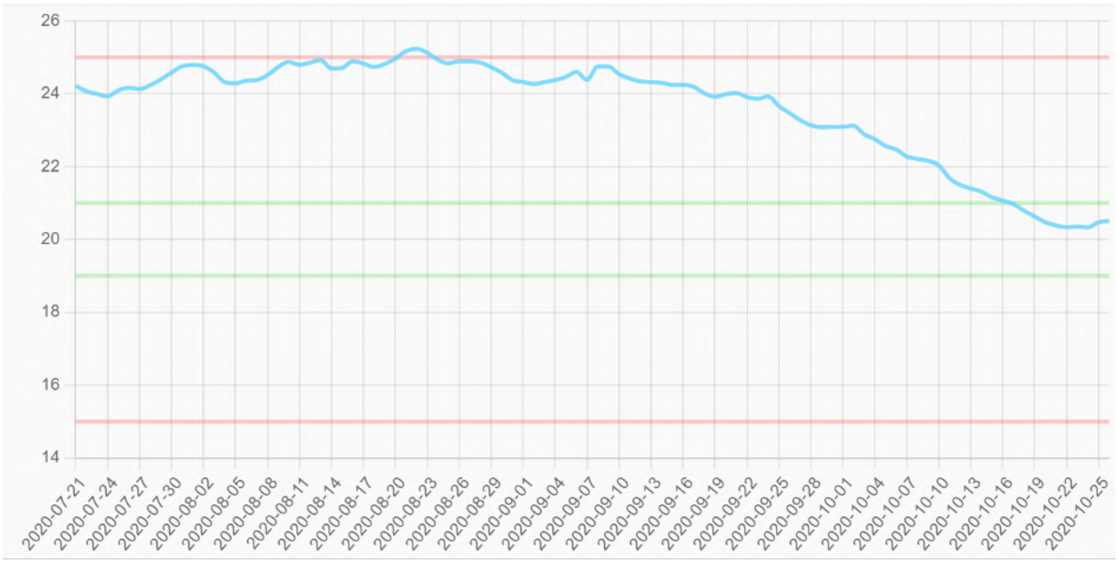


Figure 4.7: Temperature data obtained from the sensor inside Noland-Birth container

Figure 4.8 shows the relative humidity data obtained from the sensor inside Noland-Birth container in the period from mid-July 2020 to the end of October

2020. From the image, it can be observed that the conditions inside the container have been critical all the time.

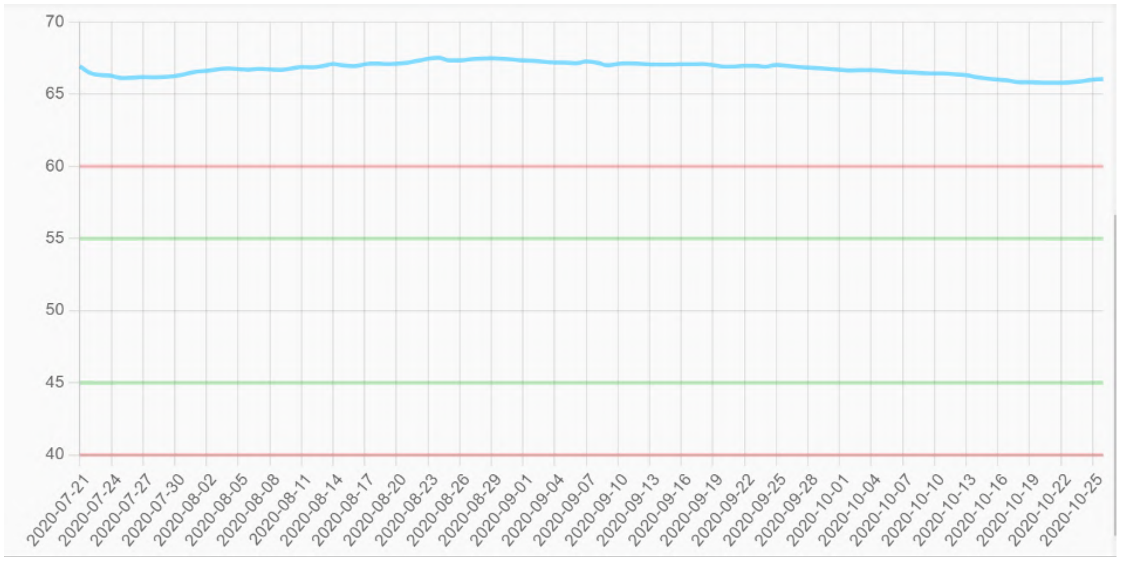


Figure 4.8: Relative humidity data obtained from the sensor inside Noland-Birth container

Figure 4.9 shows data obtained from the sensor inside Noland-Birth container in the period from mid-June 2020 to mid-March 2021 for Acetic acid, Formic acid, Formaldehyde, Acetaldehyde and $NO_2$. It can be observed that conditions inside the container have been critical all the time for Acetic and Formic Acid, while good for Formaldehyde and Acetaldehyde. $NO_2$ values instead dropped from critical to good.

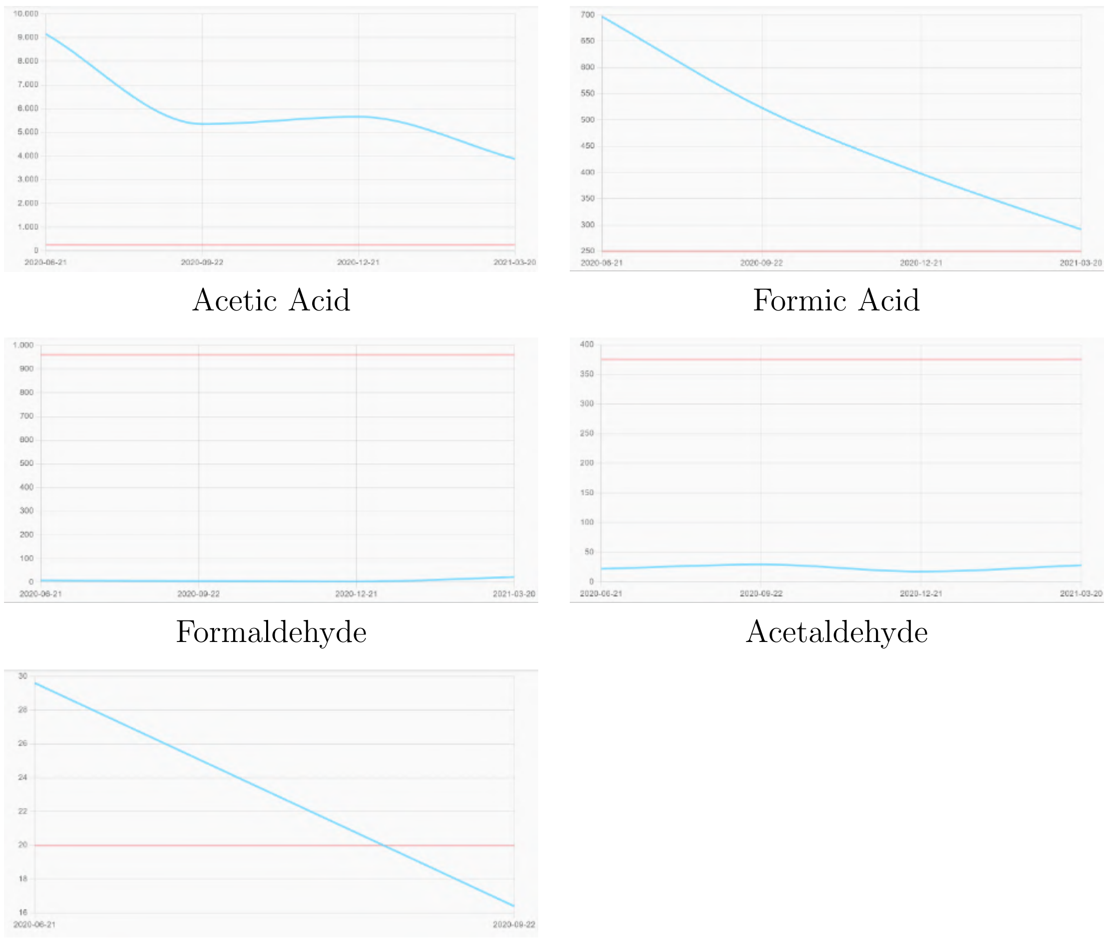
Table 4.3 shows the suggested preventive measures for Noland-Birth container.

Table 4.3: Suggested preventive measures for Noland-Birth container by agent of deterioration

| Category | Name | Score |
|---|---|---|
| Light | Dimmerable | 5 |
| | Fluorescent tubes | 5 |
| | Display facsimiles | 5 |
| Temperature | Conservation heating | 6 |
| | Remove objects from external walls | 5 |
| | Cold storage (below 0°C) for unsuitable materials | 5 |
| Relative humidity | Dehumidifiers | 6 |
| | Insulate from walls | 5 |
| | Super absorbents polymers (SAPs) dispersed in heptane for humidity sorption | 4 |
| Pollutants | Graphene-based aerogels | 5 |
| | Air quality control (absorbents for VOCs) | 4 |
| | Absorbents for pollutants | 4 |

### Boccioni - Dinamismo di un cavallo in corsa

Boccioni - Dinamismo di un cavallo in corsa has high isolation and contains materials such as paper, metals, wood, and oil painting. Humidity, temperature, and



Acetic Acid

Formic Acid

Formaldehyde

Acetaldehyde

$NO_2$

Figure 4.9: All VOCs data obtained from the sensor inside Noland-Birth container

VOCs (Volatile Organic Compounds) sensors both inside and outside the display case are present.

Figure 4.10 shows the temperature data obtained from the sensor inside Boccioni-Dinamismo di un cavallo in corsa container in the period from mid-January 2020 to mid-February 2021. From the image it can be observed that the conditions inside the container have been optimal between January and February 2020 and good until early May 2020. During the summer period between May 2020 and early October 2020 they have been critical and then returned good from October 2020 to mid-February 2021.

Figure 4.11 shows the relative humidity data obtained from the sensor inside Boccioni-Dinamismo di un cavallo in corsa container in the period from mid-January 2020 and mid-February 2021. From the image it can be observed that the conditions have been good for most of the time, sometimes optimal, and critical between January 2020 and February 2020 and in the summer period between May 2020 and September 2020.

Figure 4.12 shows data obtained from the sensor inside Boccioni-Dinamismo di un cavallo in corsa container in the period from mid-June 2020 to mid-March 2021 for Acetic acid, Formic acid, Formaldehyde, Acetaldehyde and $NO_2$. It can be observed that conditions inside the container have been critical all the time for Acetic acid while for Formic Acid went up and then down again. All other VOCs were good all along the period.

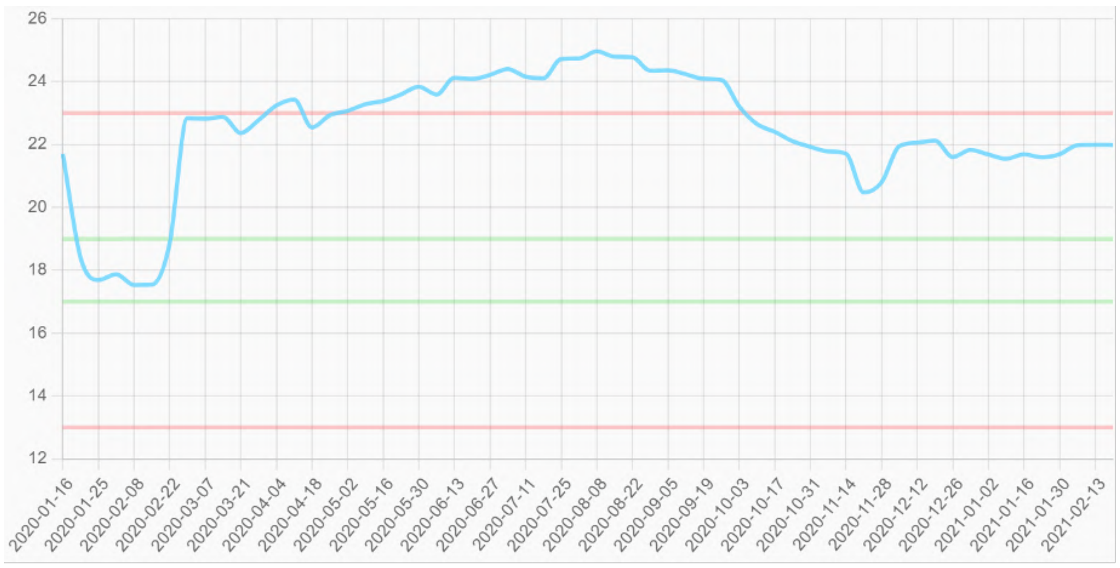


Figure 4.10: Temperature data obtained from the sensor inside Boccioni-Dinamismo di un cavallo in corsa container

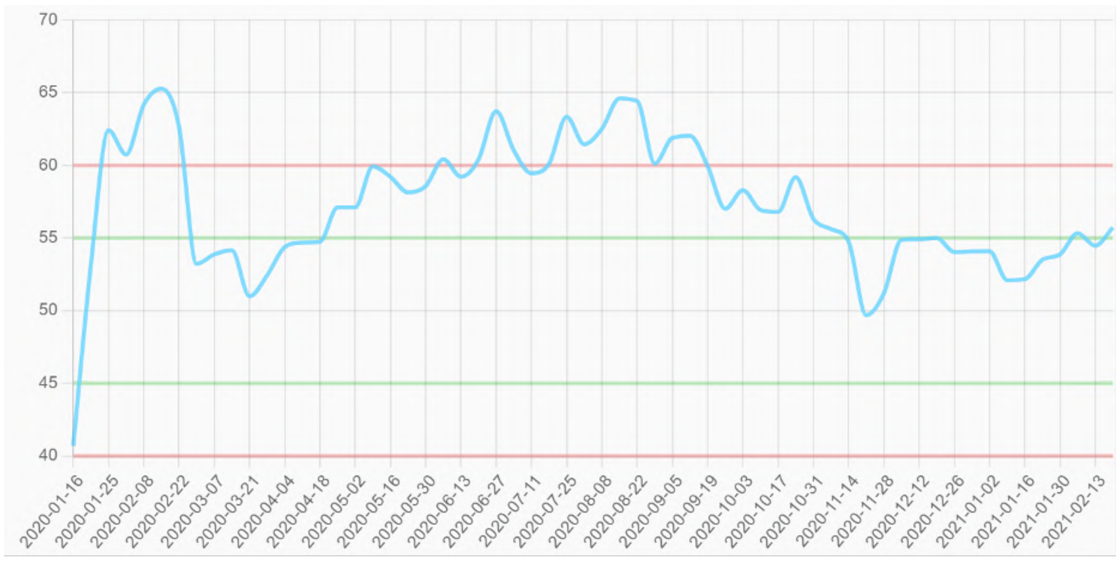


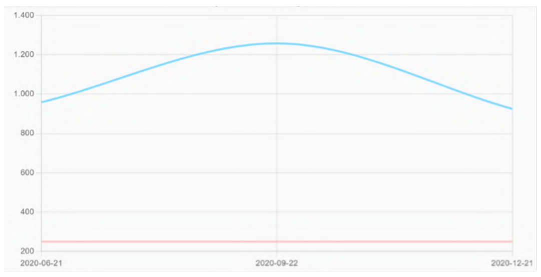
Figure 4.11: Relative humidity data obtained from the sensor inside Boccioni-Dinamismo di un cavallo in corsa container

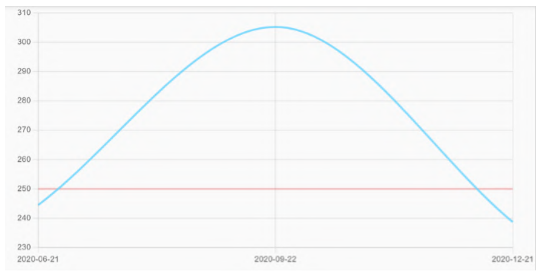
Table 4.4 shows the suggested preventive measures for Boccioni-Dinamismo di un cavallo in corsa container.

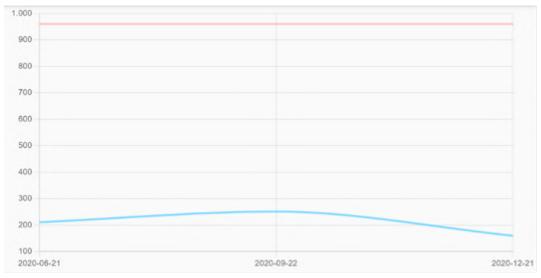
### 4.2.2 National Museum of Slovenia

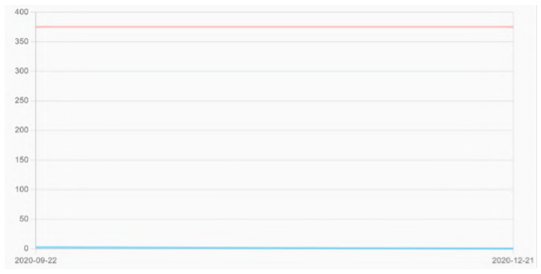
The National Museum of Slovenia (in Figure 4.13) is housed in a centrally located neo-Renaissance palace built between 1883 and 1885 to designs by Viljem Treo. The art objects considered for this case study are located in two places named 'Muzejska' and 'Metelkova'. For each place one container was selected, the Hasan's cloak glass display case for Muzejska and a Storage drawer for Metelkova. The two selected containers were chosen as being the most representative having enough sensor data.

The conditions inside and outside the containers were very variable both pos-


Acetic Acid


Formic Acid


Formaldehyde


Acetaldehyde

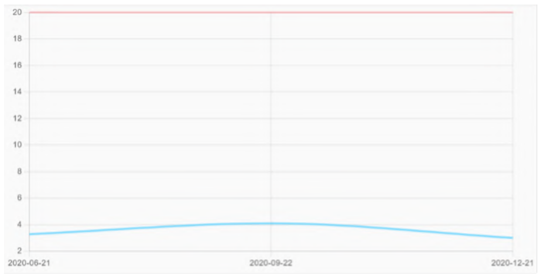

$NO_2$

Figure 4.12: All VOC data obtained from the sensor inside Boccioni-Dinamismo di un cavallo in corsa container

Table 4.4: Suggested preventive measures for Boccioni-Dinamismo di un cavallo in corsa container by agent of deterioration

| Category | Name | Score |
|---|---|---|
| Light | Dimmerable | 5 |
| | Fluorescent tubes | 5 |
| | Display facsimiles | 5 |
| Temperature | Conservation heating | 6 |
| | Remove objects from external walls | 5 |
| | Cold storage (below 0°C) for unsuitable materials | 5 |
| Relative humidity | Dehumidifiers | 6 |
| | Insulate from walls | 5 |
| | Super absorbents polymers (SAPs) dispersed in heptane for humidity sorption | 4 |
| Pollutants | Graphene-based aerogels | 5 |
| | Air quality control (absorbents for VOCs) | 4 |
| | Absorbents for pollutants | 4 |

itively and negatively in terms of temperature and relative humidity. Of the five VOCs analyzed, only one reached a level that resulted critical outside of both containers. Inside the containers the conditions have always been optimal.

Figure 4.13: National Museum of Slovenia

### Hasan cloak display case

The Hasan cloak display case contains materials such as textiles and metals and it has a hight isolation. Humidity, temperature, and VOCs (Volatile Organic Compounds) sensors both inside and outside the display case are present.

Figure 4.14 shows the temperature data obtained from the sensor inside Hasan cloak display case container in the period from the end of December 2019 to mid-October 2021. From the image it can be observed that the conditions inside the container have been good and optimal most of the time except during the winter period between November and February 2020 when the condition became critical.

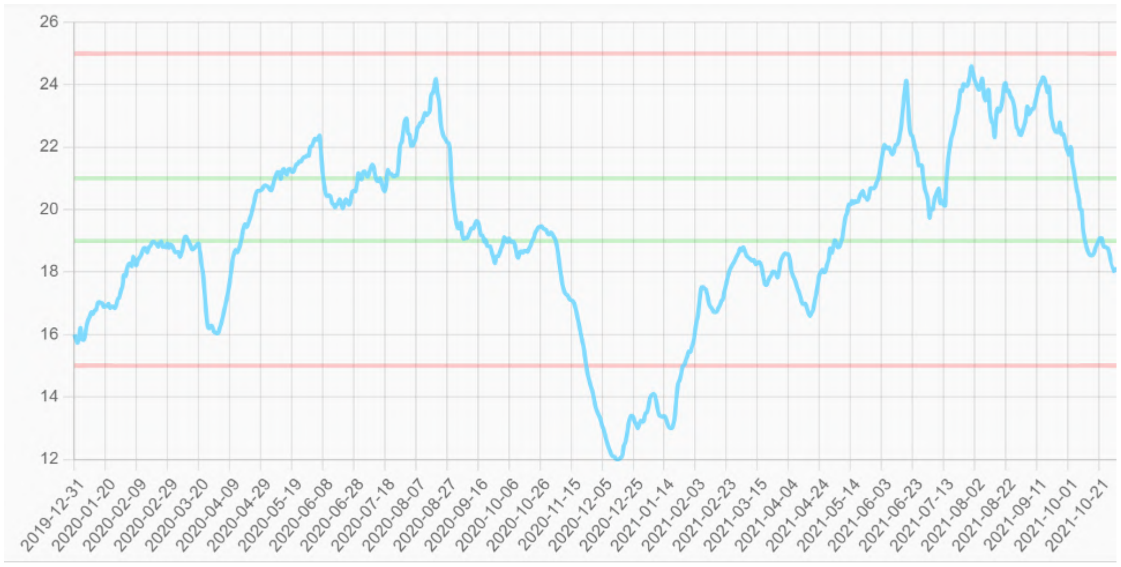


Figure 4.14: Temperature data obtained from the sensor inside Hasan cloak display case container

Figure 4.15 shows the relative humidity data obtained from the sensor inside Hasan cloak display case container in the period from the end of December 20219 to mid-October 2021. From the image it can be observed that the conditions have been good for most of the time, sometimes optimal, and critical between the winter period between December 2020 and April 2021, in the summer period between June 2020 and September 2020 and again in the winter period between February 2021 and April 2021.

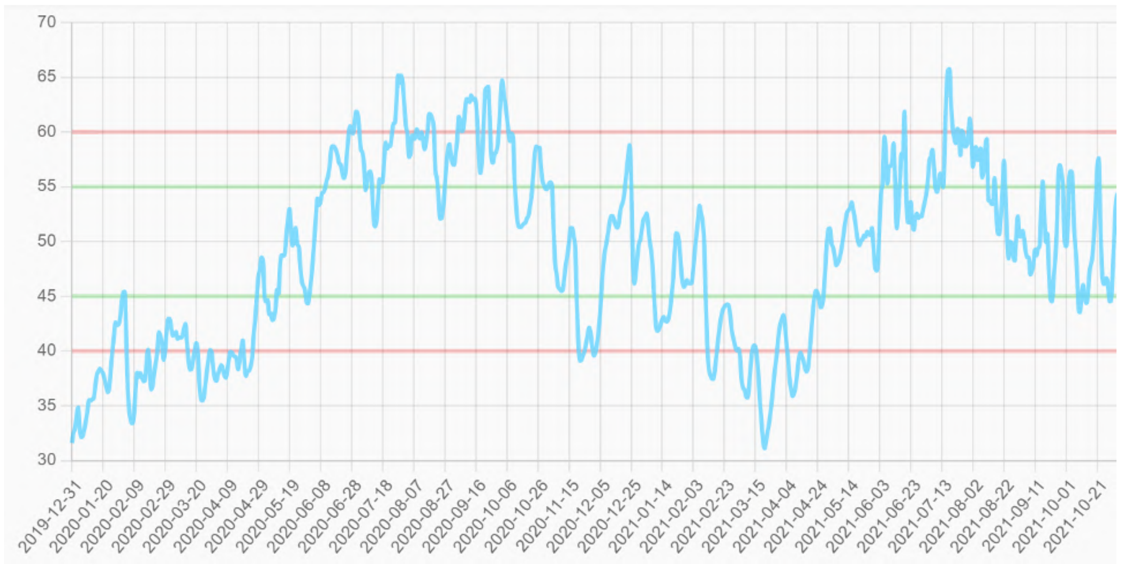


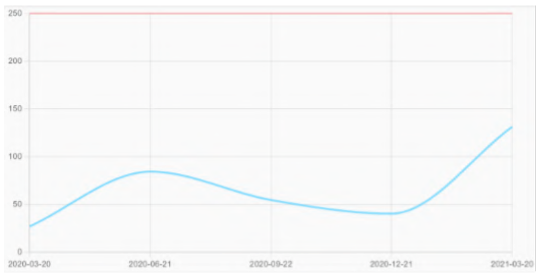
Figure 4.15: Relative humidity data obtained from the sensor inside Hasan cloak display case container

Figure 4.16 shows data obtained from the sensor inside Hasan cloak display case container in the period from mid-March 2020 to mid-March 2021 for Acetic acid, Formic acid, Formaldehyde, Acetaldehyde and $NO_2$. It can be observed that conditions inside the container have been good all the time for all VOCs.

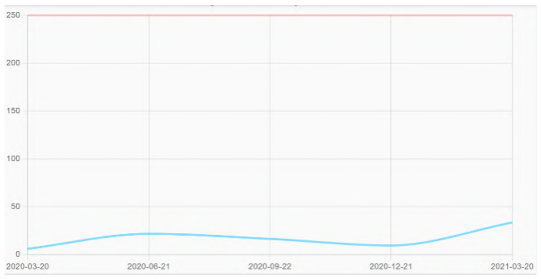
Table 4.5 shows the suggested preventive measures for Hasan cloak display case container.

Table 4.5: Suggested preventive measures for Hasan cloak display case container by agent of deterioration

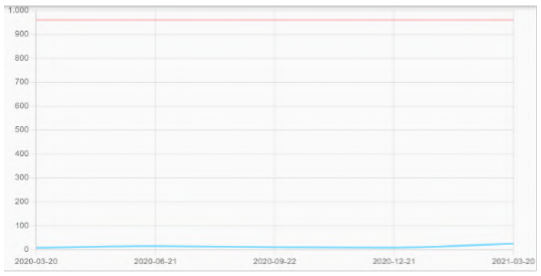
| Category | Name | Score |
|---|---|---|
| Light | Natural light shielding system | 6 |
| | Avoid areas of direct sunlight | 6 |
| | Cover windows | 6 |
| Temperature | Conservation heating | 6 |
| | Remove objects from external walls | 5 |
| | Cold storage (below 0°C) for unsuitable materials | 5 |
| Relative humidity | Dehumidifiers | 6 |
| | Insulate from walls | 5 |
| | Super absorbents polymers (SAPs) dispersed in heptane for humidity sorption | 4 |
| Pollutants | Graphene-based aerogels | 5 |
| | Air quality control (absorbents for VOCs) | 4 |
| | Absorbents for pollutants | 4 |



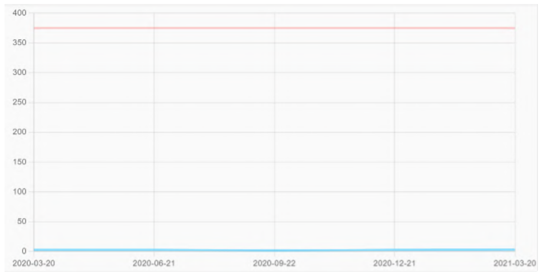
Acetic Acid



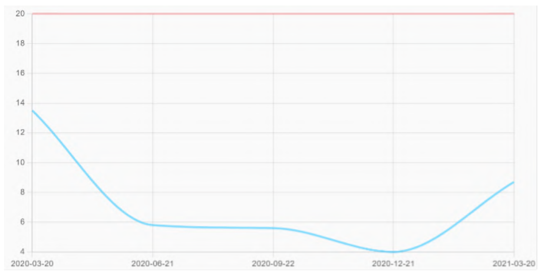
Formic Acid



Formaldehyde



Acetaldehyde



$NO_2$

Figure 4.16: All VOC data obtained from the sensor inside Hasan cloak display case container

### Storage drawner

The Storage Drawer contains materials such as paper, stone and plaster and it has a medium isolation. Humidity, temperature, and VOCs (Volatile Organic Compounds) sensors both inside and outside the drawer are present.

Figure 4.17 shows the temperature data obtained from the sensor inside Storage Drawer container in the period from the end of February 2020 and the end of May 2021. From the image it can be observed that the conditions inside the container have been good and optimal all the time.

Figure 4.18 shows the relative humidity data obtained from the sensor inside Storage Drawer container in the period from the end of February 2020 and the end of May 2021. From the image it can be observed that the conditions have been critical most of the time, sometime good and optimal during May and October 2020 and from April 2021.

Figure 4.19 shows data obtained from the sensor inside Storage Drawer container in the period from mid-March 2020 to mid-March 2021 for Acetic acid, Formic acid, Formaldehyde, Acetaldehyde and $NO_2$. It can be observed that conditions inside the container have been good all the time for all VOCs.

Table 4.6 shows the suggested preventive measures for Storage Drawer container.

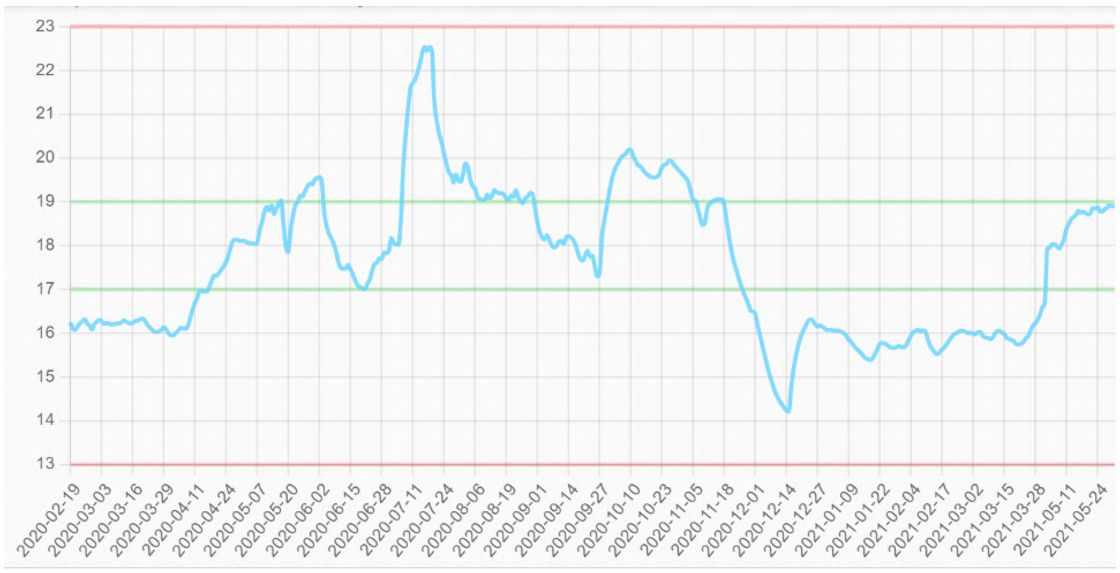


Figure 4.17: Temperature data obtained from the sensor inside Storage Drawer container

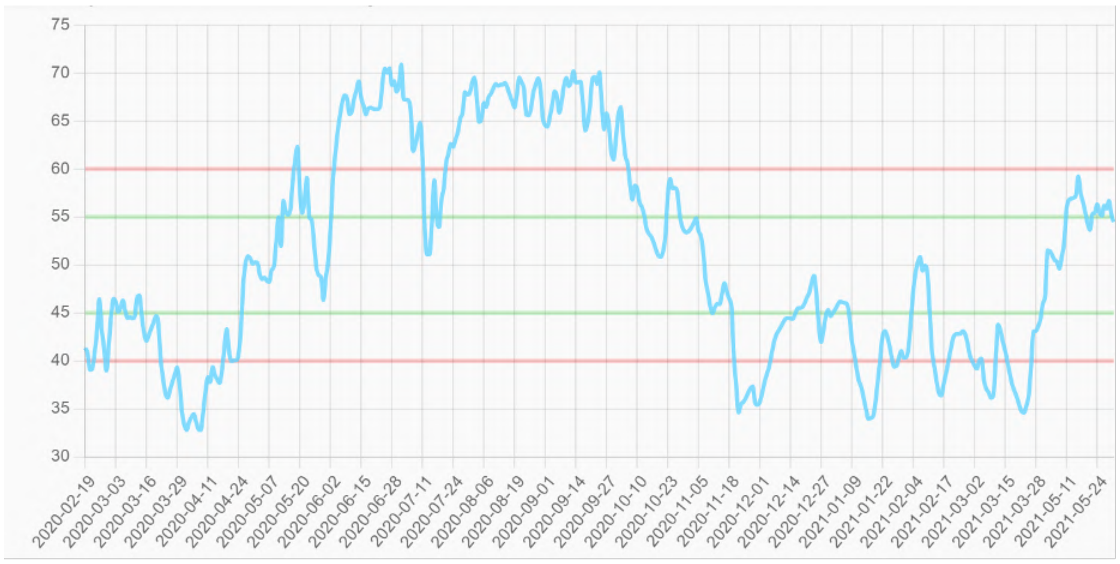
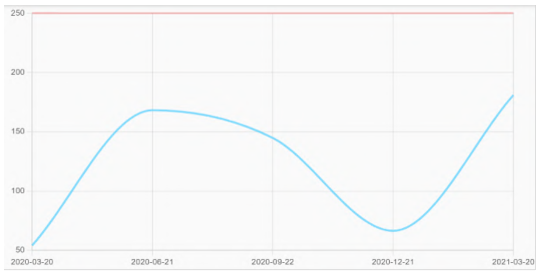


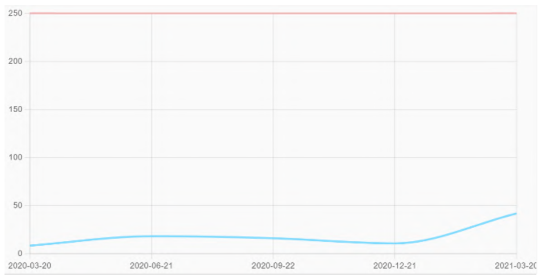
Figure 4.18: Relative humidity data obtained from the sensor inside Storage Drawer container

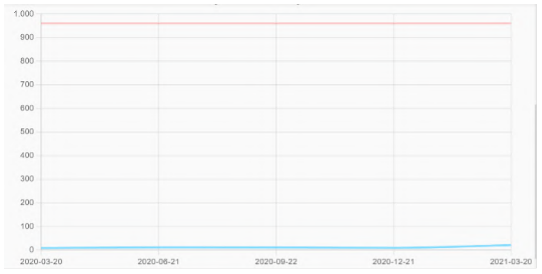
### 4.2.3 Hungarian National Museum

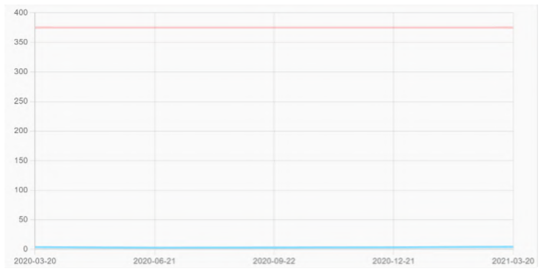
The Hungarian National Museum (Hungarian: Magyar Nemzeti Múzeum) (in Figure 4.20) was founded in 1802 and is the national museum for the history, art, and archaeology of Hungary. The art objects considered for this case study are located in two places named 'Museum' and 'Storage'. Four containers were selected for the first place, Bronze age object, Library room, Weapon Storage Cabinet and The Szechenyi Hall, and one for the second, Collections of historical photographs.

Conditions have often been critical both inside and outside the container Library room both for temperature and relative humidity. They have often been good and optimal for the container Collection of historical photographs. Of the five VOCs analyzed, one reached a level that resulted critical outside both containers and within the container Collection of historical photographs.


Acetic Acid


Formic Acid


Formaldehyde


Acetaldehyde

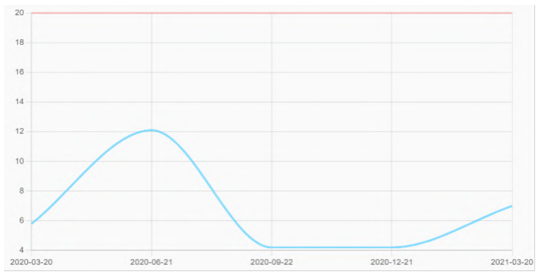

$NO_2$

Figure 4.19: All VOC data obtained from the sensor inside Storage Drawer container

Table 4.6: Suggested preventive measures for Storage Drawer container by agent of deterioration

| Category | Name | Score |
|---|---|---|
| Light | Natural light shielding system | 6 |
| | Avoid areas of direct sunlight | 6 |
| | Cover windows | 6 |
| Temperature | Conservation heating | 6 |
| | Remove objects from external walls | 5 |
| | Cold storage (below 0°C) for unsuitable materials | 5 |
| Relative humidity | Dehumidifiers | 6 |
| | Insulate from walls | 5 |
| | Closed boxes (Conventional Archive Boxes) | 5 |
| Pollutants | Closed boxes (Conventional Archive Boxes) | 5 |
| | Storage in pollutant barrier bags | 4 |
| | Layered double hydroxides (LDH) dispersed in heptane for acid sorption | 4 |

### Library room

The Library room contains materials such as paper, glass, textiles and metals and it has a medium isolation. Humidity, temperature, and VOCs (Volatile Organic Compounds) sensors both inside and outside the display case are present.

Figure 4.20: Hungarian National Museum

Figure 4.21 shows the temperature data obtained from the sensor inside Library room container in the period from July 2020 to end of June 2021. From the image it can be observed that the conditions inside the container have been critical most of the time except during the winter period between November and January 2021 when conditions sometime became good.

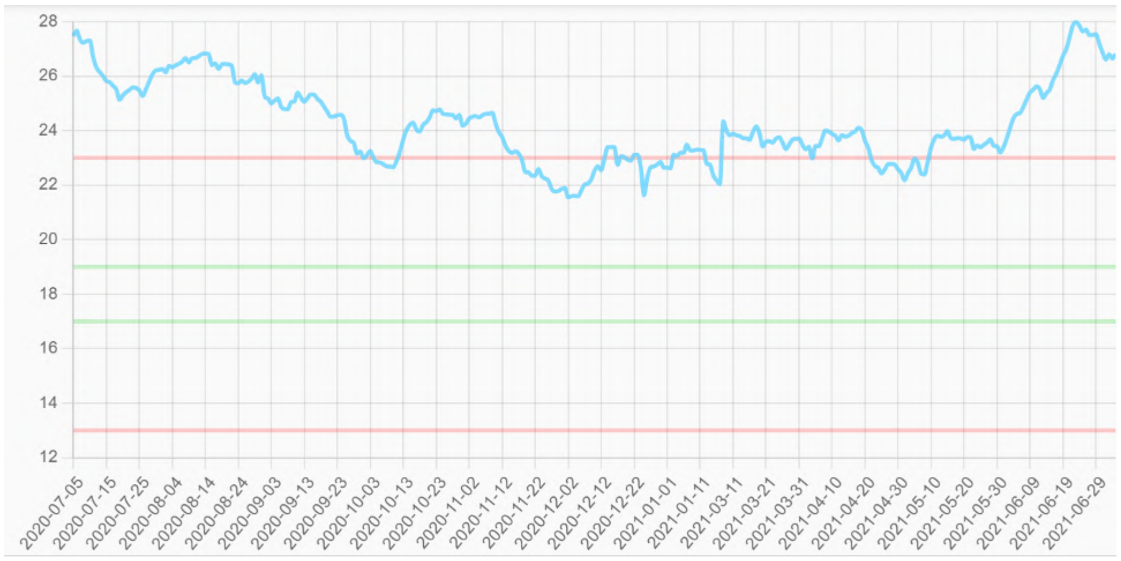


Figure 4.21: Temperature data obtained from the sensor inside Library room container

Figure 4.22 shows the relative humidity data obtained from the sensor inside

Library room container in the period from July 2020 to end of June 2021. From the image it can be observed that the conditions have been good and optimal until mid-November when became critical, then good and optimal again from June 2021.

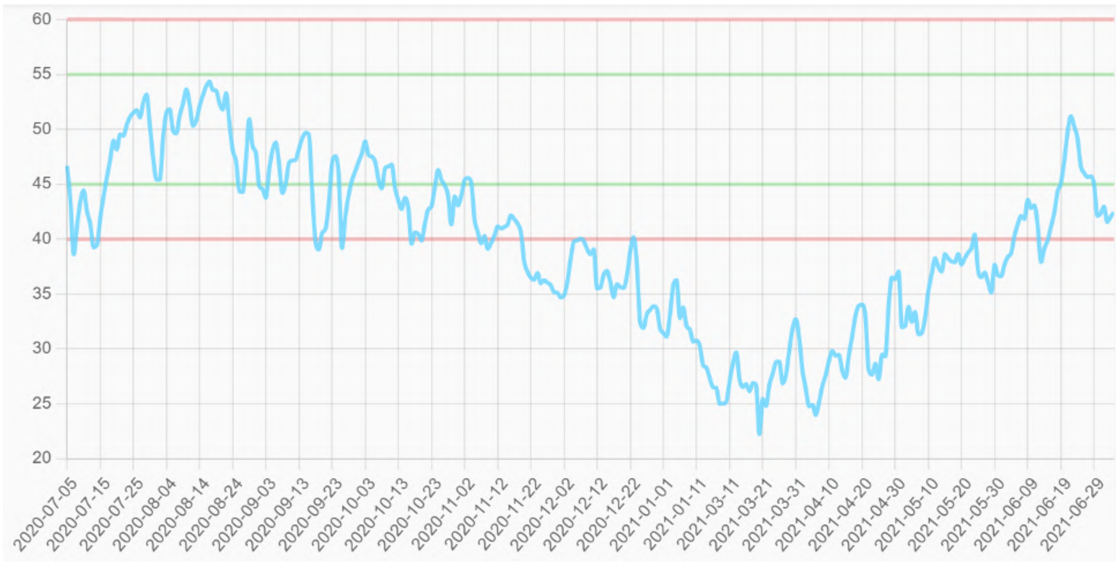


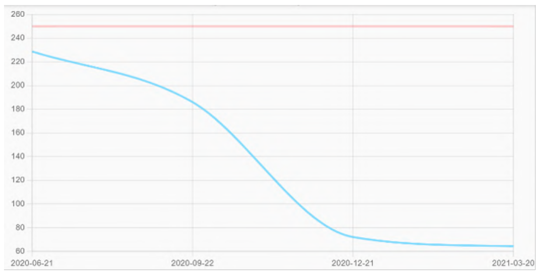
Figure 4.22: Relative humidity data obtained from the sensor inside Library room container

Figure 4.23 shows data obtained from the sensor inside Library room container in the period from mid-June 2020 to mid-March 2021 for Acetic acid, Formic acid, Formaldehyde, Acetaldehyde and $NO_2$. It can be observed that conditions inside the container have been good all the time for all VOCs.

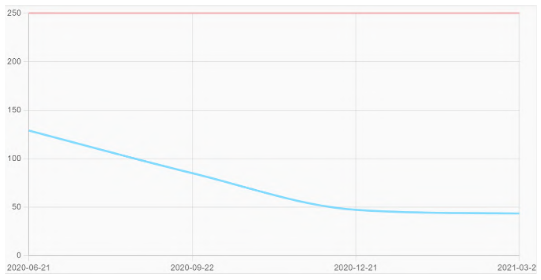
Table 4.7 shows the suggested preventive measures for Library room container.

Table 4.7: Suggested preventive measures for Library room container by agent of deterioration

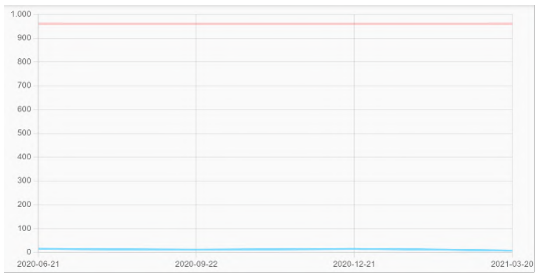
| Category | Name | Score |
|---|---|---|
| Light | Natural light shielding system | 6 |
| | Avoid areas of direct sunlight | 6 |
| | Cover windows | 6 |
| Temperature | Conservation heating | 6 |
| | Remove objects from external walls | 5 |
| | Cold storage (below 0°C) for unsuitable materials | 5 |
| Relative humidity | Dehumidifiers | 6 |
| | Insulate from walls | 5 |
| | Compensation of humidity fluctuations using graphene-based coatings | 4 |
| Pollutants | Layered double hydroxides (LDH) dispersed in heptane for acid sorption | 4 |
| | Layered double hydroxides (LDH) for acid sorption | 4 |
| | Layered double hydroxides (LDH) in PVA membrane for acid sorption | 4 |



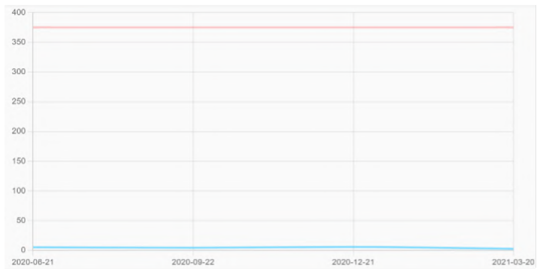
Acetic Acid



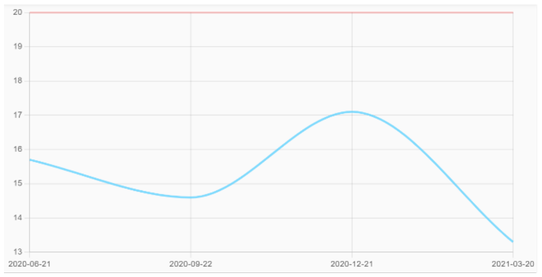
Formic Acid



Formaldehyde



Acetaldehyde



$NO_2$

Figure 4.23: All VOC data obtained from the sensor inside Storage Drawer container

### Collection of historical photographs

The Collection of historical photograph contains materials such as paper and photographic materials and it has a medium isolation. Humidity, temperature, and VOCs (Volatile Organic Compounds) sensors both inside and outside the display case are present.

Figure 4.24 shows the temperature data obtained from the sensor inside Collection of historical photographs container in the period from July 2020 to July 2021. From the image it can be observed that the conditions inside the container have been good until August 2020 when became critical and then optimal until the last measure in July 2021.

Figure 4.25 shows the relative humidity data obtained from the sensor inside Collection of historical photographs container in the period from July 2020 to July 2021. From the image it can be observed that the conditions have been good and optimal until January 2021 when became critical and then returned good.

Figure 4.26 shows data obtained from the sensor inside Collection of historical photographs container in the period from mid-June 2020 to mid-March 2021 for Acetic acid, Formic acid, Formaldehyde, Acetaldehyde and $NO_2$. It can be observed that conditions inside the container have been good all the time for all VOCs excluding Acetic Acid where were initially critical then became good.

Table 4.8 shows the suggested preventive measures for Collection of historical

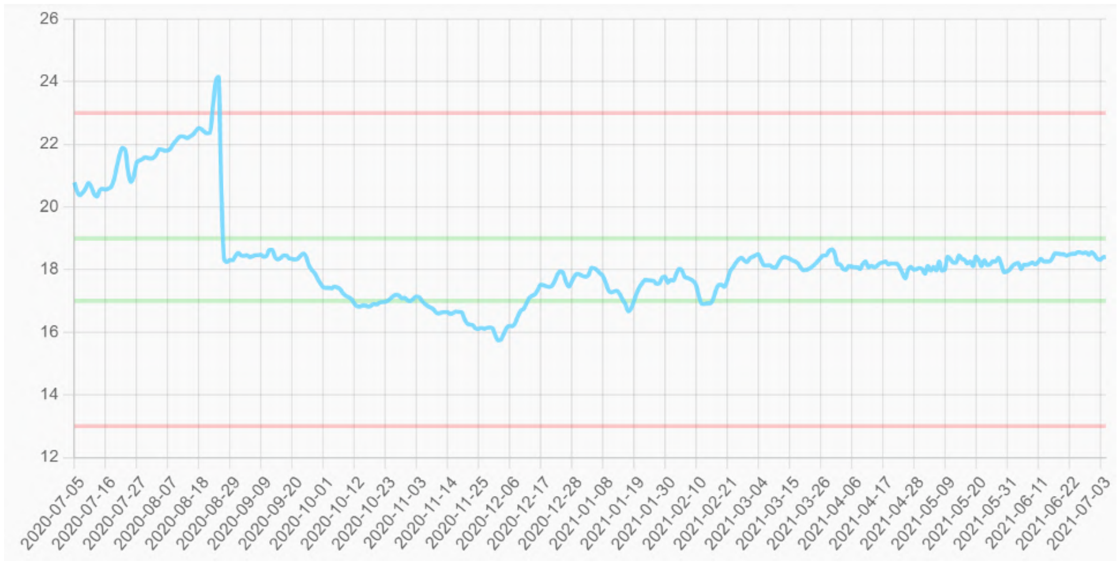


Figure 4.24: Temperature data obtained from the sensor inside Collection of historical photographs container

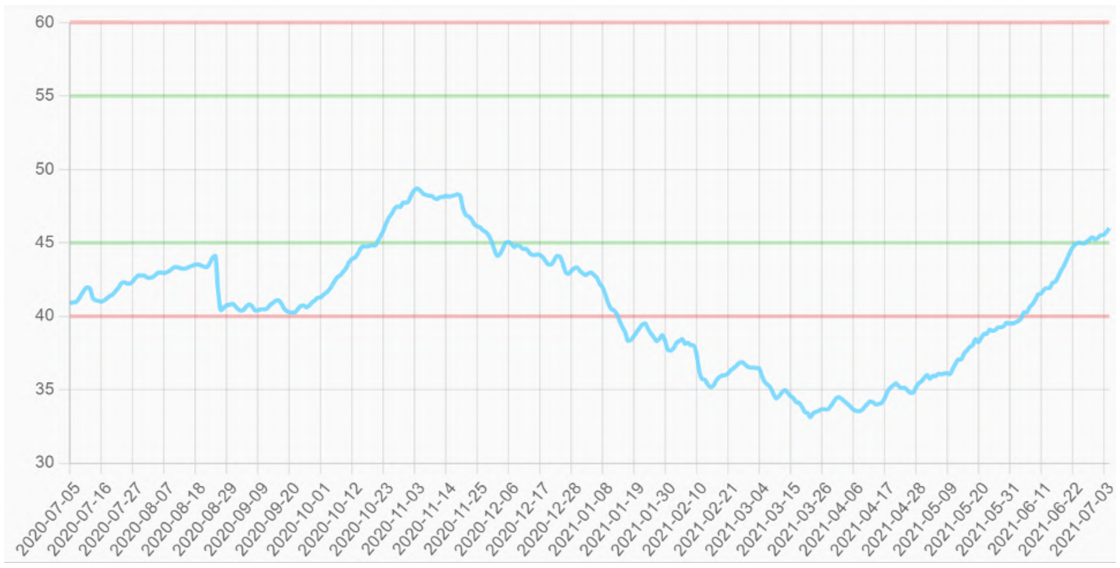


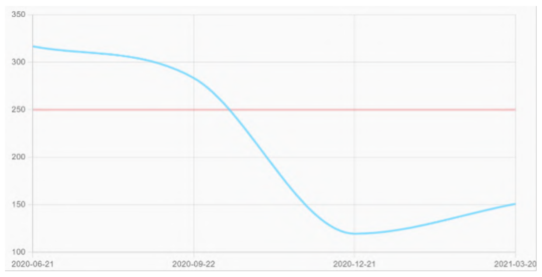
Figure 4.25: Relative humidity data obtained from the sensor inside Collection of historical photographs container

photographs container.

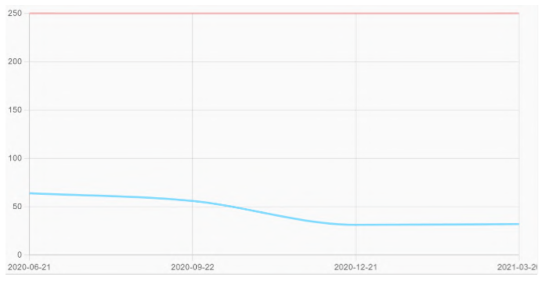
### Bronze Age objects

The Bronze Age object container contains materials such as metals, and it has a high isolation. Sensors are not present in this container, therefore only preventive measures derived by static data (data entered by collectors that are not dynamic / variable like those obtained from sensors) are reported.

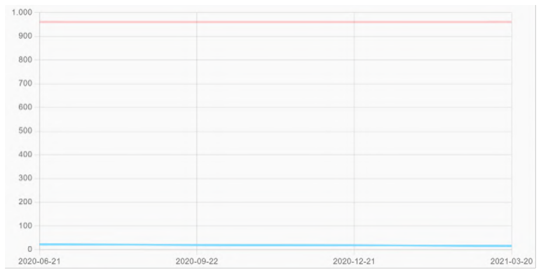
Table 4.9 shows the suggested preventive measures for Bronze age object container.



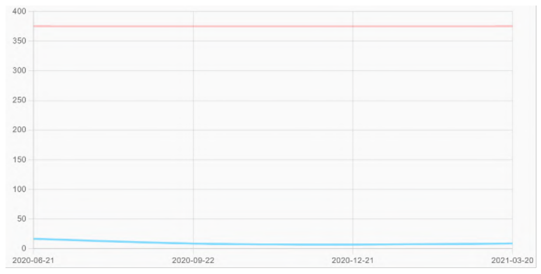
Acetic Acid



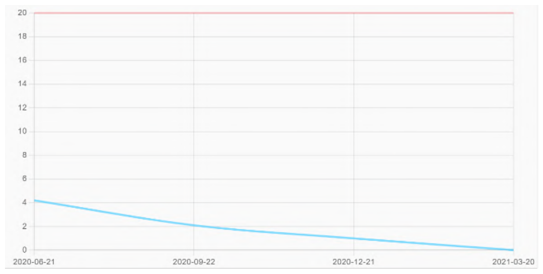
Formic Acid



Formaldehyde



Acetaldehyde



$NO_2$

Figure 4.26: All VOC data obtained from the sensor inside Collection of historical photographs container

Table 4.8: Suggested preventive measures for Collection of historical photographs container by agent of deterioration

| Category | Name | Score |
|---|---|---|
| Light | Natural light shielding system | 6 |
| | Avoid areas of direct sunlight | 6 |
| | Cover windows | 6 |
| Temperature | Conservation heating | 6 |
| | Remove objects from external walls | 5 |
| | Cold storage (below 0°C) for unsuitable materials | 5 |
| Relative humidity | Dehumidifiers | 6 |
| | Insulate from walls | 5 |
| | Compensation of humidity fluctuations using graphene-based coatings | 4 |
| Pollutants | Closed boxes (Conventional Archive Boxes) | 5 |
| | Storage in pollutant barrier bags | 4 |
| | Layered double hydroxides (LDH) dispersed in heptane for acid sorption | 4 |

### Weapon Storage Cabinet

The Weapon Storage Cabinet container contains materials such as wood, stone, plaster and metals and it has a medium isolation. Sensors are not present in this

Table 4.9: Suggested preventive measures for Bronze age object container by agent of deterioration

| Category | Name | Score |
|---|---|---|
| Light | Natural light shielding system | 6 |
| | Avoid areas of direct sunlight | 6 |
| | Cover windows | 6 |
| Temperature | Conservation heating | 6 |
| | Temperature monitoring | 6 |
| | Remove objects from external walls | 5 |
| Relative humidity | Dehumidifiers | 6 |
| | Relative humidity monitoring | 6 |
| | Insulate from wall | 5 |
| Pollutants | Graphene-based aerogels | 5 |
| | Air quality control (absorbents for VOCs) | 4 |
| | Absorbents for pollutants | 4 |

container, therefore only preventive measures derived by static data (data entered by collectors that are not dynamic / variable like those obtained from sensors) are reported.

Table 4.10 shows the suggested preventive measures for Weapon Storage Cabinet container.

Table 4.10: Suggested preventive measures for Weapon Storage Cabinet container by agent of deterioration

| Category | Name | Score |
|---|---|---|
| Light | Natural light shielding system | 6 |
| | Avoid areas of direct sunlight | 6 |
| | Cover windows | 6 |
| Temperature | Conservation heating | 6 |
| | Temperature monitoring | 6 |
| | Remove objects from external walls | 5 |
| Relative humidity | Dehumidifiers | 6 |
| | Relative humidity monitoring | 6 |
| | Insulate from wall | 5 |
| Pollutants | Layered double hydroxides (LDH) dispersed in heptane for acid sorption | 4 |
| | Layered double hydroxides (LDH) for acid sorption | 4 |
| | Layered double hydroxides (LDH) in PVA membrane for acid sorption | 4 |

### The Szechenyi Hall

The Szechenyi Hall container contains materials such as paper, wood, glass and acrylic paint and it has a medium isolation. Sensors are not present in this container, therefore only preventive measures derived by static data (data entered by collectors that are not dynamic / variable like those obtained from sensors) are reported.

Table 4.11 shows the suggested preventive measures for Szechenyi Hall container.

Table 4.11: Suggested preventive measures for Szechenyi Hall container by agent of deterioration

| Category | Name | Score |
|---|---|---|
| Light | Natural light shielding system | 6 |
| | Avoid areas of direct sunlight | 6 |
| | Cover windows | 6 |
| Temperature | Conservation heating | 6 |
| | Temperature monitoring | 6 |
| | Remove objects from external walls | 5 |
| Relative humidity | Dehumidifiers | 6 |
| | Relative humidity monitoring | 6 |
| | Insulate from wall | 5 |
| Pollutants | Layered double hydroxides (LDH) dispersed in heptane for acid sorption | 4 |
| | Layered double hydroxides (LDH) for acid sorption | 4 |
| | Layered double hydroxides (LDH) in PVA membrane for acid sorption | 4 |

### 4.2.4 Ministry of Culture, Italy (MiC)

The Ministry of Culture in Italy (in Figure 4.27) exercises the powers of the State in matters of protection of cultural and environmental assets and implementation of policies relating to tourism. The art objects considered for this case study are located in the Chieti state archive. Five containers were selected for this place, Compactable metal racks semi-open, 1800's historical books, Metal shelves, Closed metal document box, Closed box of synthetic material in metal shelves.

Figure 4.27: Ministry of Culture (MiC)

Conditions have often been good both inside and outside the containers for temperature, while they were often critical for relative humidity (relative humidity has been always critical in the two containers Compactable metal racks semi-open and Closed metal document box). The conditions have always been good both inside and outside the containers.

### Compactable metal racks semi-open

The Compactable metal racks semi-open container contains materials such as paper and it has a medium isolation. Temperature, relative humidity, and VOCs (Volatile Organic Compounds) sensors are present outside the metal bookshelf.

Figure 4.28 shows the temperature data obtained from the sensor outside Compactable metal racks semi-open container in the period from the end of February 2020 to January 2021. From the image it can be observed that the conditions outside the container have been good and optimal most of the time except during some days in August 2020 when conditions became critical.

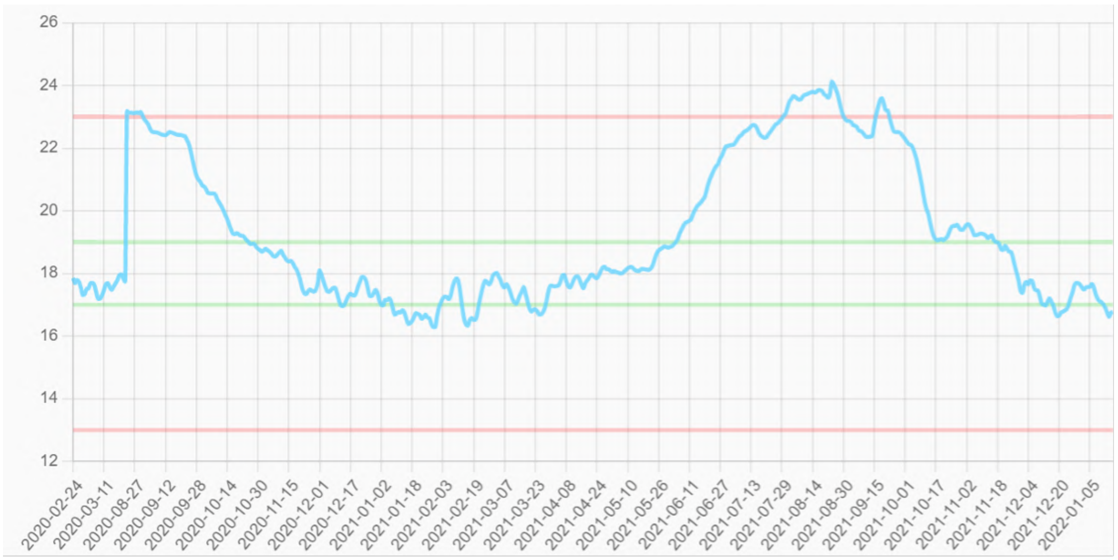


Figure 4.28: Temperature data obtained from the sensor outside Compactable metal racks semi-open container

Figure 4.29 shows the relative humidity data obtained from the sensor outside Compactable metal racks semi-open container in the period from the end of February 2020 to January 2021. From the image it can be observed that the conditions have always been critical.

Figure 4.30 shows data obtained from the sensor outside Compactable metal racks semi-open container in the period from mid-June 2020 to mid-March 2021 for Acetic acid, Formic acid, Formaldehyde, Acetaldehyde and $NO_2$. It can be observed that conditions outside the container have been good all the time for all VOCs.

Table 4.12 shows the suggested preventive measures for Compactable metal racks semi-open container.

### 1800's historical books

The 1800's historical books container contains materials such as paper and it has a low isolation. Temperature, Relative humidity sensors and VOCs (Volatile Organic Compounds) sensors inside the container are present.

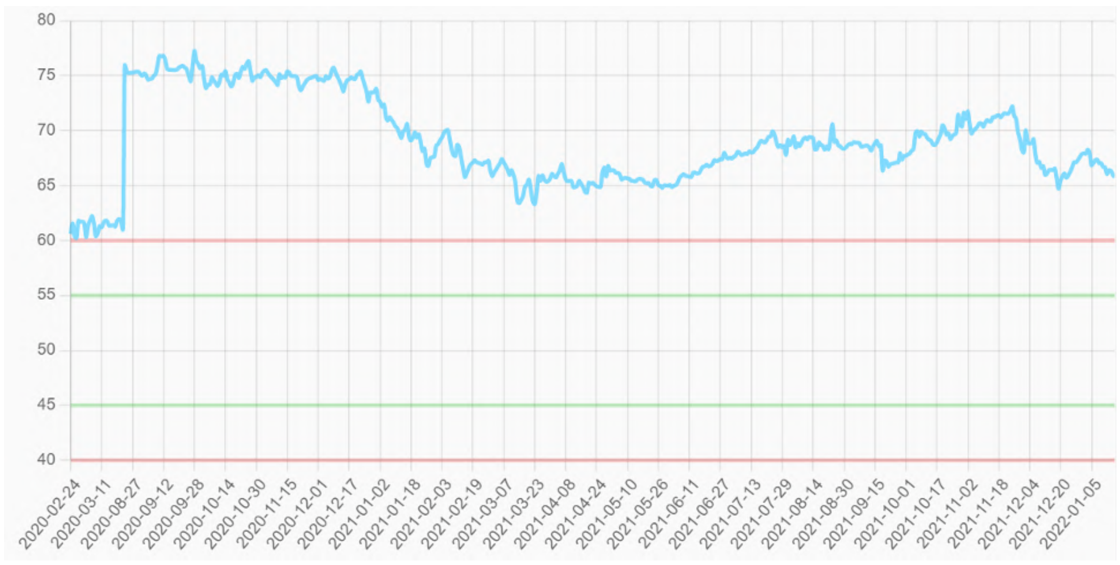


Figure 4.29: Relative humidity data obtained from the sensor outside Compactable metal racks semi-open container

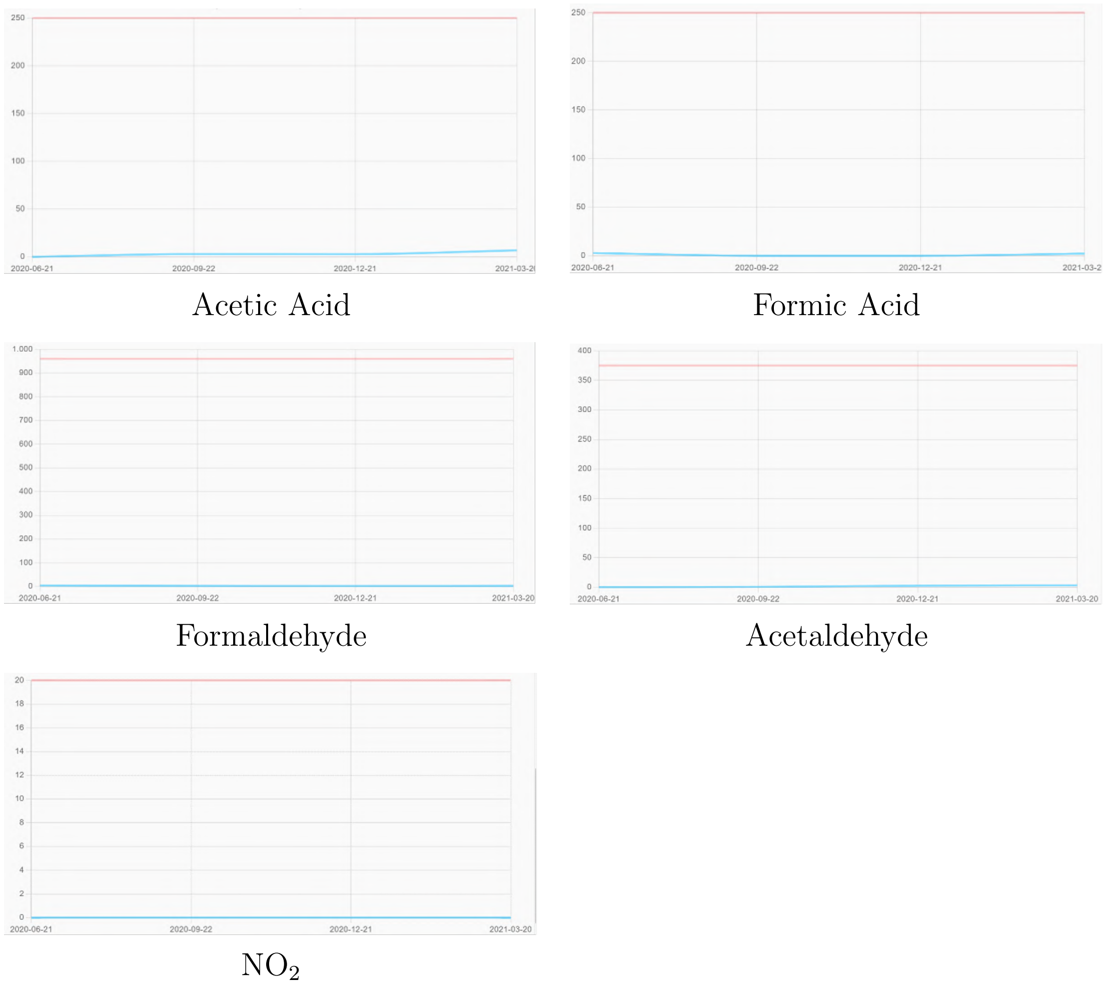


Figure 4.30: All VOC data obtained from the sensor outside Compactable metal racks semi-open container

Figure 4.31 shows the temperature data obtained from the sensor inside 1800's historical books container in the period from the end of February 2020 to January 2021. From the image it can be observed that the conditions inside the container

Table 4.12: Suggested preventive measures for Compactable metal racks semi-open container by agent of deterioration

| Category | Name | Score |
|---|---|---|
| Light | Natural light shielding system | 6 |
| | Avoid areas of direct sunlight | 6 |
| | Cover windows | 6 |
| Temperature | Conservation heating | 6 |
| | Remove objects from external walls | 5 |
| | Cold storage (below 0°C) for unstable materials | 5 |
| Relative humidity | Dehumidifiers | 6 |
| | Insulate from wall | 5 |
| | Closed boxes (Conventional Archive Boxes) | 5 |
| Pollutants | Closed boxes (Conventional Archive Boxes) | 5 |
| | Storage in pollutant barrier bags | 4 |
| | Layered double hydroxides (LDH) dispersed in heptane for acid sorption | 4 |

have been initially critical, good until June 2020, critical for two months and then optimal and good.

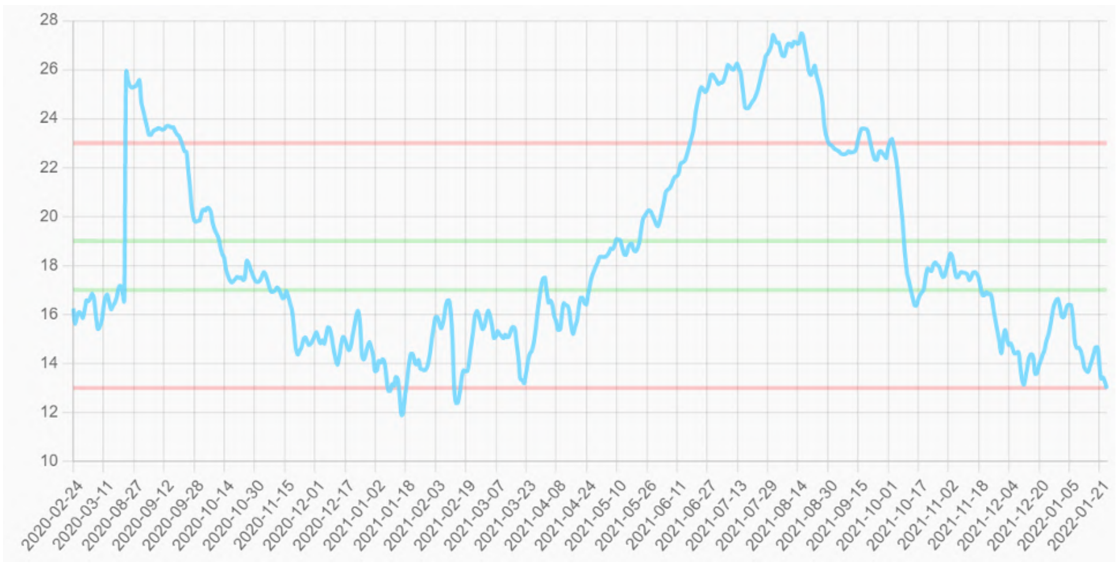


Figure 4.31: Temperature data obtained from the sensor inside 1800's historical books container

Figure 4.32 shows the relative humidity data obtained from the sensor inside 1800's historical books container in the period from the end of February 2020 to the end of January 2022. From the image it can be observed that the conditions have been initially good, critical during summer 2021, good until November 2021 when became critical and then good and optimal.

Figure 4.33 shows data obtained from the sensor inside 1800's historical books container in the period from mid-June 2020 to mid-March 2021 for Acetic acid, Formic acid, Formaldehyde, Acetaldehyde and $NO_2$. It can be observed that conditions outside the container have been good all the time for all VOCs.

Table 4.13 shows the suggested preventive measures for 1800's historical books container.

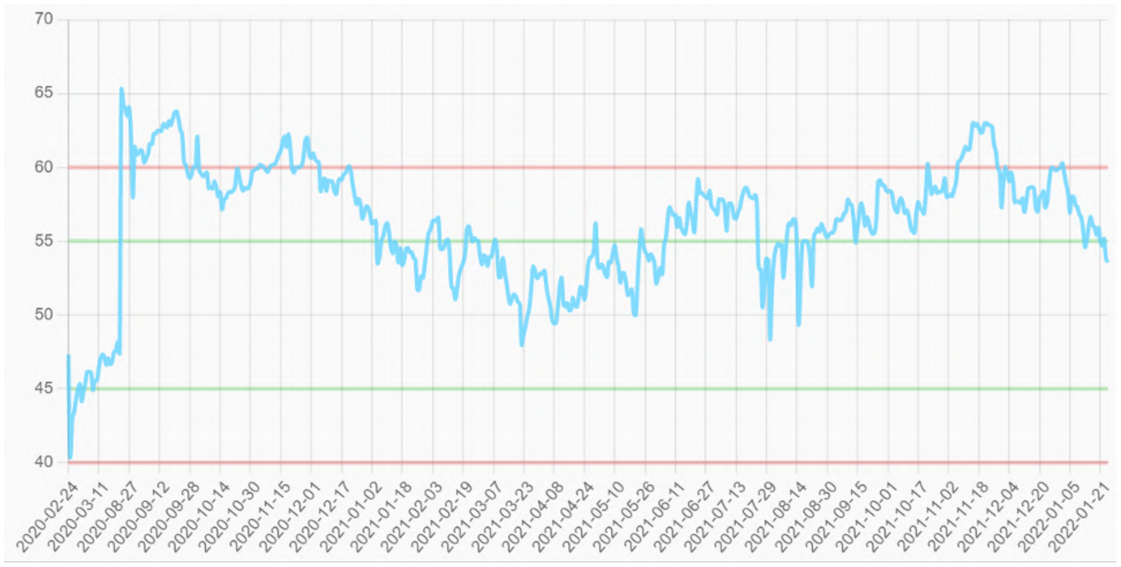


Figure 4.32: Relative humidity data obtained from the sensor inside 1800's historical books container

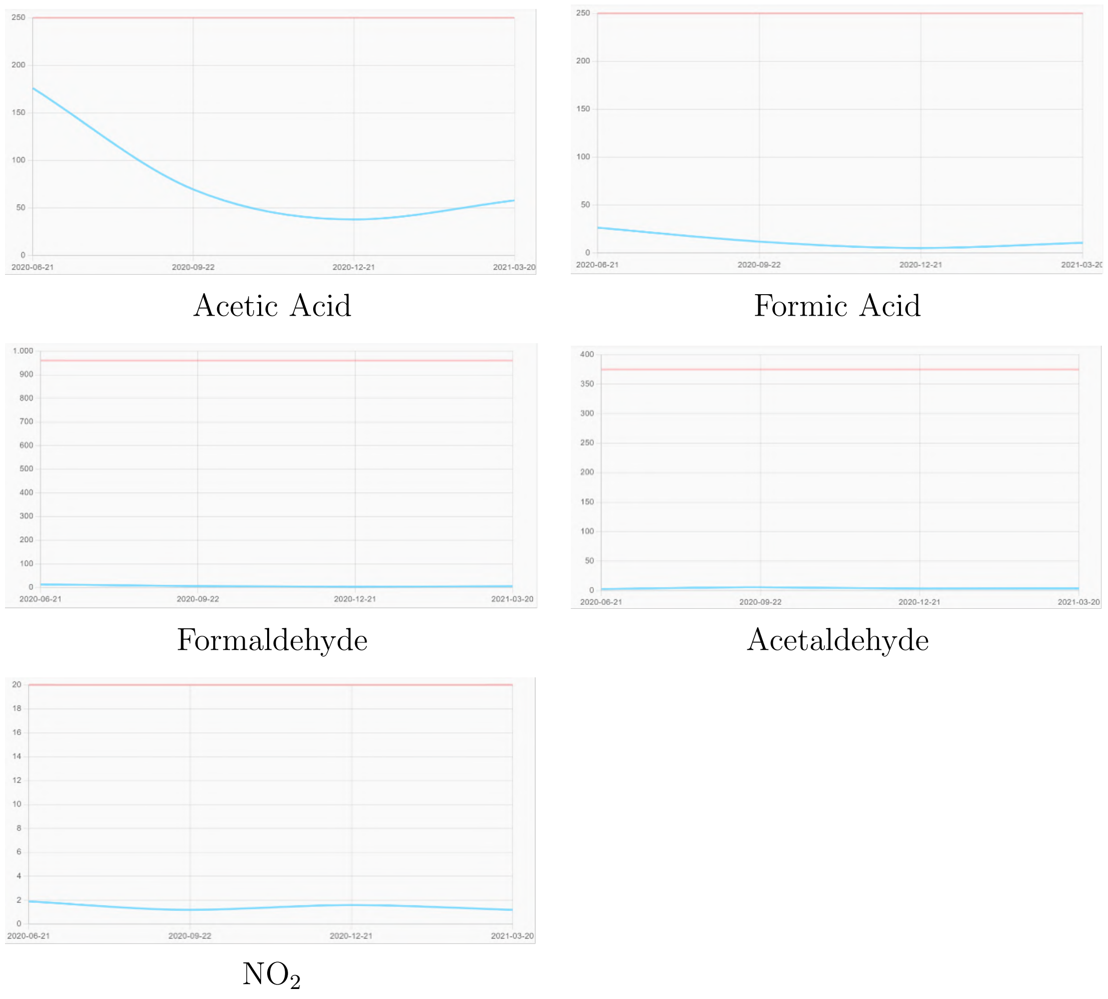


Figure 4.33: All VOC data obtained from the sensor inside 1800's historical books container

## Closed metal document box

The Closed metal document box contains materials such as paper and it has a medium isolation. Temperature, Relative humidity VOCs (Volatile Organic Com-

Table 4.13: Suggested preventive measures for 1800's historical books container by agent of deterioration

| Category | Name | Score |
|---|---|---|
| Light | Natural light shielding system | 6 |
| | Avoid areas of direct sunlight | 6 |
| | Cover windows | 6 |
| Temperature | Conservation heating | 6 |
| | Remove objects from external walls | 5 |
| | Cold storage (below 0°C) for unstable materials | 5 |
| Relative humidity | Dehumidifiers | 6 |
| | Insulate from wall | 5 |
| | Closed boxes (Conventional Archive Boxes) | 5 |
| Pollutants | Closed boxes (Conventional Archive Boxes) | 5 |
| | Storage in pollutant barrier bags | 4 |
| | Layered double hydroxides (LDH) dispersed in heptane for acid sorption | 4 |

pounds) sensors inside the container are present.

Figure 4.34 shows the temperature data obtained from the sensor inside Closed metal document box container in the period from the end of February 2020 to January 2022. From the image it can be observed that the conditions inside the container have been mostly good until July 2021 when became critical for one months, then good and optimal again.

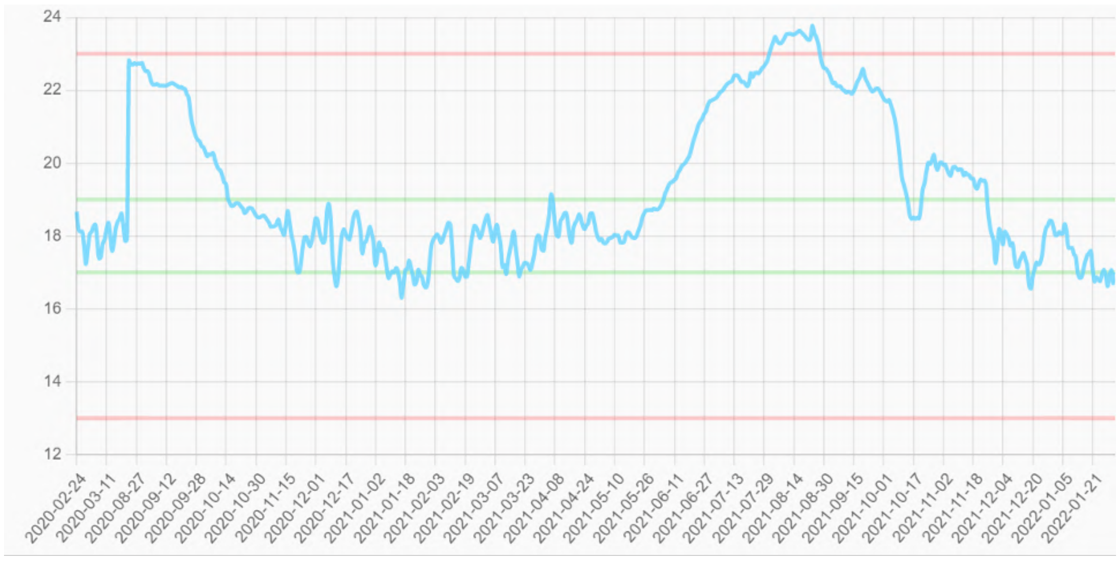


Figure 4.34: Temperature data obtained from the sensor inside Closed metal document box container

Figure 4.35 shows the relative humidity data obtained from the sensor inside Closed metal document box container in the period from the end of February 2020 to the end of January 2022. From the image it can be observed that the conditions have always been critical.

Figure 4.36 shows data obtained from the sensor inside Closed metal document box container in the period from mid-June 2020 to mid-March 2021 for Acetic acid, Formic acid, Formaldehyde, Acetaldehyde and $NO_2$. It can be observed that conditions inside the container have been good all the time for all VOCs.

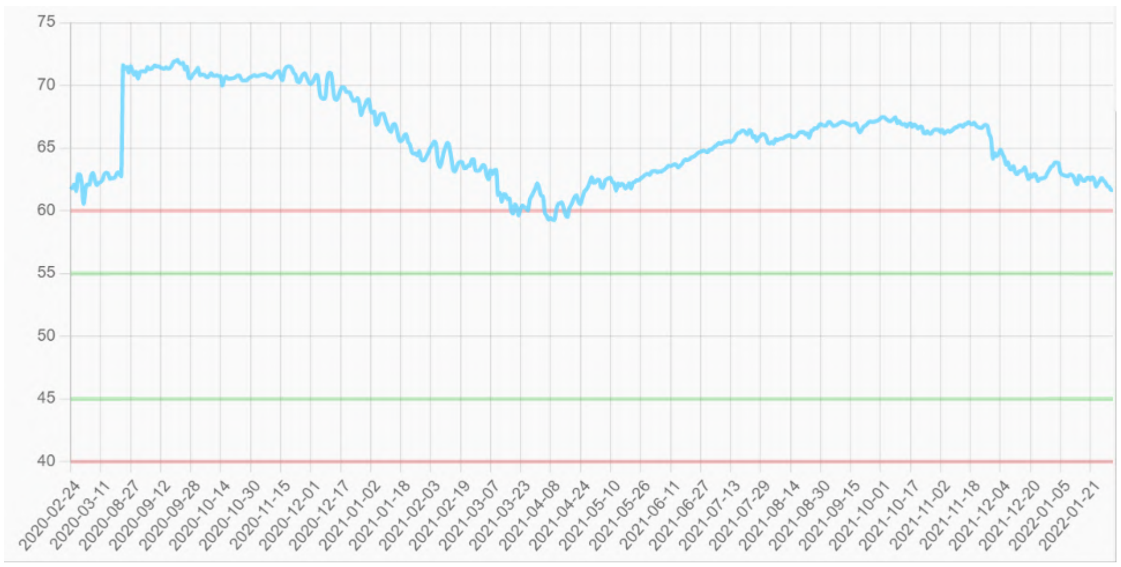
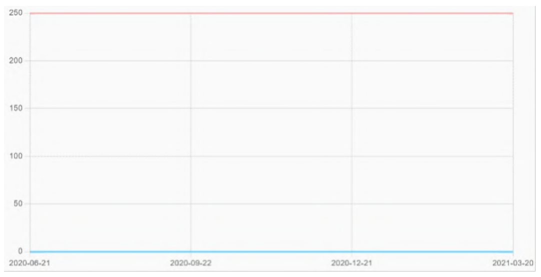
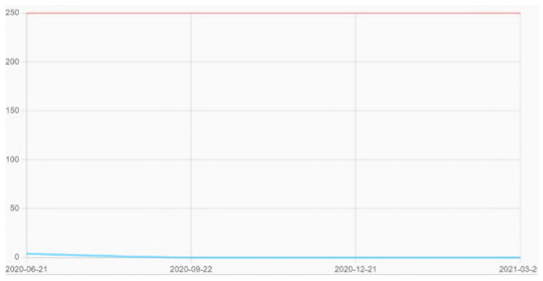


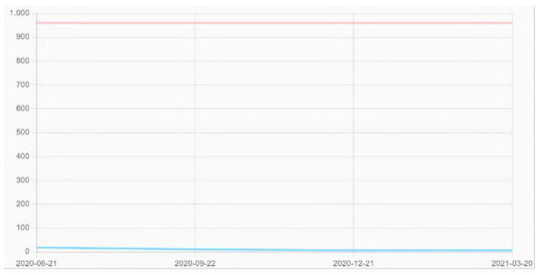
Figure 4.35: Relative humidity data obtained from the sensor inside Closed metal document box container



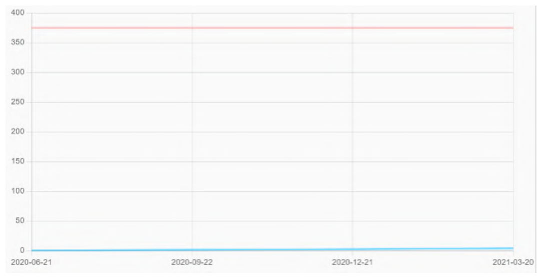
Acetic Acid



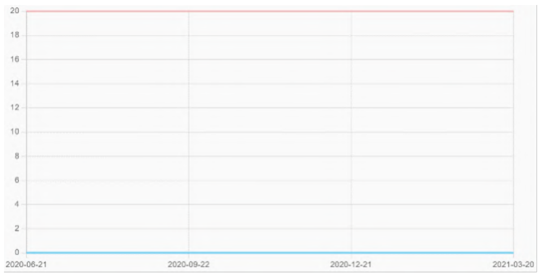
Formic Acid



Formaldehyde



Acetaldehyde



$NO_2$

Figure 4.36: All VOC data obtained from the sensor inside Closed metal document box container

Table 4.14 shows the suggested preventive measures for Closed metal document box container.

Table 4.14: Suggested preventive measures for Closed metal document box container by agent of deterioration

| Category | Name | Score |
|---|---|---|
| Light | Natural light shielding system | 6 |
| | Avoid areas of direct sunlight | 6 |
| | Cover windows | 6 |
| Temperature | Conservation heating | 6 |
| | Remove objects from external walls | 5 |
| | Cold storage (below 0°C) for unstable materials | 5 |
| Relative humidity | Dehumidifiers | 6 |
| | Insulate from wall | 5 |
| | Closed boxes (Conventional Archive Boxes) | 5 |
| Pollutants | Closed boxes (Conventional Archive Boxes) | 5 |
| | Storage in pollutant barrier bags | 4 |
| | Layered double hydroxides (LDH) dispersed in heptane for acid sorption | 4 |

### 4.2.5 Fondazione Scienza e Tecnica

The Science and Technology Foundation of Florence (in Figure 4.37), founded in 1987, aims to promote and disseminate scientific, historical and technological culture, through the recovery and enhancement of its extraordinary historical and scientific heritage. The art objects considered for this case study are located in two places named ”Museum” and ”Storage”. For the first place two containers were selected, the Scientific instruments and Models of plants, and for the second place one container was selected, Technological and pharmaceutical herbarium.

Figure 4.37: Fondazione Scienza e Tecnica

The conditions were often good or optimal both inside and outside the containers for temperature and relative humidity. Inside two out of three containers, a VOC presented critical levels. Outside a container, two VOCs presented critical levels.

### Scientific instruments

The Scientific instruments container contains materials such as glass, ceramics, wood, stone, plaster and metals and it has a medium isolation. Humidity, temperature, and VOCs (Volatile Organic Compounds) sensors both inside and outside the container are present.

Figure 4.38 shows the temperature data obtained from the sensor inside Scientific instruments container in the period from mid-December 2019 to October 2021. From the image it can be observed that the conditions inside the container have been good and optimal most of the time except during the summer period between June and October 2020 and June and October 2021 when the condition became critical.

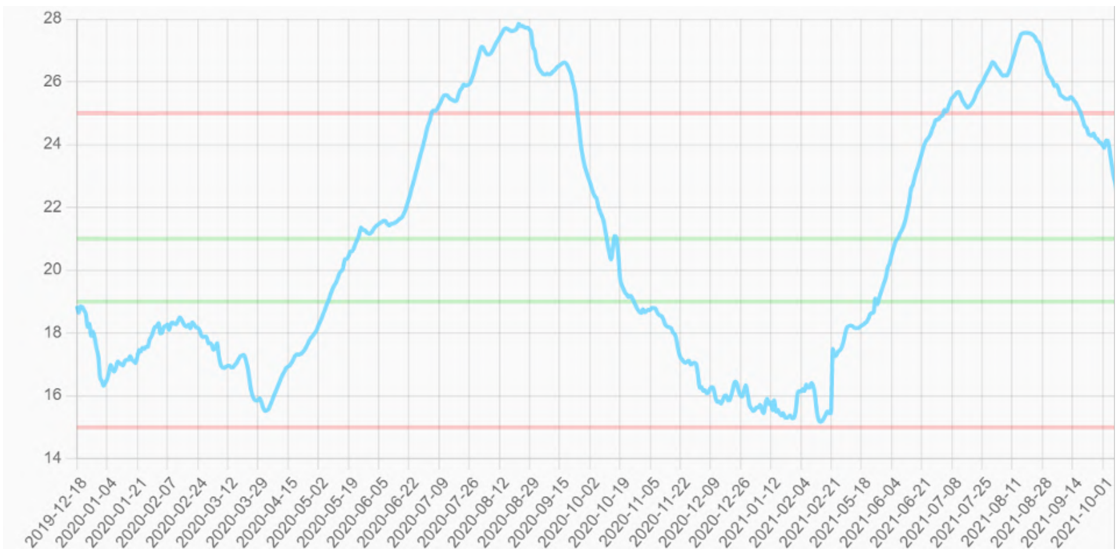


Figure 4.38: Temperature data obtained from the sensor inside Scientific instruments container

Figure 4.39 shows the relative humidity data obtained from the sensor inside Scientific instruments container from mid-December 2019 to October 2021. It can be observed that the conditions inside the container have been good and optimal most of the time except during the summer period between June and August 2020, in November 2020 and between June and August 2021 when the condition became critical.

Figure 4.40 shows data obtained from the sensor inside Scientific instruments container in the period from mid-June 2020 to mid-March 2021 for Acetic acid, Formic acid, Formaldehyde, Acetaldehyde and $NO_2$. It can be observed that conditions inside the container have been good all the time for all VOCs excluding Acetic Acid where initially were critical and then became good.

Table 4.15 shows the suggested preventive measures for Scientific instruments container.

### Models of plants

The Models of plants contains materials such as paper, glass, ceramics, wood, stone, plaster and metals and it has a hight isolation. Humidity, temperature, and VOCs (Volatile Organic Compounds) sensors both inside and outside the container are present.

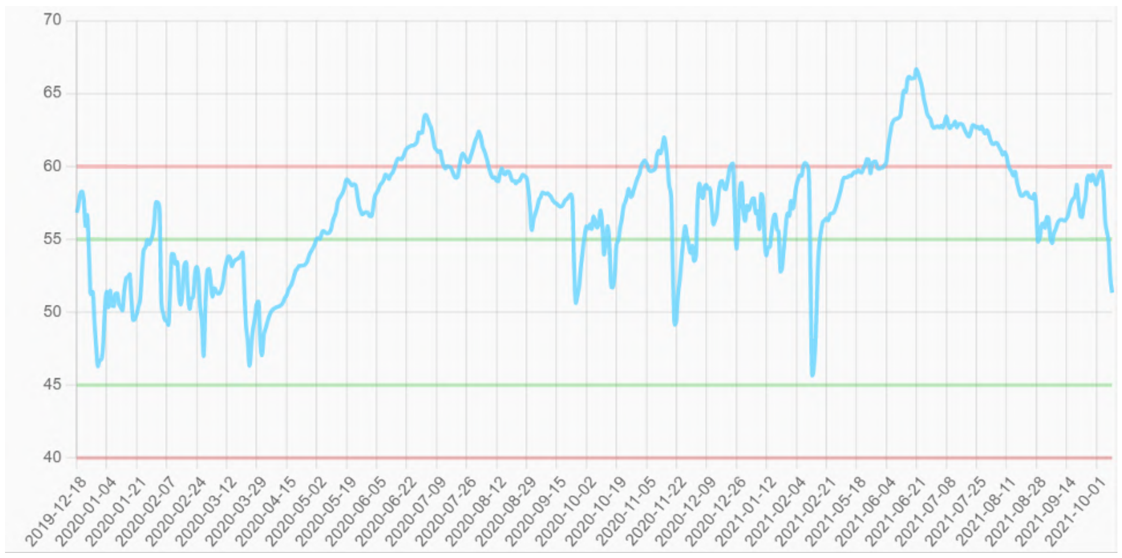


Figure 4.39: Temperature data obtained from the sensor inside Scientific instruments container

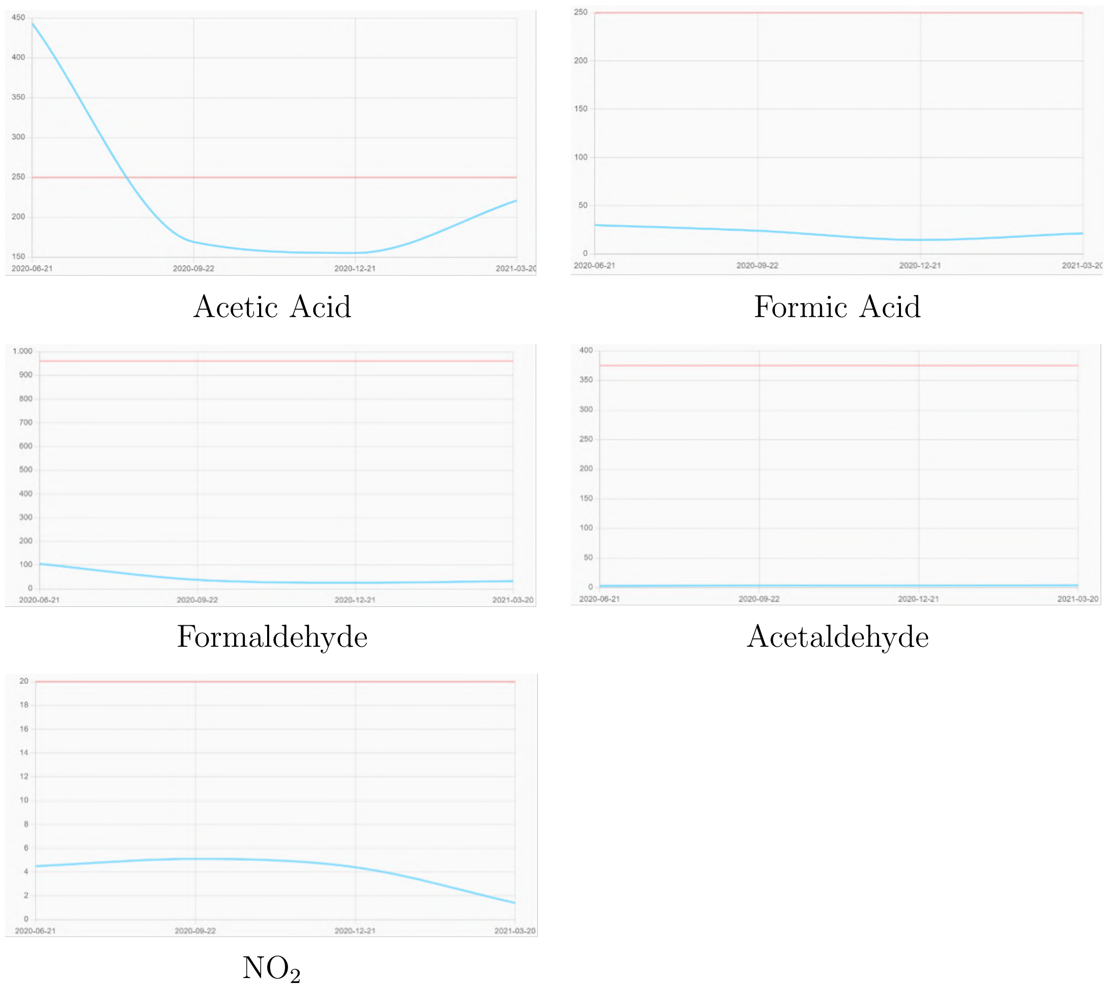


Figure 4.40: All VOC data obtained from the sensor inside Scientific instruments container

Figure 4.41 shows the temperature data obtained from the sensor inside Models of plants container in the period from mid-December 2019 to October 2021. From the image it can be observed that the conditions inside the container have been

Table 4.15: Suggested preventive measures for Scientific instruments container

| Category | Name | Score |
|---|---|---|
| Light | Natural light shielding system | 6 |
| | Avoid areas of direct sunlight | 6 |
| | Cover windows | 6 |
| Temperature | Conservation heating | 6 |
| | Remove objects from external walls | 5 |
| | Cold storage (below 0°C) for unstable materials | 5 |
| Relative humidity | Dehumidifiers | 6 |
| | Insulate from wall | 5 |
| | Compensation of humidity fluctuations using graphene-based coatings | 5 |
| Pollutants | Layered double hydroxides (LDH) dispersed in heptane for acid sorption | 4 |
| | Layered double hydroxides (LDH) for acid sorption | 4 |
| | Layered double hydroxides (LDH) in PVA membrane for acid sorption | 4 |

good and optimal most of the time except during the summer period between June and October 2020 and June and October 2021 when the condition became critical.

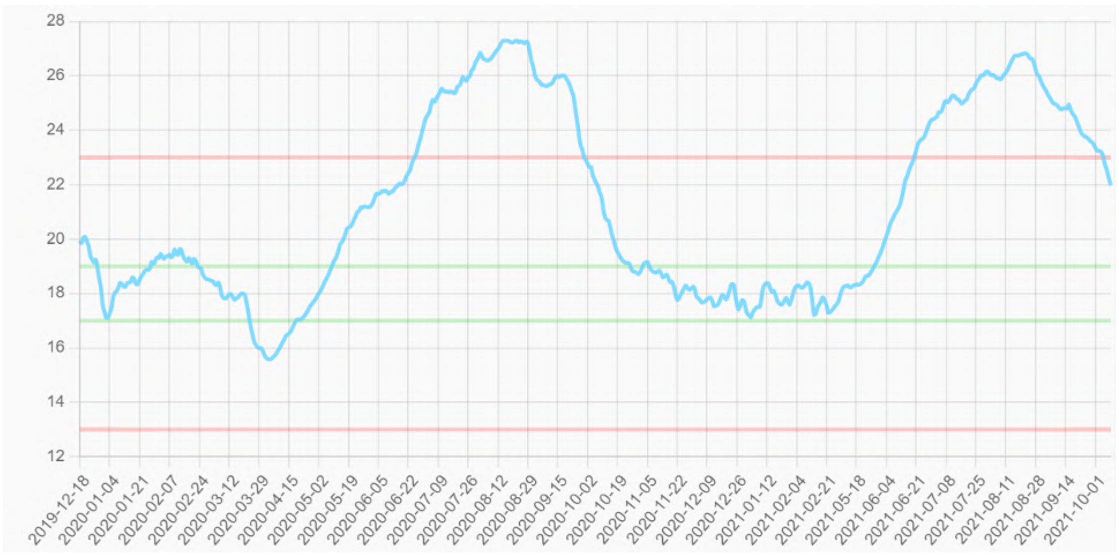


Figure 4.41: Temperature data obtained from the sensor inside Models of plants container

Figure 4.42 shows the relative humidity data obtained from the sensor inside Models of plants container from mid-December 2019 to October 2021. It can be observed that the conditions inside the container have been good and optimal most of the time except during the summer period between May and July 2020, in November 2020, from February to July 2021 and from September 2021 when the conditions became critical.

Figure 4.43 shows data obtained from the sensor inside Models of plants container in the period from mid-June 2020 to mid-March 2021 for Acetic acid, Formic

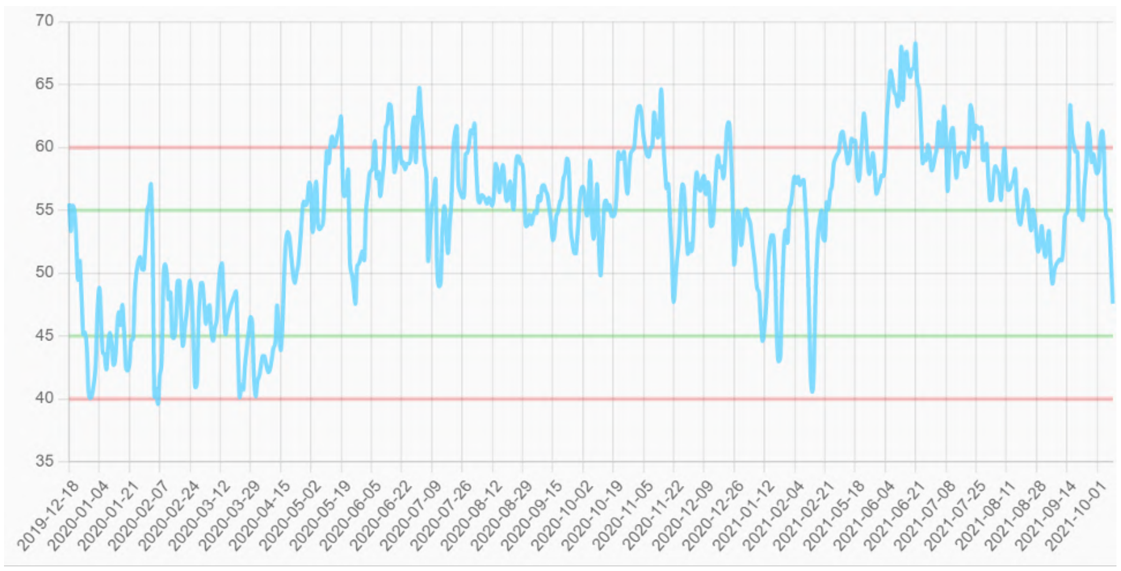
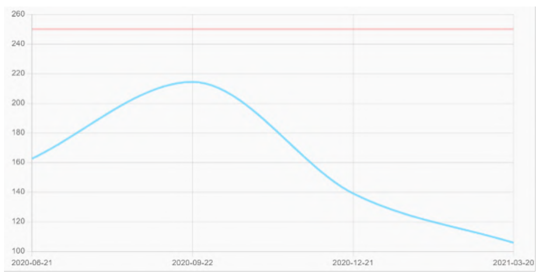
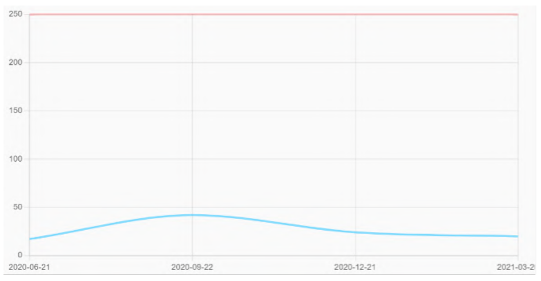


Figure 4.42: Relative humidity data obtained from the sensor inside Models of plants container

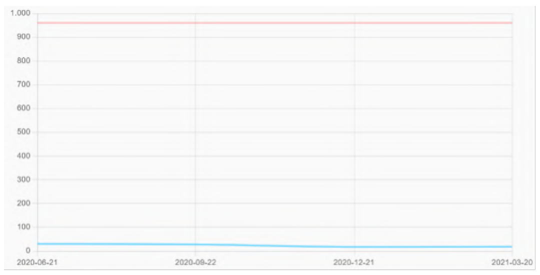
acid, Formaldehyde, Acetaldehyde and $NO_2$. It can be observed that conditions inside the container have been good all the time for all VOCs.



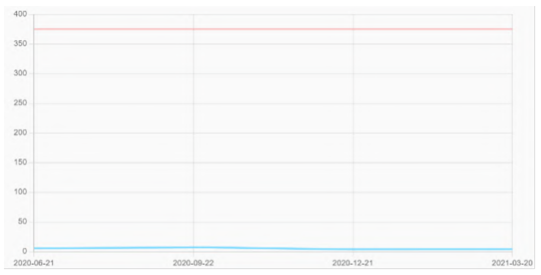
Acetic Acid



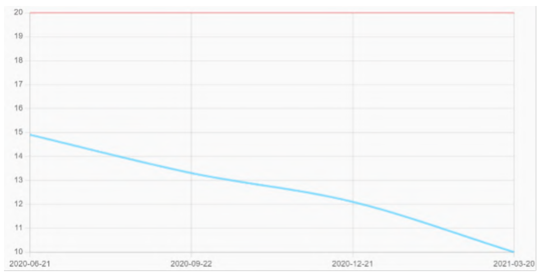
Formic Acid



Formaldehyde



Acetaldehyde



$NO_2$

Figure 4.43: All VOC data obtained from the sensor inside Models of plants container

Table 4.16 shows the suggested preventive measures for Models of plants container.

Table 4.16: Suggested preventive measures for Models of plants container

| Category | Name | Score |
|---|---|---|
| Light | Natural light shielding system | 6 |
| | Avoid areas of direct sunlight | 6 |
| | Cover windows | 6 |
| Temperature | Conservation heating | 6 |
| | Remove objects from external walls | 5 |
| | Cold storage (below 0°C) for unstable materials | 5 |
| Relative humidity | Dehumidifiers | 6 |
| | Insulate from wall | 5 |
| | Super absorbent polymers (SAPs) dispersed in heptane for humidity sorption | 4 |
| Pollutants | Graphene-based aerogels | 5 |
| | Air quality control (absorbents for VOCs) | 4 |
| | Absorbents for pollutants | 4 |

### Technological and pharmaceutical herbarium

The Technological and pharmaceutical herbarium contains materials such as paper, and it has a medium isolation. Humidity, temperature, and VOCs (Volatile Organic Compounds) sensors both inside and outside the container are present.

Figure 4.44 shows the temperature data obtained from the sensor inside Technological and pharmaceutical herbarium container in the period from mid-December 2019 to the end of September 2021. From the image it can be observed that the conditions inside the container have been good and optimal most of the time except during the summer period between June and October 2020 and June and September 2021 when the condition became critical.

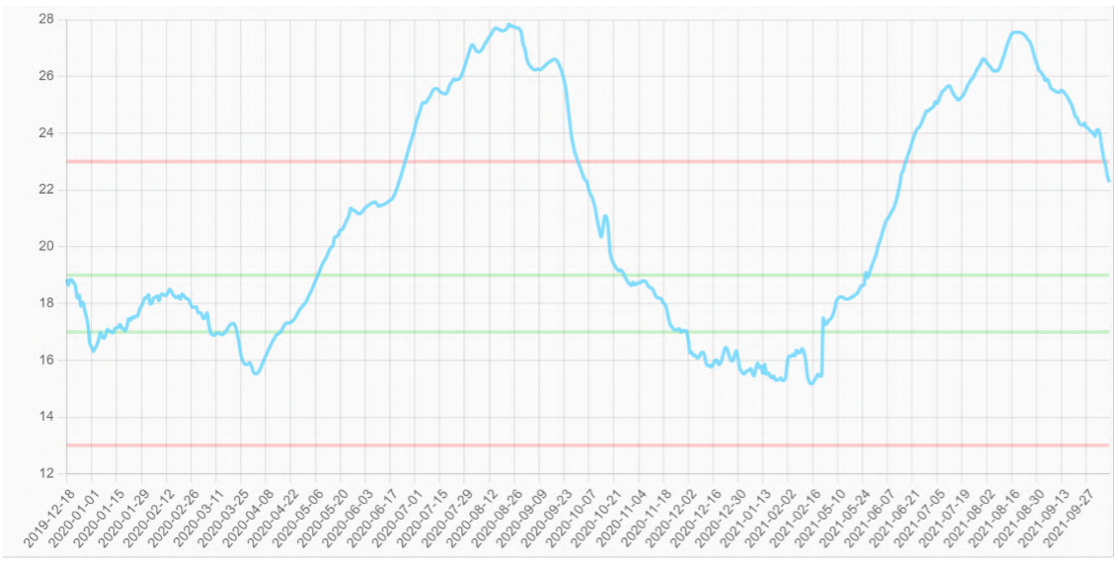


Figure 4.44: Temperature data obtained from the sensor inside Technological and pharmaceutical herbarium container

Figure 4.45 shows the relative humidity data obtained from the sensor inside Technological and pharmaceutical herbarium container from mid-December 2019 to the end of September 2021. It can be observed that the conditions inside the container have been good and optimal most of the time except during the summer period between June and August 2020, in November 2020 and between June and August 2021 when the condition became critical.

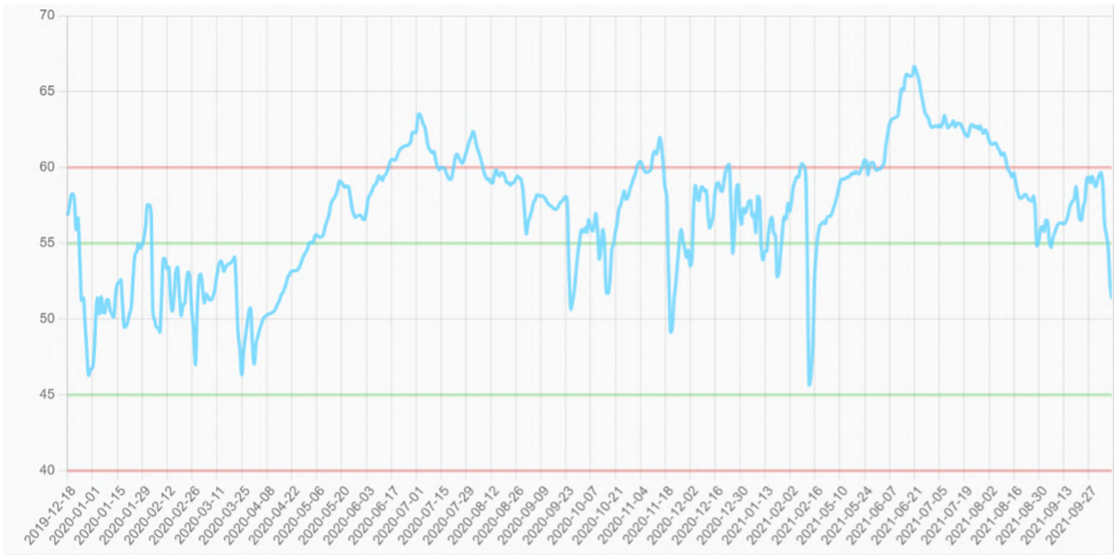


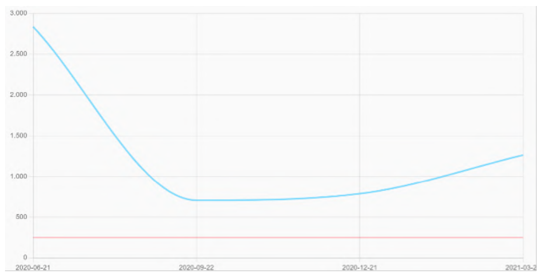
Figure 4.45: Relative humidity data obtained from the sensor inside Technological and pharmaceutical herbarium container

Figure 4.46 shows data obtained from the sensor inside Technological and pharmaceutical herbarium container in the period from mid-June 2020 to mid-March 2021 for Acetic acid, Formic acid, Formaldehyde, Acetaldehyde and $NO_2$. It can be observed that conditions inside the container have been good all the time for all VOCs excluding Acetic Acid that has been critical all the time.

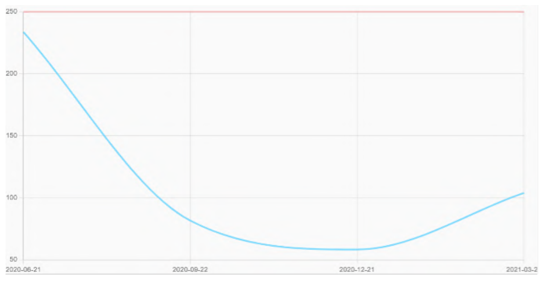
Table 4.17 shows the suggested preventive measures for Technological and pharmaceutical herbarium container.

Table 4.17: Suggested preventive measures for Technological and pharmaceutical herbarium container

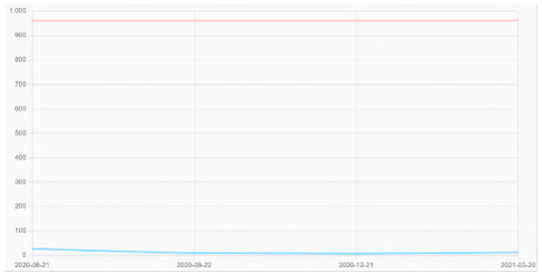
| Category | Name | Score |
|---|---|---|
| Light | Natural light shielding system | 6 |
| | Avoid areas of direct sunlight | 6 |
| | Cover windows | 6 |
| Temperature | Conservation heating | 6 |
| | Remove objects from external walls | 5 |
| | Cold storage (below 0°C) for unstable materials | 5 |
| Relative humidity | Dehumidifiers | 6 |
| | Insulate from wall | 5 |
| | Closed boxes (Conventional Archive Boxes) | 5 |
| Pollutants | Closed boxes (Conventional Archive Boxes) | 5 |
| | Storage in pollutants barrier bags | 4 |
| | Layered double hydroides (LDH) dispersed in heptane for acid sorption | 4 |



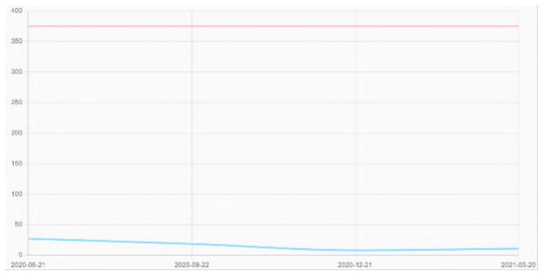
Acetic Acid



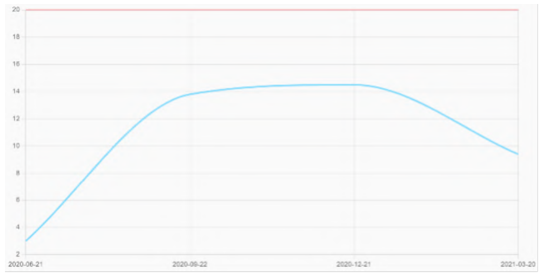
Formic Acid



Formaldehyde



Acetaldehyde



$NO_2$

Figure 4.46: All VOC data obtained from the sensor inside Technological and pharmaceutical herbarium container

### 4.2.6 Centre Pompidou

Centre national d'art et de culture Georges-Pompidou (in Figure 4.47), commonly known as the Centre Pompidou is located in the Beaubourg area of the 4th arrondissement of Paris. The Centre is famous for its iconic 20th-century building, designed by Renzo Piano and Richard Rogers, which is immediately recognizable by its exterior escalators and enormous coloured tubing. It is home to the National Museum of Modern Art and is internationally renowned for its 20th and 21st century art collections. The art objects considered for this case study are located in two places named 'Museum' and 'Storage'. For the first place two containers were selected, the Mur Breton and Boite valise Duchamp, and for the second place one container was selected, Storage.

The conditions were often good or optimal as regards temperature and humidity inside the containers. In all three containers, one or more VOCs have reached levels such as to result in the critical condition.

#### Mur Breton

The Mur Breton container contains materials such as paper, glass, ceramics, basketry, ivory, wood, sone and plaster, skin and taxidermy, textiles, metals, and it has a hight isolation. Humidity, temperature, and VOCs (Volatile Organic Com-

Figure 4.47: Centre Pompidou

pounds) sensors inside the container are present. In this container more than one sensor for type was present therefore only more relevant data has been reported. Since the sensors were present in the same room almost all the sensors reported the same data. Thus, only for Acetic acid VOC two graphs have been reported.

Figure 4.48 shows the temperature data obtained from the sensor inside Mur Breton container in the period from October 2020 to mid-July 2021. From the image it can be observed that the conditions inside the container have been good most of the time except during some days in October 2020 when the condition became critical.

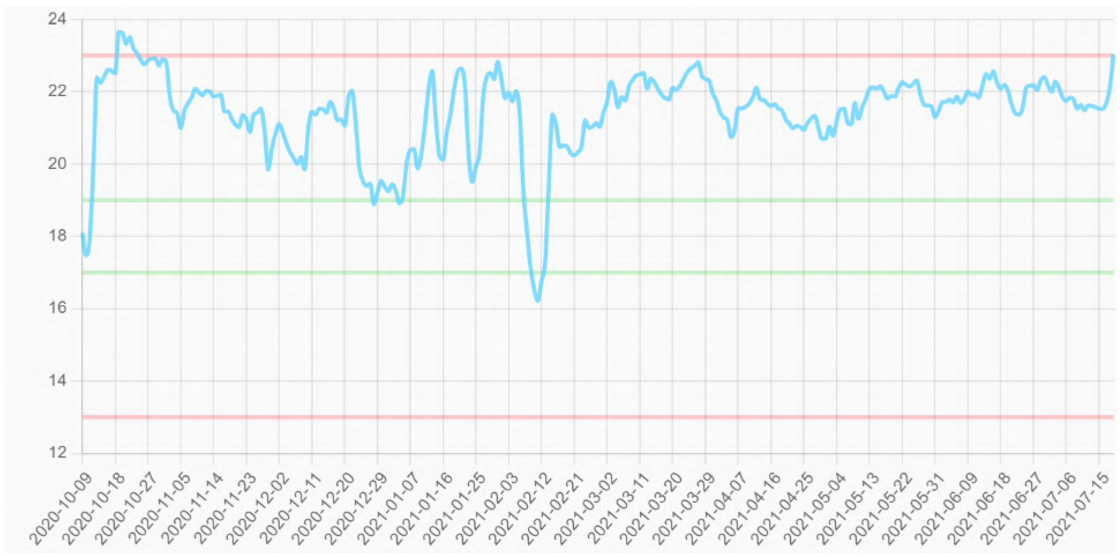


Figure 4.48: Temperature data obtained from the sensor inside Mur Breton container

Figure 4.49 shows the relative humidity data obtained from the sensor inside Mur Breton container from October 2020 to mid-July 2021. It can be observed that the conditions inside the container have been good and optimal most of the time.

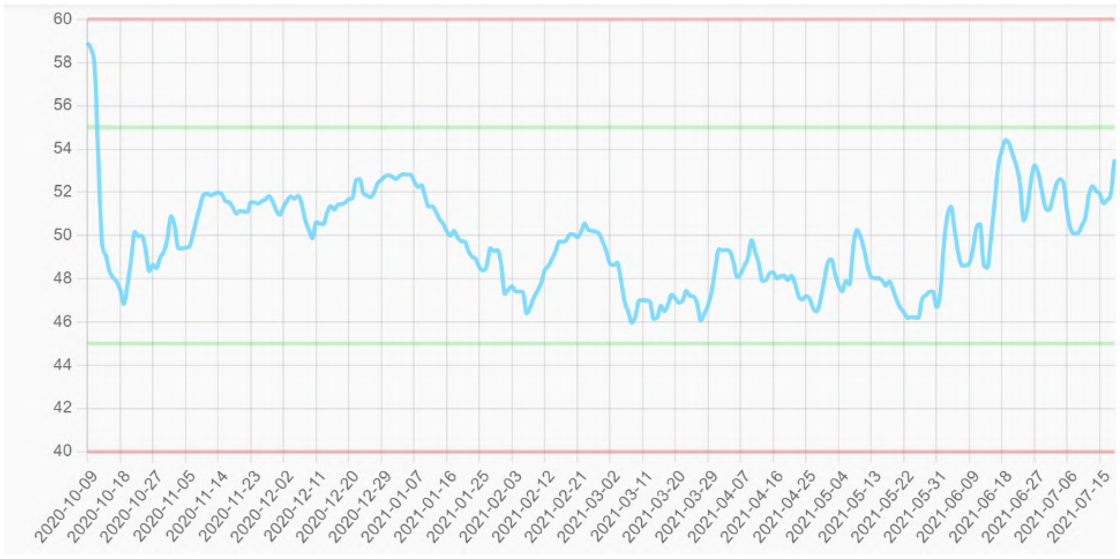


Figure 4.49: Relative humidity data obtained from the sensor inside Mur Breton container

Figure 4.50 shows data obtained from the sensor inside Mur Breton container in the period from the end of September 2020 to mid-June 2021 for Acetic acid, Formic acid, Formaldehyde, Acetaldehyde and $NO_2$. It can be observed that conditions inside the container have been good for Formaldehyde, Acetaldehyde and $NO_2$, were critical for Acetic acid, and for both Formic acid 1 and Formic acid 2 were critical in the beginning but then improved for Formic acid 2 when they first improved and then worsened for Formic acid 2.

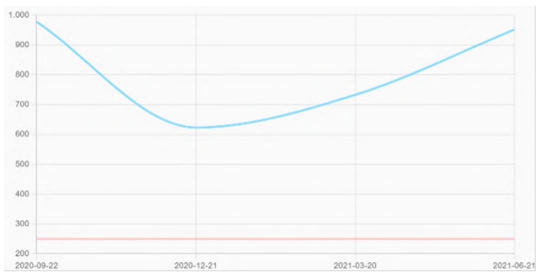
Table 4.18 shows the suggested preventive measures for Mur Breton container.

Table 4.18: Suggested preventive measures for Mur Breton container

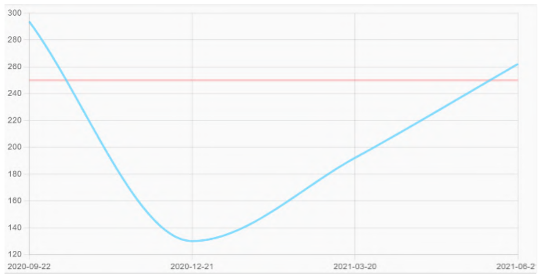
| Category | Name | Score |
|---|---|---|
| Light | Natural light shielding system | 6 |
| | Avoid areas of direct sunlight | 6 |
| | Cover windows | 6 |
| Temperature | Conservation heating | 6 |
| | Remove objects from external walls | 5 |
| | Cold storage (below 0°C) for unstable materials | 5 |
| Relative humidity | Dehumidifiers | 6 |
| | Insulate from wall | 5 |
| | Super absorbent polymers (SAPs) dispersed in heptane for humidity sorption | 4 |
| Pollutants | Graphene-based aerogels | 5 |
| | Air quality control (absorbents for VOCs) | 4 |
| | Absorbents for pollutants | 4 |

### Boite valise Duchamp

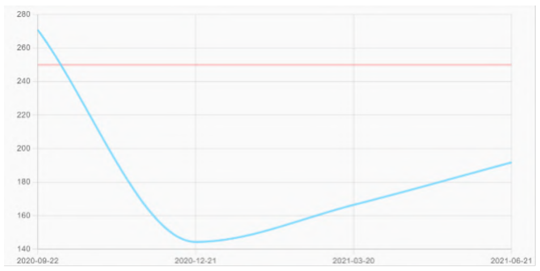
The Boite valise Duchamp container contains materials such as paper, glass, ceramics, wood, textiles, metals, and it has a hight isolation. Humidity, temperature,



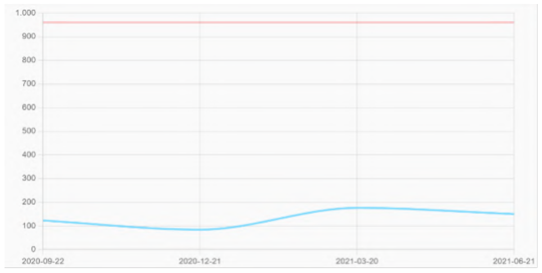
Acetic Acid



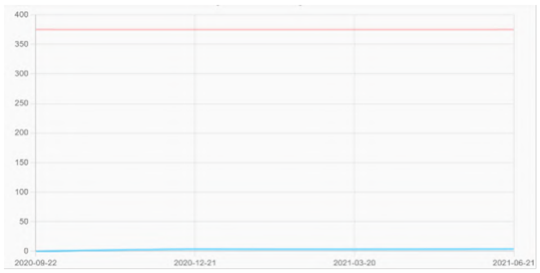
Formic Acid 1



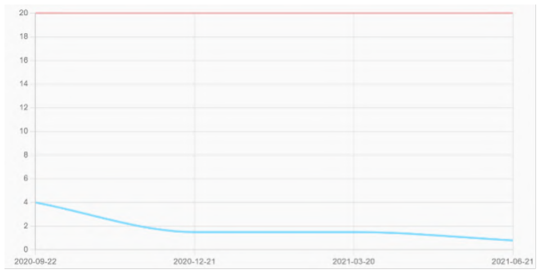
Formic Acid 2



Formaldehyde



Acetaldehyde 2



$NO_2$

Figure 4.50: All VOC data obtained from the sensor inside Mur Breton container

and VOCs (Volatile Organic Compounds) sensors inside the container are present.

Figure 4.51 shows the temperature data obtained from the sensor Boite valise Duchamp container in the period from October 2020 to mid-July 2021. From the image it can be observed that the conditions inside the container have been good most of the time except in the period from March to April 2021 when the condition became critical.

Figure 4.52 shows the relative humidity data obtained from the sensor inside Boite valise Duchamp on container from October 2020 to mid-July 2021. It can be observed that the conditions inside the container have always been good and optimal.

Figure 4.53 shows data obtained from the sensor inside Boite valise Duchamp container in the period from the end of September 2020 to mid-June 2021 for Acetic acid, Formic acid, Formaldehyde, Acetaldehyde and $NO_2$. It can be observed that conditions inside the container have been good all the time for all VOCs excluding Acetic Acid that has been critical in the beginning and then became good.

Table 4.19 shows the suggested preventive measures for Boite valise Duchamp container.

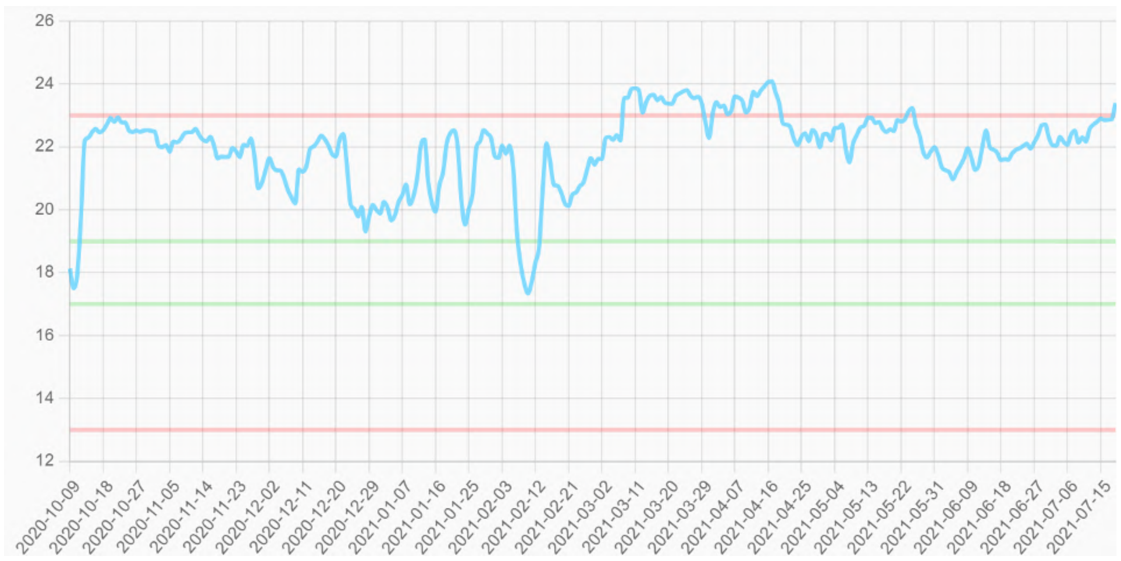


Figure 4.51: Temperature data obtained from the sensor inside Boite valise Duchamp container

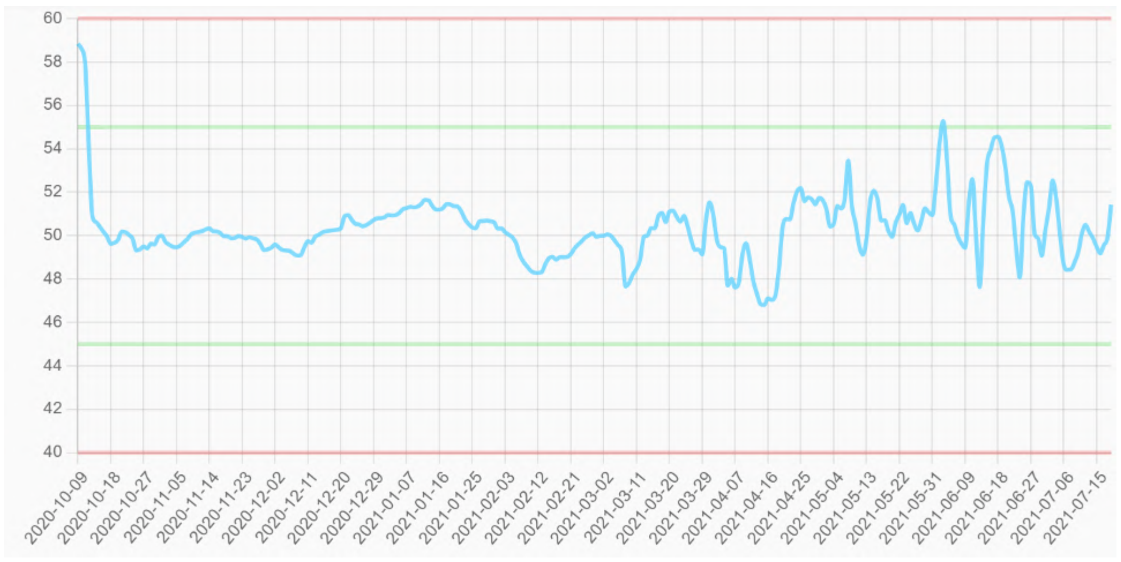


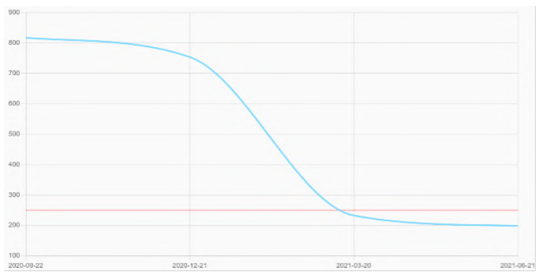
Figure 4.52: Relative humidity data obtained from the sensor inside Boite valise Duchamp container

### Storage

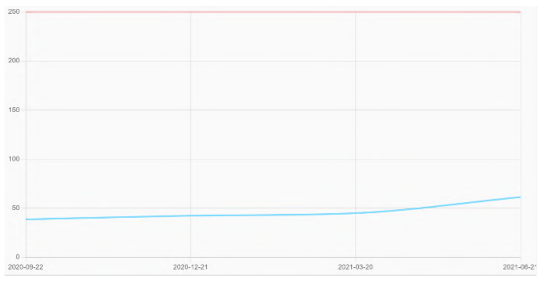
The Storage container contains materials such as paper, stone, and plaster, and it has a medium isolation. Only VOCs (Volatile Organic Compounds) sensors inside the container are present.

Figure 4.54 shows data obtained from the sensor inside Storage container in the period from the end of September 2020 to mid-March 2021 for Acetic acid, Formic acid, Formaldehyde, Acetaldehyde and $NO_2$. It can be observed that conditions inside the container have always been good for all VOCs excluding Acetic Acid which has been initially good, then became critical and then good again.

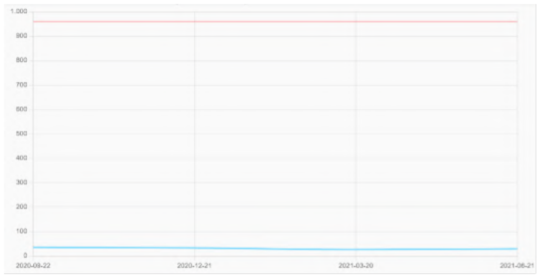
Table 4.20 shows the suggested preventive measures for Storage container.



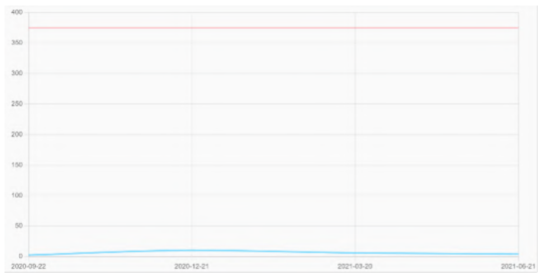
Acetic Acid



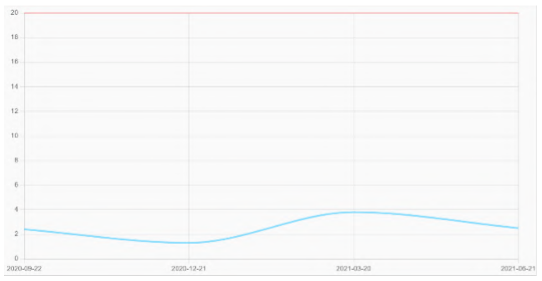
Formic Acid



Formaldehyde



Acetaldehyde



$NO_2$

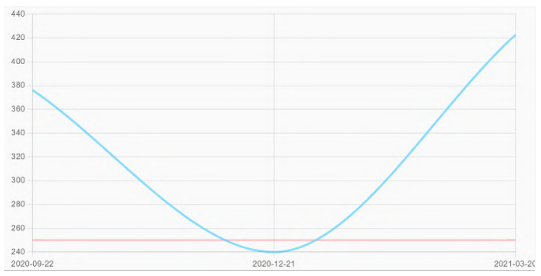
Figure 4.53: All VOC data obtained from the sensor inside Boite valise Duchamp container

Table 4.19: Suggested preventive measures for Boite valise Duchamp container

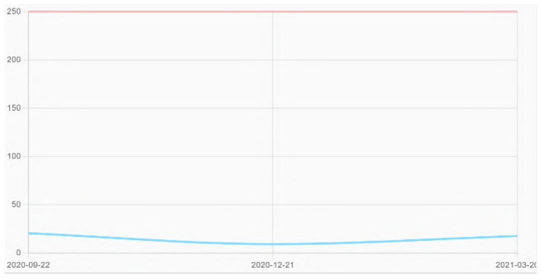
| Category | Name | Score |
|---|---|---|
| Light | Natural light shielding system | 6 |
| | Avoid areas of direct sunlight | 6 |
| | Cover windows | 6 |
| Temperature | Conservation heating | 6 |
| | Remove objects from external walls | 5 |
| | Cold storage (below 0°C) for unstable materials | 5 |
| Relative humidity | Dehumidifiers | 6 |
| | Insulate from wall | 5 |
| | Super absorbent polymers (SAPs) dispersed in heptane for humidity sorption | 4 |
| Pollutants | Graphene-based aerogels | 5 |
| | Air quality control (absorbents for VOCs) | 4 |
| | Absorbents for pollutants | 4 |

## 4.3 Conclusions

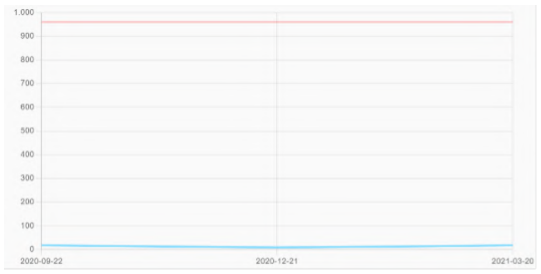
The DSS contains both a visualization tool to display the environmental data measured by temperature, relative humidity, and VOCs sensors and many preventive



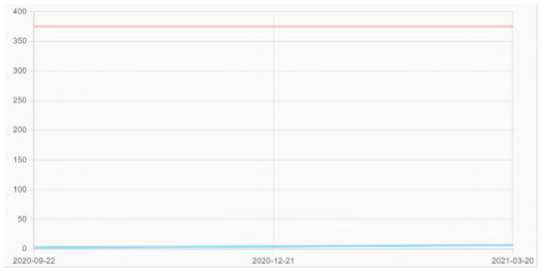
Acetic Acid



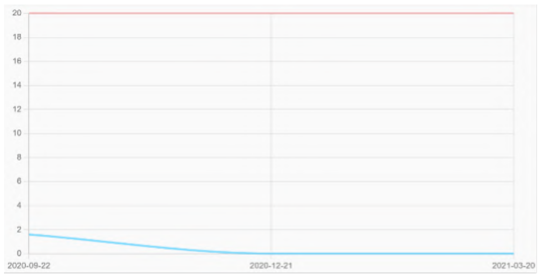
Formic Acid



Formaldehyde



Acetaldehyde



$NO_2$

Figure 4.54: All VOC data obtained from the sensor inside Storage container

Table 4.20: Suggested preventive measures for Storage container

| Category | Name | Score |
|---|---|---|
| Light | Natural light shielding system | 6 |
| | Avoid areas of direct sunlight | 6 |
| | Cover windows | 6 |
| Temperature | Conservation heating | 6 |
| | Remove objects from external walls | 5 |
| | Cold storage (below 0°C) for unstable materials | 5 |
| Relative humidity | Dehumidifiers | 6 |
| | Relative humidity monitoring | 6 |
| | Insulate from wall | 5 |
| Pollutants | Closed boxes (Conventional Archive Boxes) | 5 |
| | Storage in pollutant barrier bags | 4 |
| | Layered double hydroxides (LDH) dispersed in heptane for acid sorption | 4 |

measures collected from APACHE experts and other sources. The APACHE DSS was applied to six case studies, each a collection of a museum. The results of the application of the DSS are reported in the form of graphs to show the data collected with the sensors, and by suggested preventive conservation measures for improved cultural heritage management. The following paragraphs summarize the results of the application of the Apache DSS to each of the 6 case studies.

For the Peggy Guggenheim collection, the internal temperature and relative humidity conditions were often critical for the two room containers. Both inside and outside the two display cases, conditions were often critical for both temperature, relative humidity and VOCs. Of the five VOCs analysed, one reached a level that resulted in the critical condition being outside the two display cases and two or more VOCs reached a level that resulted in the critical condition within the display cases.

For the National Museum of Slovenia, the conditions inside and outside the containers were very variable both positively and negatively in terms of temperature and relative humidity. Of the five VOCs analysed, only one reached a level that resulted critical outside of both containers. Inside the containers, the conditions have always been optimal.

For Hungarian National Museum, conditions have often been critical both inside and outside the container Library room, both for temperature and relative humidity. They have often been good and optimal for the container Collection of historical photographs. Of the five VOCs analysed, one reached a level that resulted critical outside both containers and within the container Collection of historical photographs.

For MiC, conditions have often been good both inside and outside the containers for temperature, while they were often critical for relative humidity (relative humidity has been always critical in the two containers compactible metal rack's semi-open and closed metal document box). The conditions have always been good, both inside and outside the containers.

For the Science and Technology Foundation of Florence, the conditions were often good or optimal both inside and outside the containers for temperature and relative humidity. Inside two out of three containers, a VOC presented critical levels. Outside a container, two VOCs presented critical levels.

For Centre Pompidou, the conditions were often good or optimal as regards temperature and humidity inside the containers. In all three containers, one or more VOCs have reached levels such as to result in the critical condition.

The results given by the DSS were evaluated by conservator experts in the APACHE project. Based on their evaluation, the DSS has proposed the most appropriate preventive measures that have been reported for each container inside the deliverable.

Thanks to this work, it was possible to learn that the conditions inside and outside the containers are rarely optimal and often critical. Even though temperature thresholds shall sometimes be raised to preserve relative humidity in particularly humid areas, the results show the need to apply preventive measures to maintain and preserve the art objects in the best way.

# Chapter 5

# Time series compression

Time series are relevant in several contexts, and the Internet of Things ecosystem (IoT) is among the most pervasive ones. IoT devices, indeed, can be found in different applications, ranging from health care (smart wearables) to industrial ones (smart grids) [4], producing a large amount of time-series data. For instance, a single Boeing 787 fly can produce about half a terabyte of data from sensors [62]. In those scenarios, characterized by high data rates and volumes, time series compression techniques are a sensible choice to increase the efficiency of collection, storage, and analysis of such data. In particular, the need to include in the analysis both information related to the recent and the history of the data stream leads to consider data compression as a solution to optimize space without losing the most important information. A direct application of time series compression, for example, can be seen in Time Series Management Systems (or Time Series Database) in which compression is one of the most significant steps [39].

There exists extensive literature on data compression algorithms, both on generic purpose ones for finite-size data and on domain-specific ones, for example, for images, video, and audio data streams. This chapter aims at providing an overview of the state-of-the-art in time series compression research, specifically focusing on general-purpose data compression techniques that are either developed for time series or working well with time series.

The proposed algorithms can deal with the continuous growth of time series over time and suitable for generic domains (as in the different applications in the IoT). Furthermore, these algorithms take advantage of the peculiarities of time series produced by sensors, such as:

- **Redundancy**: some segments of a time series can frequently appear inside the same or other related time series;
- **Approximability**: sensors in some cases produce time series that can be approximated by functions;
- **Predictability**: some time series can be predictable, for example using deep neural network techniques.

Moreover, the chapter illustrates the taxonomy proposed in [13] of time series compression techniques based on their approach (dictionary-based, functional approximation, autoencoders, sequential, others) and their properties (adaptiveness, lossless reconstruction, symmetry, tuneability of max error or minimum compression ratio), anticipated in visual form in Figure 5.1 and discussed in Section 5.1, that will guide the description of the selected approaches. Finally, Section 5.4 recapitulates the results of performance measurements indicated in the described studies.

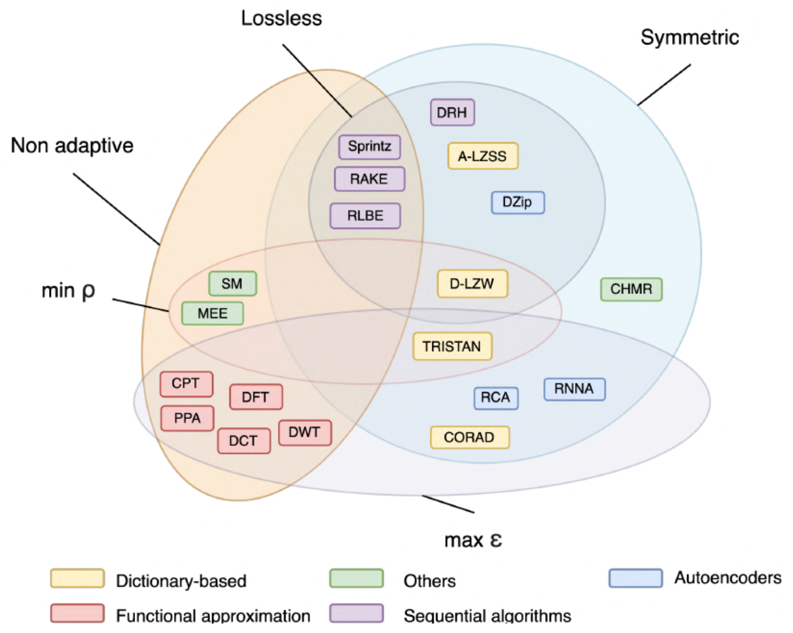


Figure 5.1: Visual classification of time series compression algorithms

## 5.1 Background

### 5.1.1 Time series

Time series are defined as a collection of data, sorted in ascending order according to the timestamp $t_i$ associated with each element. They are divided into:

- Univariate Time Series (UTS): elements inside the collection are real values;
- Multivariate Time Series (MTS): elements inside the collections are arrays of real values, in which each position in the array is associated with a time series feature.

For instance, the temporal evolution of the average daily price of a commodity as the one represented in the plot in Figure 5.2 can be modeled as a univariate time series, whereas the summaries of daily exchanges for a stock (including opening price, closing price, the volume of trades and other information) can be modeled as a multivariate time series.

Using a formal notation, time series can be written as:

$$TS = [(t_1, x_1), \ldots, (t_n, x_n)], x_i \in R^m \tag{5.1}$$

where $n$ is the number of elements inside a time series and $m$ is the vector dimension of multivariate time series. For univariate time series, $m = 1$. Given $k \in [1, n]$, $TS[k]$ indicates the $k$-th element $(t_n, x_n)$ of the time series $TS$.

A time series can be divided into segments, defined as a portion of the time series without any missing elements and ordering preserved:

$$TS_{[i,j]} = [(t_i, x_i), \ldots, (t_j, x_j)] \tag{5.2}$$

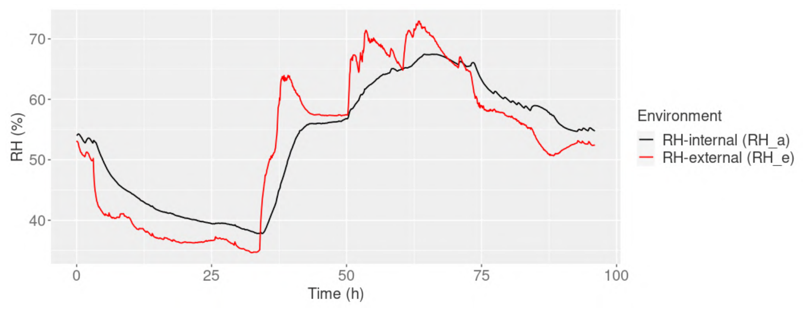


Figure 5.2: Example of an MTS representing relative humidity fluctuations inside and outside an artwork's container

where $\forall k \in [i, j], TS[k] = TS_{[i,j]}[k - i + 1]$.

### 5.1.2 Compression

Data compression, also known as *source coding*, is defined in [65] as "the process of converting an input data stream (the source stream or the original raw data) into another data stream (the output, the bitstream, or the compressed stream) that has a smaller size". This process can take advantage of the *Simplicity Power* (SP) theory, formulated in [87], in which the compression goal is to remove redundancy while having high descriptive power.

The decompression process, complementary to the compression one, is indicated also as *source decoding*, and tries to reconstruct the original data stream from its compressed representation.

Compression algorithms can be described with the combination of different classes, shown in the following list:

- **Non-adaptive - adaptive**: a non-adaptive algorithm that doesn't need a training phase to work efficiently with a particular dataset or domain since the operations and parameters are fixed, while an adaptive one does;

- **Lossy - lossless**: algorithms can be lossy if the decoder doesn't return a result that is identical to original data, or lossless if the decoder result is identical to original data;

- **Symmetric - non-symmetric**: an algorithm is symmetric if the decoder performs the same operations of the encoder in reverse order, whereas a non-symmetric one uses different operations to encode and decode a time series.

In the particular case of time series compression, a compression algorithm (*encoder*) takes in input one Time Series $TS$ of size $s$ and returns its compressed representation $TS'$ of size $s'$, where $s' < s$ and the size is defined as the bits needed to store the time series: $E(TS) = TS'$. From the compressed representation $TS'$, using a *decoder*, it is possible to reconstruct the original time series: $D(TS') = \overline{TS}_s$. If $\overline{TS} = TS_s$ then the algorithm is lossless, otherwise it is lossy.

In Section 5.2, there are shown the most relevant categories of compression techniques and their implementation.

### 5.1.3 Quality indices

To measure the performances of a compression encoder for time series, three characteristics are considered: compression ratio, speed, and accuracy.

**Compression ratio** This metric measures the effectiveness of a compression technique, and it is defined as:

$$\rho = \frac{s'}{s} \tag{5.3}$$

where $s'$ is the size of the compressed representation and $s$ is the size of the original time series. Its inverse $\frac{1}{\rho}$ is named *compression factor*. An index used for the same purpose is the *compression gain*, defined as:

$$c_g = 100 \log_e \frac{1}{\rho} \tag{5.4}$$

**Accuracy**, also called distortion, measures the fidelity of the reconstructed time series respect to the original. It is possible to use different metrics to determine fidelity [66]:

- **Mean Squared Error**: $MSE = \frac{\sum_{i=1}^{n}(x_i - \overline{x}_i)^2}{n}$
- **Root Mean Squared Error**: $RMSE = \sqrt{MSE}$
- **Signal to Noise Ratio**: $SNR = \frac{\sum_{i=1}^{n} x_i^2 n}{MSE}$
- **Peak Signal to Noise Ratio**: $PSNR = \frac{x_{pick}^2}{MSE}$ where $x_{peak}$ is the maximum value in the original time series.

## 5.2 Compression algorithms

This section shows the most relevant time series compression algorithms by describing in short summaries their principles and peculiarities. Also, a pseudo-code is provided for each approach, focusing more on style homogeneity than on the faithful reproduction of the original code proposed by the authors. Below, the full list of algorithms described in this section, divided by approach:

1. Dictionary-Based (DB):
    1.1. TRISTAN;
    1.2. CORAD;
    1.3. A-LZSS;
    1.4. D-LZW.
2. Functional Approximation (FA)
    2.1. Piecewise Polynomial Approximation (PPA);
    2.2. Chebyshev Polynomial Transform (CPT);
    2.3. Discrete Wavelet Transform (DWT);
    2.4. Discrete Fourier Transform (DFT);
    2.5. Discrete Cosine Transform (DCT).

3. Autoencoders:
    3.1. Recurrent Neural Network Autoencoder (RNNA);
    3.2. Recurrent Convolutional Autoencoder (RCA);
    3.3. DZip.
4. Sequential Algorithms (SA):
    4.1. Delta encoding, Run-length and Huffman (DRH);
    4.2. Sprintz;
    4.3. Run-Length Binary Encoding (RLBE);
    4.4. RAKE.
5. Others:
    5.1. Major Extrema Extractor (MEE);
    5.2. Segment Merging (SM);
    5.3. Continuous Hidden Markov Chain (CHMC).

RNNA algorithm can be applied both to univariate and multivariate time series. The other algorithms can also handle multivariate time series by extracting each feature as an independent time series. All the algorithms accept time series represented with real values.

The considered methods span a temporal interval of more than 20 years, with a significant outlier dating back to the '80s. Figure 5.3 graphically represents the temporal trend of adoption of the different approaches for the considered methods.

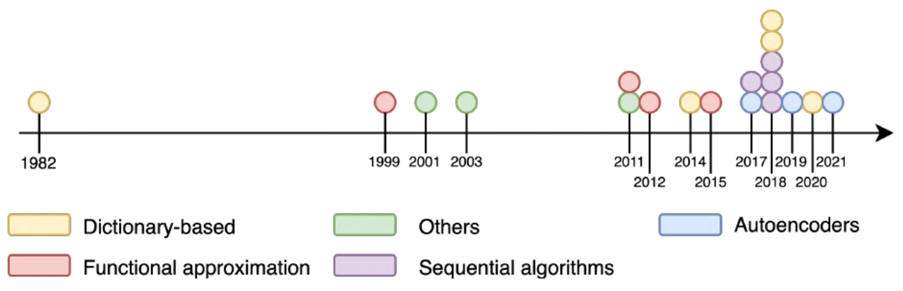


Figure 5.3: Chronological trend of compression techniques for time series

### 5.2.1 Dictionary-Based (DB)

This approach is based on the principle that time series share some common segments, without considering timestamps. These segments can be extracted into atoms, such that a time series segment can be represented with a sequence of these atoms. Atoms are then collected into a dictionary that associates each atom with a univocal key used both in the representation of time series and to search efficiently their content. The choice of atoms length should guarantee a low decompression error and maximize the compression factor at the same time. Algorithm 5.1 shows how the training phase works at a high level: createDictionary function computes a dictionary of segments given a dataset composed by time series and a threshold value `th`.

Algorithm 5.1: Dictionary based - Training

```
createDictionary(Stream S, Threshold th, int segmentLength) {
    Dictionary d;
    Segment s;
    while (S.isNotEmpty()) {
        s.append(S.read());
        if (s.length == segmentLength) {
            if (d.find(s, th)) {
                d.merge(s);
            } else {
                d.add(s);
            }
            s.empty();
        }
    }
    return d;
}
```

The `find` function searches in the dictionary if there exists a segment which is similar to the segment `s` with a distance lower than threshold `th`; a possible index of distance can be $MSE$. If a match is found, the algorithm merges the segment `s` with the matched one to achieve generalization. Larger `th` value results in higher compression and lower reconstruction accuracy, and a dictionary with lower dimension.

After the segment dictionary is created, the compression phase takes in input one time series, and each segment is replaced with its key in the dictionary. If some segments are not present in the dictionary, they are left uncompressed, or a new entry is added to the dictionary. Algorithm 5.2 shows how the compression phase works: the compress function takes in input a time series to compress, the dictionary created during the training phase, a threshold value, and returns the compressed time series as a list of indices and segments.

Algorithm 5.2: Dictionary based - Compression

```
compress(Stream S, Dictionary d, Threshold th, int segmentLength)
    {
    Segment s;
    while (S.isNotEmpty()) {
        s.append(S.read());
        if (s.length == segmentLength) {
            if (d.find(s, th)) {
                send(d.getIndex(s));
            } else {
                send(s);
            }
            s.empty();
        }
    }
}
```

The compression achieved by this technique can be either lossy or lossless, depending on the implementation.

The main challenges for this architecture are to:

- maximize the searching speed to find time series segments in the dictionary;
- make the time series segments stored in the dictionary more general as possible to minimize the distance in the compression phase.

### TRISTAN

One implementation of the Dictionary-Based architecture is TRISTAN [50], an algorithm divided into two phases: the learning phase and the compression phase.

**Learning phase** The dictionary used in this implementation can be created by domain experts that add typical patterns or by learning a training set. To learn a dictionary from a training set $T = [t_1, \dots, t_n]$ of $n$ segments, the following minimization problem has to be solved:

$$
\begin{gathered}
D = \arg\min_D \sum_{i=1}^{n} \|w_i D - t_i\|_2 \\
\text{Under the constraint: } \|w_i\|_0 \leq sp
\end{gathered}
\tag{5.5}
$$

where $D$ is the obtained dictionary, $sp$ is a fixed parameter representing sparsity, $w_i$ is the compressed representation of segment $i$, and $w_i D$ is the reconstructed segment. The meaning of this formulation is that the solution to be found is a dictionary that minimizes the distance between original and reconstructed segments.

The problem shown in equation 5.5 is NP-hard [50], thus an approximate result is computed. A technique for approximating this result is shown in [48].

**Compression phase** Once the dictionary is built, the compression phase consists of finding $w$ such that:

$$
s = w \cdot D \tag{5.6}
$$

where $D$ is the dictionary, $s$ is a segment and $w \in \{0,1\}^k$, and $k$ is the length of the compressed representation. An element of $w$ in position $i$ is 1 if the $a_i \in D$ is used to reconstruct the original segment, 0 otherwise.

Finding a solution for Equation 5.6 is an NP-hard problem and, for this reason, the matching pursuit method [49] is used to approximate the original problem:

$$
\begin{gathered}
\overline{s} = \arg\min_w \|wD - s\|_2 \\
\text{Under the constraint: } \|\overline{s}\|_0 \leq sp
\end{gathered}
\tag{5.7}
$$

where $sp$ is a fixed parameter representing sparsity.

**Reconstruction phase** Having a dictionary $D$ and a compressed representation $w$ of a segment $s$, it is possible to compute $\overline{s}$ as:

$$
\overline{s} = wD \tag{5.8}
$$

### CORAD

This implementation extends the idea presented in TRISTAN [42]. The main difference is that it adds autocorrelation information to get better performances in terms of compression ratio and accuracy.

Correlation between two time series $TS^A$ and $TS^B$ is measured with the Pearson correlation coefficient:

$$
r = \frac{\sum_i^n (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_i^n (x_i - \overline{x})^2}\sqrt{\sum_i^n (y_i - \overline{y})^2}} \tag{5.9}
$$

where $x_i$ is an element of $TS_n^A$, $y_i$ is an element of $TS_n^B$ and $\overline{x}, \overline{y}$ are mean values of the corresponding time series. This coefficient can be applied also to segments that have different ranges of values and $r \in [-1, q]$ where 1 expresses the maximum linear correlation, -1 the maximum linear negative correlation, and 0 no linear correlation.

Time series are divided into segments and time windows are set. For each window, correlation is computed between each segment belonging to it, and results are stored in a correlation matrix $M \in R^{n \times n}$ where $n$ is the segment number of elements.

**Compression phase** During this phase, segments are sorted from the less correlated to the most correlated. To do this, a correlation matrix $M$ is used. The metric used to measure how much one segment is correlated with all the others is the absolute sum of the correlations, computed as the sum of each row of $M$.

Knowing correlation information, a dictionary is used only to represent segments that are not correlated with others, as in TRISTAN implementation. While the other segments are represented solely using correlation information.

**Reconstruction phase** The reconstruction phase starts with the segment represented with dictionary atoms while the others are reconstructed looking at the segment to which they are correlated.

This process is very similar to the one proposed in TRISTAN, with the sole difference that segments represented in the dictionary are managed differently than those that are not.

### Accelerometer LZSS (A-LZSS)

Accelerometer LZSS is an algorithm built on top of the LZSS algorithm [76] for searching matches [59]. A-LZSS algorithm uses Huffman codes, generated offline using frequency distributions. In particular, this technique considers blocks of size $s = 1$ to compute frequencies and build the code: an element of the time series will be replaced by a variable number of bits. Moreover, larger blocks can be considered and, in general, having larger blocks gives better compression performances at the cost of larger Huffman code tables.

The implementation of this technique is shown in Algorithm 5.3, where:

- `minM`: the minimum match length, which is asserted to be `minM` $> 0$;
- `Ln`: determines the lookahead distance, as $2^{Ln}$;
- `Dn`: determines the dictionary atoms length, as $2^{D}n$;
- `longestMatch`: is a function that returns the index $I$ of the found match and the length $L$ of the match. If the length of the match is too small, then the Huffman code representation of `s` is sent as the compression representation, otherwise, the index and the length of the match are sent, and the next $L$ elements are skipped.

This implementation uses a brute-force approach, with complexity $O(2^{Dn} \cdot 2^{Ln})$ but it is possible to improve over it by using hashing techniques.

Algorithm 5.3: A-LZSS algorithm

```
compress(Stream S, int minM, Dictionary d, int Ln, int Dn) {
    foreach(s in S) {
        I, L = d.longestMatch(s, Ln, Dn);
        if (L < minM) {
            send(getHuffCode(s));
        } else {
            send((I, L));
            s.skip(L);
        }
    }
}
```

### Differential LZW (D-LZW)

The core of this technique is the creation of a very large dictionary that grows over time: once the dictionary is created, if a buffer block is found inside the dictionary it is replaced by the corresponding index, otherwise, the new block is inserted in the dictionary as a new entry [82].

Adding new blocks guarantees lossless compression, but has the drawback of having too large dictionaries. This makes the technique suitable only for particular scenarios (i.e. input streams composed by words/characters or when the values inside a block are quantized).

Another drawback of this technique is how the dictionary is constructed: elements are simply appended to the dictionary to preserve the indexing of previous blocks. For a simple implementation of the dictionary, the complexity for each search is $O(n)$ where $n$ is the size of the dictionary. This complexity can be improved by using more efficient data structures.

## 5.2.2 Function Approximation (FA)

The main idea behind function approximation is that a time series can be represented as a function of time. Since finding a function that approximates the whole time series is infeasible due to the presence of new values that cannot be handled, the time series is divided into segments and for each of them, an approximating function is found.

Exploring all the possible functions $f : T \to X$ is not feasible, thus implementations consider only one family of functions and try to find the parameters that better approximate each segment. This makes the compression lossy.

A point of strength is that it does not depend on the data domain, so no training phase is required since the regression algorithm considers only single segments in isolation.

### Piecewise Polynomial Approximation (PPA)

This technique divides a time series into several segments of fixed or variable length and tries to find the best polynomials that approximate segments.

Despite the compression is lossy, a maximum deviation from the original data can be fixed a priori to enforce a given reconstruction accuracy.

The implementation of this algorithm is described in [19] where the authors apply a greedy approach and three different online regression algorithms for approximating constant functions, straight lines, and polynomials. These online algorithms are:

- the PMR-Midrange algorithm, that approximates using constant functions [44];
- the optimal approximation algorithm, described in [16], that uses linear regression;
- the randomized algorithm presented in [70], that approximates using polynomials.

The algorithm used in [19] for approximating a time series segment is explained in Algorithm 5.4.

Algorithm 5.4: PPA algorithm

```
compress(int ρ, Stream S, float ϵ) {
    Segment s = S.read();
    while (S.isNotEmpty()) {
        Polynomial p;
        int l = 0;
        foreach (k in [0 : ρ]) {
            float currErr;
            Polynomial currP;
            bool continueSearch = true;
            int i = 0;
            int currL;
            while (continueSearch && i < len(s)) {
                currErr, currP = approx(k, s.getPrefix(i));
                if (currErr < ϵ) {
                    i += 1;
                    currL = i;
                } else {
                    continueSearch = false;
                }
            }
            while (continueSearch && S.isNotEmpty()) {
                s.append(S.read());
                currErr, currP = approx(k, s);
                if (currErr < ϵ) {
                    currL = len(s);
                } else {
                    currL = len(s) - 1;
                    continueSearch = false;
                }
            }
            if (currErr < ϵ && currL > l) {
                p = currP;
                l = currL;
            }
        }
        if (len(s) > 0) {
            send(p, l);
            s.removePrefix(l);
        } else {
            throw "Error: exceeded error threshold value";
        }
    }
}
```

Where:

- $S$ is the input time series;
- $\rho$ is the maximum polynomial degree;
- $\epsilon$ is the error threshold.

This algorithm finds repeatedly the polynomial of degree between 0 and a fixed maximum that can approximate the longest segment within the threshold error, yielding the maximum local compression factor. After a prefix of the stream has been selected and compressed into a polynomial, the algorithm analyzes the following stream segment. A higher value of $\epsilon$ returns higher compression and lower reconstruction accuracy. The fixed maximum polynomial degree $\rho$ affects compression speed, accuracy, and compression ratio: higher values slow down compression and reduce the compression factor, but return higher reconstruction accuracy.

**Chebyshev Polynomial Transform (CPT)**

Another implementation of polynomial compression can be found in [33]. In this article, the authors show how a time series can be compressed in a sequence of finite Chebyshev polynomials.

The principle is very similar to the one shown in subsection 5.2.2 but based on the use of a different type of polynomial. Chebyshev Polynomials are of two types, $T_n(x)$, $U_n(x)$, defined as [47]:

$$T_n(x) = \frac{n}{2}\sum_{k=0}^{n/2}(-1)^k\frac{(n-k-1)!}{k!(n-2k)!}(2x)^{n-2k}, |x| < 1 \tag{5.10}$$

$$U_n(x) = \sum_{k=0}^{n/2}(-1)^k\frac{(n-k)!}{k!(n-2k)!}(2x)^{n-2k}, |x| < 1 \tag{5.11}$$

where $n \geq 0$ is the polynomial degree.

**Discrete Wavelet Transform (DWT)**

Discrete wavelet transform uses wavelet functions to transform time series. Wavelets are functions that, similarly to a wave, start from zero, and end with zero after some oscillation. An application of this technique can be found in [2].

This transformation can be written as:

$$\Psi_{m,n}(t) = \frac{1}{\sqrt{a^m}}\Psi\left(\frac{t-nb}{a^m}\right) \tag{5.12}$$

where $a > 1$, $b > 0$, $m, n \in Z$.

To recover one transformed signal, the following formula can be applied:

$$x(t) = \sum_{m\in Z}\sum_{n\in Z}\langle x, \Psi_{m,n}\rangle \cdot \Psi_{m,n}(t) \tag{5.13}$$

### Discrete Fourier Transform (DFT)

Together with the Discrete wavelet transform, the Discrete Fourier transform is commonly used for signals compression. Given a time series $[(t_1, x_1), \ldots, (t_n, x_n)]$, it is defined as:

$$X_k = \sum_{n=1}^{N} x_n e^{\frac{-i2\pi kn}{N}}$$

Where $e^{\frac{i2\pi}{N}}$ is a primitive $N$th root of 1.

To efficiently compute this transformation, the Fast Fourier transform algorithm can be used, as the one introduced here [29]. As described in [40], compressing a time series can be done after applying the DFT by cutting coefficients that are close to zero. This can be done also by fixing a compression ratio and discarding all the coefficients after a certain index.

### Discrete Cosine Transform (DCT)

Given a time series $[(t_1, x_1), \ldots, (t_n, x_n)]$, the Discrete Cosine Transform is defined as:

$$c_k = \sum_{n=0}^{N} x_n \cdot \cos\left[\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right]$$

The computation of this transformation is computationally expensive, so efficient implementations can be used, as the one presented in [36].

Similarly to the DFT, compression of time series can be achieved by cutting the coefficients that are close to zero or by fixing a compression ratio and discarding all the coefficients after a certain index. An example of this technique can be found at [52].

## 5.2.3 Autoencoders

An autoencoder is a particular neural network that is trained to give as output the same values passed as input. Its architecture is composed of two symmetric parts: encoder and decoder. Giving an input of dimension $n$, the encoder gives as output a vector with dimensionality $m < n$, called code, while the decoder reconstructs the original input from the code, as shown in Figure 6.1 [30].

### Recurrent Neural Network Autoencoder (RNNA)

RNNA compression algorithms exploit recurrent neural networks [71] to achieve a compressed representation of a time series. Figure 5.5 shows the general unrolled structure of a recurrent neural network encoder and decoder. The encoder takes in input time series elements, which are combined with hidden states. Each hidden state is then computed starting from the new input and the previous state. The last hidden state of the encoder is passed as the first hidden state of the decoder, which applies the same mechanism, with the only difference that each hidden state provides an output. The output provided by each state is the reconstruction of the relative time series element and is passed to the next state.

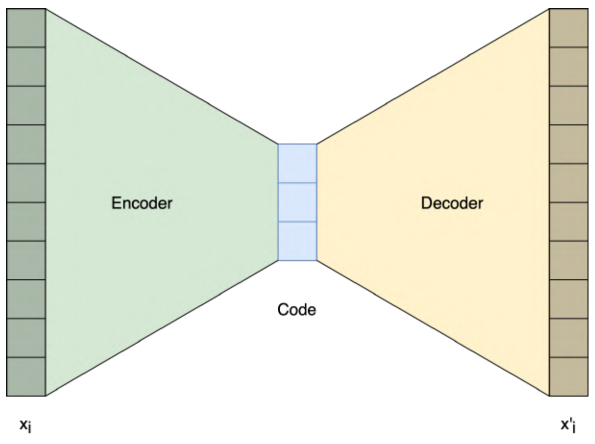


Figure 5.4: Simple autoencoder architecture

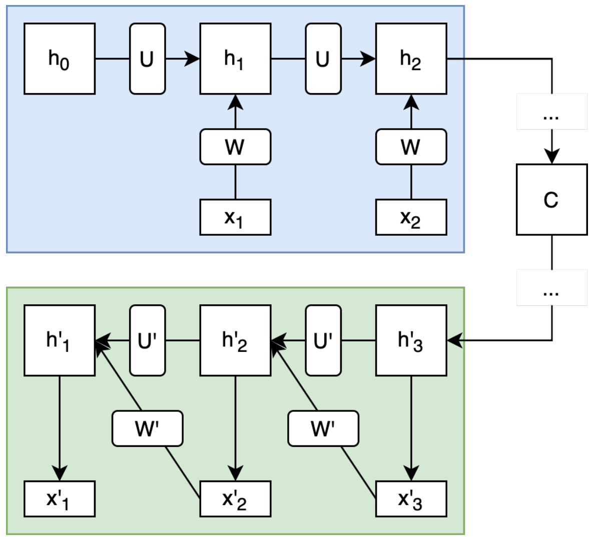


Figure 5.5: General structure of a RNN encoder and decoder

New hidden state $h_t$ is obtained by applying:

$$h_t = \phi(Wx_t + Uh_{t-1}) \tag{5.14}$$

where $\phi$ is a logistic sigmoid function or the hyperbolic tangent.

One application of this technique is shown in [35], in which also Long Short-Term Memory [48] is considered. This implementation compresses time series segments of different lengths using autoencoders. The compression achieved is lossy and a maximal loss threshold $\epsilon$ can be enforced.

The training set is preprocessed considering temporal variations of data applying:

$$\Delta(\mathcal{L}) = \sum_{t \in \mathcal{L}} |x_t - x_{t-1}| \tag{5.15}$$

where $\mathcal{L}$ is the local time window.

The value obtained by Equation 5.15 is then used to partition the time series, such that each segment has a total variation close to a predetermined value $\tau$.

Algorithm 5.5 shows an implementation of the RNN autoencoder approach, with an error threshold $\epsilon$, where:

- `RAE` is the recurrent autoencoder trained on a training set, composed of an encoder and a decoder;

- `getError` computes the reconstruction error between the original and reconstructed segment;
- $\epsilon$ is the error threshold value.

Algorithm 5.5: RNN compression algorithm

```
compress(Stream S, float ϵ, RAE a) {
    Segment s = Null;
    while (S.isNotEmpty()) {
        Segment aux = s;
        Element e = S.read();
        aux.append(e);
        if (getError(aux, a.decode(a.encode(aux)) < ϵ) {
            s = aux
        } else {
            send(a.encode(s));
            s.empty();
            s.append(e);
        }
    }
}
```

### Recurrent Convolutional Autoencoder (RCA)

A variation of the technique presented in the previous subsection is proposed in [89] and consists of adding convolutional layers. Convolutional layers are used mainly for images data to extract local features. In the case of time series, this layer allows extracting local fluctuations, addressing complex inputs.

### DZip

DZip is a lossless recurrent autoencoder [31]. This model also uses the prediction technique: it tries to predict the next symbol, having, in this case, a fixed vocabulary. To achieve a lossless compression, it combines two modules: the bootstrap model and the supporter model, as shown in Figure 5.6.

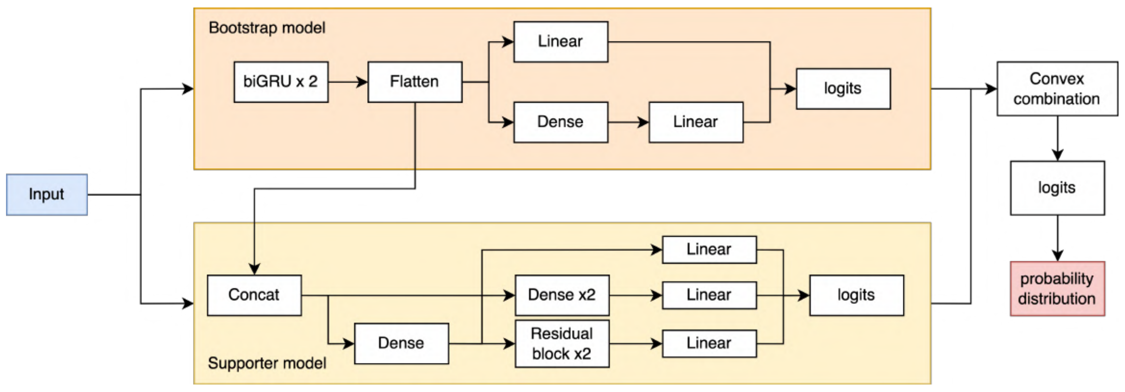


Figure 5.6: DZip model infrastructure

The bootstrap model is composed by stacking two bidirectional gated recurrent units, followed by linear and dense layers. The output of this module is a probability distribution, that is, the probability of each symbol in the fixed vocabulary to be the next one.

The supporter model is used to better estimate the probability distribution respect to the bootstrap model. It is composed by stacking three neural networks which act as independent predictors of varying complexity.

The two models are then combined by applying the following equation:

$$o = \lambda o_b + (1 - \lambda) o_s$$

Where $o_b$ is the output of the bootstrap model, $o_s$ is the output of the supporter model, and $\lambda \in [0, 1]$ is a learnable parameter.

### 5.2.4 Sequential algorithms

This architecture is characterized by combining sequentially several simple compression techniques. Some of the most used techniques are:

- Huffman coding;
- Delta encoding;
- Run-length encoding;
- Fibonacci binary encoding.

These techniques, summarized below, are the building blocks of the methods presented in the following subsections.

**Huffman coding** Huffman coding is the basis of many compression techniques since it is one of the necessary steps, as for the algorithm shown in Subsection 5.2.4.

The encoder creates a dictionary that associates each symbol with a binary representation and replaces each symbol of the original data with the corresponding representation. The compression algorithm is shown in Algorithm 5.6 [66].

The decoder algorithm is very simple since the decoding process is the inverse of the encoding process: the encoded bits are searched in the dictionary and replaced with the corresponding original symbols.

**Delta encoding** This technique encodes a target file with respect to one or more reference files [79]. In the particular case of time series, each element at time $t$ is encoded as $\Delta(x_t, x_{t-1})$.

**Run-length** In this technique, each run (a sequence in which the same value is repeated consecutively) is substituted with the pair $(v_t, o)$ where $v_t$ is the value at time $t$ and $o$ is the number of consecutive occurrences [32].

**Fibonacci binary encoding** This encoding technique is based on the Fibonacci sequence, defined as:

$$F(N) = \begin{cases} F(N-1) + F(N-2), & \text{if } N >= 1 \\ N, & \text{N = -1, N = 0} \end{cases} \tag{5.16}$$

Having $F(N) = a_1 \dots a_p$ where $a_j$ is the bit at position $j$ of the binary representation of $F(N)$, the Fibonacci binary coding is defined as:

$$F_E(N) = \sum_{i=0}^{j} a_i \cdot F(i) \tag{5.17}$$

Where $a_i \in \{0, 1\}$ is the i-th bit of the binary representation of $F(N)$ [84].

Algorithm 5.6: Huffman Code compression

```
createDictionary(StreamPrefix S) {
    Tree T = new Tree();
    PriorityList L = createPriorityList(S);
    foreach (i in [0 : L.length − 1]) {
        Node n = new Node();
        Element el1 = L.extractMin();
        Element el2 = L.extractMin();
        n.frequency = el1.frequency + el2.frequency;
        T.addTree(n, (el1, el2));
        L.add(n);
    }
    return L.toDictionary();
}

compress(Stream S, int prefixLen) {
    StreamPrefix s = S.prefix(prefixLen);
    Dictionary D = createDictionary(s);
    CompressedRepresentation R = [];
    while (S.isNotEmpty()) {
        Element e = S.read();
        send(D[e]);
    }
}
```

### Delta encoding, Run-length, and Huffman (DRH)

This technique combines together three well-known compression techniques [53]: delta encoding, run-length encoding, and Huffman code. Since these techniques are all lossless, the compression provided by DRH is lossless too if no quantization is applied.

Algorithm 5.7 describes the compression algorithm, where $Q$ is the quantization level and is asserted to be $\geq 1 : ifQ = 1$ the compression is lossless, and increasing values of $Q$ return higher compression factor and lower reconstruction accuracy.

The decompression algorithm is the inverse of the compression one: once data is received, it is decoded using the Huffman code and reconstructed using the repetition counter.

Since this kind of algorithm is not computationally expensive, the compression phase can be performed also by low resource computational units, such as sensor nodes.

### Sprintz

Sprintz algorithm [6] is designed for the IoT scenario, in which energy consumption and speed are important factors. In particular, the goal is to satisfy the following requirements:

- Handling of small blocks size
- High decompression speed
- Lossless data reconstruction

The proposed algorithm is a coder that exploits prediction to achieve better results. In particular, it is based on the following components:

Algorithm 5.7: DRH nodel level

```
compress(Stream S, float Q) {
 Element lastValue = null;
 Element lastDelta = null;
 int counter = 0;
 foreach (s in S) {
   if (last != null && lastDelta != null) {
      float delta = 0;
      if (Q > 1) {
         delta = (int)(lastValue - s) / Q;
      } else {
         delta = lastValue - s;
      }
      lastValue = s;
      if (delta == lastDelta) {
         counter += 1;
      } else {
         encoded = huffmanEncode(lastDelta);
         send((encoded, counter));
         lastDelta = delta;
         counter = 0;
      }
   } else {
      lastValue = s;
      lastDelta = 0;
   }
 }
}
```

- **Forecasting**: used to predict the difference between new samples and the previous ones through delta encoding or FIRE algorithm [6];
- **Bit packing**: packages are composed of a payload that contains prediction errors and a header that contains information that is used during reconstruction;
- **Run-length encoding**: if a sequence of correct forecasts occurs, the algorithm does not send anything until some error is detected and the length of skipped zero error packages is added as information;
- **Entropy coding**: package headers and payloads are coded using Huffman coding, presented in Subsection 5.2.4.

### Run-Length Binary Encoding (RLBE)

This lossless technique is composed of 5 steps, combining delta encoding, run-length, and Fibonacci coding, as shown in Figure 5.7 [75].

This technique is developed specifically for devices characterized by low memory and computational resources, such as IoT infrastructures.

### RAKE

RAKE algorithm, presented in [9], exploits sparsity to achieve compression. It is a lossless compression algorithm with two phases: preprocessing and compression.

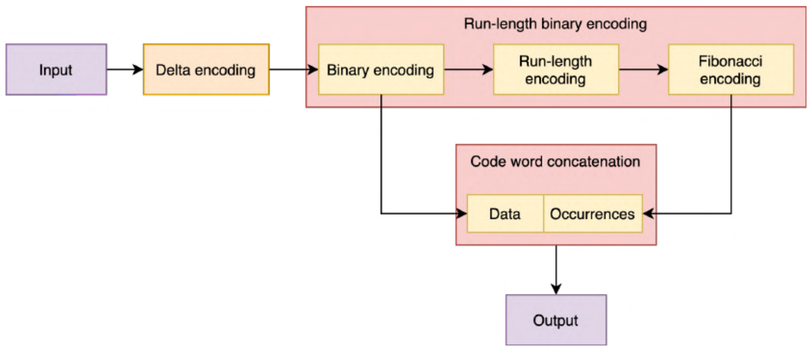


Figure 5.7: RLBE compression steps

Table 5.1: RAKE dictionary

| symbol | code | length |
|---|---|---|
| -1 | 1 | 1 |
| +1 | 01 | 2 |
| -2 | 001 | 3 |
| +2 | 0001 | 4 |
| ... | ... | ... |
| -R | 0...1 | 2R - 1 |
| +R | 00...1 | 2R |
| 0 | all zeros | 2R |

**Preprocessing** In this phase, a dictionary is used to transform original data. For this purpose, many algorithms can be used, such as the Huffman coding presented in Subsection 5.2.4, but since the aim is that of obtaining sparsity, the RAKE dictionary uses a code similar to unary coding thus every codeword has at most one bit set to 1. This dictionary doesn't depend on symbol probabilities, so no learning phase is needed. Table 5.1 shows a simple RAKE dictionary.

**Compression** This phase works, as suggested by the algorithm name, as a rake of $n$ teeth. Figure 5.8 shows an execution of the compression phase, given a preprocessed input and $n = 4$.

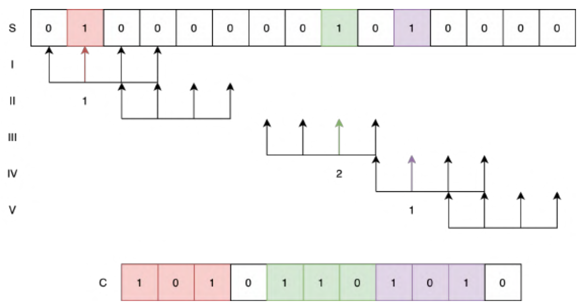


Figure 5.8: RAKE algorithm execution

The rake starts at position 0, in case there is no bit set to 1 in the rake interval, then 0 is added to the code, otherwise 1 and followed by the binary representation of relative index for the first bit set to 1 in the rake (2 bits in the example in Figure 5.8). After that, the rake is shifted right of $n$ positions for the first case or starting right to the first found bit. In the figure, the rake is initially at position 0, and the first bit set to 1 is in relative position 1 (output: 1 followed by 01), then the rake advances of 2 positions (after the first 1 in the rake); all bits are set to zero (output: 0) thus that rake is moved forward by 4 places; the first bit set to 1 in the rake has relative index 2 (output: 1 followed by 10) thus the rake is advanced by 3 places and the process continues for the two last rake positions (output: 101 and 0 respectively).

**Decompression** The decompression processes the compressed bit stream replacing 0 with $n$ occurrences of 0 and 1 followed by an offset with a number 0 equal to the offset followed by a bit set to 1. The resulting stream is decoded on the fly using the dictionary.

### 5.2.5 Others

This subsection presents other time series compression algorithms that cannot be grouped in the previously described categories.

#### Major Extrema Extractor (MEE)

This algorithm is introduced in [23] and exploits time series features (maxima and minima) to achieve compression. For this purpose, strict, left, right, and flat extrema are defined. Considering a time series $TS = [(t_0, x_0), \dots, (t_n, x_n)]$, $x_i$ is a minimum if it follows these rules:

- **strict**: if $x_i < x_{i-1} \wedge x_i < x_{i+1}$
- **left**: if $x_i < x_{i-1} \wedge \exists k > i : \forall j \in [i, k], x_j = x_i \vee x_i < x_{k+1}$
- **right**: if $x_i < x_{i+1} \wedge \exists k < i : \forall j \in [k, i], x_j = x_i \vee x_i < x_{k-1}$
- **flat**: if $(\exists k > i : \forall j \in [i, k], x_j = x_i \vee x_i < x_{k+1}) \wedge (\exists k < i : \forall j \in [k, i], x_j = x_i \vee x_i < x_{k-1})$

For maximum points, they are defined similarly.

After defining minimum and maximum extrema, the authors introduce the concept of importance, based on a distance function *dist* and a compression ratio $\rho$: $x_i$ is an important minimum if:

$$\begin{aligned} \exists i_l < i < i_r : \quad & x_i \text{ is minimum in } \{x_l, \dots, x_r\} \wedge \\ & dist(x_l, x_i) < \rho \wedge dist(x_i, x_r) < r \end{aligned}$$

Important maximum points are defined similarly.

Once important extrema are found, they are used as a compressed representation of the segment. This technique is a lossy compression technique, as the Segment Merging one described in the next section. Although the compressed data can be used to obtain original data properties useful for visual data representation (minimum and maximum), it is impossible to reconstruct the original data.

#### Segment Merging (SM)

This technique, presented in [41] and reused in [28] and [46], considers time series with regular timestamps and repeatedly replaces sequences of consecutive elements (segments) with a summary consisting of a single value and a representation error, as shown in Figure 5.9 where the error is omitted.

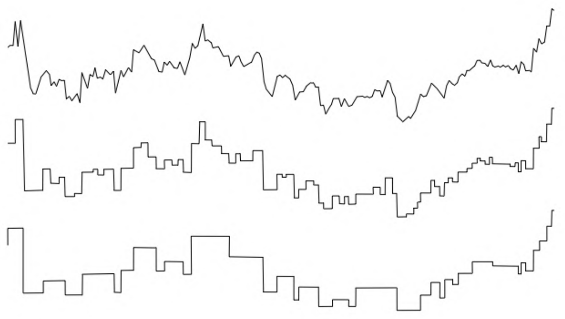


Figure 5.9: Example of segment merging technique, with increasing value time interval

After compression, segments are represented by tuples $(t, y, \delta)$ where $t$ is the starting time of the segment, $y$ is the constant value associated with the segment, and $\delta$ is the segment error. The merging operation can be applied either to a set of elements or to a set of segments, to further compress a previously compressed time series. The case of elements is an instance of the case of segments, and it is immediate to generalize from two to more elements/segments. The description is limited to the merge of two consecutive segments represented by the tuples $(t_i, y_i, \delta_i)$ and $(t_j, y_j, \delta_j)$, where $i < j$, into a new segment represented by the tuple $(t, y, \delta)$ computed as:

$$t = t_i$$

$$y = \frac{\Delta t_i \cdot y_i + \Delta t_j \cdot y_j}{\Delta t_i + \Delta t_j}$$

$$\delta = \sqrt{\frac{\Delta t_i \cdot (y_i^2 + \delta_i^2) + \Delta t_j \cdot (y_i^2 + \delta_i^2)}{\Delta t_i + \Delta t_j} - y^2}$$

where $\Delta t_x$ is the duration of the segment $x$, thus $\Delta t_i = t_j - t_i$, $\Delta t_j = t_k - t_j$ and $t_k$ is the timestamp of the segment after the one starting at $t_j$.

The sets of consecutive segments to be merged are chosen to minimize segment error with constraints on a maximal acceptable error and maximal segment duration.

This compression technique is lossy and the result of the compression phase can be considered both as the compressed representation and as the reconstruction of the original time series, without any additional computations executed by a decoder.

### Continuous Hidden Markov Chain (CHMC)

The idea behind this algorithm is that the data generation process follows a probabilistic model and can be described with a Markov chain [38]. This means that a system can be represented with a set of finite states $S$ and a set of arcs $A$ for transition probabilities between states.

Once the hidden Markov chain is found using known techniques, a lossy reconstruction of the original data can be obtained by following the chain probabilities.

### 5.2.6 Algorithms summary

When the taxonomy described in Section 5.1.2 and Section 5.2 is applied to the above techniques, the result is the classification reported in Table 5.2 that summarizes the properties of the different implementations.

Table 5.2: Compression algorithms classification

| Dictionary based techniques | | | | | |
|---|---|---|---|---|---|
| | Non-adaptive | Lossless | Symmetric | min $\rho$ | max $\epsilon$ |
| TRISTAN | - | - | ✓ | ✓ | ✓ |
| CORAD | - | - | ✓ | - | ✓ |
| A-LZSS | - | ✓ | ✓ | - | NA |
| D-LZW | - | ✓ | ✓ | ✓ | NA |
| **Function Approximation** | | | | | |
| | Non adaptive | Lossless | Symmetric | min $\rho$ | max $\epsilon$ |
| PPA | ✓ | - | - | ✓ | ✓ |
| CPT | ✓ | - | - | ✓ | ✓ |
| DWT | ✓ | - | - | ✓ | ✓ |
| DFT | ✓ | - | - | ✓ | ✓ |
| DCT | ✓ | - | - | ✓ | ✓ |
| **Autoencoders** | | | | | |
| | Non adaptive | Lossless | Symmetric | min $\rho$ | max $\epsilon$ |
| RNNA | - | - | ✓ | - | ✓ |
| RCA | - | - | ✓ | - | ✓ |
| DZip | - | ✓ | ✓ | - | NA |
| **Sequential algorithms** | | | | | |
| | Non adaptive | Lossless | Symmetric | min $\rho$ | max $\epsilon$ |
| DRH | - | ✓ | ✓ | - | NA |
| SPRINTZ | ✓ | ✓ | ✓ | - | NA |
| RLBE | ✓ | ✓ | ✓ | - | NA |
| RAKE | ✓ | ✓ | ✓ | - | NA |
| **Others** | | | | | |
| | Non adaptive | Lossless | Symmetric | min $\rho$ | max $\epsilon$ |
| MEE | ✓ | - | - | ✓ | - |
| SM | ✓ | - | - | ✓ | - |
| CHMC | - | - | ✓ | - | - |

Where ✓ indicates if the related property is true, - if it is not, and NA if it is not applicable. Min $\rho$ and max $\epsilon$ indicate respectively the possibility to set a minimum compression ratio or a maximum reconstruction error.

## 5.3 Performances comparison

This section shows the different performances of the techniques in Section 5.2, as reported by their authors. To ensure a homogeneous presentation of the different techniques and for copyright reasons, the figures are redrawn using the same graphical style for all of them and maintain a faithful reproduction with respect to the represented values.

The experiments presented in those studies were based on the following datasets:

- ICDM challenge [86] – Traffic flows;
- RTE France [1] – Electrical power consumptions;
- ACSF1[2] – Power consumptions;
- BAFU[3], – Hydrogeological data;
- GSATM[4] – Dynamic mixtures of carbon monoxide (CO) and humid synthetic air in a gas chamber;
- PigAirwayPressure[5]– vital signs;
- SonyAIBORobotSurface2[5] – X-axis of robot movements;
- SMN [74] – Seismic measures in Nevada;
- HAS [73] – Human activity from smartphone sensors;
- PAMAP [60] – Wearable devices motion and heart frequency measurements;
- MSRC-12[6] – Microsoft Kinect gestures;
- UCI gas dataset[4] – Measuring gas concentration during chemical experiments;
- AMPDs[7] – Measuring maximum consumption of water, electricity, and natural gas in houses.

### 5.3.1 Algorithms

To foster the comparison of the different algorithms, the accuracy, and compression ratio obtained in the experiments are reported synoptically as described by their authors.

#### TRISTAN

**Accuracy** The creation of the dictionary depends on the dictionary sparsity parameter that is directly related to its size and the accuracy can be influenced by this value, as shown in Equation 5.5. Experimental results are shown in Figure 5.10, where sparsity is related to RMSE.
**Compression ratio** The sparsity parameter also affects the compression ratio. To assess this dependency, the authors execute experiments on two different datasets both containing one measurement per minute with daily segments: RTE and ICDM, using dictionaries consisting respectively of 16 and 131 atoms.

The resulting compression ratios are $\rho = 0.5$ for RTE and $\rho = 0.05$ for ICDM, thus higher sparsity values (larger dictionaries) allow for better compression ratios.

[1]https://data.rte-france.com
[2]http://www.timeseriesclassification.com
[3]https://www.bafu.admin.ch
[4]https://archive.ics.uci.edu
[5]https://www.cs.ucr.edu/
[6]https://www.microsoft.com
[7]http://ampds.org

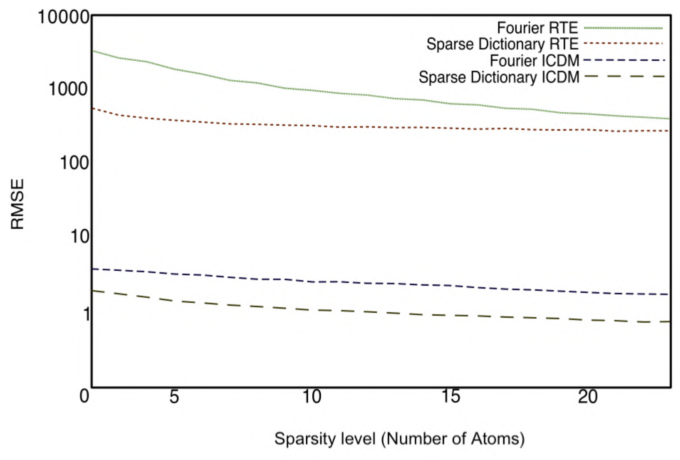


Figure 5.10: RMSE results depending on sparsity for TRISTAN[50]

### CORAD

**Accuracy** The authors of CORAD measure accuracy using MSE and the best-reported results for their algorithms are 0.03 for the BAFU dataset, 0.11 for the GSATM dataset, and 0.04 for the ACSF1 dataset. As for TRISTAN, accuracy depends on time series segments redundancy and correlation.
**Compression ratio** The best reported results for CORAD compression ratio are $\rho = 0.04$ for the ACSF1 dataset, $\rho = 0.07$ for the BAFU dataset and $\rho = 0.06$ for the GSATM dataset.

The compression ratio is affected by the parameters used during training and encoding: error threshold, sparsity, and segment length. Their effects are depicted in the three plots in Figure 5.11, that represent the compression factor $(\frac{1}{\rho})$ obtained for different values of the parameters.

### A-LZSS

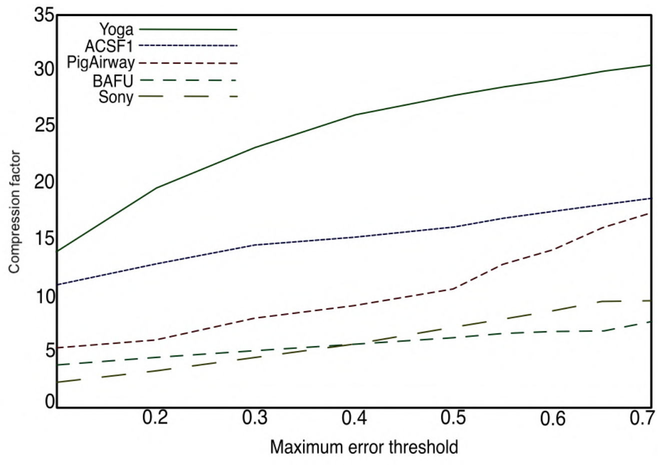
The compression ratio obtained by the A-LZSS algorithm depends on two parameters: $L_n$ and $D_n$, which are respectively the lookahead distance, and the dictionary atoms length. From the experiments conducted by the authors in [59], the best compression ratio is obtained setting $L_n = 5$ and $D_n = 10$. Increasing $L_n$ and decreasing $D_n$ penalize this quality index, as values of $L_n < 4$.

### D-LZW

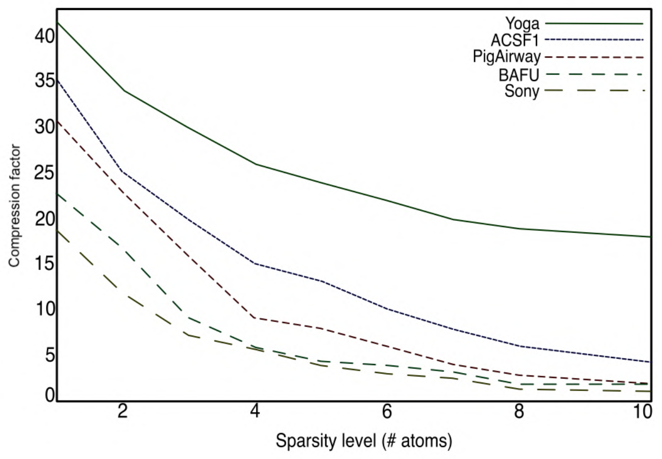
The authors didn't provide a detailed study on the compression performances, limiting the experimental results only to EGC and PPG signals. Nevertheless, they did a comparison with Bzip2 and Gzip, giving the results shown in Table 5.3.

Table 5.3: D-LZW compression ratio comparison

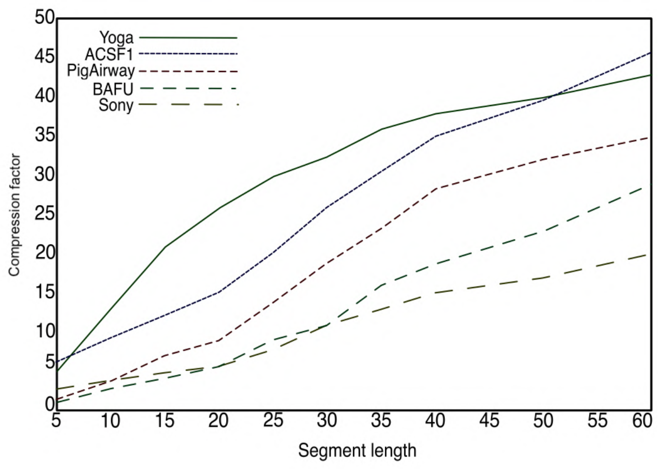
| | D-LZW | Bzip2 | Gzip |
|---|---|---|---|
| Compression ratio (%) | 76.47 | 60.52 | 53.63 |



(a) Varying error threshold



(b) Varying dictionary atoms



(c) Varying segment length

Figure 5.11: Compression factor ($\frac{1}{\rho}$) with varying parameters for CORAD [42]

### Piecewise Polynomial Approximation (PPA)

Accuracy and compression ratio of PPA are affected by the maximum polynomial degree and the maximum allowed deviation, the parameters of the method as described in Subsection 5.2.2.

**Accuracy**

Errors are bounded by the maximum deviation parameter and the maximum polynomial degree can affect accuracy: deviation is always under the threshold and using higher degree polynomials yield more precise reconstructions.

Figure 5.12 shows the relation between maximum deviation and MSE.

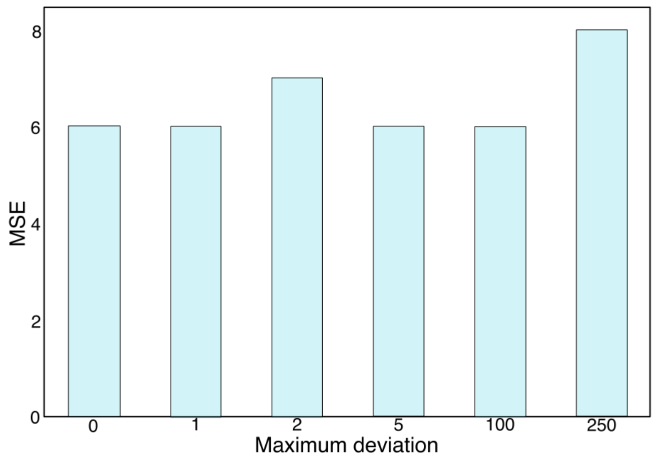


Figure 5.12: Relation between MSE and maximum deviation for PPA[35]

**Compression ratio**

Both the maximum deviation and polynomial degree parameters affect the compression ratio, as it can observed in Figure 5.13 where the compression factor ($\frac{1}{\rho}$) is reported for different values of parameters. The best compression ratios, $\rho = 0.06$ for REDD and $\rho = 0.02$ for Office, are achieved for the highest degree polynomials. As expected, also for increasing maximum deviation the compression factor is monotonically increasing.

### Chebyshev Polynomial Transform (CPT)

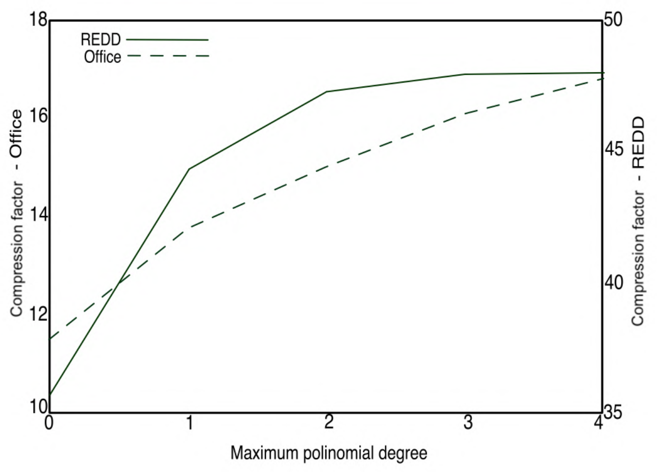
Despite the absence of experimental results in the original paper, some considerations can be done. The algorithm has three parameters:

- Block size;
- Maximum deviation;
- Number of quantization bits.

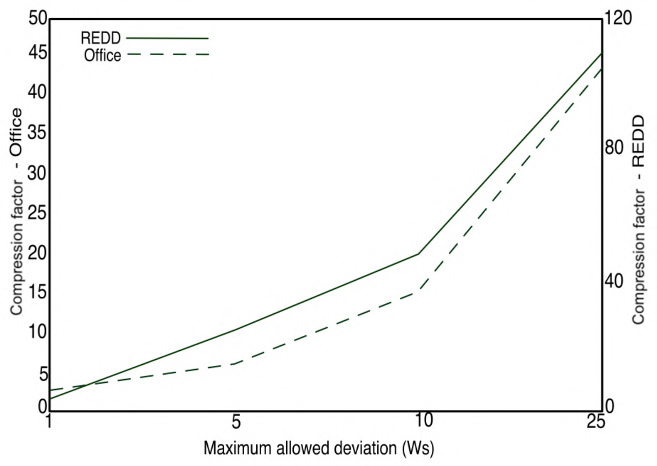
Similarly to PPA, in CPT it is possible to obtain higher compression ratios at the cost of higher MSE by increasing the maximum deviation parameter. The same holds for the block size: larger blocks correspond to better compression ratios and higher decompression errors [33]. Similarly, it is possible to suppose that using fewer quantization bits would entail a less accurate decompression and a better compression ratio.

### Discrete Wavelet Transform (DWT)

As for CPT, experimental results are not provided, but also DWT performances follow the same rules as for PPA since the compression ratio and threshold errors



(a) Varying maximum polynomial degree



(b) Varying maximum deviation

Figure 5.13: PPA [19] compression factor ($\frac{1}{\rho}$) for varying parameters

are strictly correlated. The threshold error can be fixed a priori to have a higher compression ratio or a higher accuracy.

### Discrete Fourier Transform (DFT)

The authors tested the performance of the DFT algorithm using three types of signals: Block, Heavy Sine, and Mishmash. The compression ratio depends on the percentage of coefficients that are removed. The optimum threshold is derived by iterative tests, and it depends on the signal type. As stated by the authors, the results obtained by DWT are comparable to the ones obtained by DWT: the choice of the algorithm highly depends on the type of signal that has to be compressed [40].

### Discrete Cosine Transform (DCT)

The accuracy obtained by this algorithm depends on the chosen compression ratio. The authors used the ARIANE 5 (temperature sensor, and vibration sensor) and AISat temperature sensor for their experiments. The results show that the performances of this algorithm are not homogeneous, as they highly depend on the input dataset. Even if the input data seems to be similar, belonging to the same domain: the performances obtained on the ARIANE 5 and AISat temperature sensors are complitely different, as shown in Table 5.4 [52].

Table 5.4: DCT experimental results

| | Compression ratio (%) | MSE |
|---|---|---|
| A5 temperature sensor | 49.60 | 0.0009 |
| | 82.60 | 0.03 |
| A5 vibration sensor | 3.10 | 0.04 |
| | 7.10 | 0.10 |
| | 22.90 | 0.30 |
| AISat temperature sensor | 2.30 | 6.50 |
| | 5.20 | 92.80 |
| | 8.80 | 317.92 |

### Recurrent Neural Network Autoencoder (RNNA)

**Accuracy** One of the parameters of the RNNA based methods is the threshold for the maximally allowed deviation, whose value directly affects accuracy. For the experiments on univariate time series presented in [35] the authors report, considering the optimal value for deviation parameter, an RMSE value in the range $[0.02, 0.07]$ for different datasets. In the same experiments, RMSE is in the range $[0.02, 0.05]$ for multivariate time series datasets. Thus, there is no significant difference between univariate and multivariate time series.
**Compression ratio** Similarly, the maximally allowed deviation affects the compression ratio. According to the results reported by the authors, in this case, there is not a significant difference between univariate and multivariate time series: $\rho$ is in the range $[0.01, 0.08]$ for the univariate time series dataset and in the range, $[0.31, 0.05]$ for the multivariate one.

### Recurrent Convolutional Autoencoder (RCA)

The experimental results reveal a slight improvement respect to the RNNA model over the traffic flow dataset[89]: without the convolutional layer, the RMSE value is 10.37, while with the convolutional layer, the RMSE decreases to 8.17.

Since the original paper is focused on the prediction of the next values only, the compression ratio is missing.

### DZip

The authors tested the model with real and synthetic datasets and compared the results with the state-of-the-art compression algorithms, including RNNA. The results revealed that it outperforms RNNA autoencoder. The compression ratio depends on the vocabulary size, that is negatively affected by larger sizes. Another factor that influences this parameter is the size of the original data: larger is the time series to be compressed, better is the compression ratio [31].

### Delta encoding, Run-Length, and Huffman (DRH)

DRH, as most of the following algorithms, is a lossless algorithm, thus the only relevant performance metric is the compression ratio.

**Compression ratio** This parameter highly depends on the dataset: run-length algorithm combined with delta encoding achieves good results if the time series is composed of long segments that are always increasing or decreasing of the same value or constant. Experimental results are obtained on temperature, pressure, and light measures datasets. For this type of data, DRH appears to be appropriate, since the values are not highly variable and have a relatively long interval with constant values. The resulting compression factors are reported in Table 5.5 [53].

Table 5.5: DRH Compression factors

| | $^1/_\rho$ |
|---|---|
| Indoor temperature | 92.2 |
| Outdoor temperature | 91.17 |
| Pressure | 82.55 |
| Light | 83.53 |

### Sprintz

**Compression ratio** This algorithm is tested over the datasets of the UCR time series archive, which contains data coming from different domains. In the boxplots in Figure 5.14, compression factors are shown for 16 and 8-bit data and compared with other lossless compression algorithms. From this graph, it is possible to see that all algorithms achieve better (higher) compression factors on the 8-bit dataset. This could be due to the fact that 16-bit data are more likely to have higher differences between consecutive values.

Another interesting result is that the FIRE forecasting technique improves this compression, especially if combined with Huffman coding when compared with Delta coding [6].

### Run-Length Binary Encoding (RLBE)

**Compression ratio** The author of this algorithm in [75] report compression ratio and processing time of RLBE with respect to those of other algorithms as shown in Figure 5.15 where ellipses represent the inter-quartile range of compression ratios and processing times. RLBE achieves good compression ratios with a 10% variability and with low processing time.

### RAKE

**Compression ratio** The authors of RAKE in [9] run experiments on sparse time series with different sparsity ratios $p$ and different algorithms. The corresponding compression factors are reported in Table 5.6. It can be noticed that the compression factor is highly dependent on sparsity and lower sparsity values correspond to higher compression factors (better compression). Thus, the RAKE algorithm is more suitable for sparse time series.

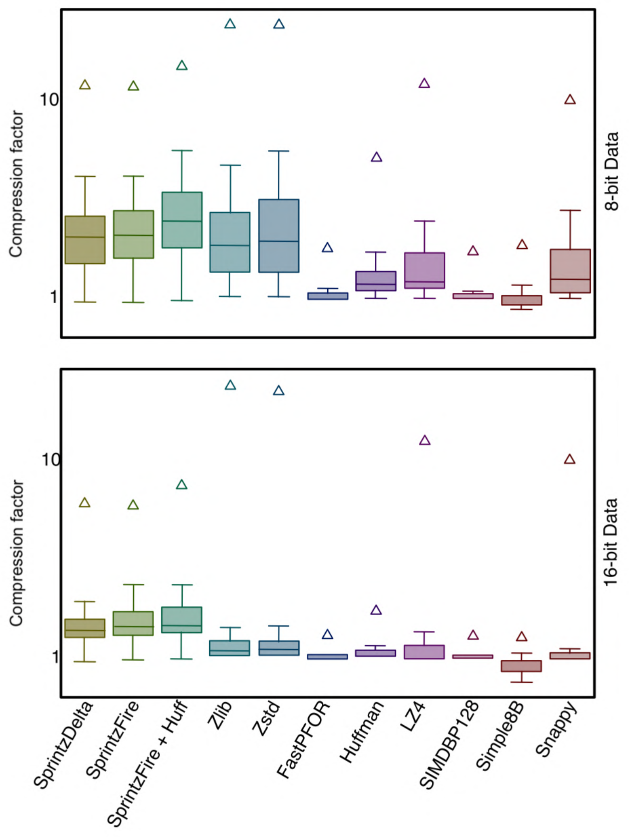


Figure 5.14: Sprintz compression factors versus other lossless compression algorithms [6]

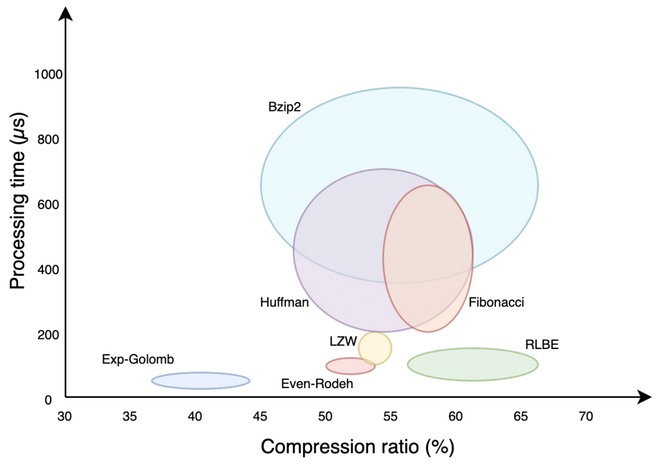


Figure 5.15: RLBE compression ratio versus other lossless compression algorithms [75]

## Major Extrema Extractor (MEE)

The compression ratio is one of the parameters of the MEE algorithm, thus the only relevant performance metric is the accuracy.

**Accuracy** Even though MEE is a lossy compression algorithm, the authors have not provided any information about accuracy. Since the accuracy depends on the compression ratio, the authors recommend choosing a value that is not smaller than the percentage of non-extremal points, to obtain an accurate reconstruction

Table 5.6: Compression factor results for RAKE versus other lossless compression algorithms

| | p | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Algorithm | 0.002 | 0.005 | 0.01 | 0.05 | 0.1 | 0.15 | 0.2 | 0.25 |
| OPT | 48.0 | 22.0 | 12.4 | 3.5 | 2.1 | 1.6 | 1.4 | 1.2 |
| RAKE | 47.4 | 21.5 | 12.0 | 3.5 | 2.1 | 1.6 | 1.4 | 1.2 |
| RAR | 22.1 | 12.1 | 7.4 | 2.6 | 1.7 | 1.4 | 1.2 | 1.2 |
| GZIP | 21.4 | 11.8 | 7.4 | 2.6 | 1.8 | 1.4 | 1.3 | 1.2 |
| BZIP2 | 26.0 | 14.2 | 8.7 | 2.6 | 1.7 | 1.3 | 1.2 | 1.1 |
| PE | 29.5 | 11.9 | 5.9 | 1.2 | 0.6 | 0.4 | 0.3 | 0.2 |
| RLE | 35.7 | 15.7 | 8.4 | 2.1 | 1.2 | 0.9 | 0.8 | 0.6 |

of the compressed data [23].

### Segment Merging (SM)

Since this algorithm is used for visualization purposes, a first evaluation can be a qualitative observation. In Figure 5.9 it can be seen how a light sensor time series is compressed considering different time series segment lengths. For this algorithm, compression ratio and error are strictly correlated, as shown in Figure 5.16 where FWC and CB-*m* correspond to different versions of the algorithm and different parameters [28]. It can be observed that compression ratio values

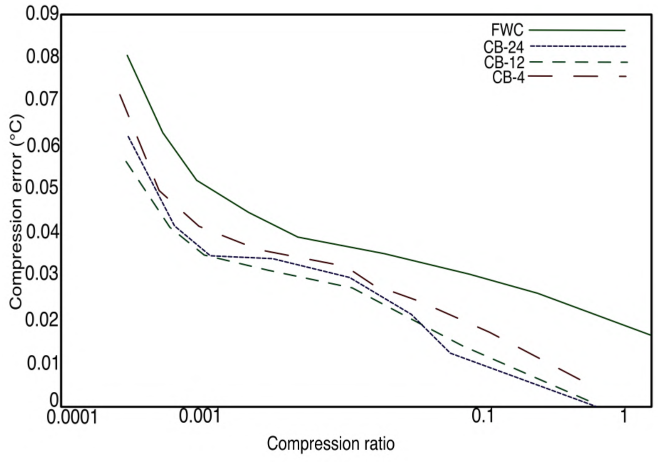


Figure 5.16: Correlation between compression ratio and error on a temperature dataset [28]

close to 1 correspond to error values close to zero.

The dependence of compression ratio and error changes for different datasets. Knowing the function that correlates the compression ratio and error for a given dataset, it is possible to set the compression ratio that corresponds to the desired maximum compression error.

### Continuous Hidden Markov Chain (CHMC)

For this algorithm, the authors haven't provided any quantitative evaluation for compression performances: their assessment is qualitative and mainly focused on the compression and reconstruction of humanoid robot motions after a learning phase [38].

### 5.3.2 Intergroup comparison

After passing through the single techniques, this subsection proposes an intergroup comparison to address the general characteristics of each group.

#### Dictionary-based

The dictionary-based approaches include both lossy and lossless compression. The compression ratio depends on the atoms size in the dictionary and on the dataset repetitiveness. For large datasets, the storage needed for the dictionary can be considered irrelevant, while for smaller datasets it can be inefficient. The training phase can be inefficient for databases with low repetitiveness.

#### Functional approximation

Functional approximation includes only lossy techniques. The compression ratio can be set in advance, as the maximum reconstruction error. They work better with regular data with low fluctuations and peaks.

#### Autoencoders

Autoencoders includes lossy and lossless techniques. The error threshold can be set in advance, while the compression ratio depends on the quality of the training phase and in the regularity of the dataset. If the autoencoder receives data with different seasonability and distribution respect to the ones found in the training set, its prediction capacity becomes low. A weak point is the computational cost of the training set: recurrent neural networks are more expensive to be trained respect to classic neural network autoencoders and need a large amount of data in the training set.

#### Sequential algorithms

The selected techniques in this group includes lossless techniques only. The compression ratio depends on the data sparsity and regularity, especially for those that use delta and run-length encoding. They don't need a training phase and work well with low-computational resources.

#### Others

This group includes lossy techniques, and their peculiarities depend on the single technique.

## 5.4 Conclusions

This chapter provides an analysis of the most relevant compression techniques for time series and proposes a taxonomy to classify them. Even if this is a meta-analysis, it is informative for the reader about the suitability of a technique for specific contexts, for example, when a fixed compression ratio or accuracy is required, or for a dataset having particular characteristics.

The selection of the most appropriate compression technique highly depends on the time series characteristics, the constraints required for the compression, or purpose.

Starting from the first case, time series can be characterized by their regularity (that means, having or not fixed time intervals between two consecutive measurements), sparsity, fluctuations, and periodicity. Considering **regularity**, only techniques based on function approximation can be applied directly to the time series without the need of transformations. This is because that approach approximates a function passing through the given points, having in the x-axis time, so the time-distance between consecutive point can be arbitrary. All the other techniques assumes the time-distance between two points to be fixed. To apply them, some transformations on the original time series are needed. Some techniques work better having in input time series with high **sparsity**, as the ones proposed in Section 5.2.4. Another characteristic to consider is **fluctuations**. The lossy compression techniques presented in this survey cannot achieve good accuracy results having in input time series with high non-regular fluctuations. This is caused by the fact that functional approximation and autoencoder techniques tend to flatten peaks, while the ones based on dictionaries would need overly large dictionaries. In the other hand, functional approximation and lossy autoencoders can be taken in consideration if it is sufficient to reconstruct the general trend of the original time series. Lastly, dictionary-based techniques work well with time series with high **periodicity**. In this scenario, a relatively small dictionary could be enough to represent the whole time series by combining atoms. Lossy autoencoders also improve their reconstruction accuracy, having in input periodical time series.

The constraints required for compression include compression ratio, reconstruction accuracy, speed, and availability of a training dataset. As shown in Table 5.2, some techniques allow to specify a minimum **compression ratio** and a maximum **reconstruction accuracy** error. This could be necessary in some contexts in which space or accuracy are given as constraints. Moreover, compression **speed** can be relevant, in particular when hardware with low computation capacity are involved. For this context, sequential algorithms can compress at high speed, without needing powerful devices. Depending on the presence or not of a **training set**, all the techniques that need a training phase must be discarded.

The different purposes for time series compression include analysis, classification, and visualization. For time series **analysis**, functional approximation and lossy autoencoders can help to reduce noise and outlayers, for those time series that have many fluctuations. Autoencoders can be used also to more deep analysis, as anomaly detections, starting from the compressed representation, as shown in [**?** ]. The CHMC technique can also be used for this aim, since time series can be represented with a probability graph. Autoencoders can be used for **classification** too, since the autoencoder can be used to reduce the dimensionality of the input. Lastly, for **visualization** purposes, MEE and SM can be used to having different definitions of long time series.

# Chapter 6

# Time series classification

## 6.1 Introduction

Due to the proliferation of connected devices as smartphones, wearables, and IoT devices [91], the amount of time series data is continuously growing [61, 72, 88]. This large volume of time series data needs to be compressed in order to reduce the required storage space and associated storage costs. Besides storage optimization, it is desirable to allow operations on the compressed data. The chapter focuses on classification tasks. This problem can be found in different applications, from biological [21], medical [37, 3], and, in general, IoT [18].

Classification algorithms are generally trained directly on the original training sets or on their dimensional reductions to achieve better performances. This means that, once time series data are compressed, they must be decompressed before being passed to a classification model. This operation has a very high computational cost, in particular for time series that grow in time [5]. In this case, in fact, new measurements may update the compressed representation and, since they may change the classification of the whole time series, to update the classification result, the time series have to be decompressed at every new added measure. This can be solved if the compressor is designed such that the compressed representation can be used both to train a classifier, and to reconstruct the original time series.

Existing time series compression models do not fit this requisite: despite they achieve good results in the compression task, the compressed representation is not suitable to train a classification algorithm. Most widely used compression techniques developed for time series [13] include dictionary-based, functional approximation, sequential algorithms, and autoencoders.

**Dictionary-based** This technique creates a dictionary during the training phase composed by segments found in the dataset that has to be compressed. The compression result is a list of indices pointing to dictionary segments, in place of the original segments. The indices assigned to segments are determined by storage efficiency strategies, or they are assigned incrementally as new segments are encountered during compression. For this reason, from the compressed representation, it is impossible to extract additional information and the only relationship between original data and the compressed representation is given by the dictionary itself: from the classification point of view, the compressed representation is similar to a random sequence. In addition, near indices can be assigned to very different segments, and vice-versa, so it is impossible to compute the distance or similarity between two time series compressed using dictionary indices.

**Functional approximation** This technique tries to find a function (polynomials, wavelets, and Fourier transforms) that has the same shape as the original data. The compressed representation is made by the function coefficients. Generally, one function cannot well approximate a long time series, especially if time series are long. For this reason, the original time series is divided into time windows of fixed or variable size. Having many segments makes it difficult to classify the whole time series, that would be represented as a list of polynomial.

**Sequential algorithms** Sequential algorithms still use a dictionary representation, Huffman or Fibonacci encoding, that, as the dictionary-based models, are not strictly related to the original data. Besides this, they compose different algorithms, that are not suitable to be used by classifier such as:

- Run-length encoding: compress identical sequential values with the same value and a counter;
- Delta-encoding: transform the original data to a sequence representing the differences between one value and the previous one. It is usually combined with run-length to compress constant ascending or descending sequences.

**Autoencoders** Autoencoders are special neural network models that are composed by an encoder $e(x)$ that receives as input the original data $x \in \mathbb{R}^{l \times f}$ where $l$ is the length of the time series and $f$ is the number of features, and returns its representation in a lower dimensionality, embedding $E \in \mathbb{R}^p, p \ll l \cdot f$, and a decoder $d(E)$ that receives the embedding $E$ generated by the encoder for a certain input and returns the reconstruction of the original data, $\hat{x}$. The network is trained in such a way that the reconstruction error $E$ between $d\left(e\left(x\right)\right)$ and $x$ is $< \epsilon$ [30].

Neural network autoencoders have a fixed size for the input and output. In addition to this, the weights of the network give a high importance to the position of the values in the training dataset, which results in a weak generalization. An example is the compression and reconstruction of time series data, in which each value may be shifted by some time stamps. To address these two problems, recurrent autoencoders are introduced.

**Recurrent autoencoders** The recurrent autoencoder is a many-to-many generative recurrent neural network [80, 30]. It is composed by an encoder and a decoder. The encoder takes as input a time series $X = [x_1, \ldots, x_l]$, and, for each $x_i \in R^f$ it updates the hidden state $h_{i+1}$ and produces an embedding $E \in R^e, e < lf$ of the processed time series. The embedding is then replicated $l$-times, as the length of the original time series, and passed as input to the decoder. The output of each recursion step in position $l + i$ is the reconstruction of the original time series in position $i$, marked as $\hat{x_i}$. The weights $W$, $U$, and $V$ are shared among the network and learned during training, as shown in Figure 6.1. For each time step, the hidden state $h$ and output are computed using Equations 6.1.

$$\begin{aligned} a_i &= b + W h_{i-1} + U x_i \\ h_i &= \tanh(a_i) \\ o_i &= c + V h_i \\ \hat{x}_i &= \text{softmax}(o_i) \end{aligned} \tag{6.1}$$

Where $b$ and $c$ are bias vectors, $W$, $U$, and $V$ are weight matrices, and $h_0$ is initialized before training. Figure 6.1 shows the unfolded computational graph of the basic structure of this network.

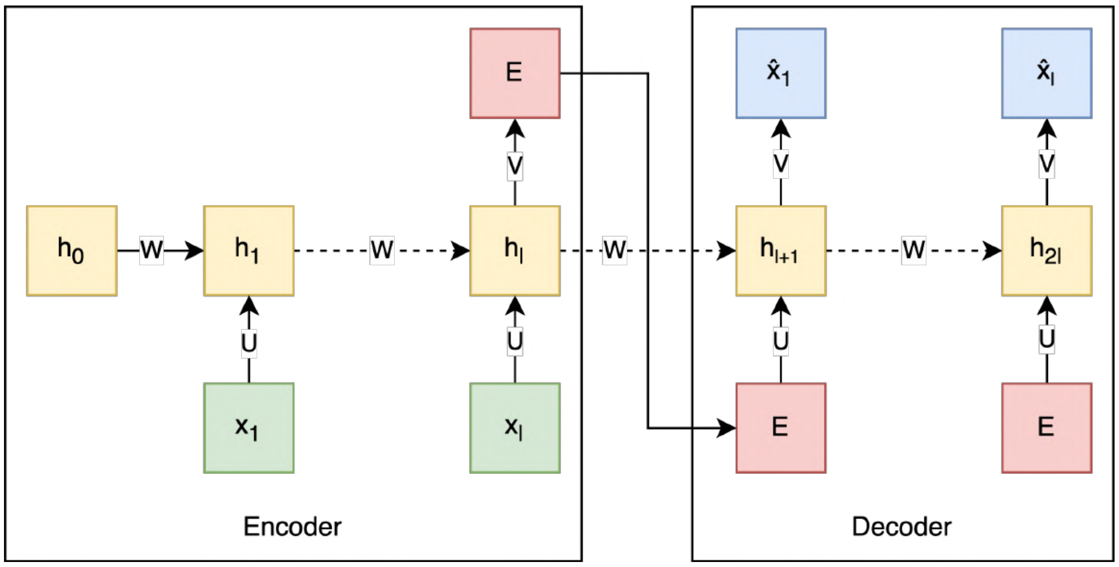


Figure 6.1: Unfolded schema of a basic recurrent autoencoder

Deep recurrent autoencoders are an extension of this model [56]: instead of having one single hidden state, they have $p$ hidden states positioned one on top of the other before the output layer. Each hidden state $h_{i,k}, k \in [1, p]$ is updated by $\tanh(b + W_K h_{i-1,k} + U_k h_{i,k-1})$.

Another way to update the hidden states of a recurrent network is applying long-short-term memory (LSTM) models [26]. The particularity of this network is that hidden states are replaced by LSTM cells that have the ability to keep important information and skip non-relevant information through internal gates.

Hence, this chapter presents a recurrent autoencoder that returns a compressed representation of the original time series that can be used both to reconstruct the original one (lossy compression) and to train a classifier, without knowing it in advance, achieving an error in the classification which is comparable with the one obtained if the classifier was trained on the original data.

The contributions of this work are the following:

- a new model that achieves good results in the classification of compressed representations of time series data;
- the model introduces a new loss computed on the embedded representation that preserve distances between the embedded vectors and, consecutively, better classification accuracy, as shown in the results section;
- being the first work stressing the importance of finding an improved compressed representation of time series that can be used both for general-purpose classification tasks and to reconstruct the original time series;

- the performances of the proposed model are demonstrated over multivariate and univariate time series datasets taken from different domains.

The rest of the chapter is organized as follows: Section 6.2 reviews some related work, Section 6.3 shows the proposed method, and Section 6.4 experiments and results.

## 6.2 Related work

In the literature review, the focus is on autoencoders that use the compressed representation both to classify and reconstruct the original time series. Relevant works that use recurrent neural network autoencoders to reach this goal were not found, so the ones that use simple autoencoders are considered.

The general model is shown in Figure 6.2. During the training phase, the model receives as input a batch of $n$ time series $B_{TS} = (X_1, \dots, X_n)$ and a batch of $n$ labels $B_L = (Y_1, \dots, Y_n)$. The autoencoder computes the embedding of each time series in $B_{TS}$ and their reconstructions derived from the corresponding embedded representations $\hat{B}_{TS}$. The embedded representations found by the autoencoder are passed as input to the NN classifier that produces as output the predicted labels for each embedded representation $\hat{B}_L$. To update the network weights, two distinct losses are computed, respectively for the reconstruction and for the classification. The two losses are then combined to get the overall loss of a training step. The weights of the autoencoder and the neural network (NN) classifier are trained together at each training epoch.

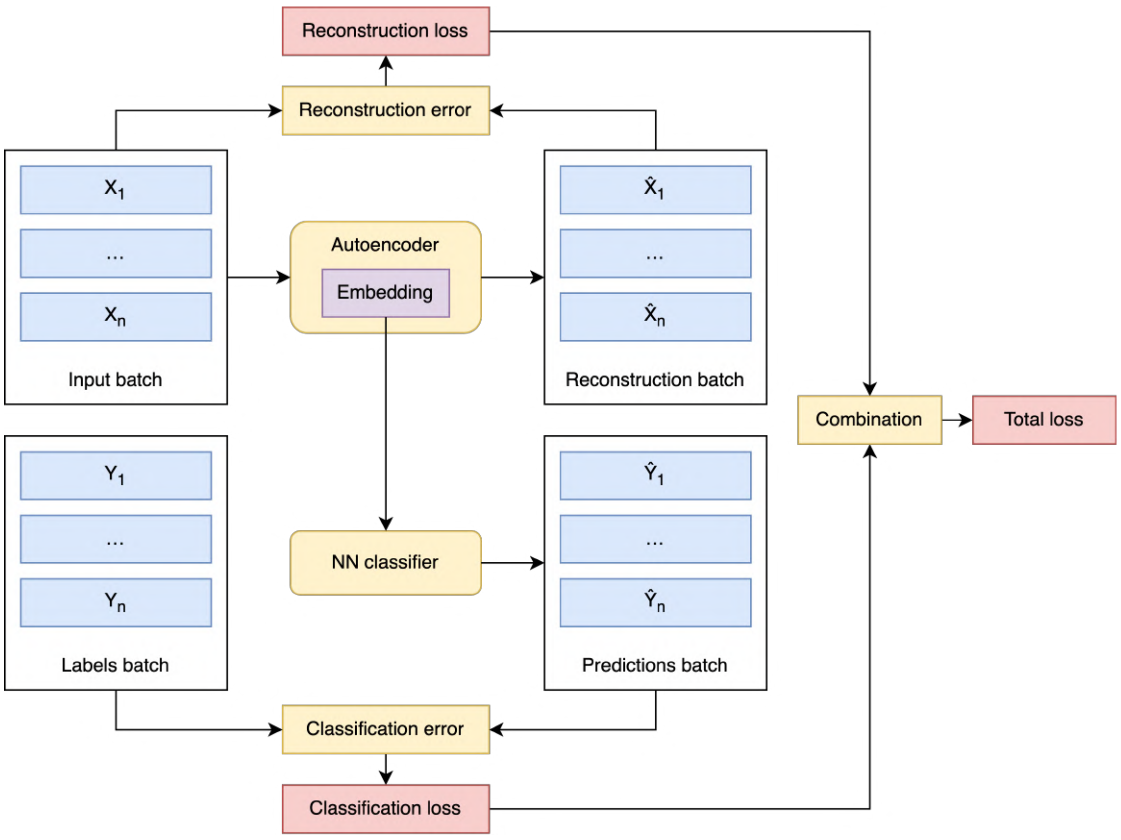


Figure 6.2: Schema of autoencoder integrated with a NN classifier

This model has also been used for different objectives as discriminative restricted Boltzmann machine and label-consistent dictionary learning. An example of this approach is applied in [27] for analyzing biomedical signals.

Though this model achieves good results, it has a problem related to generalization. The model is trained for a specific classification task and cannot be generalized for different tasks. If a new classification task is needed, or the one used in the training phase needs some updates (new labels are added), it could be necessary to train the whole model from scratch, including the autoencoder. In addition to this, all the previous time series that are already compressed, need to be decompressed and compressed again using the new encoder to be classified by the new classification model. This process, in some contexts, could be infeasible in terms of computational costs.

Using a feed forward autoencoder has a restriction: new time series that are passed in input to the model must have the same length as the ones given in input during training. A common approach is to split the time series in different time windows of fixed size. Doing that, the classifier can classify only the single time windows independently, and this can fail to classify correctly the whole time series.

Other works use learning vector quantization to combine compression and classification, as shown in [5]. In [21] authors propose the compression and classification of string using the Universal Similarity Metric [45]. Another approach is shown in [81] and consists of features extraction aimed at achieving good results for the classification task.

## 6.3 The proposed method

This section presents the Recurrent Autoencoder for Time-series Compression and Classification (RAT-CC) model, shown in Figure 6.3, that consists of a many-to-many recurrent autoencoder [43] with two loss functions that are combined.

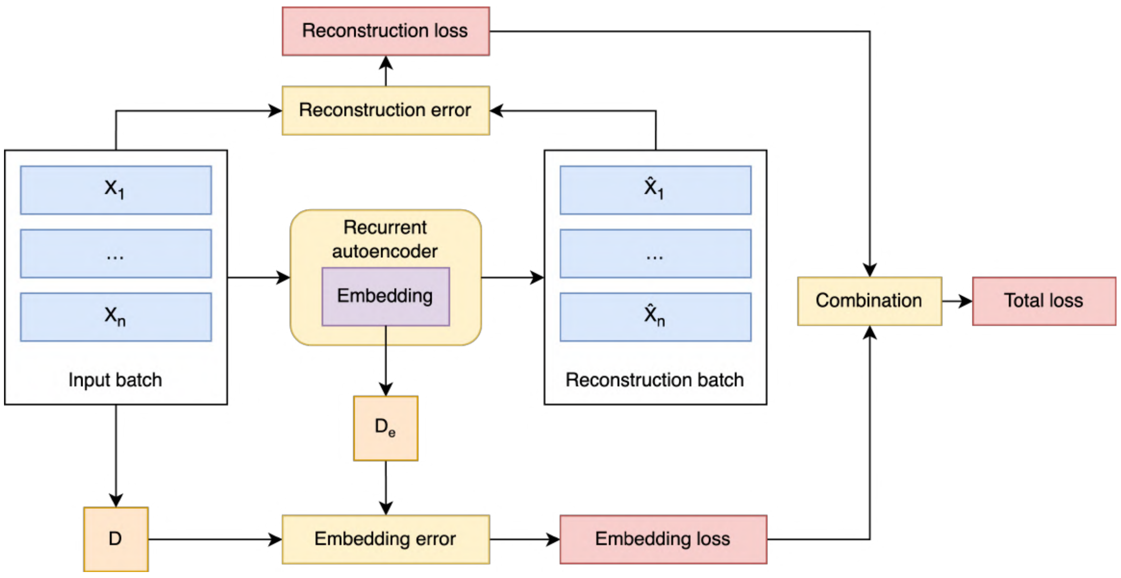


Figure 6.3: Schema of the RAT-CC model

The reconstruction loss function is computed at the last layer and measures the reconstruction error, as classic recurrent autoencoders do. The embedding loss is applied at the embedding layer. Unlike the mentioned work, the embedded representation is not connected to a classifier, so, this loss does not measure the classification accuracy for a specific task. The embedding loss is added to force the distances between time series in the original space and in the embedded space to be similar.

To do so, the embedding loss, during the training phase, receives a batch input composed of original time series and a batch input of their corresponding embeddings. The distance matrix is measured for both the original time series batch and their embeddings, respectively $D$, and $D_E$. The two matrices are normalized separately so that the maximum distance in the matrix is 1, as show in Equation 6.2. The embedding error is computed between the two matrices.

$$D_{norm} = \frac{D - \min(D)}{\max(D) - \min(D)} \tag{6.2}$$

Before computing the distance matrix over the original input batch, since its shape has 3 dimensions ($n \times l \times f$, where $n$ is the batch size, $l$ is the length of time series, and $f$ is the number of features of time series), it is reshaped to 2 dimensions ($n \times l \cdot f$). To do so, the third dimension of the features are concatenated, as in the following example: $Batch_{original} = [[[1, 2], [3, 4], [5, 6]], \ldots]$, $Batch_{reshaped} = [[1, 2, 3, 4, 5, 6], \ldots]$.

The overall loss is the sum of the two losses, as shown in Equation 6.3.

$$loss = \underbrace{\lambda \cdot MAPE(X)}_{\text{Reconstruction loss}} + \underbrace{(1 - \lambda) \cdot MSE(e(X))}_{\text{Embedding loss}}, \lambda \in [0, 1] \tag{6.3}$$

Where the value of $\lambda$ can be changed to give more weight to the reconstruction accuracy or to the embedding similarity accuracy, the Mean Absolute Percentage Error (MAPE) is the reconstruction loss, and the Mean Squared Error (MSE) is the embedding loss. Both the MAPE and MSE losses are defined below respectively in Equation 6.4 and Equation 6.5.

$$MAPE = \frac{\sum_{i=1}^{l} \sum_{j=1}^{f} |x_{i,j} - \hat{x}_{i,j}|}{l} \tag{6.4}$$

Where $X = [x_1, \ldots, x_l]$ is the original time series, $\hat{X} = [\hat{x_1}, \ldots, \hat{x_l}]$ is the reconstructed time series starting from its embedding, and $l$ is the length of time series.

$$MSE = \frac{1}{k} \sum_{i=1}^{k} (D_i - \hat{D}_i)^2 \tag{6.5}$$

Where $k$ is the number of time series inside a batch, $D_i \in R^{k \times k}$ is the distance matrix computed between the original time series $\hat{D} \in R^{k \times k}$ is the distance matrix computed between the time series' embeddings.

The architecture of the network is shown in Figure 6.4. It is composed by an encoder and a decoder: the encoder takes as input a sequence and returns a vector of size $e$ (embedding). The encoder takes as input the embedding replicated $l$ times, as the length of the original time series, and returns a sequence, which is the reconstructed time series. The complexity of the model can be adjusted by setting the $k$ and $d$ parameters which are, respectively, the number of stacked LSTM layers, and the length of output vectors at each time step returned by LSTM layers. The decoder and encoder architectures for recurrent neural networks are shown separately in [30].

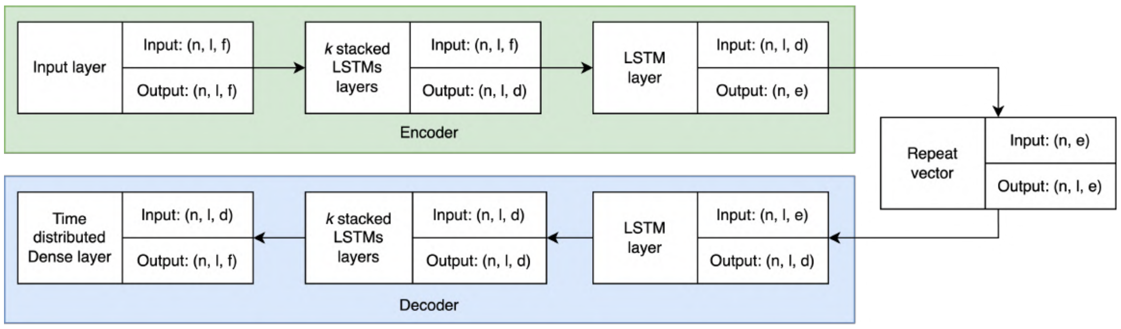


Figure 6.4: Schema of the architecture of the network

## 6.4 Experiments

To evaluate the performance of RAT-CC, 4 time series datasets from different domains are selected. In the literature, there are no extensive studies on classification of compressed time series, for this reason there are not reference datasets to compare results. The selected datasets are labelled, so it is possible to train a classification algorithm and evaluate its accuracy. The tests include both supervised and unsupervised classifiers: the Support Vector Machine (SVM) [15], Spectral Clustering [54], feed forward neural network [25]. In the following subsections, the datasets and results are presented.

### 6.4.1 Datasets

**Museums** The museum dataset is composed by environmental measures of temperature and relative humidity taken by sensors placed inside boxes containing artworks. The sensors registered new data points every 15 minutes from 2020/05/07 to 2021/07/05. Measurements are then split by day, resulting in 4050 time series of 96 points and 2 features each (temperature and relative humidity). The labels assigned to each time series indicate if the environmental conditions of one day is dangerous or not for the artworks inside the aforementioned box (i.e., the mean of temperature is not between 15 °C and 25 °C, and the mean of humidity is not between 40% and 60%). The dataset is collected by partners in the APACHE project, funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 814496[1]. For privacy reasons, the data is not publicly available.

**Activity recognition** The activity recognition dataset collects the measurements of sensors wore by people to track their movements while performing some simple activities [55]. The sensors are placed on people's chest and ankles and the activities that were monitored are: bending (two different types), cycling, lying, sitting, standing, and walking (7 classes). The dataset is composed by 88 time

[1]https://www.apacheproject.eu

series composed by 479 points, and 3 features (mean value for 250ms interval of chest, left ankle, and right ankle sensor measurements).

**Handwritten characters** Differently to other handwritten character recognition datasets, this dataset is not composed by images, but multivariate time series representing the movement recorded by a WACOM tablet (x, y, and pressure axis) [85]. It is composed by 2858 samples, 20 classes, and 205 points for each sample. The set of characters is not completed, because the ones that require to lift the pen from the tablet are excluded. The classification task consists of recognizing the written characters.

**Chipless tags** The dataset consists of measurements of 4 tags with symmetrical circular ring resonators (CRRs) milled out of FR-4 laminate that resonate within the 2-3.5 GHz band. The measurements are taken with background subtraction, and contain the magnitude values (in dB) of the S11 of the Tx/Rx antenna when irradiating the chipless tags, with a resolution of 1601 univariate samples per measurement. Measurements are taken varying the distance between the tag and the antenna (range 50-160 cm), the tag alignment with the antenna (axial alignment and inclination) and the presence of clutter-generating elements. The number of measurements for this dataset is 5600 [24].

### 6.4.2 Evaluation metrics

The evaluation of the performances of RAT-CC has to include two metrics: one for the evaluation of the reconstruction accuracy and one for the evaluation of the accuracy of the classifiers trained on the compressed representation.

For the evaluation of the classifiers, two metrics are used, depending on the classifier. If the classifier is supervised, the accuracy is evaluated as the ratio between the number of time series that are classified correctly and the total number of time series.

Since clustering models assign labels randomly, metrics used to evaluate the accuracy have to evaluate if data is separated similarly between the predicted labels and the actual ones. To do so, the Normalized Mutual Information (NMI) defined in Equation 6.6 [77] is used.

$$NMI(Y,\hat{Y}) = \frac{MI(Y,\hat{Y})}{mean(H(Y),H(\hat{Y}))} \tag{6.6}$$

Where:

- $MI(Y,\hat{Y}) = \sum_{i=1}^{N}\sum_{j=1}^{N} \frac{|Y_i \cap \hat{Y}_j|}{N} \log\left(\frac{N|Y_i \cap \hat{Y}_j|}{|Y_i||\hat{Y}_j|}\right)$
- $H(Y) = -\sum_{i=1}^{N} \frac{|Y_i|}{N} \log\left(\frac{|Y_i|}{N}\right)$

### 6.4.3 Setup

The classifiers used for the experiments are taken from the scikit-learn library[2]. To choose the best parameters for the classifiers, all the possible attributes are tested.

[2]https://scikit-learn.org/stable/index.html

For the SVM classifier, tests includes:

- **Kernel**: linear, polynomial, Gaussian, and sigmoid
- **Gamma**: scale, auto
- **Degree**: 2, 3, 4, 5, 6, 7, 8, 9, 10 (used only for the polynomial kernel)

The multilayer perceptron (MLP) has 100 neurons in the hidden layer and is trained with 1000 epochs.

The hyperparameters tested for the spectral clustering includes:

- **Affinity**: nearest neighbors (construct the affinity matrix by computing a graph of nearest neighbors) and Gaussian (construct the affinity matrix using a radial basis function kernel)
- **Label assignment**: k-means, discretize, and cluster-qr [17]

Since the time series in the training set have slightly variable lengths for the activity dataset, and the Keras library requires that time series in the same batch have the same length, to train the model time series are cut to the minimum length. This is done only for training, while for testing the model time series are used with their original length.

To evaluate the accuracy of the model, the original datasets are divided into training, testing, and validation datasets. The splits are set to 70% and 30 % of the original dataset for training dataset and testing dataset, and 20% of the training dataset is used for validation. The validation set is used to monitoring the training curve and force early stopping if the loss begins to grow for this set and for finding the best hyperparameters.

In the original dataset, time series are sorted by their labels. This implies that the division between training, testing, and validation datasets makes the classes unbalanced as the testing datasets can be populated by elements assigned to classes that are not present in the training dataset. For this reason, the datasets are shuffled before being divided.

The same dataset division is applied to train the supervised classifiers (SVM and multilayer perceptron). The spectral clustering instead received as input received the whole dataset since, being an unsupervised model, it has not a training phase.

### 6.4.4 Results

This section shows the results obtained for the aforementioned datasets. For all of them, the value of the compression ratio $\rho$ and $\lambda$ are fixed: $\rho = 0.8$ and $\lambda = 0.5$, having $\rho$ defined in Equation 6.7 and $\lambda$ defined in Equation 6.3.

$$\rho = 1 - \frac{e}{l \cdot f} \tag{6.7}$$

Where $e$ is the size of the embedding and $l, f$ are respectively the length of a time series and the number of features.

Other hyperparameters used to build the network are the size of each hidden layer $d$ and the number of staked LSTM layers $k$. The values used in RAT-CC are shown in Table 6.1.

Table 6.1: Input parameters used for training the different datasets

| | Museums | Activities | Characters | Tags |
|---|---|---|---|---|
| Hidden layer size | 20 | 55 | 35 | 85 |
| Hidden layers | 3 | 4 | 2 | 4 |

The reconstruction error is computed both with RAT-CC and the basic autoencoder to compare if some notable differences emerge between the two models. To evaluate the reconstruction error, the MSE measures is used. Results are shown in Table 6.2.

Table 6.2: MSE computed on the reconstructed time series over the original ones

| | Museums | Activities | Characters | Tags |
|---|---|---|---|---|
| Basic autoencoder | 3.2 E-4 | 5.6 E-3 | 1.0 E-2 | 0.45 |
| RAT-CC | **1.7 E-4** | **5.2 E-3** | **1.0 E-2** | **0.21** |

Table 6.3 shows the classification error for the different datasets. The evaluated classifiers are: for the supervised models, used SVM [58] and a multilayer perceptron classifier [63], for the unsupervised classifiers, Spectral clustering [54].

Table 6.3: Classification accuracy comparison expressed in percentage(%)

| | Museums | Activities | Characters | Tags |
|---|---|---|---|---|
| SVM | | | | |
| Original data | 98.1 | 42.5 | **96.7** | **99.2** |
| Basic autoencoder | 95.6 | 21.3 | 92.4 | 59.9 |
| RAT-CC | **99.4** | **58.1** | 96.6 | 97.1 |
| Multilayer perceptron | | | | |
| Original data | 82.6 | 50.2 | **98.3** | 82.5 |
| Basic autoencoder | 62.5 | 42.5 | 66.4 | 70.1 |
| RAT-CC | **94.4** | **62.3** | 78.4 | **89.9** |
| Spectral clustering | | | | |
| Original data | 62.9 | **63.8** | 63.3 | 72.5 |
| Basic autoencoder | 13.8 | 23.8 | 47.6 | 59.8 |
| RAT-CC | **77.3** | 62.7 | **89.2** | **82.7** |

Also, the execution time taken by RAT-CC to compress and reconstruct time series is evaluated. The average speed is $5.78e^{-7} \cdot N \cdot F$ where $N$ is the number of points and $F$ number of features for each point. The code was executed on a MacBook Pro with M1 Pro chip with 9-core CPU and 14-core GPU, 16-core neural engine, and 16Gb RAM.

### 6.4.5 Hyperparameters evaluation

This section illustrates how hyperparameters can influence the reconstruction error and classification accuracy. To do so, some experiments are ran on the museums dataset, testing different values for $\lambda$ and for the compression ratio $\rho$. In these experiments, the tested values are: $\lambda \in \{0.25, 0.5, 0.75\}$ having $\rho = 0.2$, and $\rho \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$, having $\lambda = 0.5$.

The results plotted in the following figures show a correlation between these values and the performance of the model. Having a higher value of $\lambda$ improves the reconstruction of the original time series (Figure 6.5) while sacrificing the classification accuracy that drops significantly (Figure 6.6). Looking at $\rho$, the reconstruction error is reduced by increasing the size of the embedding (Figure 6.5). Having an embedding with lower dimensionality is preferable to improve classification accuracy (Figure 6.6). The same data are data in Figure 6.7 with different representations.

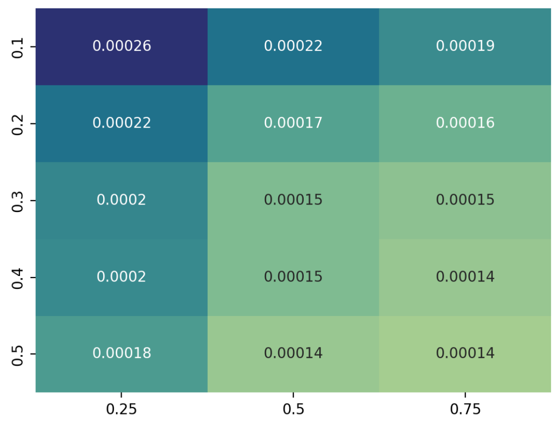


Figure 6.5: MSE computed for different combinations of $\lambda$ and $\rho$ for the museums dataset

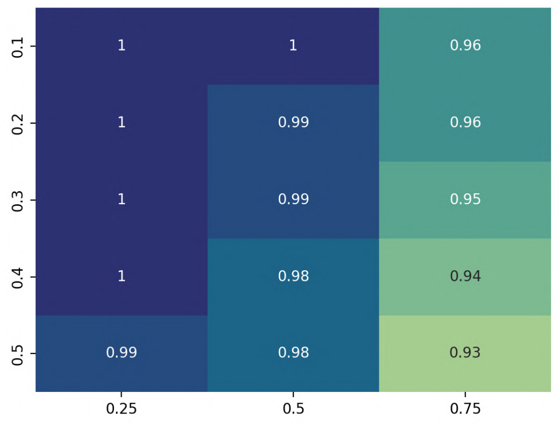


Figure 6.6: Classification accuracy of SVM classifier computed for different combinations of $\lambda$ and $\rho$ for the museums dataset

To test the consistency of the RAT-CC model, 10 different seeds are tried to shuffle the datasets before dividing them into training, testing, and validation. The mean and standard deviation of the results obtained for the SVM and MLP classifiers are shown in Table 6.4.

Table 6.4: Mean of accuracy and standard deviation for different seeds for the museums

| | SVM | MLP |
|---|---|---|
| mean | 0.989 | 0.957 |
| std | 0.010 | 0.017 |

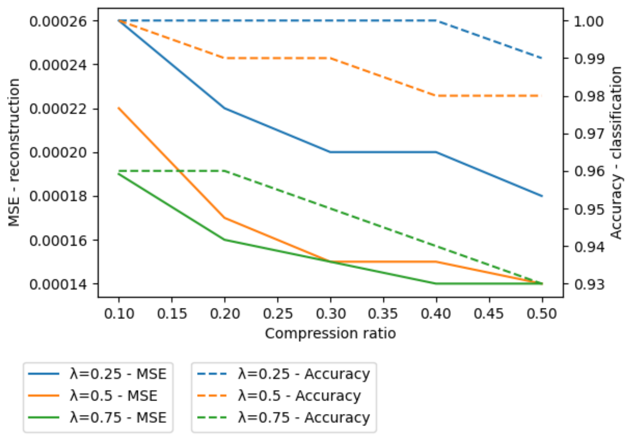
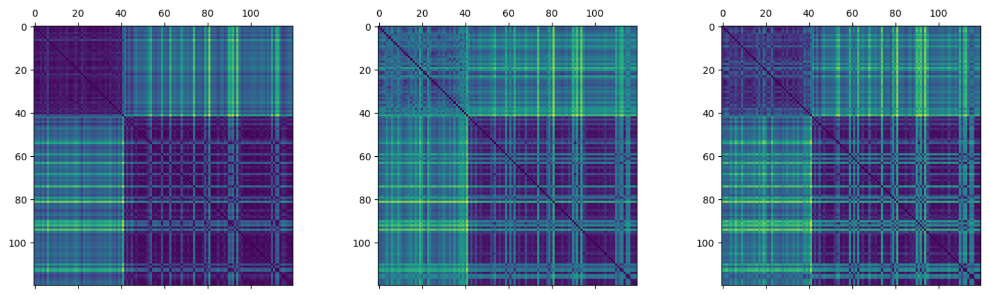


Figure 6.7: Plot of the reconstruction MSE error and classification accuracy for SVM classifier for different combinations of $\lambda$ and $\rho$ for the museums dataset

### 6.4.6 Visual evaluation

To evaluate the benefits of the loss applied to the embedding, the distance matrices for the museums dataset computed on the original data (a), in their embedding generated by the basic autoencoder (b), and their embeddings generated with the RAT-CC model (c) are plotted in Figure 6.8. These plots show a net separation of the two classes for the distance matrix computed on the original data and their embedding computed with RAT-CC, while the separation is weaker for the embedding computed with the basic autoencoder.



a. Original data  b. Embedding of basic autoencoder  c. Embedding of RAT-CC

Figure 6.8: Distance matrices for the museums datasets

Similarly, if the dimensionality of the data is reduced and plotted on a plane using the t-SNE algorithm [83], it is possible to see a weaker separation of the two classes for the embedding computed with the basic autoencoder, as shown in Figure 6.9

To conclude, Figure 6.10 shows the result of the decoder for the museums' dataset. The result obtained with the RAT-CC model is closer to the original data.

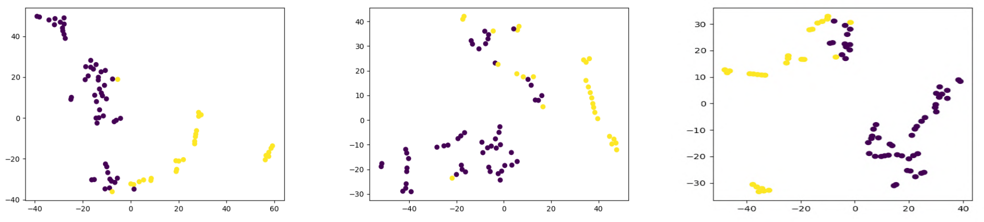

a. Original data    b. Embedding of basic autoencoder    c. Embedding of RAT-CC

Figure 6.9: Plot of the museums dataset reduced in 2 dimensions with the t-SNE algorithm and colored by class

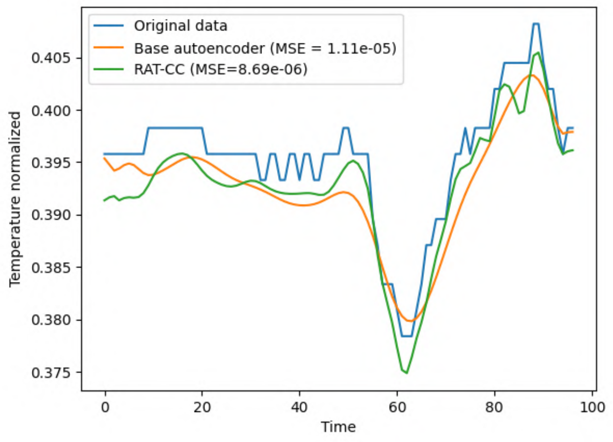


Figure 6.10: Plot of the temperature features of the museums' dataset after reconstruction

## 6.5 Conclusions

This section presents a new model (RAT-CC) based on recurrent neural networks to compress time series in such a way that the compressed representation can be used as input of a general-task classifier. Although in the literature some methods have been proposed, they work well only for supervised classifiers and for the specific task for which they are trained.

RAT-CC is tested with 4 different datasets taken from very different domains and compared it with classic autoencoder trained without custom loss applied to the embedding representation. The results reveal an incrementation of the performances in the classification task, without losing accuracy for the reconstruction of the original time series.

RAT-CC is implemented with Tensorflow [1] and the source code is available on GitHub at: `https://github.com/ChJ4m3s/TS_compression_and_classification`.

# Chapter 7

# Conclusions

This chapter summarizes the achievements obtained by the works presented in this thesis.

Starting with the APACHE project, its main goal was to develop and implement a framework to help museums to preserve cultural objects. To do so, the software had to deliver all the necessary features as:

- giving general advices about materials and agents of deteriorations;
- monitoring environmental conditions;
- firing alarms if threats are found;
- suggesting the most suitable preventive measures for a specific container.

Moreover, the user interface had to be simple, and all the connections to work seamlessly.

To achieve these goals, the framework that is developed is composed by two main parts: the preventive measures open repository and the APACHE DSS. The preventive measures open repository stores the preventive measures collected from the preventive measures producers in the APACHE projects, the user group composed by partner museums, and from the literature. Being an open repository, registered users have the possibility to add comments, suggestions, and proposals.

The APACHE DSS is also divided into two parts: tier 1 and tier 2. Tier 1 is designed as a wiki that stores general advices, divided by materials and agent of deteriorations. Tier 2 requires users to characterize their collection and is connected with wireless sensors to monitor environmental data and delivers a ranked list of suggested preventive measures, based on this data.

Both the preventive measures open repository and the APACHE DSS are implemented as a reactive web app and are already available online to be used.

The APACHE DSS is then tested by the user group with real data collected during the project. All the feedback received by the user group were taken in account and implemented when useful. The monitoring and alerts system inside the APACHE DSS helped museums to find out harmful situation that weren't known.

Due to the nature of the data that have to be stored coming from wireless sensors, this thesis presented a state-of-the-art survey on compression techniques developed for time series. This survey shows the main characteristics of each technique, their limits, and a comparison among them.

In many applications, as the APACHE DSS, compressed data have to be analysed to extract more information. To do so, this thesis presents a model developed to compress time series in such a way that the compressed representation can be used for classification tasks. The model is then tested with 4 datasets taken from different domains to evaluate its performances, showing promising results.

## 7.1 Future directions

The work presented in this thesis is open to future developments. Starting with the framework designed for the APACHE DSS, a possible future extension is related to remedial conservation and restoration actions. As the framework is designed in a modular structure and since damage functions can provide information about physical damages, it could be possible, in the future, to add a repository of remedial conservation and restoration proposals and suggest appropriate actions for restoration, in the same way as for the suggestion of appropriate preventive measures.

Moreover, as the framework is designed to be easily extended without the need to modify the overall structure. For instance, the list of materials, damage functions and agents of deterioration can be updated. Considering them more in detail, also their structure can be updated to implement new features:

- **Materials**: they are now structured as a plain list. This can be updated with a tree structure to model different types of one material (e.g., papyrus is a specific type of paper). This would let specific types to easily inherit the characteristics of their super class while adding more specific details. Moreover, this would improve searchability;
- **Damage functions**: current damage functions are able to understand if current environmental conditions are optimal for a certain material. If new damage functions addressing damage detection will be available, they can be added to the framework. This requires the structure to be updated to be able to differentiate the two types of functions. The resulting index would then be used to understand the best restoration measure from a repository of restoration measures;
- **Agents of deterioration**: currently, the APACHE DSS handles only 3 over the 10 agents of deterioration defined by the Canadian Conservation Institute. If the ones implemented in the APACHE DSS cause damages in the long-term, others as fire and floods have an immediate disruptive effect. For this reason, they should be modelled such that immediate actions are suggested.

The performances of the APACHE DSS can be further improved by implementing the model proposed in Chapter 6. In this way, it is possible to optimize the space needed to store sensor's data and analyse past measurements efficiently without needing to recover the original data.

Considering the model illustrated in Chapter 6, possible developments include the possibility to consider the correlation between the features of a multivariate time series to improve compression capabilities, while preserving the reconstruction accuracy.

# Bibliography

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

[2] Mohammed Abo-Zahhad. ECG Signal Compression Using Discrete Wavelet Transform. In Juuso T. Olkkonen, editor, *Discrete Wavelet Transforms - Theory and Applications*. InTech, April 2011. ISBN 978-953-307-185-5. doi: 10.5772/16019. URL `http://www.intechopen.com/books/discrete-wavelet-transforms-theory-and-applications/ecg-signal-compression-using-discrete-wavelet-transform`.

[3] A. S. Alvarado, C. Lakshminarayan, and J. C. Principe. Time-Based Compression and Classification of Heartbeats. *IEEE Transactions on Biomedical Engineering*, 59(6):1641–1648, June 2012. ISSN 0018-9294, 1558-2531. doi: 10.1109/TBME.2012.2191407. URL `http://ieeexplore.ieee.org/document/6172220/`.

[4] Parvaneh Asghari, Amir Masoud Rahmani, and Hamid Haj Seyyed Javadi. Internet of Things applications: A systematic review. *Computer Networks*, 148:241–261, January 2019. ISSN 13891286. doi: 10.1016/j.comnet.2018.12.008. URL `https://linkinghub.elsevier.com/retrieve/pii/S1389128618305127`.

[5] J.S. Baras and S. Dey. Combined compression and classification with learning vector quantization. *IEEE Transactions on Information Theory*, 45(6):1911–1920, 1999. doi: 10.1109/18.782112.

[6] Davis Blalock, Samuel Madden, and John Guttag. Sprintz: Time Series Compression for the Internet of Things. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(3):1–23, September 2018. ISSN 24749567. doi: 10.1145/3264903. URL `http://arxiv.org/abs/1808.02515`. arXiv: 1808.02515.

[7] Lukasz Bratasz, Tim White, Susan Butts, Catherine Sease, Nathan Utrup, Richard Boardman, and Stefan Simon. Toward sustainable collections management in the Yale Peabody Museum: Risk assessment, climate management, and energy efficiency. *Bulletin of the Peabody Museum of Natural History*, 59(2):249–268, 2018. ISSN 21624135. doi: 10.3374/014.059.0206.

[8] Mirjam Brusius and Kavita Singh. *Museum storage and meaning: Tales from the crypt.* 2017. ISBN 978-1-315-15939-3. doi: 10.4324/9781315159393. Publication Title: Museum Storage and Meaning: Tales from the Crypt.

[9] Giuseppe Campobello, Antonino Segreto, Sarah Zanafi, and Salvatore Serrano. RAKE: A simple and efficient lossless compression algorithm for the Internet of Things. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 2581–2585, Kos, Greece, August 2017. IEEE. ISBN 978-0-9928626-7-1. doi: 10.23919/EUSIPCO.2017.8081677. URL `http://ieeexplore.ieee.org/document/8081677/`.

[10] C. Caple. *Conservation Skills*. Taylor and Francis, Oxford, 2012. URL https://www.perlego.com/book/1624066/conservation-skills-pdf.

[11] C. Caple. *Preventive conservation in museums*. Leicester readers in museum studies. Routledge, Oxford, January 2012. URL http://dro.dur.ac.uk/18307/.

[12] Shoue Chen, Sandrayee Brahma, Jonathon Mackay, Changyong Cao, and Bahar Aliakbarian. The role of smart packaging system in food supply chain. *Journal of Food Science*, 85 (3), 2020. ISSN 17503841. doi: 10.1111/1750-3841.15046.

[13] Giacomo Chiarot and Claudio Silvestri. Time series compression survey. *ACM Comput. Surv.*, aug 2022. ISSN 0360-0300. doi: 10.1145/3560814. URL https://doi.org/10.1145/3560814. Just Accepted.

[14] Jørgen Erik Christensen and Christos Georgios Kollias. Hygrothermal evaluation of a museum storage building based on actual measurements and simulations. In *Energy Procedia*, volume 78, pages 651–656, 2015. doi: 10.1016/j.egypro.2015.11.051. ISSN: 18766102.

[15] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3): 273–297, September 1995. ISSN 0885-6125, 1573-0565. doi: 10.1007/BF00994018. URL http://link.springer.com/10.1007/BF00994018.

[16] M. Dalai and R. Leonardi. Approximations of one-dimensional digital signals under the$l^infty$norm. *IEEE Transactions on Signal Processing*, 54(8):3111–3124, 2006. doi: 10.1109/TSP.2006.875394.

[17] Anil Damle, Victor Minden, and Lexing Ying. Simple, direct and efficient multi-way spectral clustering. *Information and Inference: A Journal of the IMA*, 8(1):181–203, 06 2018. ISSN 2049-8772. doi: 10.1093/imaiai/iay008. URL https://doi.org/10.1093/imaiai/iay008.

[18] Matteo Danieletto, Nicola Bui, and Michele Zorzi. Improving Internet of Things communications through compression and classification. In *2012 IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 284–289, Lugano, Switzerland, March 2012. IEEE. ISBN 978-1-4673-0907-3 978-1-4673-0905-9 978-1-4673-0906-6. doi: 10.1109/PerComW.2012.6197496. URL http://ieeexplore.ieee.org/document/6197496/.

[19] Frank Eichinger, Pavel Efros, Stamatis Karnouskos, and Klemens Böhm. A time-series compression technique and its application to the smart grid. *The VLDB Journal*, 24(2): 193–218, April 2015. ISSN 1066-8888, 0949-877X. doi: 10.1007/s00778-014-0368-8. URL http://link.springer.com/10.1007/s00778-014-0368-8.

[20] Kristian Fabbri and Anna Bonora. Two new indices for preventive conservation of the cultural heritage: Predicted risk of damage and heritage microclimate risk. *Journal of Cultural Heritage*, 47:208–217, January 2021. ISSN 12962074. doi: 10.1016/j.culher.2020.09.006. URL https://linkinghub.elsevier.com/retrieve/pii/S1296207420304453.

[21] Paolo Ferragina, Raffaele Giancarlo, Valentina Greco, Giovanni Manzini, and Gabriel Valiente. Compression-based classification of biological sequences and structures via the Universal Similarity Metric: experimental assessment. *BMC Bioinformatics*, 8(1):252, July 2007. ISSN 1471-2105. doi: 10.1186/1471-2105-8-252. URL https://doi.org/10.1186/1471-2105-8-252.

[22] José Figueira, Salvatore Greco, and Matthias Ehrgott. Multiple criteria decision analysis, state of the art surveys. *Multiple Criteria Decision Analysis: State of the Art Surveys*, 78, 01 2005. doi: 10.1007/b100605.

[23] Eugene Fink and Harith Suman Gandhi. Compression of time series by extracting major extrema. *Journal of Experimental & Theoretical Artificial Intelligence*, 23(2):255–270, June 2011. ISSN 0952-813X, 1362-3079. doi: 10.1080/0952813X.2010.505800. URL http://www.tandfonline.com/doi/abs/10.1080/0952813X.2010.505800.

[24] Jafait Junior Fodop Sokoudjou. Chipless RFID Tag Implementation and Machine Learning Workflow for Robust Identification. preprint, May 2022. URL https://www.techrxiv.org/articles/preprint/Chipless_RFID_Tag_Implementation_and_Machine_Learning_Workflow_for_Robust_Identification/19906756/1.

[25] Raphael Féraud and Fabrice Clérot. A methodology to explain neural network classification. *Neural Networks*, 15(2):237–246, 2002. ISSN 0893-6080. doi: https://doi.org/10.1016/S0893-6080(01)00127-7. URL https://www.sciencedirect.com/science/article/pii/S0893608001001277.

[26] Felix A. Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to Forget: Continual Prediction with LSTM. *Neural Computation*, 12(10):2451–2471, October 2000. ISSN 0899-7667, 1530-888X. doi: 10.1162/089976600300015015. URL https://direct.mit.edu/neco/article/12/10/2451-2471/6415.

[27] Anupriya Gogna, Angshul Majumdar, and Rabab Ward. Semi-supervised Stacked Label Consistent Autoencoder for Reconstruction and Analysis of Biomedical Signals. *IEEE Transactions on Biomedical Engineering*, 64(9):2196–2205, September 2017. ISSN 0018-9294, 1558-2531. doi: 10.1109/TBME.2016.2631620. URL http://ieeexplore.ieee.org/document/7752836/.

[28] Rhys Goldstein, Michael Glueck, and Azam Khan. REAL-TIME COMPRESSION OF TIME SERIES BUILDING PERFORMANCE DATA. page 8.

[29] I. J. Good. Introduction to Cooley and Tukey (1965) An Algorithm for the Machine Calculation of Complex Fourier Series. In Samuel Kotz and Norman L. Johnson, editors, *Breakthroughs in Statistics*, pages 201–216. Springer New York, New York, NY, 1997. ISBN 978-0-387-94989-5 978-1-4612-0667-5. doi: 10.1007/978-1-4612-0667-5_9. URL http://link.springer.com/10.1007/978-1-4612-0667-5_9. Series Title: Springer Series in Statistics.

[30] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[31] Mohit Goyal, Kedar Tatwawadi, Shubham Chandak, and Idoia Ochoa. DZip: improved general-purpose loss less compression based on novel neural network modeling. In *2021 Data Compression Conference (DCC)*, pages 153–162, Snowbird, UT, USA, March 2021. IEEE. ISBN 978-1-66540-333-7. doi: 10.1109/DCC50243.2021.00023. URL https://ieeexplore.ieee.org/document/9418692/.

[32] S M Hardi, B Angga, M S Lydia, I Jaya, and J T Tarigan. Comparative Analysis Run-Length Encoding Algorithm and Fibonacci Code Algorithm on Image Compression. *Journal of Physics: Conference Series*, 1235:012107, June 2019. ISSN 1742-6588, 1742-6596. doi: 10.1088/1742-6596/1235/1/012107. URL https://iopscience.iop.org/article/10.1088/1742-6596/1235/1/012107.

[33] S. Edward III Hawkins and Edward Hugo Darlington. Algorithm for Compressing Time-Series Data. June 2012. URL https://ntrs.nasa.gov/search.jsp?R=20120010460.

[34] Jetmir Haxhibeqiri, Eli De Poorter, Ingrid Moerman, and Jeroen Hoebeke. A survey of lorawan for iot: From technology to application. *Sensors*, 18(11), 2018. ISSN 1424-8220. doi: 10.3390/s18113995. URL https://www.mdpi.com/1424-8220/18/11/3995.

[35] Daniel Hsu. Time Series Compression Based on Adaptive Piecewise Recurrent Autoencoder. *arXiv:1707.07961 [cs]*, August 2017. URL http://arxiv.org/abs/1707.07961. arXiv: 1707.07961.

[36] Yuh-Ming Huang and Ja-Ling Wu. A refined fast 2-d discrete cosine transform algorithm. *IEEE Transactions on Signal Processing*, 47(3):904–907, 1999. doi: 10.1109/78.747801.

[37] Hyejung Kim, R.F. Yazicioglu, P. Merken, C. Van Hoof, and Hoi-Jun Yoo. ECG Signal Compression and Classification Algorithm With Quad Level Vector for ECG Holter System. *IEEE Transactions on Information Technology in Biomedicine*, 14(1):93–100, January 2010. ISSN 1089-7771. doi: 10.1109/TITB.2009.2031638. URL `http://ieeexplore.ieee.org/document/5256175/`.

[38] T. Inamura, H. Tanie, and Y. Nakamura. Keyframe compression and decompression for time series data based on the continuous hidden Markov model. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, volume 2, pages 1487–1492, Las Vegas, NV, USA, 2003. IEEE. ISBN 978-0-7803-7860-5. doi: 10.1109/IROS.2003.1248854. URL `http://ieeexplore.ieee.org/document/1248854/`.

[39] Søren Kejser Jensen, Torben Bach Pedersen, and Christian Thomsen. Time series management systems: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 29(11): 2581–2600, 2017. doi: 10.1109/TKDE.2017.2740932.

[40] Samsul Ariffin Abdul Karim, Mohd Hafizi Kamarudin, Bakri Abdul Karim, Mohammad Khatim Hasan, and Jumat Sulaiman. Wavelet transform and fast fourier transform for signal compression: A comparative study. In *2011 International Conference on Electronic Devices, Systems and Applications (ICEDSA)*, pages 280–285, 2011. doi: 10.1109/ICEDSA.2011.5959031.

[41] Eamonn Keogh, Kaushik Chakrabarti, Sharad Mehrotra, and Michael Pazzani. Locally Adaptive Dimensionality Reduction for Indexing Large Time Series Databases. page 12.

[42] Abdelouahab Khelifati, Mourad Khayati, and Philippe Cudre-Mauroux. CORAD: Correlation-Aware Compression of Massive Time Series using Sparse Dictionary Coding. page 10.

[43] Tung Kieu, Bin Yang, Chenjuan Guo, and Christian S. Jensen. Outlier Detection for Time Series with Recurrent Autoencoder Ensembles. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 2725–2732, Macao, China, August 2019. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-0-9992411-4-1. doi: 10.24963/ijcai.2019/378. URL `https://www.ijcai.org/proceedings/2019/378`.

[44] I. Lazaridis and S. Mehrotra. Capturing sensor-generated time series with quality guarantees. In *Proceedings 19th International Conference on Data Engineering (Cat. No.03CH37405)*, pages 429–440, 2003. doi: 10.1109/ICDE.2003.1260811.

[45] Ming Li, Xin Chen, Xin Li, Bin Ma, and P.M.B. Vitanyi. The similarity metric. *IEEE Transactions on Information Theory*, 50(12):3250–3264, 2004. doi: 10.1109/TIT.2004.838101.

[46] Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Bill Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, DMKD '03, page 2–11, New York, NY, USA, 2003. Association for Computing Machinery. ISBN 9781450374224. doi: 10.1145/882082.882086. URL `https://doi.org/10.1145/882082.882086`.

[47] Xingxing Lv and Shimeng Shen. On Chebyshev polynomials and their applications. *Advances in Difference Equations*, 2017(1):343, December 2017. ISSN 1687-1847. doi: 10.1186/s13662-017-1387-8. URL `http://advancesindifferenceequations.springeropen.com/articles/10.1186/s13662-017-1387-8`.

[48] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *J. Mach. Learn. Res.*, 11:19–60, mar 2010. ISSN 1532-4435.

[49] S.G. Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993. doi: 10.1109/78.258082.

[50] Alice Marascu, Pascal Pompey, Eric Bouillet, Michael Wurst, Olivier Verscheure, Martin Grund, and Philippe Cudre-Mauroux. TRISTAN: Real-time analytics on massive time series using sparse dictionary compression. In *2014 IEEE International Conference on Big Data (Big Data)*, pages 291–300, Washington, DC, USA, October 2014. IEEE. ISBN 978-1-4799-5666-1. doi: 10.1109/BigData.2014.7004244. URL `http://ieeexplore.ieee.org/document/7004244/`.

[51] Maryanne McCubbin, Alice Cannon, Caroline Carter, Dermot Henry, Helen Privett, Nancy Ladas, Dean Leggett, Ruth Leveson, Melanie Raberts, Laura Stedman, and Robert Waller. Improving risk assessment methods in a complex setting: Museum Victoria's collection risk assessment. *ICOM-CC 17th triennial conference Melbourne 15-19 September 2014: preprints*, 2014. ISBN: 978-92-9012-410-8.

[52] Jan-Gerd Meß, Robert Schmidt, Goerschwin Fey, and Frank Dannemann. On the compression of spacecraft housekeeping data using discrete cosine transforms. In *2016 International Workshop on Tracking, Telemetry and Command Systems for Space Applications (TTC)*, pages 1–8, 2016.

[53] Hussein Sh. Mogahed and Alexey G. Yakunin. Development of a Lossless Data Compression Algorithm for Multichannel Environmental Monitoring Systems. In *2018 XIV International Scientific-Technical Conference on Actual Problems of Electronics Instrument Engineering (APEIE)*, pages 483–486, Novosibirsk, October 2018. IEEE. ISBN 978-1-5386-7054-5. doi: 10.1109/APEIE.2018.8546121. URL `https://ieeexplore.ieee.org/document/8546121/`.

[54] Mariá C.V. Nascimento and André C.P.L.F. de Carvalho. Spectral methods for graph clustering – a survey. *European Journal of Operational Research*, 211(2):221–231, 2011. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2010.08.012. URL `https://www.sciencedirect.com/science/article/pii/S0377221710005497`.

[55] Filippo Palumbo, Claudio Gallicchio, Rita Pucci, and Alessio Micheli. Human activity recognition using multisensor data fusion based on reservoir computing. *J. Ambient Intell. Smart Environ.*, 8(2):87–107, jan 2016. ISSN 1876-1364. doi: 10.3233/AIS-160372. URL `https://doi.org/10.3233/AIS-160372`.

[56] Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. How to Construct Deep Recurrent Neural Networks, April 2014. URL `http://arxiv.org/abs/1312.6026`. arXiv:1312.6026 [cs, stat].

[57] Gianluca Pastorelli, Costanza Cucci, Oihana García, Giovanna Piantanida, Abdelrazek Elnaggar, May Cassar, and Matija Strli. Environmentally induced colour change during natural degradation of selected polymers. *Polymer Degradation and Stability*, 107:198–209, 2014.

[58] John Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Adv. Large Margin Classif.*, 10, 06 2000.

[59] James Pope, Antonis Vafeas, Atis Elsts, George Oikonomou, Robert Piechocki, and Ian Craddock. An accelerometer lossless compression algorithm and energy analysis for IoT devices. In *2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pages 396–401, Barcelona, April 2018. IEEE. ISBN 978-1-5386-1154-8. doi: 10.1109/WCNCW.2018.8368985. URL `https://ieeexplore.ieee.org/document/8368985/`.

[60] Attila Reiss and Didier Stricker. Towards global aerobic activity monitoring. In *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '11, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450307727. doi: 10.1145/2141622.2141637. URL `https://doi.org/10.1145/2141622.2141637`.

[61] Sean Rhea, Eric Wang, Edmund Wong, Ethan Atkins, and Nat Storer. LittleTable: A Time-Series Database and Its Uses. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 125–138, Chicago Illinois USA, May 2017. ACM. ISBN 978-1-4503-4197-4. doi: 10.1145/3035918.3056102. URL https://dl.acm.org/doi/10.1145/3035918.3056102.

[62] Jussi Ronkainen and Antti Iivari. Designing a Data Management Pipeline for Pervasive Sensor Communication Systems. *Procedia Computer Science*, 56:183–188, 2015. ISSN 18770509. doi: 10.1016/j.procs.2015.07.193. URL https://linkinghub.elsevier.com/retrieve/pii/S1877050915016749.

[63] D.W. Ruck, S.K. Rogers, M. Kabrisky, M.E. Oxley, and B.W. Suter. The multilayer perceptron as an approximation to a Bayes optimal discriminant function. *IEEE Transactions on Neural Networks*, 1(4):296–298, December 1990. ISSN 10459227. doi: 10.1109/72.80266. URL http://ieeexplore.ieee.org/document/80266/.

[64] Morten Ryhl-Svendsen, Lars Aasbjerg Jensen, and Poul Klenz Larsen. Air quality in low-ventilated museum storage buildings. *ICOM-CC 17th triennial conference Melbourne 15-19 September 2014: preprints*, (2007), 2014. ISBN: 978-92-9012-410-8.

[65] D. Salomon. *Data compression: the complete reference*. Springer, London, 4th ed edition, 2007. ISBN 978-1-84628-602-5.

[66] Khalid Sayood. *Introduction to data compression*. Morgan Kaufmann series in multimedia information and systems. Elsevier, Amsterdam ; Boston, 3rd ed edition, 2006. ISBN 978-0-12-620862-7.

[67] Dirk Schaefer and Wai M. Cheung. Smart Packaging: Opportunities and Challenges. In *Procedia CIRP*, volume 72, pages 1022–1027, 2018. doi: 10.1016/j.procir.2018.03.240. ISSN: 22128271.

[68] Eva Schito and Daniele Testi. Integrated maps of risk assessment and minimization of multiple risks for artworks in museum environments based on microclimate control. *Building and Environment*, 123:585–600, 2017. ISSN 0360-1323. doi: https://doi.org/10.1016/j.buildenv.2017.07.039. URL https://www.sciencedirect.com/science/article/pii/S0360132317303384.

[69] Eva Schito and Daniele Testi. Integrated maps of risk assessment and minimization of multiple risks for artworks in museum environments based on microclimate control. *Building and Environment*, 123:585–600, 2017. ISSN 03601323. doi: 10.1016/j.buildenv.2017.07.039.

[70] Raimund Seidel. Small-dimensional linear programming and convex hulls made easy. *Discrete & Computational Geometry*, 6(3):423–434, September 1991. ISSN 1432-0444. doi: 10.1007/BF02574699. URL https://doi.org/10.1007/BF02574699.

[71] Alex Sherstinsky. Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. *Physica D: Nonlinear Phenomena*, 404:132306, March 2020. ISSN 01672789. doi: 10.1016/j.physd.2019.132306. URL https://linkinghub.elsevier.com/retrieve/pii/S0167278919305974.

[72] Jin Shieh and Eamonn Keogh. iSAX: disk-aware mining and indexing of massive time series datasets. *Data Mining and Knowledge Discovery*, 19(1):24–57, August 2009. ISSN 1384-5810, 1573-756X. doi: 10.1007/s10618-009-0125-6. URL http://link.springer.com/10.1007/s10618-009-0125-6.

[73] Muhammad Shoaib, Stephan Bosch, Ozlem Durmaz Incel, Hans Scholten, and Paul J. M. Havinga. Fusion of smartphone motion sensors for physical activity recognition. *Sensors*, 14(6):10146–10176, 2014. ISSN 1424-8220. doi: 10.3390/s140610146. URL https://www.mdpi.com/1424-8220/14/6/10146.

[74] K. Smith, D. Depolo, J. Torrisi, N. Edwards, G. Biasi, and D. Slater. The 2008 Mw 6.0 Wells, Nevada Earthquake Sequence. In *AGU Fall Meeting Abstracts*, volume 2008, pages S51B–1741, December 2008.

[75] Julien Spiegel, Patrice Wira, and Gilles Hermann. A Comparative Experimental Study of Lossless Compression Algorithms for Enhancing Energy Efficiency in Smart Meters. In *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*, pages 447–452, Porto, July 2018. IEEE. ISBN 978-1-5386-4829-2. doi: 10.1109/INDIN.2018.8471921. URL https://ieeexplore.ieee.org/document/8471921/.

[76] James A. Storer and Thomas G. Szymanski. Data compression via textual substitution. *Journal of the ACM (JACM)*, 29(4):928–951, October 1982. ISSN 0004-5411, 1557-735X. doi: 10.1145/322344.322346. URL http://dl.acm.org/doi/10.1145/322344.322346.

[77] Alexander Strehl and Joydeep Ghosh. Cluster Ensembles – A Knowledge Reuse Framework for Combining Multiple Partitions.

[78] Matija Strli, Carlota M. Grossi, Catherine Dillon, Nancy J. Bell, Kalliopi Fouseki, Peter Brimblecombe, Eva Menart, Kostas Ntanos, William Lindsay, David R. Thickett, Fenella G. France, and Gerrit de Bruin. Damage function for historic paper. part iii: Isochrones and demography of collections. *Heritage Science*, 3:1–11, 2015.

[79] Torsten Suel. Delta Compression Techniques. In Sherif Sakr and Albert Zomaya, editors, *Encyclopedia of Big Data Technologies*, pages 1–8. Springer International Publishing, Cham, 2018. ISBN 978-3-319-63962-8. doi: 10.1007/978-3-319-63962-8_63-1. URL http://link.springer.com/10.1007/978-3-319-63962-8_63-1.

[80] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA, 2014. MIT Press.

[81] Tzu-Wei Tseng, Kai-Jiun Yang, C.-C. Jay Kuo, and Shang-Ho Tsai. An Interpretable Compression and Classification System: Theory and Applications. *IEEE Access*, 8:143962–143974, 2020. ISSN 2169-3536. doi: 10.1109/ACCESS.2020.3014307. URL https://ieeexplore.ieee.org/document/9159554/.

[82] Tuong Ly Le and Minh-Huan Vo. Lossless Data Compression Algorithm to Save Energy in Wireless Sensor Network. page 4, 2018.

[83] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL http://jmlr.org/papers/v9/vandermaaten08a.html.

[84] Jirí Walder, Michal Krátký, and Jan Platos. Fast fibonacci encoding algorithm. volume 567, pages 72–83, 01 2010.

[85] Ben H. Williams, Marc Toussaint, and Amos J. Storkey. Extracting motion primitives from natural handwriting data. In *Proceedings of the 16th International Conference on Artificial Neural Networks - Volume Part II*, ICANN'06, page 634–643, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3540388710. doi: 10.1007/11840930_66. URL https://doi.org/10.1007/11840930_66.

[86] Marcin Wojnarski, Pawel Gora, Marcin Szczuka, Hung Son Nguyen, Joanna Swietlicka, and Demetris Zeinalipour. Ieee icdm 2010 contest: Tomtom traffic prediction for intelligent gps navigation. In *2010 IEEE International Conference on Data Mining Workshops*, pages 1372–1376, 2010. doi: 10.1109/ICDMW.2010.51.

[87] J. G. Wolff. Simplicity and Power - Some Unifying Ideas in Computing. *The Computer Journal*, 33(6):518–534, June 1990. ISSN 0010-4620, 1460-2067. doi: 10.1093/comjnl/33.6.518. URL https://academic.oup.com/comjnl/article-lookup/doi/10.1093/comjnl/33.6.518.

[88] Fangjin Yang, Eric Tschetter, Xavier Léauté, Nelson Ray, Gian Merlino, and Deep Ganguli. Druid: a real-time analytical data store. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, pages 157–168, Snowbird Utah USA, June 2014. ACM. ISBN 978-1-4503-2376-5. doi: 10.1145/2588555.2595631. URL `https://dl.acm.org/doi/10.1145/2588555.2595631`.

[89] Xia Zhao, Xiao Han, Weijun Su, and Zhen Yan. Time series prediction method based on Convolutional Autoencoder and LSTM. In *2019 Chinese Automation Congress (CAC)*, pages 5790–5793, Hangzhou, China, November 2019. IEEE. ISBN 978-1-72814-094-0. doi: 10.1109/CAC48633.2019.8996842. URL `https://ieeexplore.ieee.org/document/8996842/`.

[90] Ian Ziraldo, Kristen E Watts, Arnold Luk, Anthony F. Lagalante, and Richard C. Wolbers. The influence of temperature and humidity on swelling and surfactant migration in acrylic emulsion paint films. *Studies in Conservation*, 61:209 – 221, 2016.

[91] Michele Zorzi, Alexander Gluhak, Sebastian Lange, and Alessandro Bassi. From today's intranet of things to a future internet of things: a wireless- and mobility-related view. *IEEE Wireless Communications*, 17(6):44–51, 2010. doi: 10.1109/MWC.2010.5675777.