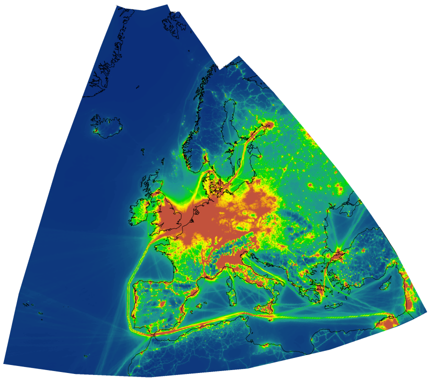


GAPS: Geo Data Portals for Air Pollution Studies

Filipe Miguel Maio Fernandes

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisor(s): Prof. Doutor João Nuno De Oliveira e Silva
Doutora Helena Cristina Serrano

Examination Committee

Chairperson: Prof. Teresa Maria Sá Ferreira Vazão Vasques
Supervisor: Prof. Doutor João Nuno De Oliveira e Silva
Member of the Committee: Prof. Doutor Bruno Emanuel Da Graça Martins

January 2021

Dedicated to my family

Declaration

I declare that this document is an original work of my own authorship and that it fulfils all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

Firstly, I am sincerely grateful to Professor João Silva for providing me with a very interesting subject for my dissertation. I would also like to thank him for his availability, advice and support.

I would like to thank the advisers Maria Alexandra Oliveira and Helena Serrano for their support and advice. They have been tireless in their availability and have always been available to help me throughout these months to clarify doubts on the subject of air pollution.

I would like to thank the eChanges (Ecology of environmental Change) research group of the cE3c (Centre for Ecology, Evolution and Environmental Changes) research centre of the University of Lisbon, that tested the developed applications.

I would like to thank my family, girlfriend and friends for their support and encouragement in completing this stage of my academic life.

Resumo

Existem muitos dados sobre poluição atmosférica ao alcance de diversos utilizadores, incluindo concentrações e deposições modeladas, e ainda observações efetuadas em estações de qualidade do ar. No entanto, existem dificuldades em integrar os dados de forma a obter a perceção de tendências espaciais e temporais a nível nacional. As dificuldades prendem-se essencialmente com as fontes de dados (muitos ficheiros, com muita informação e em formatos distintos) e com o facto do processamento destes dados ser moroso e pouco prático quando o período temporal em análise aumenta. Para além disso, o processamento de dados espácio-temporais requer a utilização de múltiplas ferramentas específicas, tal como software de sistemas de informação geográfica (ArcGIS) e estatístico (R), pouco acessíveis a utilizadores não especializados.

A solução proposta passa por desenvolver bibliotecas que recolhem e agregam diferentes tipos de dados relacionados com a poluição do ar. Para além de permitir integrar diferentes dados, as bibliotecas também são a base para se desenvolver aplicações web. As bibliotecas vão ser responsáveis por recolher dados disponibilizados pela Agência Portuguesa do Ambiente (APA) e resultados de modelação efetuada para a Europa (EMEP). As aplicações desenvolvidas estenderam as bibliotecas desenvolvidas, e permitiram o processamento de dados e a sua representação através de Dashboards, que vão dar acesso facilitado a diversos utilizadores não especializados.

As bibliotecas implementadas permitem o desenvolvimento de projetos que necessitem de aceder às mesmas fontes de dados. Os Dashboards permitem o acesso simplificado aos dados e a realização de estudos até agora fora do alcance dos investigadores.

Palavras-chave: Agregação espácio-temporal, Poluição atmosférica, Modelo EMEP, Python, Aplicações Web, Dashboard

Abstract

There is a wealth of data on air pollution within several users' reach, including modelled concentrations and depositions as well as observations from air quality stations. However, data integration to perceive spatial and temporal trends at national level, is a complex undertaking. The difficulties are mainly related to the data sources (many files, with a lot of information and in different formats). In addition, the processing of this data is time-consuming and impractical when the time period under analysis increases. Furthermore, the processing of spatial-temporal data requires the use of multiple specific tools, such as geographic information systems software (ArcGIS) and statistical (R), which are not readily accessible to non-specialised users.

The proposed solution is to develop libraries that are responsible for aggregating different types of data related with pollution. In addition to allowing the integration of different data, the libraries are also the basis for developing web applications. The libraries allow the collection of data made available by the Portuguese Environment Agency (APA) and official modelling results for Europe (EMEP). The applications developed will extend the libraries to allow data processing and representation through Dashboards. The Dashboards provide access to non-specialised users, namely about the trends of pollutants in each region.

The implemented libraries allow the development of projects that need access to the same data sources. Dashboards allow for simplified access to data and studies that have so far been beyond the reach of researchers.

Keywords: Space-time aggregation, Pollutant Concentrations, EMEP model, Python, Web Applications, Dashboard

Contents

List of Tables

List of Figures

List of Acronyms

APA *Agência Portuguesa do Ambiente*

COS *Cartografia de Uso e Ocupação do Solo*

CSUQ Computer System Usability Questionnaire

CWS Catalogue Service for the Web Service

DBMS DataBase Management Systems

DGT *Direção-Geral do Território*

EEA European Environment Agency

EMEP European Monitoring and Evaluation Programme

FAC2 Fraction of predictions within a factor of two of observations

FB Fractional Bias

GEOS Geometry Engine, Open Source

GIS Geographic Information System

JSON JavaScript Object Notation

MAES Mapping and Assessment of Ecosystems and their Services

MSC-W Meteorological Synthesizing Centre-West

NMSE Normalized Mean Square Error

OGC Open Geospatial Consortium

QUIS Questionnaire for User Interface Satisfaction

SDI Spatial Data Infrastucture

SUS System Usability Scale

TIFF Tagged Image File Format

WCS Web Coverage Service

WFS Web Feature Service

WMS Web Map Service

XML Extensible Markup Language

Chapter 1

Introduction

This chapter provides an overview of the work developed, and a brief description of the motivation, the problem addressed and the solution provided.

1.1 Motivation

Air pollution is a global problem recognised internationally (e.g., UN Sustainable Development Goals of the United Nations [1]) that affects both the environment and human health. Because of these harmful effects, there is a considerable investment made by the European Commission and member states in determining the concentration/deposition of pollutants, the cause-and-effect relationship between the concentration of pollutants and the problems it can cause to the environment and to living beings [2].

To discover the cause-and-effect relationships, as previously mentioned, it is necessary to have access to spatially and temporally explicit information, preferably in the form of observations (certified measurements from air quality stations) of pollutants. However, due to the limited number and unrepresentative character of air quality stations [3], alternative information must be used, such as pollution concentration/deposition provided by air pollution chemical transport models [4]. These air pollution chemical transport models take into account the meteorological conditions (e.g., wind direction and speed, precipitation) as well as the main known emission sources at different heights, and simulate chemical reactions in the atmosphere, pollutant concentrations and deposition in different landcovers [5]. As a result, models provide the spatial distribution of concentration and deposition for different pollutants, and their change in time. In Europe, major concerns regarding air pollution are addressed in the EU Directive 2016/2284 [2], and mostly focus on nitrogen, sulphur, particles and ozone. European air policy assessments rely on data provided by the EMEP/MSC-W (European Monitoring and Evaluation Programme/Meteorological Synthesizing Center-West) transport model [5], herein referred to as EMEP.

In order for a model to be used by decision makers, it must first be evaluated. This evaluation is usually based in a comparison between observations and model predictions by using a set of statistical measures that determine bias, data scatter and error [6].

In Portugal, air pollution concentrations are measured in air quality stations and made available by the environmental Portuguese agency, *Agência Portuguesa do Ambiente* (APA). Although, these stations do not fully represent air pollution throughout Portugal, observations can be used to evaluate model results. The data is stored in tabular form, in the old version excel format, .xls. The excel file can have different formats. Each excel file can represent one station, containing the respective pollutants or it can represent one pollutant, containing the values of the stations where it was captured. So there are many different ways to store this data. The size of each excel varies, but normally is less than 1MB, even though the information is divided in multiple excels.

Model results used in this work are provided by EMEP and extracted for mainland Portugal. EMEP model results are provided as a raster (array of geographical pixel values) in NetCDF format. The EMEP files can have very different sizes between 70 MB and 20 GB, approximately, depending on the temporal resolution (year, month, day, and hour) and the number of pollutants that the model produces.

To complement, the observation and simulation data, it is useful to provide information about the land use, or the ecosystem classification. An ecosystem is a community of living organisms in conjunction with the nonliving components of their environment. Air pollution affects ecosystems, so knowing what type of ecosystem exists at a particular location, it is possible to determine whether or not the pollutant causes harmful effects at that location.

For land use, COS data set [7, 8], provides high resolution information in GIS vector file format (geometrical shapes often associated with a tabular database that describes their attributes) within the Portuguese territory. MAES data set [9, 10] provides high resolution information about the type of ecosystem in a raster file format (each pixel has an index value that, with a table, identifies the type of ecosystem) within the European territory and it has a maximum resolution of 100 m x 100 m.

1.2 Problem

Technological advances have allowed to store large amounts of data and perform faster calculations. Thus, it is possible to store data that were collected over time by air quality stations distributed throughout the territory. On the other hand, it has also become easier to model and simulate concentrations of pollutants in increasingly smaller areas (higher resolution) [11]. The transference of large files is faster and more stable because of the reliability and transference rate of the connections between different machines. Therefore, the access to this type of data has become reachable to anyone. However, as there are different types of data, the processing depends on the format of the data, consequently it implies that the learning process is differentiated for each format and, therefore, the treatment of data is cumbersome.

This data can be treated by two type of users: environmental researchers/decision makers and programmers. Many researchers and decision makers have little knowledge of how to efficiently aggregate data to obtain better results. Hence, they have to spend a lot of time learning, making their study more time consuming. There are challenges to aggregate all this data. First of all, there are many data sources that work and store data differently. In addition, the files are of complex types, and can be large in size or in quantity. Therefore, data processing, from aggregation to spatial overlap and comparison, differs for each source.

The data sets have large dimensions or/and can be divided in multiple files, and even though some users have knowledge on some of the required applications, it can be impractical, if the data access process is manual, having to download each file, upload to the program by indicating a path for each file. In addition, if the files have large dimensions, then the process will be very time consuming to download and also to be used in the available applications to get the results.

For example, a file from the EMEP server can have 20 GB, that takes at least 1 hour to be downloaded. In contrast, the data from the APA server may be distributed over approximately 500 files, making a manual process too complicate due to a large number of files that have to be organised for better access. These problems can be easily solved by developing a system that automatically collects the data from the different sources and runs the calculation in the second plan of the system. With this type of system, the results are generated automatically and the user does not have to wait that long for the results. The programmer must have in account several things when creating a system for data extraction and processing. A good system performs background calculations so that the final user does not have to wait for the results. In addition, for the process to be automated, the system must be able to integrate different tools that handle the data derived from different sources, in different forms and formats.

However, there are currently no known libraries/modules that work with the data for observations of air pollutant concentrations in Portuguese air quality stations and EMEP model results, described above. Due to the non-existence of these libraries, there is a lot of repetitive work to obtain the data and many researchers codifying similar processes to obtain the data.

1.3 Thesis outcome

One of the outcome of this work is a set of libraries that simplifies the access to EMEP model predictions and concentration measured data.

The another outcome of this work is to use a portal that allows the integration of libraries to process, aggregate and present spatial and temporal data of air pollution. With the integration of the developed libraries, a set of demonstration applications are developed that solve some of the problems of data handling, overlap and visualisation for non-specialised users.

To solve the challenges that were announced in section 1.2, a set of libraries are developed and placed on a middleware that is placed between the demonstration applications and the operating system. This middleware is an aggregation of standard tools like Django, GeoServer, PostgreSQL.

With Django, PostgreSQL and GeoServer, it is possible to develop a platform that can allocate web applications. The architecture of the platform is represented in figure 1.1. This architecture allows the integration of modules/libraries which can provide application programming interface (API) that facilitates the access of resources through browser requests. An API is an information gateway that allows the back end of software and the services to communicate with each other.

With the libraries that already freely available to use it is possible to create the architecture, lacking

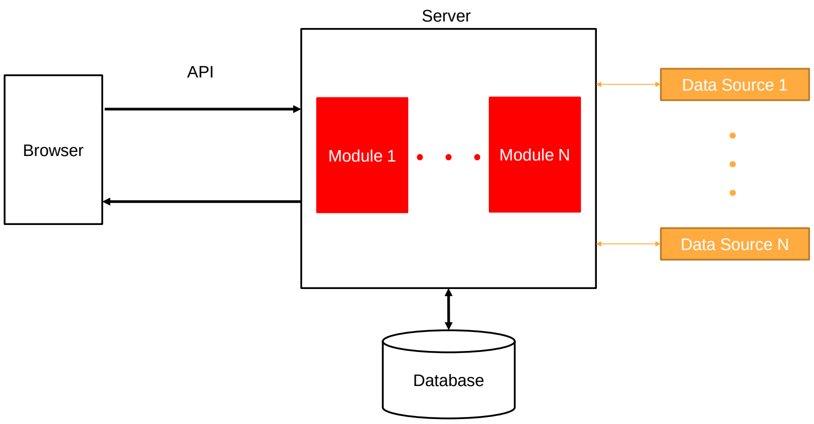


Figure 1.1: Generic architecture of a platform which allows the integration of several modules providing APIs which are accessed through browser requests. The modules are normally responsible for communicating with a data source to collect the data and store it in a database.

only the development of modules to assist in the study of air pollution. As mentioned above, the solution uses Django framework, as a result the following modules are developed in Python:

- GeoExcel - Responsible for storing excel files on the database and on the GeoServer. This application offers a method for the programmers that already know the structure of the excels files. In addition, it offers a graphical interface that uploads any excel to the database.

  GeoExcel is available for public use in a git repository https://github.com/FMMFHD/GeoExcel.

- WebAPA - Responsible for getting the data from the APA database (hourly observations from air quality stations) and to store it on the database of the platform. Also it is responsible for managing the stored concentration measured data. This application offers a graphical interface to upload observations data. An API that can be accessed by web applications with geospatial data is also available.

  WebAPA is available for public use in a git repository https://github.com/FMMFHD/WebAPA.

- WebEMEP - Responsible for getting the data from the EMEP server and to store it on the GeoServer. Also it is responsible for managing the stored EMEP model prediction, which can be accessed trough a proxy. It offers automatic mechanisms to update the most recent EMEP model predictions. An API that can be accessed by web applications with geospatial data is also available.

  WebEMEP is available for public use in a git repository https://github.com/FMMFHD/WebEMEP.

Two demonstration applications that solve some of the non-specialised users' problems are developed using the platform modules:

- Concentration Dashboard - The user interface, that allows to visualise the spatial / temporal variation of pollutants in a region based on the modelled pollutant concentration, model evaluation measures based on temporally aggregated monitoring concentration data for that location / period. In addition, there is the possibility to download the data shown on the web page.

- Deposition Dashboard - With the user interface, it is possible to visualise the spatial / temporal variation of pollutants in a region based on the modelled pollutant deposition. In addition, there is the possibility to download the data shown on the web page.

- Evaluation of EMEP data - Evaluation of EMEP model results by comparing with spatially overlapping measurements from air quality stations using a set of statistical measures.

Results comprise middleware that simplifies access to EMEP model predictions and concentration measured data. It allows the rapid creation of new applications in addition to those enumerated in this section. Furthermore, the developed modules are available as a public resource, allowing the integration of these modules into other systems.

Through the development of the applications listed above, it is possible to verify the improvement of time in access to information. At this time, the information comprises processed 19-year time series (2000-2018), and it is relatively simple to further expand the time-series with additional data from upcoming years. In addition, the processing and the corresponding generation of statistics are carried out quickly and autonomously, with no need for user intervention.

The developed applications were tested by several interested environmental researchers, which answered questionnaire after testing, providing a good feedback of the implemented application. Furthermore, the developed applications were presented at a scientific congress, XIX National Ecology Meeting organised by SPECO (Portuguese Ecology Society, in Portuguese *Sociedade Portuguesa de Ecologia*).

1.4 Outline

The rest of the document is structured as follows:

- In Chapter 2, essential topics related to this work are discussed. First, some concepts of ecology are introduced. Then, the infrastructure/tools available on the market are explained. Finally, the different types of data used are described.

- In Chapter 3, the challenges that may exist in studies on air pollution are described.

- In Chapter 4, the requirements for a programmer to work with pollution data are presented. Then, architecture is presented in a generic way that meets the requirements. Then, each component of the architecture is specified for the solution presented. Finally, the libraries that have been developed are described.

- In Chapter 5, applications developed using the libraries presented in Chapter 3 are presented.

- In Chapter 6, an evaluation of the applications developed is presented, describing and presenting the results of the survey. Finally, the requirements evaluation and the performance evaluation are presented.

- In Chapter 7, the present work is concluded, drawing conclusions, and pointing out aspects to be developed for future work.

Chapter 2

Related Work

This Chapter explains the concepts which were briefly presented in the introduction. In addition, it presents solutions already on the market as well as technologies that are used in the final solution. The different types of data used are also explained. Finally, the difficulties that environmental researchers face when conducting a study are presented.

2.1 Ecology Terminology

In this section relevant concepts on ecology are presented.

- Ecosystem - Integration of the biotic community and its physical environment, within a hierarchy of physical systems that span the range from atom to universe, cited by Tyson (1935) [12].

- Concentration - A measure of the amount of a polluting substance in a given amount of water, soil, air, other medium, EEA (European Environment Agency) Glossary [13].

- Deposition - A measure of the amount of particles in surfaces, including vegetation, surface water or soil, that are transferred from the atmosphere by dry or wet processes , EEA Glossary [13]. Deposition can be a major environmental issue resulting in acidification and eutrophication of natural ecosystems. Increased concentrations of pollutants in the atmosphere due to human activities results in more atmospheric deposition of pollutants, with negative effects on human health, and ecosystems [14].

- Eutrophication - In terrestrial ecosystems, eutrophication can be defined has shifts in plant species composition, towards nitrophilic species (tolerant to high nitrogen content), due to an increase in the availability of nitrogen. This shift can lead to the disappearance of rare or important plants and reduction of biodiversity [15, 16].

- Air pollution - Presence of pollutant substances in the air at a concentration that interferes with human health, or produces other harmful environmental effects, EEA Glossary [13].

- Critical Level - Indicates the concentration of pollutants in the atmosphere above which direct adverse effects on receptors such as plants or ecosystems may occur. This indicator is based on effects observed over periods of one day to several years. The methods used to determine the guidelines for critical levels rely on analysis either of field studies along pollution gradients or experimental studies in the laboratory or in field chambers [17].

- Critical Load - Indicates a quantitative estimate of an exposure, in the form of deposition, to one or more pollutants, below which significant harmful effects on specified sensitive elements of the environment do not seem to occur. This indicator relates to effects on ecosystem structure and functioning, and is expressed as annual depositions. The methods used to determine the guidelines for critical loads rely on analysis of field experiments, modelling or comparisons of sites with different deposition rates [17].

2.2 Geographical Data

In this section, the different services/tools with potential to solve the challenges addressed in section 1.2, and readily available for use, are described. Furthermore, services that can solve part of the challenges in question and that can be adopted in the final solution are also described.

2.2.1 Data Formats

The vector format allows to represent geographic features such as points, lines and polygons with great precision. Generally, each point is represented as a single coordinate pair, while lines and polygons are represented as ordered lists of vertices/points. Attributes are associated with each vector feature [18]. This spatial data format is preferably used when there is a need to compare spatial data especially when they have many attributes.

The raster format allows to store spatial information in a grid format, where the locations are represented by an array of cells (pixels) [19]. This format is used, for example, to represent model results.

An ESRI shapefile is a vector data storage format for storing shapes, and attributes of geographic features [18]. It is stored in a set of related files and contains one feature class.

The Network Common Data Form (NetCDF) is a file format for storing multidimensional scientific data [20]. This format is divided in two parts [21]: a header that contains all the information about dimensions, attributes, and variables; a data part that comprises fixed-size data, containing the data for variables that don't have an unlimited dimension, and variable-size data, containing the data for variables that have an unlimited dimension.

Tagged Image File Format (TIFF) is an industry standard format for handling raster or bitmapped images, that can be saved in a variety of colours [22]. Based on TIFF format, there is a similar format used for georeferenced raster imagery, named GeoTIFF [23].

2.2.2 Geographical Information System

Geographical Information System (GIS) is a framework to aggregate, manage and analyse different types of data, namely spatial data. By using this tool, it is possible to analyse geographic data and organise it in different layers of information to be visualised as maps [24]. With GIS, the spatial data are no longer represented in image/painting format, as is done in the conventional maps (paper map), but are now represented as digital information. Because it is possible to save the data on a computer, there is higher flexibility in data representation. The data can be represented as an image containing cartographic information (e.g., roads, urban/forest/agricultural areas, ecosystems representing assemblages of specific species, etc.), or as set of statistical tables that later can be converted into graphical content, such as scatter plots. The data is saved in a structured format which makes it easier to collect, analyse and save [25, 26].

As the data, whether spatial or not, is stored in a database, it can be related. That is, it is feasible to relate spatial data with non-spatial data (e.g., tabular data) through queries to generate a map or to obtain textual attributes.

The data with geographic attributes can be represented by two different formats, vector and raster.

ArcGIS is a platform to create, manage, share and analyse spatial data, making data viewing and processing more accessible. This platform can be run locally or on cloud through cloud services [27]. The platform architecture consists of several components: the *Portal*, the Apps, the Infrastructure and the External Systems and Services, that allow to create a flexible solution. Figure 2.1 shows the architecture of ArcGIS.

As shown in figure 2.1, the Apps is a set of platform components that most users interact with. The solutions that use this architecture component can have one or more of the following usage patterns: mapping and visualisation, data management, and monitoring, among others. Therefore, the Apps is responsible for connecting people and their business flows to the platform.

In the *Portal*, it is possible to organise the different users and connect them to the appropriate content and resources for defined privileges. Thus, with the *Portal*, it is possible to deliver the right content, to the right user, at the right time.

The *External Systems and Services* are responsible for offering or consuming services from the platform. Thus, it is possible to ensure that the platform's resources are accessible to other systems in different locations.

The *Infrastructure* component is where the core of the platform is located, meaning where the hardware, the software, the services and the data repositories are allocated. This unit consists of several components, each with different roles. These components are (figure 2.1): Visualisation, Analysis, Data Management, and Data and Storage. *Visualisation* represents the web services that create and consume visual information, such as maps. This information can be generated on request, or is pre-rendered and cached, allowing faster access to this kind of information. As the name indicates, *Analysis* is responsible for analysing the data, which can be a simple geometric function to a complex geoprocessing. With *Data Management* it is possible to create, maintain and transform geographic data.

In ArcGIS, *Data and Storage* is divided into three models. The first model is a data store managed by

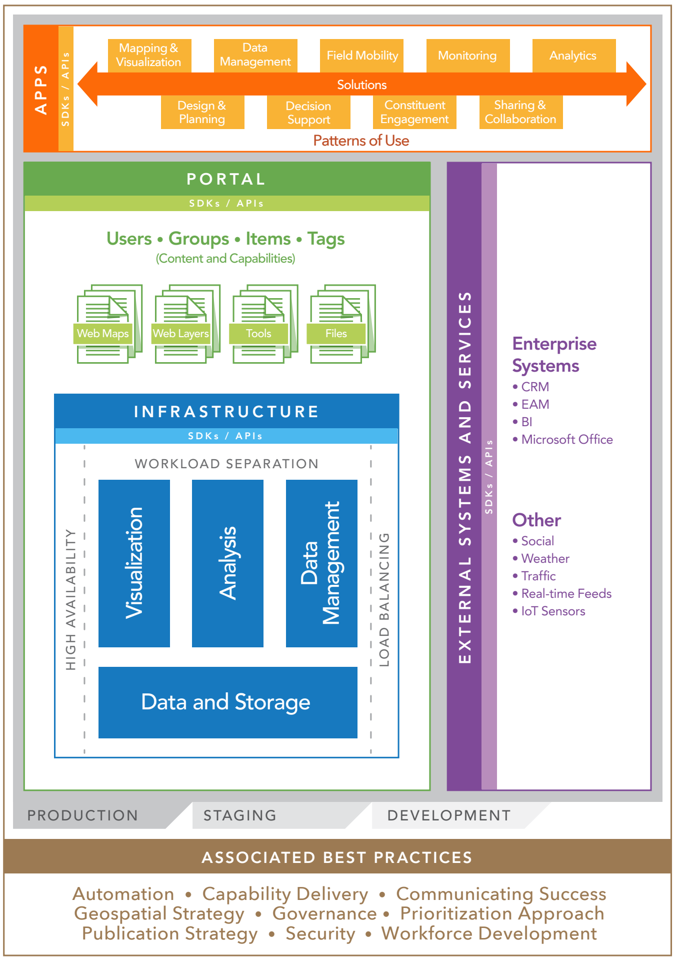


Figure 2.1: Components of the ArcGIS Platform architecture: Apps (orange), Portal (green), Infrastructure (blue), External Systems and Services (purple) [28].

ArcGIS. This data store can be relational to store the vector data, a block of caches to save the produced maps or a temporal space data store for large data. The second model is a model of relational objects for storing geographic data, which can be a file-based structure, a relational database management system, or a local memory system, which can reduce I/O interface requests. The third model is to connect to externally data sources.

To ensure smooth operations in the platform, the following approaches can be applied to the system [28]: simplifies the growth of the system as the distribution of information; ensures that the system meets an operational performance level for a specific period of time; keeps the system reliable and available, protecting the platform from negative impacts related with system changes.

To ensure that no software installation is necessary, ArcGIS has a service that offers conditions to run the system in the cloud. Thus, the programmer does not need to worry about the maintenance of the Infrastructure hardware. Also, with this service, in a short period of time it is possible for the system to increase or decrease its time responsiveness, depending changing needs [29].

ArcGIS platform has several applications. The central application is ArcMap, where it is possible to display and explore GIS data sets for a study area, and where map layouts for printing or publication are created. ArcMap is also the application that it is used to create and edit data sets [30].

2.2.3 Data Storage

Data can be saved on the server in several formats, it may be possible to store data in tables or in files using two separate models.

Spatial DataBase Management Systems (spatial DBMS) was not developed as a software solution, but it is developed to offer the ability to store cartographic data in DBMS. One of the possible DMBS to use is PostgreSQL, but this service does not have the capability to work with geographical attributes. PostGIS is a plugin that allows to save spatial data types as well as functions for analysis and for processing. This plugin is used to expand the PostgreSQL database by providing the ability to save spatial data on a open-source object-relational database management system [31–33].

GeoServer is a three-tier client-server architecture. The server consists of Web Server, Web GIS software, and database. GeoServer allows to share and edit geospatial data. The system was designed to have interoperability, allowing to publish data of any spatial type. The purpose of this software is to make any geospatial information available as much as possible [34].

As previously mentioned, besides being able to save data, GeoServer is also used to publish data, meaning it allows exposing spatial information to any user. This exposure is possible through the implementation of Open Geospatial Consortium (OGC) standards. The OGC standards consist of more than 30 standards that are responsible to cover services, which include standards for distributing spatial and tabular data. There are also standards that allow editing data style. Web Map Service (WMS), Web Feature Service (WFS), Web Coverage Service (WCS) are important OGC standards [35]. With WMS it is possible to access and create maps in different output formats. On the other hand, through the WFS and WCS standards it is possible to share and edit the style of the data that is used to generate the maps, and for the WFS standard it is also possible to edit the data content. WCS is used for data in raster format, while WFS works with vector formats. Through its standards, GeoServer is thus a service that bridges the gap between the various sources of spatial data and the different services for each type of data (figure 2.2).

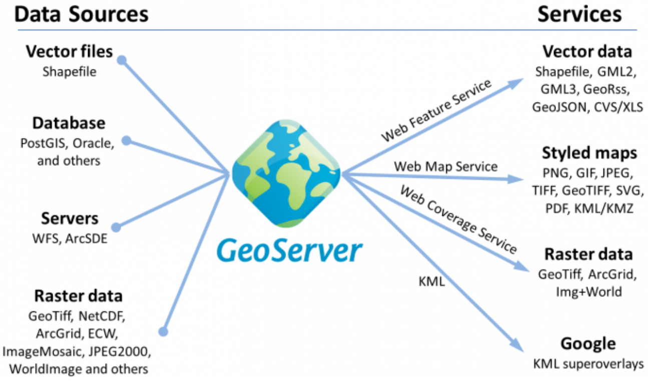
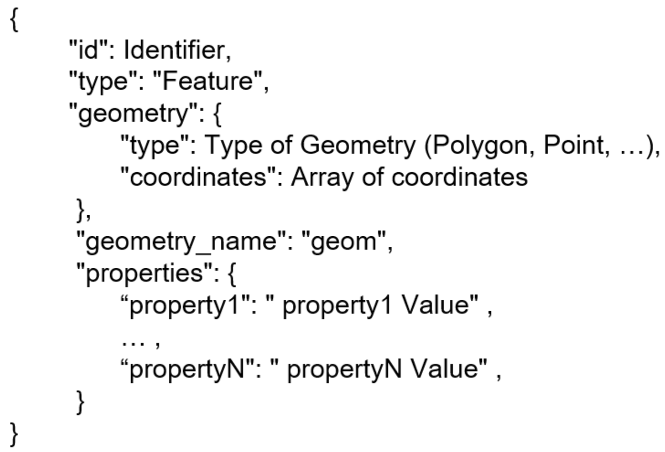

Data Sources
Services
Vector files
Shapefile
Database
PostGIS, Oracle, and others
Servers
WFS, ArcSDE
Raster data
GeoTiff, NetCDF, ArcGrid, ECW, ImageMosaic, JPEG2000, WorldImage and others
GeoServer
Web Feature Service
Web Map Service
Web Coverage Service
KML
Vector data
Shapefile, GML2, GML3, GeoRss, GeoJSON, CVS/XLS
Styled maps
PNG, GIF, JPEG, TIFF, GeoTIFF, SVG, PDF, KML/KMZ
Raster data
GeoTiff, ArcGrid, Img+World
Google
KML superoverlays

Figure 2.2: GeoServer Diagram [36].

Therefore, through its architecture, GeoServer is used as a database and to create maps from the data stored on the system. With the OGC standards, it is easy to share the data with any kind of system, and also it is easy to perform queries to obtain the desired information from the stored data.

The data, that is stored in the GeoServer, is accessed by the GeoServer API. Depending on the OGC protocol, the data may be represented by an image (WMS), by features (WFS - vector data) and by coverages (WCS - raster data). If the data corresponds to an image or a coverage, the format of the data that GeoServer makes available is a file. On the other hand, if the data is represented by features, then the format that GeoServer provides can be either a file or a list of dictionaries. A feature can be represented as a dictionary with a format similar to the one represented in figure 2.3.



```
{
      "id": Identifier,
      "type": "Feature",
      "geometry": {
           "type": Type of Geometry (Polygon, Point, ...),
           "coordinates": Array of coordinates
       },
       "geometry_name": "geom",
       "properties": {
           “property1": " property1 Value" ,
           ... ,
           “propertyN": " propertyN Value" ,
       }
}
```

Figure 2.3: Example of a GeoServer feature format.

The GeoServer provides some operations that take into account the desired standard. If the desired standard is WMS, then GeoServer provides the following operations:

- GetMap - Gets the generated map image for a specified area and pollutant [37];
- GetCapabilities - Gets list of the available layers [37];
- GetLegendGraphic - Gets legend for the generated map [37];
- GetFeatureInfo - Gets the geometry and the attribute values for a pixel location on a map [37];

The manual [37] provides more information on the operations according to each standard (WFS and WCS).

2.2.4 Data Management

Spatial Data Infrastructures (SDI) is a framework that analyses cartographic data and has an application for users and suppliers to communicate. It also facilitates the access to the geographic data because it uses a minimal set of practices, protocols, and standards [38].

GeoNode [39, 40] is a geospatial data management and publishing platform, meaning, a geospatial content management system. The data management tools developed in GeoNode allow to integrate data, metadata creation, and map visualisation. The system consists of several stable open-source projects and it has an easy-to-use interface that allows unskilled users to share data and create interactive maps [41].

GeoNode is a web-based platform that implements GIS and SDI through an open-source framework. As interoperability is one of the main concepts of SDI, it implies that GeoNode is based on technologies with standards provided by OGC. Thus, GeoNode offers the following features:

- spatial search for data and metadata;
- management and sharing of data as well as the management of sharing policies;
- data visualisation as well as the integration of different sources that are stored in external infrastructures and services. The visualisation is accessed through WMS standards;

This technology is a web application developed in Django. Django is a high-level Python Web framework, which allows to easily integrate new modules (applications), and also to access resources from the web. The framework Django allows the development of applications. With this technology, it is possible to provide the features listed above, because it uses several components based on robust and well-known open-source products, (figure 2.4).

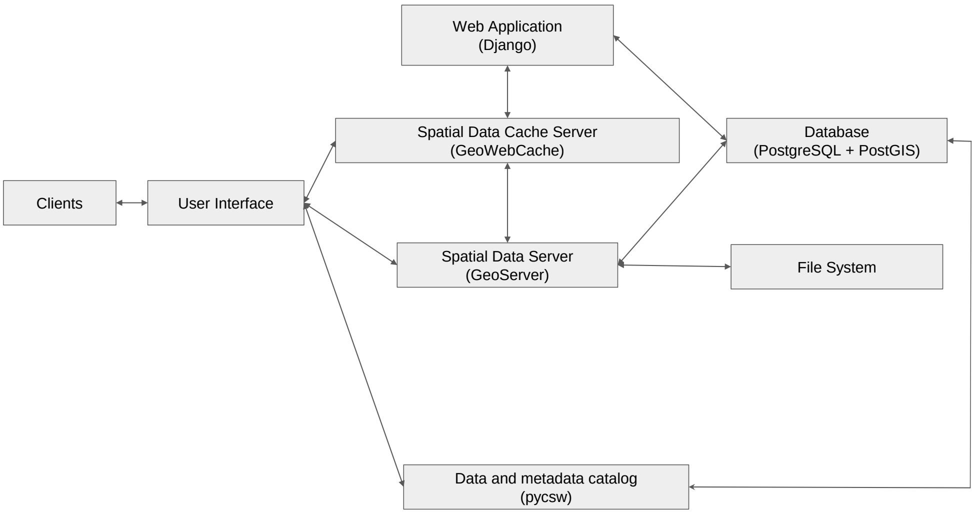


Figure 2.4: GeoNode Diagram, based on [39, 40]

These components, and their respective products, are the following:

- the user interface;
- a spatial data server, where the default option is GeoServer but it can also use QGIS Server. This server uses standards defined by OGC and allows GeoNode to have great flexibility in creating maps and sharing data;
- a spatial data cache server, based on GeoWebCache technology. This server uses the cache to speed up and optimise the distribution of map images. So it saves a lot of processing time as it eliminates processes derived from redundant requests;
- a catalogue which is an OGC CSW server, based on pycsw, allowing interoperability mechanisms to find data and metadata;

- a file system, which allows to save raster data formats;
- a database which allows to save vector data and can be used by several applications. In GeoNode, the database is made with PostgreSQL and the plugin PostGIS.

The web application is responsible for enabling the interaction between the client and the various components of the system. The information model is named ResourceBase which consists of a set of properties, such as: a unique and universal identifier (uuid), a title, a flow of the status of publications, and properties of the metadata. The most relevant resources are layers and maps. The layer represents the geospatial data set in the system. This data set can be saved in two formats, the vector and raster, or it can be provided by an external service. Maps can consist of spatial representation of information that can be collected in several layers. The layers used to create maps are traced by name in the database, in which the corresponding properties are selected, such as visibility, opacity and styles. The cartographic base of the maps (information shown in the background that facilitates user interpretation) can be provided by several organisations, such as OpenStreetMap, Google Maps or MapBox.

GeoNode is able to provide a user interface using the Django templates and JavaScript libraries / frameworks. One of the frameworks used is jQuery which is used, for example, in search boxes, and to provide dynamism to the web page. Ajax is another technology that is used to make requests to the application during the interaction with the client. For example, Ajax is used to obtain the most recent data stored in the server and to print this result on the page. Finally, this user interface includes technology based on OpenLayers or Leaflet to create interactive / dynamic maps.

In addition to all these features, there may be a need to perform operations that take too much time to finish. In this case, a good approach is to perform these operations asynchronously from HTTP requests and responses. GeoNode has already a service that solves this problem, named task queue. For this purpose, GeoNode uses the task queue to process the tasks asynchronously, such as, email notifications or data collection from remote services.

Because GeoNode is a web application, it is possible to run on a web server, such as, nginx or Apache. With a web server it is possible to make the content available to anyone that can access the Internet.

2.2.5 Supporting Libraries

NetCDF4 is an interface to the NetCDF C library [42]. NetCDF is a set of libraries that supports the creation, access, and sharing of array-oriented scientific data [43]. This interface only manipulates NetCDF files. NetCDF file is an array-oriented scientific data with time dimension [44].

Pandas is a software library created for data manipulation and analysis. In particular, it offers structures and operations for manipulating number tables and time series [45].

Numpy is a software library that supports large, multi-dimensional arrays and matrices, which are operated with a large collection of high-level mathematical functions [46].

Rasterio is a library that offers access to many different types of raster data file [47].

Shapely is a package for analysis and manipulation of features, geometric objects, using functions from the GEOS library. GEOS is the geometry engine of the PostGIS. There is restriction with data formats or coordinate systems [48].

Fiona is a library that helps to integrate geographic information systems with other computer systems by reading and writing geographic data files [49].

Leaflet is a JavaScript library, that facilitates the visualisation of georeferenced data in web applications. Leaflet is an open source library for interactive maps that works efficiently across all major desktop/mobile platforms [50].

GDAL is library for raster and vector geospatial data formats [51]. It is responsible for data translation between coordinate systems, or between data formats. It is also used for data processing [52].

2.3 Data Sources for pollution studies

Air quality studies require a large set of data to reach conclusions regarding temporal/spatial trends. Essentially, this requires theoretical data represented by the EMEP model predictions, and the observations from air quality stations made available by APA. In this case, the monitoring data does not represent the whole of mainland Portugal, but model results do. Figure 2.5 shows EMEP model predictions covers all of Europe and the air quality stations in Portugal are scarce and do not cover the whole country.

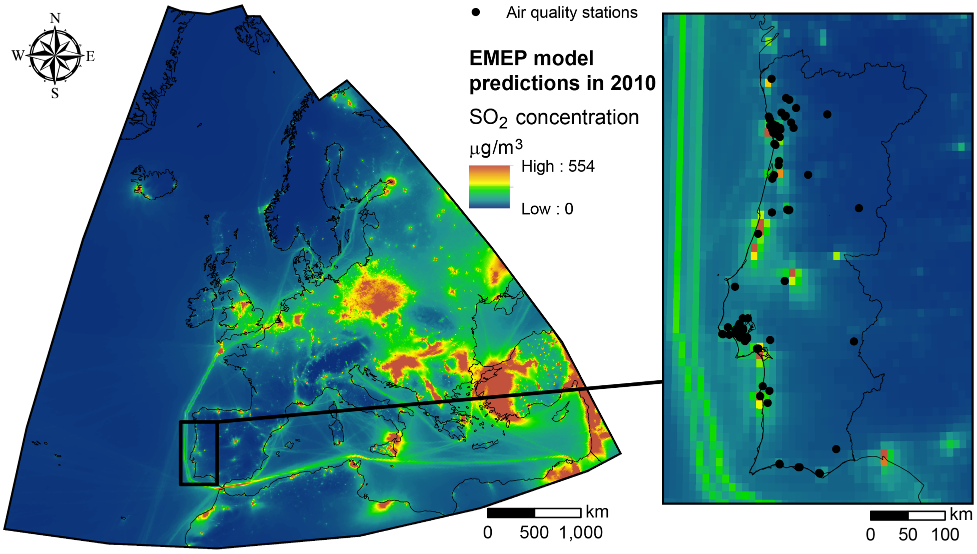


Figure 2.5: The map on the left shows the special variation of EMEP Model Predictions in Europe. The map on the right shows the special variation of EMEP Model Predictions in mainland Portugal, where the black points are the air quality stations.

To monitor air pollution and deposition trends in wider regions, models must be used. However, model quality must be ascertained prior to its use. As stated in the introduction, this can be done with observations in scattered locations (evaluation points). There are some limitations to this method. Foremost, as the spatial distribution of air quality stations is partly controlled by population density, observations cover a very small part of the territory. Furthermore, some air quality stations have been deactivated due to limited funding and observations are only available for parts of the timeline. However, the information used is the best information available and, even with all these limitations.

In addition to data regarding air pollution, cartographic data, data providing additional information such as land use (COS) and ecosystems (MAES) at a given location, are also used.

2.3.1 Observations

There are about 97 stations spread across the country, including the Archipelagos of Madeira and Azores. These stations are divided into 3 types: traffic, industrial, and background. As the name indicates, the traffic stations are in places with a lot of road traffic so the pollution data captured is very dependent on the amount and the type of vehicles that circulate at a given moment. Industrial stations are influenced by emissions from factories and other industries, such as those related with the production of energy. On the other hand, background stations are not located near relevant pollution sources. For this reason, only background stations are used in model evaluation, as they reflect overall air quality and not specific localised emission sources.

The data from the Portuguese Environment Agency, APA, are collected by several air quality stations spread across the territory (figure 2.5). Observations are divided in several excel files with less than 1MB. These excel files can either contain data about all pollutants for a given station and year, or from all stations and a specific pollutant in a given year [53]. However, these files do not contain the locations of the stations only their name. So it is also necessary to download the information of the active stations from APA website [54], and for the deactivated stations, it is necessary to do an exhaustive search through a form [55].

In addition to these differences due to the type of station, sensors sometimes do not collect data 24 hours a day, 365 days a year, due to malfunction. Stations with few values collected throughout the year cannot be used to validate model performance, to avoid incurring a gross error in the final results, derived from hourly/seasonal considerable changes in emissions related with human activity. For example, during the summer, there are no emissions related with domestic heating, mostly comprising nitrogen oxides [56]. So, if an air quality station only collects data during the summer or winter, the average annual concentration of nitrogen oxides will not be representative. Furthermore, air quality stations captures values every hour, but the model only provides yearly/monthly values. Because of this discrepancy, it is necessary to average the observed values according to the studied temporal resolution.

To produce accurate results, observation data must be averaged. Therefore, by using averages without considering data capture, model evaluation may not be credible, due to a low number of samples. The more samples there are, the more it translates the reality. Therefore, in order to have credibility in the results, it is required that the station captures a certain number of values. For example, if the station captured values every hour, then ideally per day the station would have captured 24 values; per month, 24 times the number of days in the month; and per year, 24 times the number of the days in the year.

Therefore, in order to evaluate model performance only background stations with a capture rate above 75% should be used for any given temporal resolution and date [4].

2.3.2 EMEP Model predictions

The EMEP model is a tool that monitors and evaluates the long-range transmission of air pollutants. This information is critical to develop measures aiming to mitigate transboundary air pollution problems [5]. From the extensive EMEP program, this work will only focus on the results generated by the MSC-W, Meteorological Synthesizing Center-West. EMEP / MSC-W is an Eulerian model that has been used to simulate the presence of chemical particles in the atmosphere and their deposition in the ecosystems [4, 5]. Results are then used to assess the air pollution policies implemented in European member states. The model was built to describe the long-range transportation of the pollution [5]. Initially, the results were represented by a matrix of cells with a 50 km resolution. Nowadays, these results are represented with a $\sim$10 km resolution. The modelling has vertical spatial discretization with 20 vertical layers, where the minimum height is 50 meters [5]. However, the value represented by the cell corresponds to the results of the lowest layer, as it is the one that is closest to the population and ecosystems, and so it is the most relevant information for management / monitoring air quality [5]. Although the result is only of the lowest layer, the model takes into account all gases released at different heights and all reactions between different gases, as well as the weather regime that existed during the time period [57, 58].

EMEP / MSC-W generates data covering the whole of Europe. Depending on the temporal resolution, it produces more than one layer each year. For example, if the chosen resolution is monthly, then the model generates 12 coverages, one for each month of the year. Thus, in order to facilitate access to data, the data from the same year are aggregated and another dimension is created, the temporal dimension. That is, to choose a value, latitude, longitude and date must be indicated. Therefore, EMEP model predictions are treated as a 3D matrix with the plug-in NetCDF4, where the dimensions are: latitude, longitude and time.

The EMEP data is in raster format. The data is accessed through a catalogue, that contains information for several years and for all the possible temporal resolutions. For a given resolution and year, it is possible to download all data in a NETCDF file, or to download part of the data through WCS and WMS services [59].

2.3.3 COS - *Carta de Uso e Ocupação do Solo*

The COS, *Carta de Uso e Ocupação do Solo*, is produced by *Direcção Geral do Território* (DGT), allowing a general view of the use of territorial resources and perceiving macro landscapes that reflect the diversity of the continental territory [7]. The information used in this work is from the land-use in 2018 and it is provided in vector format comprising polygons in the official Portuguese projected coordinate system (ETRS89/PT-TM06). This means that a location is defined by X and Y coordinates (instead of longitude and latitude) in meters (instead of degrees).

COS comprises polygons that represent occupation/use units of the soil. A unit is any area of land greater than or equal to the defined minimum cartographic unit (1 ha) with a distance between lines of 20 m or more, whose percentage of a given occupancy/land-use class is greater than or equal to 75% of the entire delimited area [7]. Each polygon is classified with only one land-use/cover code selected from the most detailed hierarchical level of the COS nomenclature. The nomenclature consists of a hierarchical system of 4 classes of land use, where level 4 is the most detailed [7].

The COS file format is a ESRI shapefile with 5 fields:

- ID - Unique Identifier;
- COS2015_n1 - Land cover/use classes code at level 1;
- COS2015_n4 - Land cover/use classes code at level 4;
- COS2015_Lg - Description of land cover/use classes;
- AREA - Polygon area;

2.3.4 MAES - Mapping and Assessment of Ecosystems and their Services

MAES, the Mapping and Assessment of Ecosystems and their Services framework [9] was created by the European Member States, each one responsible for mapping and assessing the state of ecosystems and of the services they provide, based on information available in large-scale land cover maps (such as Corine Land Cover, CLC) and linked to ecosystems, using the European Nature Information System (EUNIS) classification [60].

MAES is provided in raster format, where each pixel has an index that indicates the type of ecosystem. This classification has 2 levels, EUNIS_L1 and EUNIS_L2, corresponding respectively to the Major ecosystem category and to Ecosystem type, for mapping and assessment [60].

Chapter 3

Challenges on Air Pollution Studies

The challenges described in this Chapter have been identified based on methods used and partly described in an article by Oliveira et al. (2020) [4]. The objectives of the work used as a case study were to provide the distribution of nitrogen and sulphur depositions, over the western Iberian Peninsula, based on theoretical models and their evaluation using observations from air quality stations in Spain and Portugal. The data used and information produced was relative to the year 2015.

Although two theoretical models were used by [4], this thesis will only focus on data collection and processing from the EMEP model, which provides much more information and is available for download. Model results were downloaded from the EMEP web page at [59]. Observations from Spain and Portugal used for model evaluation, were downloaded from the European Environment Agency website; from the Norwegian Institute for Air Research website; and from the *Agência Portuguesa do Ambiente* website, at [61].

3.1 Data Download

Before the user starts calculating the results, it is necessary to download data from the APA and EMEP websites. In the APA website, the data is downloaded by year/pollutant/station corresponding to hourly concentrations of pollutants containing nitrogen, sulphur, ozone and also to concentration of particles smaller than 10 μm (PM10) and smaller than 2.5 μm (PM2.5). Data from the EMEP model are available with annual or monthly resolution, and comprise nitrogen and sulphur concentrations and depositions, ozone, PM10, and PM2.5 concentrations in the air. There are numerous data files to download, namely one file for each combination of station and year. Therefore, if the time period under analysis is (e.g.) 10 years, then the user must download ten times the number of files. Although, each file is usually small, together they comprise a large amount of data. There are approximately 97 stations, so, for a period of 10 years, the user must download 970 files. On the other hand, the EMEP website divides files by years and temporal resolution. So, for a period of time of 10 years, the user must download 10 files for each resolution. To process the data with a monthly resolution the user must download one NetCDF file per month/year. Therefore, this step is less practical the longer the time interval under analysis.

3.2 Data Processing

Once the user has downloaded all the data to the local drive then he/she has to do the following steps to evaluate the model, and to produce a scatter plot such as those represented in figure 3.1. The difficulties on each step are as well described bellow.

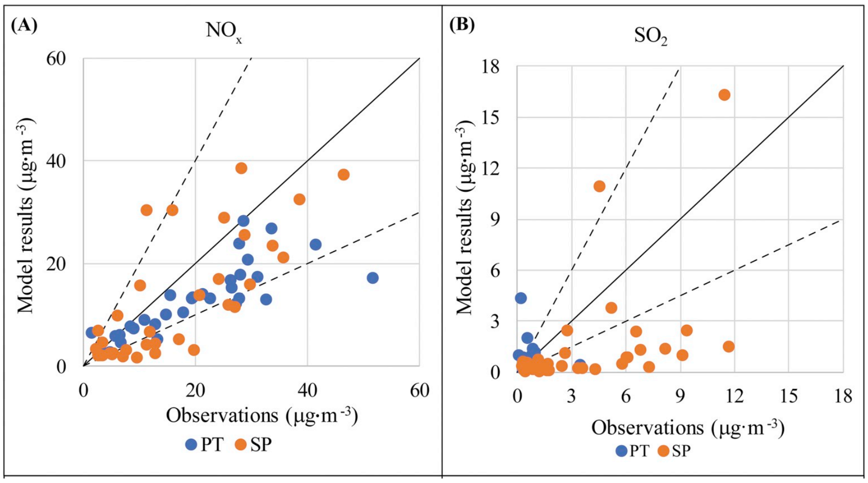


Figure 3.1: Scatter plots showing annual average modelled concentrations against observations; PT represents stations in Portugal and SP in Spain. Extracted from [4], figure 4.

Create a spatial file with the locations of APA stations

In step one, a spatial file with the locations of the APA air quality stations is created. This step is critical so that the coordinates of the stations are compatible with the coordinates used by EMEP, in decimal degrees.

Transformation of the observations

In step two, observations from the APA website (CSV file without spatial attributes) are transformed in annual / monthly concentrations to compare with model results. For this, it is necessary to aggregate the information per unit of time (e.g., annual / monthly). It is also important to exclude stations with large periods of absent data due to malfunction, and guarantee a data capture of 75% [4]. For concentrations of pollutants in the air, the annual / monthly average and the data capture are determined; for depositions of pollutants, effects in ecosystems result from the accumulation in time in a given region, so the sum is calculated. That is, for each combination of station/pollutant, two values will be determined, one representing the temporal average and the other the percentage of the data capture.

This data processing was done using R statistical software [62]. The developed script was prepared to read and use specific files, for the year 2015, that were previously downloaded and preprocessed in Excel. The final results were one excel file for each pollutant. This step was thought only for annual resolution, and so, the script cannot be applied to shorter time-periods. In other words, there is no single procedure where the only thing that changes is the input data. Thus, when the user needs results with monthly resolution, the script must be changed accordingly. For example, for each pollutant, there are two values per year (average and data capture) with annual resolution but 24 per year if the resolution is monthly, so data with monthly resolution is 12 times larger.

Transform excel files in spatial data

In step three, the excels in tabular format built in the previous step, are converted to spatial data, meaning each excel is converted to a feature class within a geodatabase, which is a data structure specific for ArcGIS. A geodatabase is a collection of geographic data sets held in a multi-user relational DBMS, a Microsoft Access database, or a common file system folder [63]. The final result is a spatial database where time-aggregated data, and its respective capture, are stored. The database is organised by pollutant, each line corresponding to point with X and Y coordinates (air quality station) in a vector file, and each column to an annual values.

Create a raster file from a NetCDF file containing EMEP model predictions

In step four, a raster file in Geotiff format per pollutant / year is produced, by extracting the gridded information from a NetCDF file containing several layers with all model results for Europe. There is a limited number of tools that can be applied to NetCDF layers in ArcMap. The extraction of pixel values to overlapping vector points, for example, cannot be done and requires the extraction of the layers to a single raster file, in this case in Geotiff format. Because of this, the size of the information increases further.

Clip the raster file into a smaller geographical domain

In step five, the raster is cut into a smaller geographical domain of interest. In the case of the work described in [4], the Iberian Peninsula was the domain of interest. For this step, data extraction and clipping was done manually in ArcMap, separately for each pollutant of interest. This procedure requires time because ArcMap is foremost a visualisation software and represents all the information prior to processing. So, the larger the region, the longer it takes to process. This tool generates new files so it is necessary to organise the files for future use in the workflow.

Overlap the observations with EMEP model predictions

In step six, the observations are overlapped with EMEP model predictions and the concentrations / depositions in the pixels that overlap the air quality stations are extracted. For this purpose, ArcMap was also used by employing a specific tool that allows to extract values from multiple rasters to points in a vector file. This tool facilitates data extraction, as it can be used to extract several years at the same time. The results comprise a spatial file in vector format containing temporally aggregated observations and model predictions in the same location, per pollutant.

Export the spatial data to produce a scatter plot

In step seven, the spatial file is exported to an excel file to produce scatter plots where observed values are plotted against model predictions for visual evaluation. Furthermore, $x = y$, $x = 2y$ and $x = \frac{1}{2}y$ lines are also added to facilitate interpretations (figure 3.1). It is necessary to export the new data to another format, such as an excel spreadsheet, because ArcMap is limited in graphical representation (e.g., scatter plots). Therefore, additional space is required to store the data. The export to an excel spreadsheet is also done separately for each pollutant, manually, in ArcMap.

Calculate model performance statistical measures

In step eight, model performance statistical measures, described in section 5.3 and [4], are calculated. This step was done by developing a script in R to run separately for each pollutant/year, resulting in an excel/txt data file for each combination. The problem in this step was the same as in the step fourth, because it requires a script to calculate model performance statistical measures. This procedure can also be done manually in an Excel spreadsheet, for example. However, the amount of files necessary makes it unpractical.

3.3 Discussion

In addition to the problems previously announced, the user must take into account the formats that are eligible for each tool. If the format produced by a tool is not eligible for another tool, then it is necessary to transform the data. The transformations are time-consuming and prone to human error. Therefore, the use of multiple tools slows down the procedure and takes even more time if we consider that the user must also learn how to use, and how to connect several tools. In addition, it makes more difficult to integrate this procedure in other machines because it is necessary to install several tools that may not be compatible with those machines.

The procedure consists of independent steps. For this reason it is necessary to remove the data from the tools and save the data in a memory, meaning, to create intermediate data, which requires additional storing space (initial data + intermediate results + final results). However, if there is an error, it is only required to run one step, except when the detected errors are from previous steps. If this procedure was undertaken in one tool/program, the memory usage would be lower, and no intermediate data would be required, and, consequently, stored.

More importantly, repetitive processes made by hand usually generate errors derived from human tiredness or lack of attention. Because of these errors, some steps must be repeated, contributing to increased execution time. Moreover, the execution time also increases due to the files and folders organisation in order to be able to repeat the previous steps in the case of errors. The procedure described above was made by hand, meaning, between the steps it is necessary human interaction to start/do the next step. Therefore, the user must be focused on this procedure for most of the time.

Chapter 4

Libraries for pollution data programming

In this chapter, the requirements, the architecture of the platform and the modules/libraries important to assist in the study of air pollution and its effect on ecosystems, developed during this project, will be described.

4.1 Requirements

The requirements indicate what will be possible to do with the libraries. These requirements are necessary for the programmer to work with data pollution. If these requirements are all implemented then there will be a set of libraries that can help to create web applications that facilitate the researchers in their studies.

The proposed requirements are:

1. The library allows the development of applications that use georeferenced data stored in excel:
   - 1.1. The library should provide a graphical interface for the submission of georeferenced excel data.
   - 1.2. The library should provide an excel data version management.
   - 1.3. The data stored in excel files should be accessible as features.
2. The library allows the collection and processing of pollution data from the Portuguese network of air pollution stations:
   - 2.1. The air pollution data should be stored locally.
   - 2.2. The library should provide an API for data collection.
   - 2.3. The library should provide an API for data access (per date/resolution/pollutant/station).
3. The library allows EMEP model predictions data management:

    3.1. The EMEP model predictions should be stored locally.

    3.2. The library should provide automatic download of new data versions.

    3.3. The library should provide an API for data access (per date/resolution/pollutant).

4. The library allows an easy access to Geonames service:

    4.1. The library should provide download management.

    4.2. The library should provide an API for access to coordinates.

5. Libraries allow the management of equivalences between EMEP model predictions and observations data.

6. Libraries must run as part of a Middleware for web/georeferenced application development.

7. Libraries must store data in geospatial database.

8. Libraries must store data on a geospatial data sharing server.

9. Libraries must use a standard API for database access.

4.2 Architecture

The system must be able to integrate the requirements announced in section 4.1. The characteristics of the system should allow easy integration of libraries. In addition, it must be as generic as possible in order for any programmer to be able to work with the system.

The framework provides a generic functionality that can be changed by additional user-written code (programmer), thus providing application-specific software. In addition, it has a standard and universal way to build and deploy applications, and it may include support code libraries that bring together all the different components. According to the requirements, the system must be able to support communication with other systems via Ethernet, and must be web-based.

The system needs to store data. The storage must be able to store data in tabular format and in file format. The Database is an additional system that allows to store data in table format, and the Spatial Data Server is responsible to manage spatial data, including data that is not possible to store in a table.

Figure 4.1 shows a generic representation of the architecture of a system, that is capable to integrate support and user-written code. Due to the framework, it is possible to connect to different storage systems and, at the same time, to develop applications that facilitate the access to results, which were previously complicated to access. The system is also capable to communicate with observations and model predictions servers.

4.3 Target system

The generic architecture presented in figure 4.1 is already implemented in GeoNode, a product that it is available to the public. The GeoNode is implemented in Python and provides a web-based platform

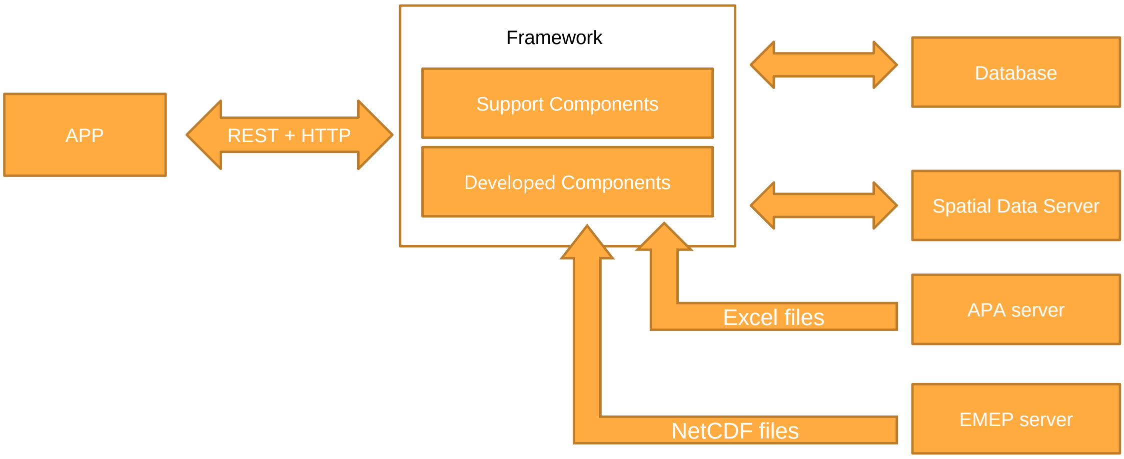


Figure 4.1: Generic Architecture.

that implements GIS and SDI.

Since GeoNode is already linked to a Database and a Spatial Data Server, there is no need to develop a new platform, because it is possible to change the behaviour of the GeoNode system by implementing modules/libraries.

Even though, it is possible to work with any systems for the Database and Spatial Data Server, these are based on the GeoNode product. The chosen Database, PostgreSQL, is a database management system that, together with the PostGIS plugin, is capable of storing data with geographical attributes. GeoServer is the chosen Spatial Data Server.

Figure 4.2 shows a detailed version of the system architecture where the systems/libraries used are instantiated. The developed components are responsible for managing observations and model predictions and, therefore, communicate with data source servers. The developed system takes advantage of some modules of the GeoNode, namely the graphical interface for the submission of georeferenced data. Therefore, this architecture is based on the GeoNode product but only uses some of its modules.

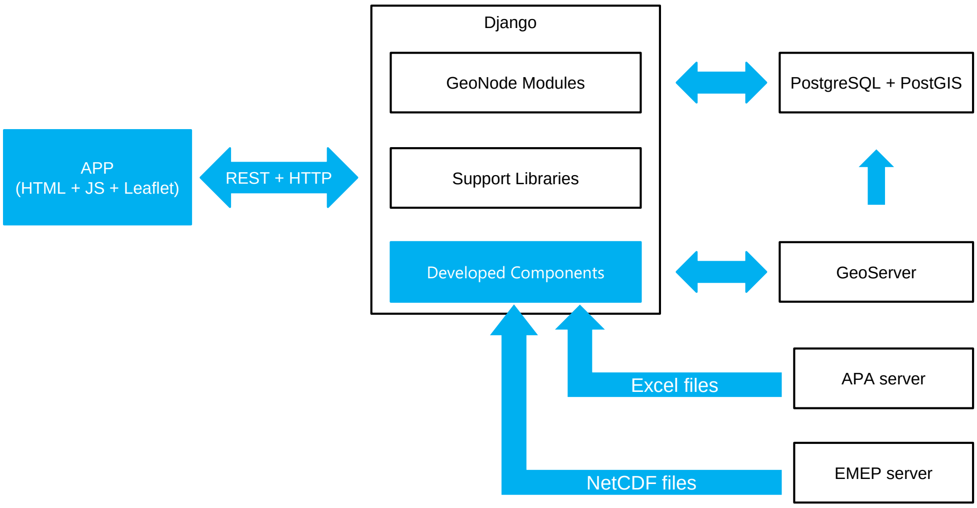


Figure 4.2: Architecture with its supporting systems. Blue boxes can be developed by the programmer, and the black bounded boxes have been developed by other entities.

GeoNode modules do not cover all the programming needs to implement the requirements. For this reason other support libraries must be added, such as: NetCDF4; Pandas; numpy; rasterio; shapely; fiona; Leaflet.

4.4 GeoExcel

GeoNode has a graphical interface where the user can upload files with spatial attributes to the GeoServer, and can, for example, upload shapefiles, geotiffs, etc. The GeoNode module responsible for uploading data to the GeoServer is named Upload. However, it has a limitation, because it cannot upload excel files to GeoServer/PostgreSQL. Figure 4.3 represents a diagram of the integration of GeoExcel in GeoNode modules. As it can be seen, the GeoExcel is inside of Upload Module because it is an extension that allows the upload of excel files. Therefore, GeoExcel was developed as an extension of GeoNode that allows the upload of excel files.

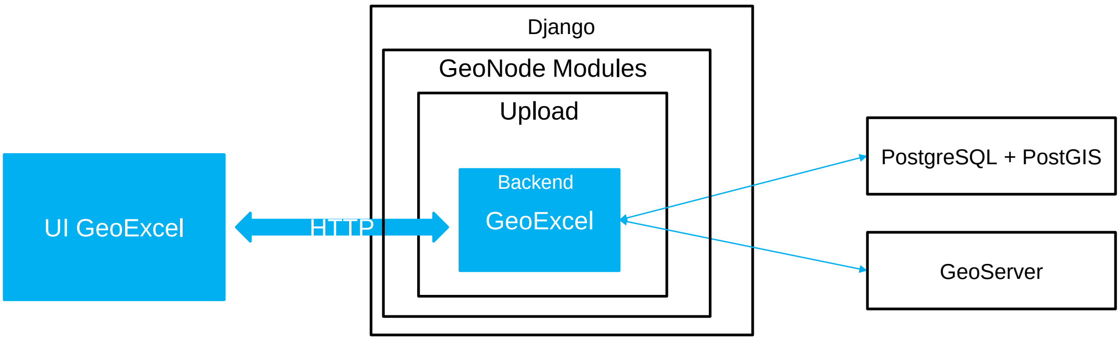


Figure 4.3: Integration of GeoExcel, where blue boxes corresponds to GeoExcel, and the black bounded boxes have been developed by other entities.

The GeoExcel component developed during this project is a Django application that can also be used in GeoNode and it is available for public use in a git repository, https://github.com/FMMFHD/GeoExcel.

4.4.1 Backend Implementation

When the GeoNode module detects that the uploaded file is an excel file, it runs the GeoExcel code. This code does the following procedures. Firstly, the excel file must be opened to access the data. Pandas is a library used to control and to access excel files. After access is established, the next step is to verify if the table exists in the database, otherwise it must be created. The final step is to copy the data from the excel file to the table in the database. As the final step depends on the type of data that is stored in the excel file, the process calls a function responsible for parsing the excel file, that is an input parameter of the method. To save the data in database the following pseudo-code is executed, *Algorithm 1*.

The method `check_table`, line 2 of the *Algorithm 1*, is responsible for verifying if the table exists, otherwise it can create a new table. The excel is opened in line 3. In the line 4, the method defined as argument, that takes the data from excel file and creates SQL statements for storing the data in the

Algorithm 1: Stores the excel data into the database.

```
Input  : Path to the excel (PathExcel); Method to stored the data on the database
         (functionInsert); Database table name (tableName; Table
         Characteristics(TableChracteristics)
Output: None
Function copy_excel_database(PathExcel, functionInsert, tableName, TableChracteristics):
    check_table(tableName, TableChracteristics, . . . )
    dataframe = pandas.read_excel(PathExcel, . . . )
    sqlStatements = functionInsert(dataframe, tableName, . . . )
    Execute(sqlStatements)
```

database, is invoked. The last line executes the SQL statements, in other words, it saves the data into the database.

The logic of the method `functionInsert` is defined by the programmer that invokes the GeoExcel method, `copy_excel_database`. Since excel files can have varied structures, so the parser (a software component that takes input data and creates a data structure) that allows to take the excel and build SQL statements to save in the database, depends on the excel file in question. To allow some freedom in the structure of excel, the `functionInsert` method is placed as an input parameter giving the programmer freedom to upload any excel file.

4.4.2 User Interface (UI)

In addition to the `copy_excel_database` method, GeoExcel offers a graphical interface that allows anyone to upload an excel. The backend of this interface is simple. The first step is to receive the file inserted into the Upload Module (one of the GeoNode modules) graphical interface. The file comes in binary format and is therefore converted to CSV, as Pandas offers this functionality.

Since excel is converted to attributes (features), it means that it is possible to either create a new table or update an existing table. So, the second step is to make the tables in the database available to the graphical interface.

If the user decides to create a new table, then it is necessary to check that it does not exist. After this verification, it is necessary to define variables so the user can configure the parser through a graphical interface. These variables are:

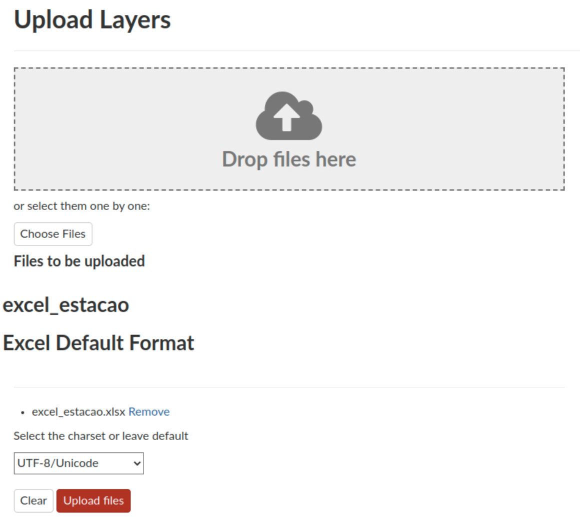
- The name of the excel columns
- Code and names of system coordinates

After the user configures the parser through the user interface, the final step is to upload the excel file to the database. To upload the excel file, it is used the method `copy_excel_database` where the input parameter `functionInsert` is the parser defined by the user in the graphical interface.

So that the user is not always configuring the parser, this is saved in a separated table in the database. This table is called `mapping_json` and has 2 columns: one of the columns is the parser used to save the excel in the database and the other column is the name of the table where the parser was used to save the excel file.

If the user decides to add new entries to an existing table, then it is only necessary to get the parser from the `mapping_json` using the table name given by the user. Having the parser and the excel file, it is invoked the method `copy_excel_database` to upload the excel to the database.

As the GeoExcel is an extension of the Upload module, the initial part of the user interface is common. This part is where the user selects the file (e.g. excel) that wants to upload. Figure 4.4 shows the original user interface for uploading files from GeoNode modules.



Upload Layers

Drop files here

or select them one by one:

Choose Files

Files to be uploaded

excel_estacao

Excel Default Format

- excel_estacao.xlsx Remove

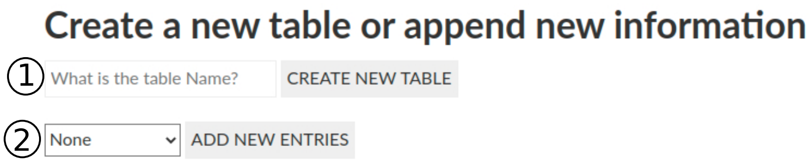
Select the charset or leave default

UTF-8/Unicode

Clear Upload files

Figure 4.4: GeoNode interface where the user can choose to upload the excel file. In this example, the user is uploading the file excel_estacao.xlsx .

After choosing an excel file, the Upload module will detect that the file is an excel and will redirect the user to the GeoExcel graphical interface where he/she will have to choose either to create a new table or to update an existing one (figure 4.5). The user must click on the first button (CREATE NEW TABLE), after typing the table name on the input text ① (figure 4.5). Clicking this button means that the user choose to create a new table to insert the contents of the file. If the user clicks on the button (ADD NEW ENTRIES) then it means the user wants to update the table selected on the list ② (figure 4.5). If the user clicks on the button (ADD NEW ENTRIES), then is redirected to the web page that belongs to the Upload module (figure 4.4). If the user clicks on the button (CREATE NEW TABLE), then is redirected to a web page similar to the one presented in figure 4.6.



Create a new table or append new information

① What is the table Name? CREATE NEW TABLE

② None ADD NEW ENTRIES

Figure 4.5: GeoExcel user interface, where the user can choose between creating a new table or updating an existing one.

The web page, figure 4.6, is a graphical interface that allows to configure the parser. This configuration is divided in 4 steps:

1. If the excel contains coordinates, then the user must select which columns represent latitude and

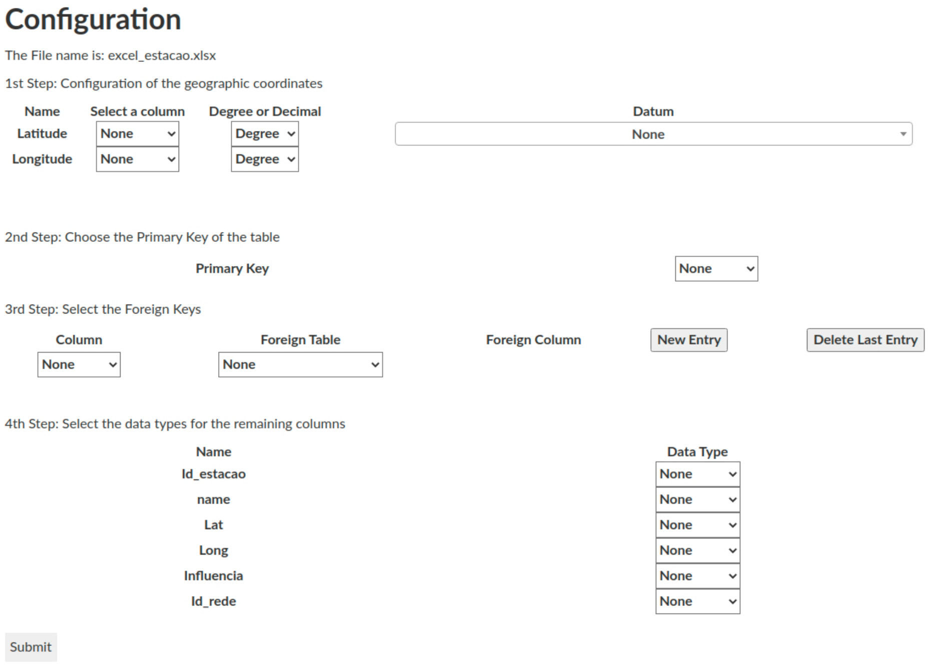
longitude. The format of the coordinates must be selected, which can be either degrees, minutes and seconds, or decimal degrees. Furthermore the user must select the coordinate system, the default being EPSG:4326, which represents the geographical position in degrees, not meters, in reference to the World Geodetic System 1984 (WGS84) datum. The latitude and longitude columns information are merged and a new column named location is created, removing the latitude and longitude columns. The user can choose the name of this new column.

2. The second step is to choose the table's primary key. If *coord* appears on the list, then it means that the coordinates column can be chosen as a primary key.

3. To connect this new table to other tables, the column that will be connected to a foreign table must be selected. In this case, the column in the foreign table must also be selected. The new table may have as many foreign keys as the number of columns in the excel file.

4. The column that represents the coordinates and the columns that are linked to foreign tables, already have the type of data defined. Therefore, the last step is to choose the data type of the remaining columns. The available data types are: Integer, Float, String and Date/Time.

After the parser is configured, the configuration is sent to the backend that starts to upload the excel to the database.



Configuration

The File name is: excel_estacao.xlsx

1st Step: Configuration of the geographic coordinates

Name	Select a column	Degree or Decimal
Latitude	None	Degree
Longitude	None	Degree

Datum: None

2nd Step: Choose the Primary Key of the table

Primary Key: None

3rd Step: Select the Foreign Keys

Column	Foreign Table	Foreign Column
None	None	

4th Step: Select the data types for the remaining columns

Name	Data Type
Id_estacao	None
name	None
Lat	None
Long	None
Influencia	None
Id_rede	None

Figure 4.6: GeoExcel user interface, where the user can configure the parser to upload the excel to the database.

The excel files, that are uploaded to the database using the graphical interface, must have a structure similar to a database table, meaning the excel header must have one line, containing only the names of the columns. Figure 4.7 shows a transformation of an excel file with spatial attributes into a database entry, where the structure of the excel file is similar to a database entry.

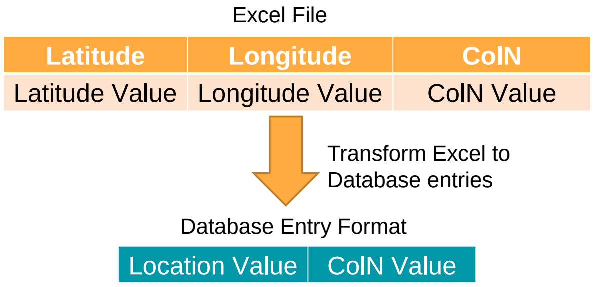


Figure 4.7: Example of uploading an excel file, the Latitude and Longitude columns are the coordinates.

4.4.3 GeoExcel Usage

In order for a programmer to add the GeoExcel Library into his/her applications, he/she must:

- Place the GeoExcel module on the same directory of the project. In addition, it is necessary to define the URL for this module, meaning it is necessary to add the code: "url(r’ˆgeoexcel/’, include(’geoexcel.urls’))" to the file *urls.py* of the respective project.

- Add the name of the component, GeoExcel, to the list of *INSTALLED_APPS* in *settings.py*, so that the project can run the GeoExcel module and import methods from this component.

- Change the file *global_settings.py* or replace this file by a similar one, changing the respective *import*. The *global_settings.py* contains the definitions of the global variables and it is located in a general module, named General_modules, that contains common methods to different components. This module is a private repository, but all the methods are commonly used in any project, so it is easy to create the methods and to replace the respective *imports*.

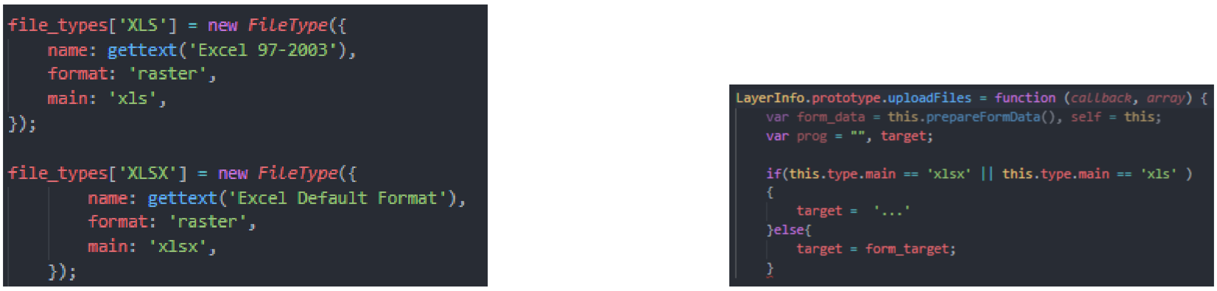
- Add the code to the file *FileTypes.js* that is locate in directory ".../geonode/static/geonode/js/upload/", as represented in figure 4.8a. This change allows the GeoNode’s graphical interface to accept excel files.

- Add a condition to the prototype of the class *LayerInfo*, located in the file *LayerInfo.js*, in the same directory as the file *FileTypes.js*. Basically, the condition will detect that the file is an excel file and, instead of following the same procedure, redirects to the GeoExcel module, to the interface illustrated in figure 4.5. Figure 4.8b shows the condition that detects if the file is an excel. The target is the URL to the view of the component GeoExcel (.../geoexcel/upload).

It is only necessary to create SQL statements to access the data uploaded by the component. It should be taken into account that all uploads end up in the same database and that the excels must be compatible with the database table.

4.5 WebAPA

The objectives of the WebAPA component are to store observations data from the APA server in a georeferenced database. Besides storage, it offers an API for developing web applications with geospa-



```
file_types['XLS'] = new FileType({
    name: gettext('Excel 97-2003'),
    format: 'raster',
    main: 'xls',
});

file_types['XLSX'] = new FileType({
        name: gettext('Excel Default Format'),
        format: 'raster',
        main: 'xlsx',
    });
```

(a) Code that defines the files types that are accept on the Upload graphical interface.

```
LayerInfo.prototype.uploadFiles = function (callback, array) {
    var form_data = this.prepareFormData(), self = this;
    var prog = "", target;

    if(this.type.main == 'xlsx' || this.type.main == 'xls' )
    {
        target =  '...'
    }else{
        target = form_target;
    }
```

(b) Code that is insert in the prototype of the class LayerInfo (LayerInfo.prototype.uploadFiles) that detects if the file is an excel.

Figure 4.8: Code that has to be inserted in files of GeoNode Upload module.

tial data.

The WebAPA solves the problems described in section 3.1, related with numerous data files that is necessary to download. In addition, the data is accessed by HTTP, because there are no web services to collect this data. To download the excel file it is necessary to select several inputs. On the other hand, these files do not have information about the station location. So, it is necessary to get the locations of each station and then relate that information with pollutant concentrations from the air quality stations.

WebAPA architecture is illustrated in figure 4.9. The backend part is responsible to get and manage the data. The User Interface allows the user to download the data for all available years.

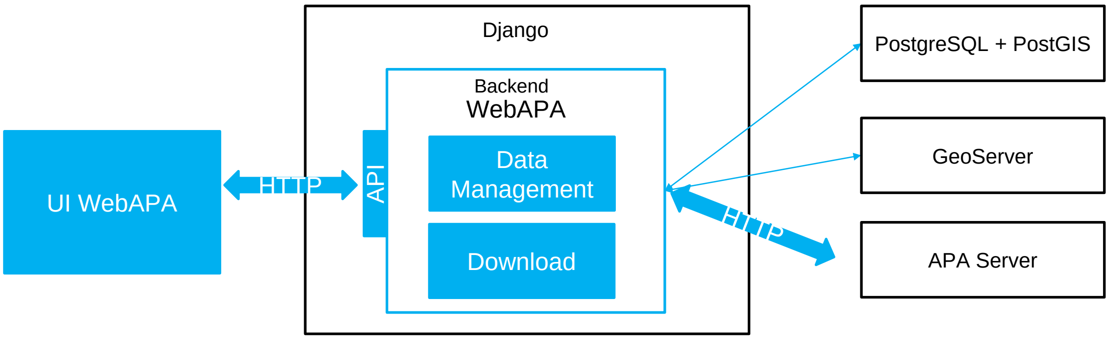


Figure 4.9: WebAPA component in a generic Django framework. Blue boxes correspond to the WebAPA, and the black bounded boxes have been developed by other entities.

The WebAPA component developed during this project is a Django application that can also be used in GeoNode and it is available for public use in a git repository, https://github.com/FMMFHD/WebAPA.

4.5.1 Backend Implmentation

WebAPA is responsible for working with data comprising concentrations of pollutants measured in air quality stations. So the backend has to collect and manage the data. The management is simple because the data is stored in the database and this already helps in the management of the data.

The WebAPA works with three types of data structures, because it is necessary to store the stations information, the observation data and the information data of pollutants. Figure 4.10 represents the relation between the three data structures. Multiple observations can be captured on the same air quality station and can refer to the same pollutant, the E-R diagram (figure 4.10) shows these relations.

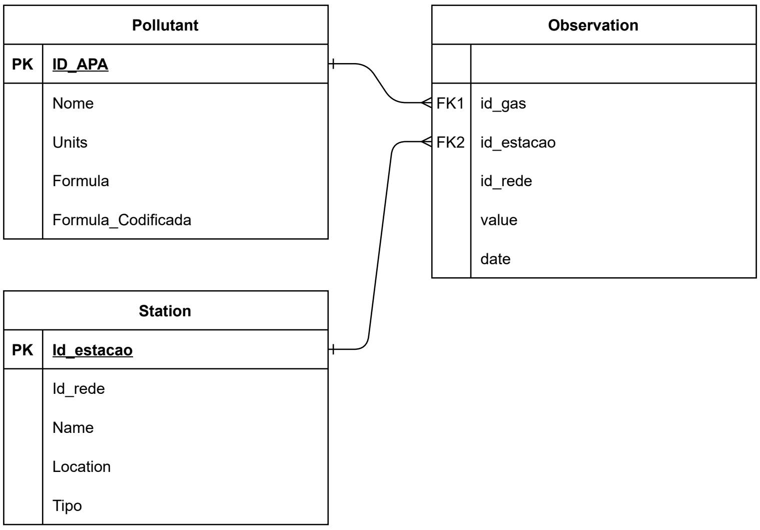


Figure 4.10: Relational model of the WebAPA's data structures.

The station data structure consists of five features:

- Id_rede - Identifier of the network where the stations belong - Integer
- Id_estacao - Identifier of the station - Integer
- Name - Name given to the station - String
- Location - Location of the station with EPSG:4326 as system coordinates - Geographic point
- *Tipo* - Identifies the type of station: background (without significant source of pollution), traffic (located near to source of pollution, vehicles) and industrial (located near to source of pollution, factories) - String

The structure of the data collected from the APA server and stored in the database, is identified by five attributes:

- id_rede - Identifier of the network where the captured station belongs - Integer
- id_estacao - Identifier of the captured station - Integer
- id_gas - Identifier of the pollutant - String
- date - Identifies the date when the value was captured - Timestamp
- value - Value of the captured pollutant - Float

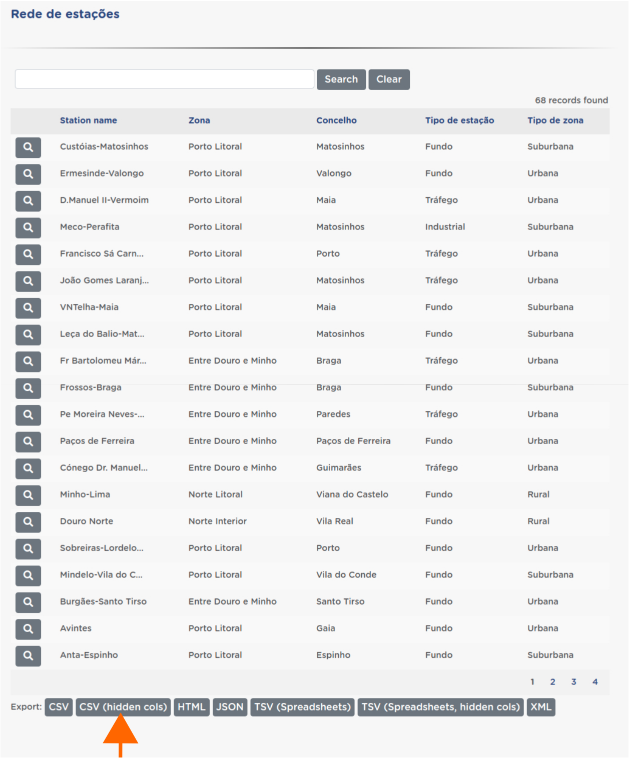
The characterisation of the pollutant has a different data structure:

- *Nome* - Name of the pollutant - String
- Units - The units of the concentrations/depositions - String

- ID_APA - Identifier of the corresponding pollutant for which the APA provides observations from the air quality stations - String
- Formula - Chemical formulation of the pollutant - String
- Formula_codificada - Coded chemical formula of the pollutant - String

Download stations information

The web page, https://qualar.apambiente.pt/qualar/estacoes, provides information from active air quality stations. At the end of this web page, there is a button to export stations' information including hidden information (figure 4.11). So, to automate the process, it is necessary to access the link [64] to export a CSV file that contains information about the active stations.



Rede de estações

Search Clear

68 records found

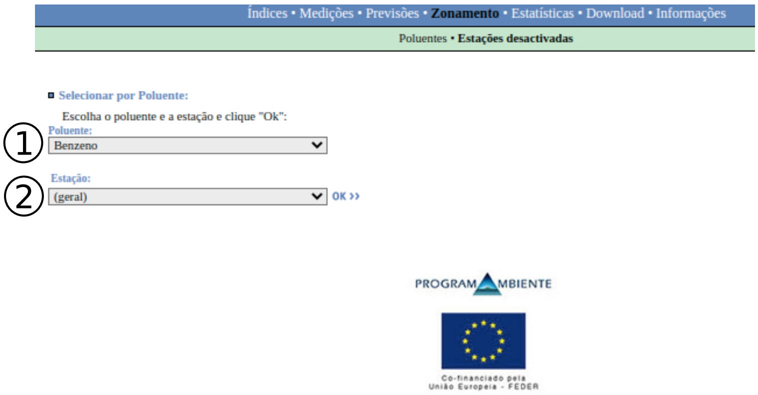
Station name	Zona	Concelho	Tipo de estação	Tipo de zona
Custóias-Matosinhos	Porto Litoral	Matosinhos	Fundo	Suburbana
Ermesinde-Valongo	Porto Litoral	Valongo	Fundo	Urbana
D.Manuel II-Vermoim	Porto Litoral	Maia	Tráfego	Urbana
Meco-Perafita	Porto Litoral	Matosinhos	Industrial	Suburbana
Francisco Sá Carn...	Porto Litoral	Porto	Tráfego	Urbana
João Gomes Laranj...	Porto Litoral	Matosinhos	Tráfego	Urbana
VNTelha-Maia	Porto Litoral	Maia	Fundo	Suburbana
Leça do Balio-Mat...	Porto Litoral	Matosinhos	Fundo	Suburbana
Fr Bartolomeu Már...	Entre Douro e Minho	Braga	Tráfego	Urbana
Frossos-Braga	Entre Douro e Minho	Braga	Fundo	Suburbana
Pe Moreira Neves-...	Entre Douro e Minho	Paredes	Tráfego	Urbana
Paços de Ferreira	Entre Douro e Minho	Paços de Ferreira	Fundo	Urbana
Cónego Dr. Manuel...	Entre Douro e Minho	Guimarães	Tráfego	Urbana
Minho-Lima	Norte Litoral	Viana do Castelo	Fundo	Rural
Douro Norte	Norte Interior	Vila Real	Fundo	Rural
Sobreiras-Lordelo...	Porto Litoral	Porto	Fundo	Urbana
Mindelo-Vila do C...	Porto Litoral	Vila do Conde	Fundo	Suburbana
Burgães-Santo Tirso	Entre Douro e Minho	Santo Tirso	Fundo	Urbana
Avintes	Porto Litoral	Gaia	Fundo	Urbana
Anta-Espinho	Porto Litoral	Espinho	Fundo	Suburbana

1 2 3 4
Export: CSV CSV (hidden cols) HTML JSON TSV (Spreadsheets) TSV (Spreadsheets, hidden cols) XML

Figure 4.11: Web page to access APA's active stations information. The orange arrow indicates the button to export the stations information, https://qualar.apambiente.pt/qualar/estacoes.

The CSV file contains more information than needed. And so, the next step is to create an excel file containing only the information of the data structure of the stations, defined in section 4.5.1. The new excel file contains the name, the identifier and the coordinates of stations and also contains information concerning the type of station (background, traffic and industrial). Having this excel, it is possible to upload the information of the active stations using GeoExcel module.

Unfortunately, the previous method only provides information about currently active stations. To get information regarding deactivated stations, it is necessary to access a different data source, different website. Figure 4.12 shows the initial page to access information about the deactivated stations. This web page has two lists: ① a list of all the available pollutants, and ② a list of deactivated stations that in the past measured the pollutant selected in ①. To facilitate the process, these lists are already inside of the HTML document. So all that is required to obtain all the pollutant-deactivated station combination, is to parse the HTML.


Índices • Medições • Previsões • Zonamento • Estatísticas • Download • Informações
Poluentes • Estações desactivadas
①
②
OK ››
PROGRAMAMBIENTE
Co-financiado pela União Europeia - FEDER

Selecionar por Poluente:

Escolha o poluente e a estação e clique "Ok":

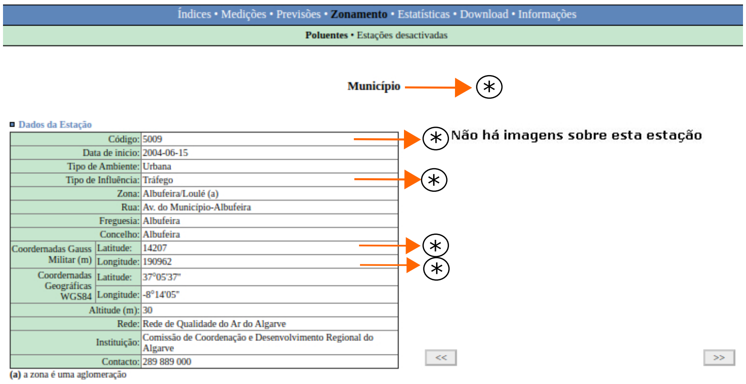
Poluente: Benzeno

Estação: (geral)

Figure 4.12: Web page to access pollutant-deactivated station combinations, https://qualar1.apambiente.pt/qualar/index.php?page=4&subpage=2.

After retrieving all possible combinations, the next step is to obtain relevant information regarding the deactivated stations. For each combination, the GET parameters are already defined, and so it is possible to consult the deactivated station web page, also in HTML (figure 4.13). The relevant information is indicated with an asterisk in figure 4.13.

It is possible to collect the deactivated station information by parsing the HTML document, noting that it is necessary to convert the geographical coordinates from sexagesimal degrees to decimal degrees. This information is added to the excel with the active stations information, and the station is marked as visited to guarantee that it is not collected again, when another combination of pollutant-deactivated station appears.


Índices • Medições • Previsões • Zonamento • Estatísticas • Download • Informações
Poluentes • Estações desactivadas
∗
∗
∗
∗
∗
<<
>>

Município

Não há imagens sobre esta estação

Dados da Estação

Código:		5009
Data de inicio:		2004-06-15
Tipo de Ambiente:		Urbana
Tipo de Influência:		Tráfego
Zona:		Albufeira/Loulé (a)
Rua:		Av. do Município-Albufeira
Freguesia:		Albufeira
Concelho:		Albufeira
Coordernadas Gauss Militar (m)	Latitude:	14207
	Longitude:	190962
Coordernadas Geográficas WGS84	Latitude:	37°05'37"
	Longitude:	-8°14'05"
Altitude (m):		30
Rede:		Rede de Qualidade do Ar do Algarve
Instituição:		Comissão de Coordenação e Desenvolvimento Regional do Algarve
Contacto:		289 889 000

(a) a zona é uma aglomeração

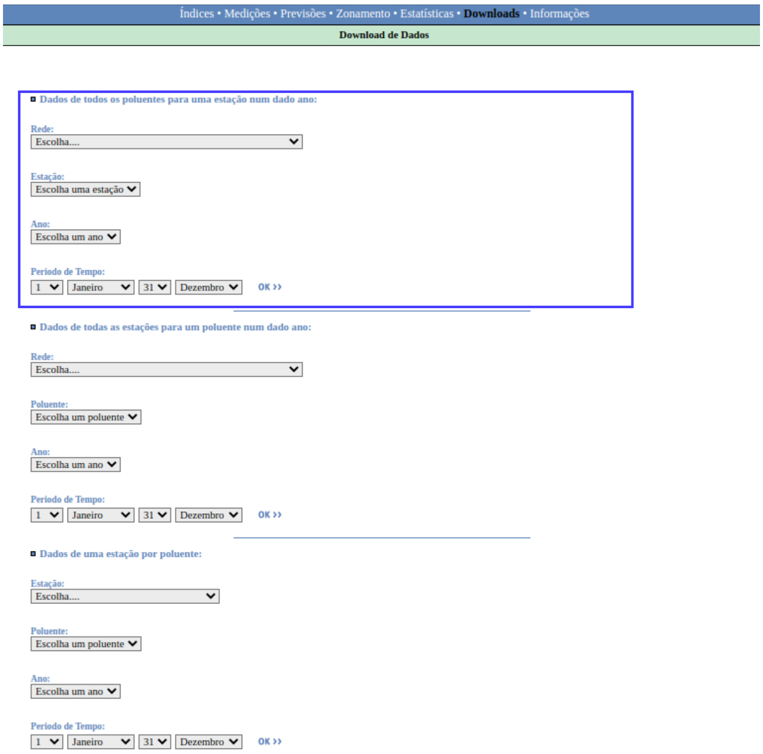
Figure 4.13: Example of information available for the deactivated station, where the fields indicated with an asterisk are relevant information for this work.

After collecting all the available information from the stations, the excel file, containing both activated and deactivated stations, is uploaded to the database using the method from the GeoExcel component, *Algorithm 1*.

Download observation data

Collecting observation data from the air quality stations is not a simple process because the data is not stored in a server with web services. It is important to remember that the APA website has three possible outputs (all pollutants per station per year, a pollutant for all stations per year, one pollutant per station per year) and, in this project, the chosen format includes all pollutants measured in a specific station, during a specific year. The only way to download is to replicate the interactions of the user through HTTP, using web scrapping techniques. Web scraping, web harvesting or web data extraction is a technique used for extracting data from websites.

Since data is not obtained by just clicking on a button in the web page, it is necessary to select a several sequential inputs, as show in figure 4.14. Figure 4.14 shows the layout of the website where observations from air quality stations are downloaded. In this website, it is possible to download in three ways, as mentioned before. In this work, the required inputs include the station network identifier, station identifier, and the year (blue rectangle in figure 4.14).


Índices • Medições • Previsões • Zonamento • Estatísticas • Downloads • Informações
OK ››
OK ››
OK ››

Download de Dados

Dados de todos os poluentes para uma estação num dado ano:

Rede: Escolha....

Estação: Escolha uma estação

Ano: Escolha um ano

Periodo de Tempo: 1 Janeiro 31 Dezembro

Dados de todas as estações para um poluente num dado ano:

Rede: Escolha....

Poluente: Escolha um poluente

Ano: Escolha um ano

Periodo de Tempo: 1 Janeiro 31 Dezembro

Dados de uma estação por poluente:

Estação: Escolha....

Poluente: Escolha um poluente

Ano: Escolha um ano

Periodo de Tempo: 1 Janeiro 31 Dezembro

Figure 4.14: Layout of the website where observations from air quality stations are downloaded [53]. Blue rectangle identifies the inputs used to download an excel file.

However, given that the existing platform has low usability, meaning that final input selection depends on previous selected inputs, it is not possible to predefined all the information required to download the data. As it can be seen in the figure 4.15a, it is not possible to choose a station without selecting a network, being the only input that initial has several options to choose from, (figure 4.15b). So to have access to air pollution data from the air quality stations, a network must be chosen. After choosing a network, for example a network *Lisboa e Vale do Tejo*, it is possible to select a station that belongs to the selected network (figure 4.15c). Therefore, to have access to the year selection, a network and a station must first be selected. When the network, station, and year inputs are selected, all that remains to download the data is pressing the "ok" button.

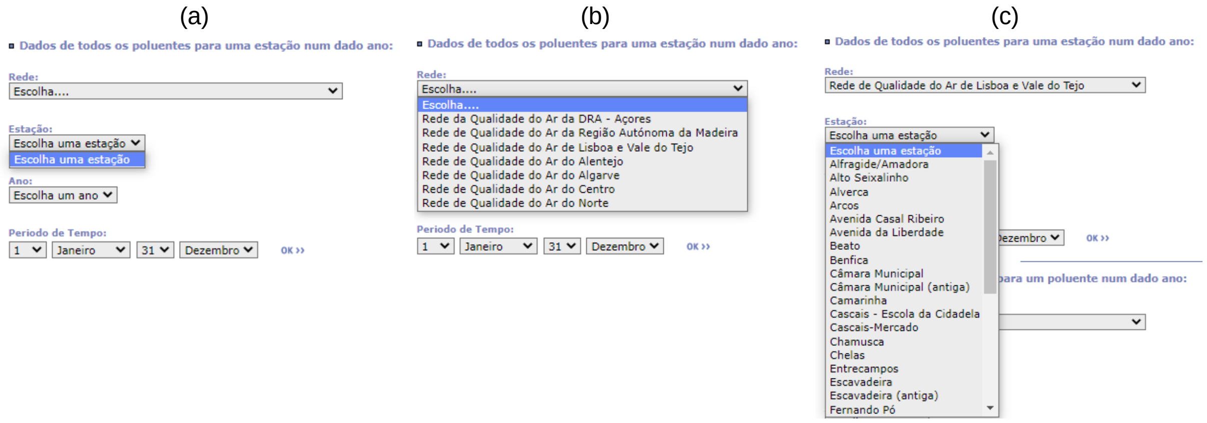

(a)
Dados de todos os poluentes para uma estação num dado ano:
Rede:
Escolha....
Estação:
Escolha uma estação
Escolha uma estação
Ano:
Escolha um ano
Periodo de Tempo:
1
Janeiro
31
Dezembro
OK ››
(b)
Dados de todos os poluentes para uma estação num dado ano:
Rede:
Escolha....
Escolha....
Rede da Qualidade do Ar da DRA - Açores
Rede de Qualidade do Ar da Região Autónoma da Madeira
Rede de Qualidade do Ar de Lisboa e Vale do Tejo
Rede de Qualidade do Ar do Alentejo
Rede de Qualidade do Ar do Algarve
Rede de Qualidade do Ar do Centro
Rede de Qualidade do Ar do Norte
Periodo de Tempo:
1
Janeiro
31
Dezembro
OK ››
(c)
Dados de todos os poluentes para uma estação num dado ano:
Rede:
Rede de Qualidade do Ar de Lisboa e Vale do Tejo
Estação:
Escolha uma estação
Escolha uma estação
Alfragide/Amadora
Alto Seixalinho
Alverca
Arcos
Avenida Casal Ribeiro
Avenida da Liberdade
Beato
Benfica
Câmara Municipal
Câmara Municipal (antiga)
Camarinha
Cascais - Escola da Cidadela
Cascais-Mercado
Chamusca
Chelas
Entrecampos
Escavadeira
Escavadeira (antiga)
Fernando Pó
ezembro
OK ››
ara um poluente num dado ano:

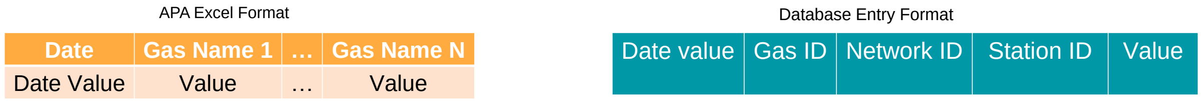
Figure 4.15: Options available for the inputs required to proceed with the download of the excel file [53].

The application has to replicate the steps of a user that wants to download the excel files. Therefore, the procedure to extract data is divided in two steps. The first step is to get all the possible combinations of network, station, and year, and the second step is to download an excel file after choosing the station, the network identifier, and the year.

To access the data without interacting with the web page, it is necessary to have access to the different values of each input. This is possible by extracting the HTML document. After knowing the identifier of the chosen network, the next step is to request via POST to APA web page [53], with a form data. The form data is an object that compiles a set of pair key/values sent in the request. The form data keys are *rede*, *estacao*, *ano*, day0, month0, day1, month1, where the value -1 represents the unknown value.

The response of the request is an HTML document, containing values regarding station input. By replicating this, it is possible to get the different values of the year input for a specific network and station. After all the inputs are chosen, the next step is to send a post to a new URL [65], with the same form data, but with all values different of -1. The response of this post request is an excel spreadsheet. With this excel file, it is possible to apply the *Algorithm 1*, developed for the GeoExcel component, to store the data in the database.

Figure 4.16a shows the structure of the excel files downloaded from the APA web site. In order to optimise queries and data management it is necessary to transform such data into a structure shown in figure 4.16b. The function named `functionInsert` referenced in the *Algorithm 1* is the one which takes the orange table (figure 4.16a) and creates several entries such as the blue one in figure 4.16b.



APA Excel Format

Date	Gas Name 1	...	Gas Name N
Date Value	Value	...	Value

Database Entry Format

Date value	Gas ID	Network ID	Station ID	Value

(a) Structure of an excel file from the APA Server.

(b) Structure of a database table that stores observations.

Figure 4.16: Data structure of Observations before and after the transformation.

The blue table in figure 4.16b represents an entry of the database containing observations. As mentioned before, observations are represented by a date, pollutant id, network id, station id, and concentration. The network id and station id are obtained from the web scraping in the APA website. The columns of the excel provide the pollutant name and, by recurring to an auxiliary table, the pollutant id, that corresponds to the name of the pollutant, is obtained.

To collect all the observations, it is necessary to iterate all the possible inputs combinations. *Algorithm 2* is a pseudo code containing the steps required to download and store the observations.

Algorithm 2: Stores data from the APA server in the database.

```
  Input  : Year to collect the data (year)
  Output: None
1 Function upload_data(year):
2     inputs_information = extract_all_information()
3     for input_info in inputs_information do
4         if year is None then
5             years = extract_year_for_station(input_info)
6             for year in years do
7                 extract_and_save_excel(input_info[0], input_info[1], year)
8         else
9             extract_and_save_excel(input_info[0], input_info[1], year)
```

The first step of the *Algorithm 2* is to aggregate all information of the available networks and stations, using web scrapping, line 2. This information is stored in array, variable named `inputs_information`, that contains all the possible combination of station, network. The cycle `for`, between line 3 and 9, is responsible to go through all possible combinations network/station. In line 4, there is an if and else condition, that verifies if the variable `year` is defined.

If the variable `year` is not defined, the next step, line 5, is to extract all the possible years for the combination network/station. The final step is to iterate all combination network/station/year to extract and store the excel.

If the variable `year` is defined, all the observation data that was measured in the year defined in the variable `year` is downloaded from the APA website (line 9).

The method `extract_and_save_excel` of the *Algorithm 2* is responsible to download and store the excel spreadsheets and it is represented as a pseudo code, *Algorithm 3*. As it was mentioned before, this process uses a method developed for the GeoExcel, *Algorithm 1*. The downloaded excel files have always the same structure, the first column is the date and the remaining columns are the pollutants' concentration, where the pollutant name is in the first line of the column. Thus, only one parser configuration is required.

In the line 2 of the *Algorithm 3*, the form data that is used to extracted the excel file is created. In this form data, the network, station and year are defined. After creating the form data, the next step is to request the excel file, line 3. If the code response of the request is 200, then the final step is to upload the excel file using the method `copy_excel_database` defined in the GeoExcel Library. If there is an error when downloading the file, the status code of the response is different from 200 and there is no

Algorithm 3: Downloads data and stores it.

```
Input  : Identifier of the network(id_network); Identifier of the station(id_station); (year)
Output: None
Function extract_and_save_excel(id_network, id_station, year):
    # Create the form data to do POST request
    form_data = create_formData(id_network, id_station, year)
    response = request_Data_External_Source('POST',
      "https://qualar1.apambiente.pt/qualar/excel_new.php?excel=1", data=form_data)
    if response.status_code == 200 then
        copy_excel_database(excel_path, . . . )
```

uploading.

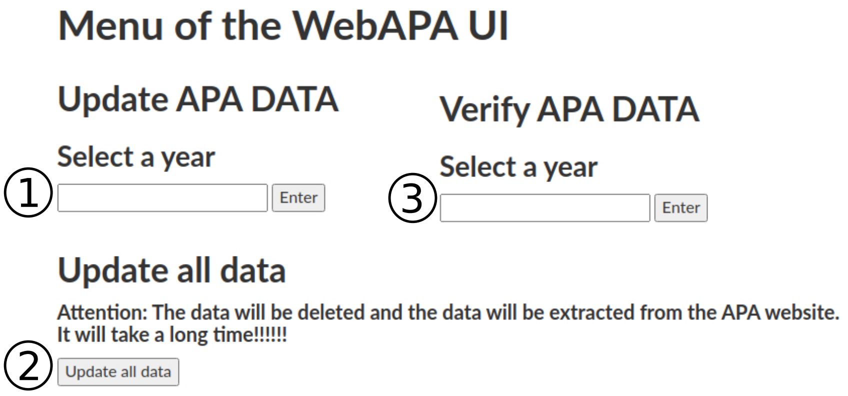
Data Management

As the data is stored in the database, its management is entrusted to the database itself. The data is only replaced when the system administrator deletes all the data for a year or decides from the user interface to remove all the data and download it again.

To access the data, it is only necessary to create SQL statements to filter data that is stored in the database. Having the station data and observations stored in the database, it is easier to obtain the location or the name of the pollutant for specific measurement, because the database offers the opportunity to relate data in different tables.

4.5.2 User Interface

The objective of the user interface is to allow the admin user to select a year to download the data from the APA server (figure 4.17). The user must select the year that will be downloaded and then click the Enter button.



Menu of the WebAPA UI

Update APA DATA

Select a year

Verify APA DATA

Select a year

Update all data

Attention: The data will be deleted and the data will be extracted from the APA website.
It will take a long time!!!!!!

Figure 4.17: User Interface to update observation from air quality data accessed trough the APA server.

If the user writes a number and clicks on button ① in figure 4.17, the system verifies if the number is valid and if data from that year is already in the database. If data is found, then a message appears giving a command line for the system administrator to use to delete the data. If there is no data, it will download all the data for the chosen year.

If the user clicks the button ② in figure 4.17, all data will be deleted and extracted from the server again. This process is time-consuming because it is necessary to download approximately 1000 files.

The communication channel between the interface and the server is not indefinitely open. So, if the time required to download is too long, the channel will close and the user might never receive a success or error message, not knowing whether the execution was a success or not. To confirm that the data has been downloaded, there is an input text and a button (③ in figure 4.17) that allows this verification.

4.5.3 API

API is an interface that the programmer can use to obtain data. With this interface, is possible to access data more easily. This component offers three access points that give information concerning the data available in the server:

- GET .../webapa/listGas
  - parameter: Id_estacao (station identifier).
  - return value type: list of dictionaries that contains two keys: Id - pollutant identifier; Name - pollutant name.
  - List of all the pollutants measured in a given station. It is necessary to filter the database tables to obtain all the pollutants for a given station. This filtering process uses the observations table to obtain the pollutant identifiers per station and the pollutant information table to obtain the name of each pollutant.
- GET .../webapa/GasValues
  - parameters: Id_estacao (station identifier); Id_gas (pollutant identifier); resolution; date.
  - return value type: HTML table and a list of dictionaries that contains two keys: date - observation date; value - observation value.
  - List of all the observations for a given station, pollutant, date, and temporal resolution. It is necessary to filter the database table of observations to get all the observations for a given station, pollutant, date, and temporal resolution to obtain the pollutant values.
- GET .../webapa/all_update
  - parameters: min_year and max_year.
  - return value type: HTTP Response that contains the message of the result of the process of updating content from the APA server.
  - This view allows the system administrator to download observations for the entire time-period available or for a specific given time interval. If the two parameters are defined then the data is collect between the *min_year* and *max_year*. If only one of the parameters is set or none is set, then all content on the APA server is downloaded.

4.5.4 WebAPA Usage

If a programmer wants to write an web application that accesses the observations data from APA server, he/she should follow the next steps to take advantage of the procedures implemented in WebAPA:

- Place the WebAPA module on the same directory of the project. In addition, it is necessary to define the URL for this module, meaning it is necessary to add the code: "url(r’ˆwebapa/’, include(’webapa.urls’))" to the file *urls.py* of the respective project.

- Add the name of the component, WebAPA, to the list of *INSTALLED_APPS* in *settings.py*, so that the project can run the WebAPA module and import methods from this component.

- Change the file *global_settings.py* or replace this file by a similar one, changing the respective *import*. The *global_settings.py* contains the definitions of the global variables and it is located in a general module, named General_modules, that contains common methods to different components. This module is a private repository, but all the methods are commonly used in any project, so it is easy to create the methods and to replace the respective *imports*. In addition, the file *global_settings.py* contains the pre-defined names of the database and the tables.

- Add the GeoExcel library to the same project, because the WebAPA component uses the method `copy_to_database` to upload the excel to the database.

- Must store the information regarding the stations (location, type, etc.) and the pollutant (name, chemical formula, etc.), because they are not automatic uploaded to the database. Inside of this component, there is a file named *script_Upload_excel.py*, which is a python script that uploads a set of excel files to the database, using the method developed by the GeoExcel component, *Algorithm 1*. The script allows to choose between obtaining stations information from an excel or from the available information given by the APA server, as described in section 4.5.1. The available script does not provide the excels. So it is necessary to create them. The script requires the excel files to have the structure illustrated in figure 4.18, where (a) corresponds to the excel containing station information and (b) information on pollutants.

(a)

Id_estacao	name	Lat	Long	Influencia	Id_rede
Station Identifier	Station Name	Latitude Value	Longitude Value	Type of Station	Network Identifier

(b)

Nome	Formula	Formula_codificada	Units	ID_APA
Pollutant Name	Chemical Formulation	Coded Chemical Formulation	Measurement Units	APA Identifier

Figure 4.18: Excel file format for: (a) Station Information, (b) Pollutant Information.

4.6 WebEMEP

The objectives of the WebEMEP component are to store all the available model predictions from the EMEP server in a local database. Besides storage, it offers access mechanisms and data management for a framework where web applications with geospatial data are developed. However, there are some issues concerning the data that contains the EMEP model results. The data that is stored in the EMEP server, is accessed by HTTP through a catalogue. To collect the data it is necessary to consult the catalogue and parser it.

WebEMEP architecture is illustrated in figure 4.19. The backend part is responsible for extracting and management of the data. There is no user interface because all the mechanisms are automatic, including the mechanism to download the data.

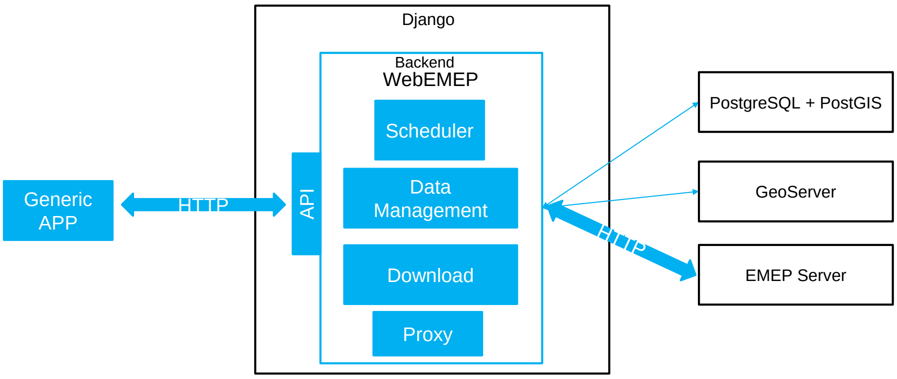


Figure 4.19: WebEMEP component in a generic Django framework. Blue boxes corresponds to the WebEMEP, and the black bounded boxes have been developed by other entities.

The WebEMEP component developed during this project is a Django application that can also be used in GeoNode and it is available for public use in a git repository https://github.com/FMMFHD/WebEMEP.

4.6.1 Backend Implementation

WebEMEP works with EMEP air pollution model results. The backend has to collect and manage the data. Model results are not stored in a conventional way, in a database, because their format does not allow conversion to a table. GeoServer is therefore used to store this data, as it stores files and so there is no need to convert the data.

The EMEP model predictions are divided in years. Each year corresponds to a raster that contains pollutant information. Every raster is clip to the smaller domain of interest, therefore, it is associate to a `clip` class that contains the information of the polygon used to clip. Figure 4.20 shows the Unified Modelling Language (UML) of the structure of EMEP model predictions. The association between the year and the raster is stored locally in JSON format, where given a pollutant and a year, it can identify the raster file.

As for the WebAPA component described in section 4.5, additional information is needed, in this case, pollutant information. This information is stored in the database, and has the following structure:

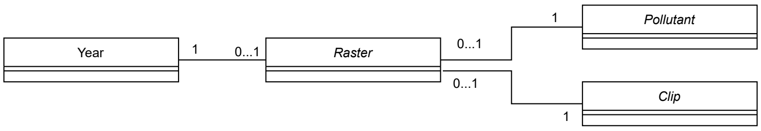


Figure 4.20: Unified Modelling Language (UML) of the EMEP Model predictions' data structures

- *Nome* - Name of the pollutant - String
- Units - The units of the concentrations/depositions - String
- ID_EMEP - Identifier of the pollutant from EMEP model predictions - String
- *Formula* - Chemical formulation of the pollutant - String
- *Formula_codificada* - Coded chemical formula of the pollutant - String

The polygon used to clip the raster files is a vector data that has a shapefile format and is locally stored on GeoServer. This file must have only one entry, that defines the region of interest, meaning that geometry type of the file is `Polygon`.

Download

As mentioned before, the WebEMEP component is responsible for retrieving data from the EMEP site and to make available the information on the platform. Instead of having an interface to update the EMEP data, a task was developed to update the data periodically. The data is available through a proxy.

The Norwegian Meteorological Institute provides a catalogue that contains all the available services, OGC standards, and its URL paths. Also, the catalogue provides the URL path to all the data sets. Having this information stored locally speeds up the update process to the GeoServer. Therefore, the first step is to download the catalogue, which corresponds to a XML file. The catalogue is converted to a dictionary using the XML parser Python library *xmltodict*.

The next step is to download and store the data. The data provided by EMEP represents all of Europe, but there is only interest in a portion of data. The default mode limits the region of interest to Portugal. So the downloaded data has to be cut by using a polygon feature delimiting the relevant spatial domain, which can be altered.

The wider EMEP spatial domain, including all Europe, is cut to the desired size and position using a rectangular shape. After, a mask represented by the polygon feature delimiting the area of interest is used to mask the data resting outside of the polygon feature. After the cut, the data is uploaded to the GeoServer, using the GeoServer API. The *Algorithm 4* is a pseudo code that summarises all the process to store EMEP model predictions in the platform.

The first step of the *Algorithm 4*, line 2, is to download and convert to a dictionary the EMEP catalogue. In line 3, the polygon that will be used to clip, is obtained from the GeoServer. The next step is to create a `mask` that will mask the data resting outside of the polygon feature, making a more detailed cut, line 4. Between line 5 and 8, there is a cycle `for` that will go through all the available data sets to download the EMEP model predictions. The raster file is downloaded from the EMEP serves in line

Algorithm 4: Stores data from the EMEP server in the GeoServer.

```
  Input  : URL path to the EMEP catalogue online (catalogue_url);
  Output: Dictionary with EMEP information(EMEP_Dict)
1 Function Upload_EMEP_DATA(catalogue_url):
2     EMEP_Dict = EMEP_Create_dict(catalogue_url)
3     polygon_path = get_shapefile_format()
4     mask = create_mask(EMEP_Dict['datasets'][0], polygon_path)
5     for dataset in EMEP_Dict['datasets'] do
6         path_dataset = download_FILE(dataset)
7         path_dataset_cut = cut_netcdf(path_dataset, mask, polygon_path)
8         uploadGeoserver(path_dataset_cut)
9     return EMEP_Dict
```

6. After downloading, the raster file is clip using the polygon and the mask and then uploaded to the GeoServer, lines 7 and 8.

The data on the EMEP server is organised by year and resolution. For each year/resolution, the file containing the EMEP model predictions has N layers. When one file is upload to the GeoServer, the N layers become independent files. So the data is organised in workspaces (folders). These workspaces identify the year and the resolution. Thus, the layer is identified with the following name: resolution-year:Layer Name. To access a pollutant with a certain time resolution in a given year, one must only request to the GeoServer the element with the identifier: resolution-year:Pollutant Identifier.

Scheduler

The Norwegian Meteorological Institute updates their catalogue once a year, normally at the same time of the year. In order to guarantee the use of model predictions made with the most recent EMEP model version, WebEMEP includes a scheduler which updates all available data on a day set by the programmer.

The scheduler is created using the Celery, a Python Library that manages task queues. This scheduler is a periodic Task manager, that kicks off tasks at regular intervals. The tasks are then executed by available worker nodes in the cluster. The programmer can choose the date and the frequency of the task, that is responsible to replace all the data that is available on EMEP server.

Data Management

The data stored in the GeoServer is accessed by using the GeoServer API. As previously mentioned, the data that is stored in the GeoServer is accessed with OGC standards. Thus, the management of the data stored in GeoServer is made through the available OGC standards and/or through the user interface made available by GeoServer, only accessed with administrative credentials.

A dictionary is created to know which EMEP models predictions are already downloaded from the EMEP server and the respective identifiers in the GeoServer. It is also used to access the raster file stored in the GeoServer when the year and the pollutant are known. The dictionary is saved in JSON format and is also used in the proxy, when the request data is EMEP model predictions data. When it is necessary to consult the dictionary, the first step is to copy it to a local variable by converting JSON to the Python Mapping Type (`dict`). This dictionary is only access by code. The dictionary is created/updated when the system starts to upload the data from the EMEP server. Thus, the catalogue information is stored in this dictionary, which contains the following information:

- services - All the sub-paths for the available web services of the EMEP server;
- resolutions - All the possible temporal resolutions that the EMEP server provides (year, month, day, hour);
- datasets - For each resolution/year indicates the sub-path to the element stored in the EMEP site and indicates the workspace where the element is stored on GeoServer;
- ListGases - All the pollutants available in EMEP model predictions for each temporal resolution;
- ListDates - All possible dates for which modelled predicted concentrations/depositions exist. These dates only include month, day, hour, and second because the dates are the same regardless of the year in which the data is shown, for example, in the dashboard;
- max_min - Maximum and minimum value for each pollutant concentration/deposition in each resolution. These values are used in the style of the generate a map of the pollutant.

The GeoServer is a private system. Thus, for security reasons, there should be no direct access from the outside. Internally, to obtain the data, it is only required to request it using a URL that obeys the OGC standards. Therefore, internally the access is similar to the database, but instead of executing SQL queries, the access is simply request from a local base URL where data restrictions are in the query string part of the URL.

4.6.2 API

The only available interface uses the proxy to redirect to the GeoServer, so that, outside users can have indirect access to the GeoServer. This proxy (see figure 4.21) is responsible to forward the requests to the GeoServer or to the EMEP website when the GeoServer is down and the requests respect the OGC standards. The EMEP website, in addition to allowing the download, also allows to consult only a part of the data through OGC standards.

The proxy will forward the requests to the EMEP website to ensure that outsider users have always EMEP model predictions even when the GeoServer is down. The forward process comprises only a change of URL of the request and to adjust the query string to match the receiving system's OGC standards. The services available on GeoServer and EMEP website are: GetMap; GetCapabilities; GetLegendGraphic; GetFeatureInfo. The proxy has another operation, named `GeneralAccess`, that gives access to other data stored on GeoServer.

This component offers one access points that give information regarding the EMEP model predictions and other data that is stored in GeoServer:

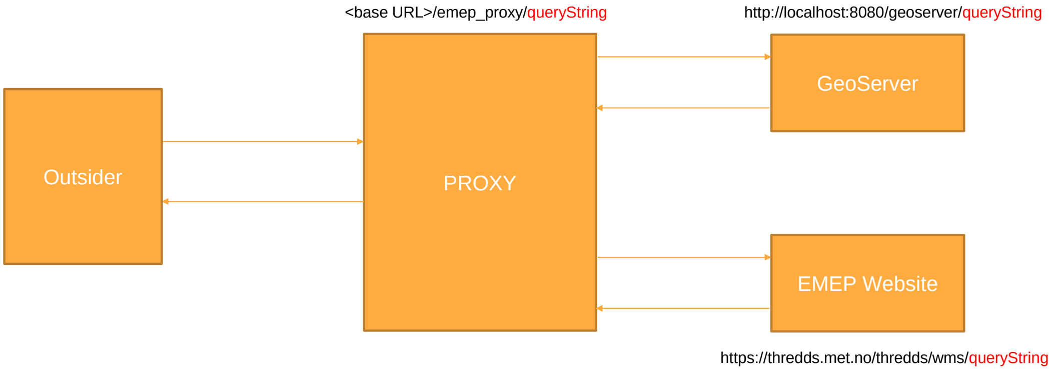

<base URL>/emep_proxy/queryString
http://localhost:8080/geoserver/queryString
Outsider
PROXY
GeoServer
EMEP Website
https://thredds.met.no/thredds/wms/queryString

Figure 4.21: Schematic of the EMEP proxy.

- GET .../webemep/emep_proxy/<information>
  - parameter: queryString.
  - return value type: HTTP Response that contains the response content.
  - Redirect the request to the appropriate spatial data server. If <information> has a value of 'geoserver_General' then the request is redirect to the GeoServer, because the outside user wants to access data that is not the EMEP model predictions. If <information> has value of temporal resolution (year, month), it means that the outside user wants to access to the EMEP model predictions, and the request is redirect to the GeoServer. If the GeoServer is down, the request is redirect to EMEP website. The queryString must have a format similar to "?&service=①&request=②&③", where ① is the type of OGC standard (WMS, WFS, WCS); ② is the name of the operation that is associated with the chosen OGC standard; ③ is the remaining parameters that belong to the chosen operation. For example, the URL to returns a list of the available layers in the GeoServer is .../webemep/emep_proxy/year?&service=WMS&request=GetCapabilities&version=1.3.0".

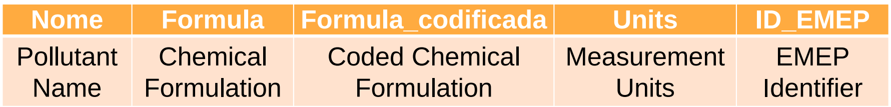
4.6.3 WebEMEP Usage

If a programmer wants to write an web application using GeoServer that accesses the EMEP model predictions from EMEP server, he/she should follow the next steps to take advantage of the procedures implemented in WebEMEP:

- Place the WebEMEP module on the same directory of the project. In addition, it is necessary to define the URL for this module, meaning it is necessary to add the code: "url(r'^webemep/', include('webemep.urls'))" to the file *urls.py* of the respective project.
- Add the name of the component, WebEMEP, to the list of *INSTALLED_APPS* in *settings.py*, so that the project can run the WebEMEP module and import methods from this component.
- Change the file *global_settings.py* or replace this file by a similar one, changing the respective *import*. The *global_settings.py* contains the definitions of the global variables and it is located in

a general module, named General_modules, that contains common methods to different components. This module is a private repository, but all the methods are commonly used in any project, so it is easy to create the methods and to replace the respective *imports*.

- Must store pollutant information, because it is not automatically uploaded to the database. Inside of this component, there is a file named *script_Upload_excel.py*, which is a python script that uploads a set of excel files to the database, using the method developed by the GeoExcel component, *Algorithm 1*. The available script does not have access to the excel file that contains depositions information (units, name, chemical formula, etc). So it is necessary to create them. The script receives the excel file with the structure illustrated in figure 4.22.



Nome	Formula	Formula_codificada	Units	ID_EMEP
Pollutant Name	Chemical Formulation	Coded Chemical Formulation	Measurement Units	EMEP Identifier

Figure 4.22: Excel file format for Pollutant Information.

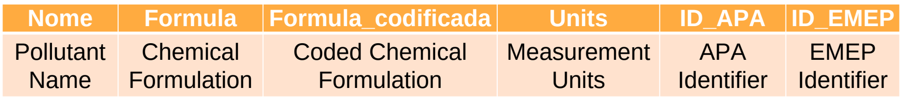
The EMEP model predictions can be accessed using the developed proxy. However, to access to the data internally is required to create the URL for accessing the content, asking directly to the GeoServer without using the proxy. The objective of the proxy is to facilitate the requests made by the developed web applications that need the maps generated from EMEP model predictions, and to secure that the credentials to the GeoServer, that can be access through a public URL, are secured.

4.7 Observations and Modelling data integration

Although EMEP Model predictions and Observations represent different information on air pollution, they have information on the same pollutant. However, there is no easy way to relate the content of the EMEP model to the observations, making it a difficult process to obtain results from the merging of these two types of data. As can be seen in sections 4.5.1 and 4.6.1, the information on pollutants is similar in both types of data, causing repetition of information if WebAPA and WebEMEP libraries are used at the same time.

The difficulties in relating the two types of data have to do with their identifiers being distinct. For example, in the case of the pollutant Ozone, the identifier on the observations side is APA_7 and on the EMEP model side is SURF_ppb_O3.

The solution is to create a table containing the two identifiers as well as the relative information of each pollutant. Figure 4.23 represents the structure the table should have.



Nome	Formula	Formula_codificada	Units	ID_APA	ID_EMEP
Pollutant Name	Chemical Formulation	Coded Chemical Formulation	Measurement Units	APA Identifier	EMEP Identifier

Figure 4.23: Table format for the integration of Observations and EMEP Model predictions information.

While observations are only on pollutant concentrations, in the case of the EMEP model, the same pollutant may have information on deposition and concentration. As mentioned before, concentrations cannot be compared with deposition, so a researcher is needed to verify that this problem does not exist.

This table can be uploaded to the database using the GeoExcel library. As this table is in the database, WebAPA and WebEMEP libraries can use this table, thus ensuring that there is no replication of information.

4.8 GeoNames

The objectives of the GeoNames component is to facilitate the conversion between toponyms and coordinates. Besides storage, it offers access mechanisms for a framework where web applications with geospatial data are developed. GeoNames architecture is illustrated in figure 4.24. The backend part is responsible for data download and for managing the access to the data.

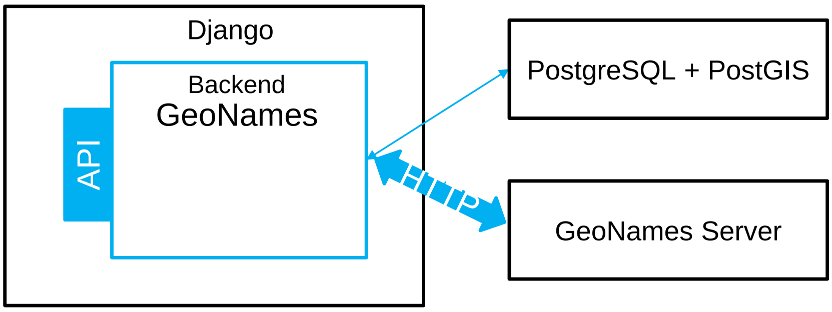


Figure 4.24: GeoNames component in a generic Django framework. Blue boxes corresponds to the GeoNames, and the black bounded boxes have been developed by other entities.

GeoNames is a geographical database that covers all countries and contains over eleven million toponyms. For each toponym it gives information about: latitude, longitude, population, etc.

The GeoNames component developed during this project is a Django application that can also be used in GeoNode and it is available for public use in a git repository
https://github.com/FMMFHD/GeoNames.

4.8.1 Backend Implementation

The backend must collect and manage the data. GeoName data is stored in the database. Basically, it is only necessary to download the zip file and copy the content to the database. A script file was created, named *update_GEONAME.sh*, that uploads the data to the database.

The first step is to the download the zip file which contains a text file already configured to work with the database parser. So it is only necessary to create the table using the *readme* of the Geonames Server [66]. On this *readme.txt*, there is a configuration of the `geoname` table that is recommended to store the data. The `geoname` table has 19 attributes, but the most important are:

- geonameid - integer id of record in geonames database;

- name - name of geographical point varchar(200);
- latitude - latitude in decimal degrees (wgs84);
- longitude - longitude in decimal degrees (wgs84);

The data is copied using the function copy of the PostgreSQL. A column with datatype Point is added to facilitate the management of georeferenced data, the table now has 20 attributes. This new column will be a junction of latitude and longitude values of their respective columns. With this new column, it is possible to use the georeferenced methods provided by the PostGIS plugin. The data is more easily accessible using SQL statements because is stored in a local database.

4.8.2 API

The API offers two access points that give information regarding the toponyms and its coordinates:

- GET .../geonames/autocomplete
  - query parameter: str_name (the sub-string that the toponyms can have).
  - return value type: an array with the possible toponyms.
  - This feature will suggest toponyms that start with the same sub-string, meaning it offers the possibility to auto complete sub-strings of toponyms. It is necessary to filter the database table of GeoNames to obtain all the possible typonoms that starts with the same string.
- GET .../geonames/location
  - query parameter: toponym.
  - return value type: an array with the all the possible coordinates.
  - This feature obtain all possible locations (several places may have the same name) for a given toponym. It is necessary to filter the database table of GeoNames to obtain all the locations for a given toponym.

4.8.3 GeoNames Usage

If a programmer wants to write an web application that needs a conversion between toponyms and coordinates, he/she should follow the next steps to take advantage of the procedures implemented in GeoNames:

- Place the GeoNames module on the same directory of the project. In addition, it is necessary to define the URL for this module, meaning it is necessary to add the code: "url(r’ˆgeonames/’, include(’geonames.urls’))" to the file *urls.py* of the respective project.
- Add the name of the component, GeoNames, to the list of *INSTALLED_APPS* in *settings.py*, so that the project can run the GeoNames module and import methods from this component.

- Change the file *global_settings.py* or replace this file by a similar one, changing the respective *import*. The *global_settings.py* contains the definitions of the global variables and it is located in a general module, named General_modules, that contains common methods to different components. This module is a private repository, but all the methods are commonly used in any project, so it is easy to create the methods and to replace the respective *imports*.

Chapter 5

Demonstration Applications

This chapter presents the applications to validate the libraries referred in the previous chapter. The objectives of this applications are to help researchers to access air pollution data (observations and modelling) and evaluate the EMEP model.

The applications developed were: Concentration Dashboard; Deposition Dashboard; and Evaluation of the EMEP model. In this chapter, topics such as the structure of the applications, their functioning and results obtained and their relevance for environmental technicians and researchers, will be addressed.

5.1 Functional Requirements

The applications were developed with the specific aim to demonstrate results in mainland Portugal. For this reason, all data was clipped and masked with a vector file uploaded to GeoServer, containing the region of interest which includes a 10km buffer around mainland Portugal and also the Portuguese margin up to a distance of approximately 400 km to the shore (figure 5.1). For example, for the EMEP model predictions, *Algorithm 4 in section 4.6.1*, the mask represented in figure 5.1 was used.

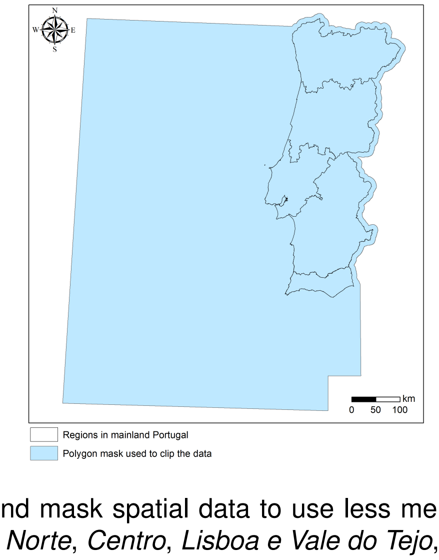

N
W
E
S
km
0 50 100
□ Regions in mainland Portugal
▨ Polygon mask used to clip the data

Figure 5.1: Mask used to clip and mask spatial data to use less memory. The 5 regions in mainland Portugal are also represented: : *Norte*, *Centro*, *Lisboa e Vale do Tejo*, *Alentejo*, and *Algarve*.

The objectives of each application are:

- Evaluation of the EMEP model - Evaluates the EMEP model using observations from air quality stations made available by the APA. The evaluation is available in graph form and can be downloaded as an excel file.
- Concentration Dashboard - Includes data sets from both observations (measured in air quality stations), and EMEP model prediction of concentrations of pollutants. In addition, land occupation/use classification (COS) is provided as an optional cartographic base. Furthermore, and exclusively for ammonia (NH_3), nitrogen oxides (NO_X), and sulphur dioxide (SO_2), an optional representation of the spatial distribution of critical level exceedances is also provided.
- Deposition Dashboard - Includes only the EMEP model predictions of wet, dry and total deposition of nitrogen (oxidised and reduced forms) and sulphur. In addition, ecosystem classification (MAES) is provided as an optional cartographic base. Furthermore, nitrogen critical load exceedances are also provided as an optional spatial representation, indicating whether the ecosystems are being affected by the amount of nitrogen being deposited, or not.

The functional requirements indicate what was decided to be included in the application based on the needs of the final users. The users of these applications are environmental researchers/decision makers that study the variation of the pollution on the environment. If these requirements are all implemented, the applications will show results that facilitate the researchers' studies.

The functional requirements for the three applications are as follows:

- H - Evaluation of the EMEP model:

  H1. The application should integrate data from the EMEP server (EMEP model predictions).

  H2. The application should integrate data from the APA server (observations).

  H3. The application should overlap the observations with EMEP model predictions and extract the concentrations in the pixels that overlap the air quality stations.

  H4. The application should use statistical formulas defined by Chang & Hanna [6] to evaluate the EMEP model.

  H5. The application should produce a bar chart using the statistical information.

  H6. The application should allow to download the statistical information.

  H7. The application should produce scatter plots where observed values are plotted against model. predictions. Furthermore, $x = y$,$x = 2y$ and $x = \frac{1}{2}y$ lines are also added to facilitate interpretations.

- D - Dashboards:

  D1. The application should provide spatial data with territorial limitation (mainland Portugal).

  D2. The application should present results defined by time resolution and by a date.

D2.1. The application should present results with annual and monthly resolution

D3. The application should integrate a search based on a location name.

D4. The application should integrate EMEP model Predictions.

D5. The application should show the temporal variation of EMEP Model Predictions.

D5.1. The application should compute the maximum and minimum value, and weighted mean of all the values contained in each region of interest (regions illustrated in figure 5.1).

D6. The application should display the information in a map.

D7. The application should display a popup, after clicking on the map.

D7.1. The application should display the coordinates.

D8. The application should allow to download the popup information.

D9. The application should allow to download the time variation of EMEP Model Predictions.

- CD - Concentration Dashboard:

CD1. The application should show pollutant concentration.

CD2. The application should present information on pollutants from two different sources:

CD2.1. The application should present pollutant information from APA server.

CD2.2. The application should present pollutant information from EMEP server.

CD2.3. The application should present pollutants that are available on both sources.

CD3. The application should integrate information on the land occupation/use.

CD4. The application should integrate information on the critical levels of some pollutants (Nitrogen Oxide - NO_X, Sulphur Dioxide - SO_2, Ammonia - NH_3).

CD5. The application should show information about the stations.

CD6. The application should show the temporal variation of observations data.

CD6.1. The application should compute the average of concentration measured by date and time resolution, including the standard deviation determined for each station.

- DD - Deposition Dashboard:

DD1. The application should show pollutant deposition.

DD2. The application should integrate information on the type of ecosystem.

DD3. The application should integrate information on Nitrogen critical loads.

5.2 Data Querying and Aggregation

The results presented in the applications require some data processing, such as determining temporal averages of observations according to the desired temporal resolution or regionally weighted averages for EMEP model predictions for the main Portuguese regions (figure 5.1). This data processing required the extension of the WebAPA and WebEMEP components to include basic statistical data processing to aggregate temporal and spatial data, frequently used in geo-temporal data analysis.

In addition to the libraries referenced in Chapter 4, the following modules have been developed to help the development of the applications described in this chapter:

- ADMIN_GRAPHICS - This module has an interface that allows to manage the content saved on the caches, namely statistics used in model evaluation and temporal variation of EMEP model predictions by region/resolution, as well as temporal variation of observations from air quality stations, by resolution (year, month);
- APA_EMEP - Compares model predictions with observations. Observations comprise a set of points, where each point represents an air quality station. The value of each pixel from EMEP predictions overlapping background air quality stations is extracted. With these two data sets it is possible to evaluate model performance based on a set of statistical measures described below;
- Dashboard - The applications Dashboards (interface) are in this module, together with the list of regions and an endpoint that returns the limit of each region;
- General_modules - Modules that contain all the common methods to all the developed libraries / modules. Also, the general definitions of the platform are in this model. As each module is associated with the General_modules, it is possible to define different states inside of the same system. This module is also used by the libraries described in Chapter 4;

In addition to these new modules, batch files were also developed to assist in the initialisation of the environment where the platform runs. These batch files are responsible to create the necessary tables in the database and also to upload all the necessary data to the GeoServer and PostgreSQL.

Figure 5.2 shows the architecture with the developed libraries described in Chapter 4 and the modules mentioned above. The blue boxes were developed in this work, and the black bounded boxes are developed by other entities.

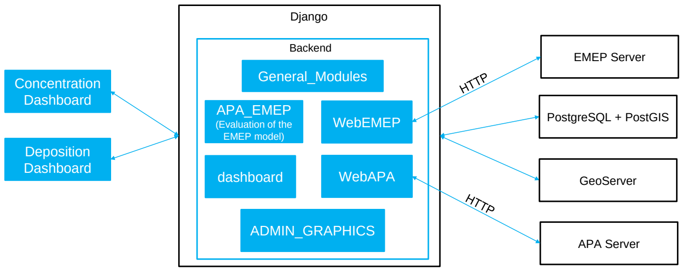


Figure 5.2: Architecture with the various components developed. Blue boxes were developed in this work, and the black bounded boxes have been developed by other entities.

5.3 Evaluation of the EMEP model

One of the defined requirements was to evaluate the EMEP model. In this work, evaluation can be shown for mainland Portugal or for each of the five regions, based on model predictions and observations falling within the polygon of interest.

So, the first step is to select the background air quality stations that fall within the selected polygon. Furthermore, from the restricted set of stations, for a given date and temporal resolution, only stations that captured more than 75% of samples of the predicted number of captured samples (one sample per hour) are used. Due to these restrictions in spatial and attribute-based queries, there will be a limited number of stations to use. Each station is, then, used to extract EMEP model predictions.

The evaluation of the EMEP model is based on the following statistical measures recommend by Chang & Hanna [6, 67]: fraction of predictions within a factor of two of observations (FAC2), fractional bias (FB) and the normalised mean square error (NMSE).

$$FAC2 \ = \ fraction\, of\, data\, that\, satisfy \quad 0.5 \ \leq \ \frac{M}{O} \ \leq \ 2.0 \tag{5.1}$$

$$FB \ = \ \frac{(\bar{O} \ - \ \bar{M})}{0.5(\bar{O} \ - \ \bar{M})} \tag{5.2}$$

$$NMSE \ = \ \frac{1}{N} \sum \frac{(O \ - \ M)^2}{\bar{O} \cdot \bar{M}} \tag{5.3}$$

Where M represents model predictions, O represents observations and overbar ($\bar{O}$ and $\bar{M}$) represents average over the data set. According to the authors [67], a model performs adequately or has "acceptable" performance when FAC2 $\geq$ 50%, FB $\leq$ 30% and NMSE $\leq$ 1.5. However, a model should not be excluded if one of the statistical measures fails to reach the indicative thresholds. For this reason Hanna and Chang [68] suggest a comprehensive acceptance criterion of 50%, meaning at least 50% of these performance criteria are met, which means that two out of three of statistical measures should be within the indicative thresholds. The *Algorithm 5* is a pseudo-code of the steps required to evaluate the data.

In the first four lines of the *Algorithm 5*, it is checked whether the result of the evaluation already exists for the given input parameters. If there are no results, then the first step is to select the background stations that belong to the given polygon as an input parameter, line 7. The evaluation covers every year where there is data, cycle for (between line 8 and 14). On line 9, the raster file is obtained from GeoServer. For each date that exists EMEP model predictions, observations are obtained and the pixels that contain the selected stations are determined (line 12). Having the observations and the respective pixels of the EMEP model predictions, the model can be evaluated (line 13). After evaluating for all years, the result is stored in memory (between line 15 and 16).

The result of the model evaluation is saved in the dictionary. This dictionary also contains the initial conditions (region, resolution, and pollutant) to ensure that, upon receiving the answer, the requester knows which initial conditions this evaluation corresponds to. This application can take from 10 minutes up to 1 hour to compute, depending on the size of the selected region. Therefore, the system saves the processed results in permanent and cache memory. The permanent memory is used to safeguard the data in case the system goes down, and the cache memory is used to decrease the response time. The

Algorithm 5: Steps to validate the EMEP data using APA data.

```
Input  : Region's Name (polygon_Name); Temporal resolution (resolution); Identifier of the Gas
         in the EMEP database (gasEMEP); Identifier of the Gas in the APA database (gasAPA)
Output: Dictionary with the initial conditions and the results of the validation(cacheData)
Function validate_EMEP(polygon_Name, resolution, gasEMEP, gasAPA):
    key = . . .
    cacheData = get_cache_Value(key)
    if cacheData is Null then
        FB = [] FAC2 = [] NMSE = [] Criteria = [] eval_data_unit_time = []
        dict_emep = get_Dict_EMEP()
        background_stations = select_background_stations(polygon_Name)
        for year in dict_emep['datasets'][resolution] do
            dataEMEP_path = getEMEPData(gasEMEP, resolution, year, . . . )
            dates = get_list_dates(resolution, dict_emep, year)
            for date in dates do
                eval_data_Oi, eval_data_MR = get_EMEP_APA_Data_station(resolution,
                  background_stations, gasAPA, gasEMEP, date, dataEMEP_path)
                compute_validation(eval_data_Oi, eval_data_MR, FB, FAC2, NMSE, Criteria)
                eval_data_unit_time.push(date)
        cacheData = save_values(polygon_Name, resolution, gasEMEP, gasAPA, FB, FAC2,
          NMSE, Criteria, eval_data_unit_time)
        set_cache_Value(key, cacheData)
    return cacheData
```

results are updated using an admin page (section 5.5), where it is possible to delete or update.

This application returns a dictionary of arrays, that are extremely relevant to users and must be shown in the Dashboard (described below, in section 5.4), and represented in a bar chart, as shown in figure 5.3. In this chart, the x axis corresponds to time axis, the bars have always the same height, and values are represented in bar colour. Each colour represents the number of statistical measures found within given thresholds. Green symbolises the best possible model performance, with all three criteria verified. Yellow indicates that the model is fit for purpose, with two out of three criteria verified. Orange indicates that the model does not perform well, but one criterion was within thresholds. The colour red means that the model does not represent the reality, all the performance criteria being outside of the acceptable limits defined by Chang & Hanna [67]. The orange box appears when the user mouses over the bar and contains the results of each criterion for a given bar.

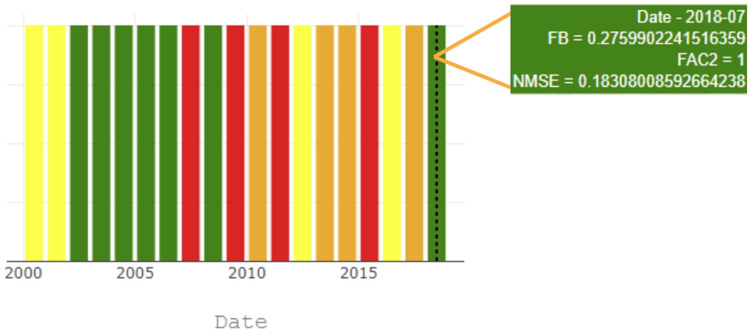


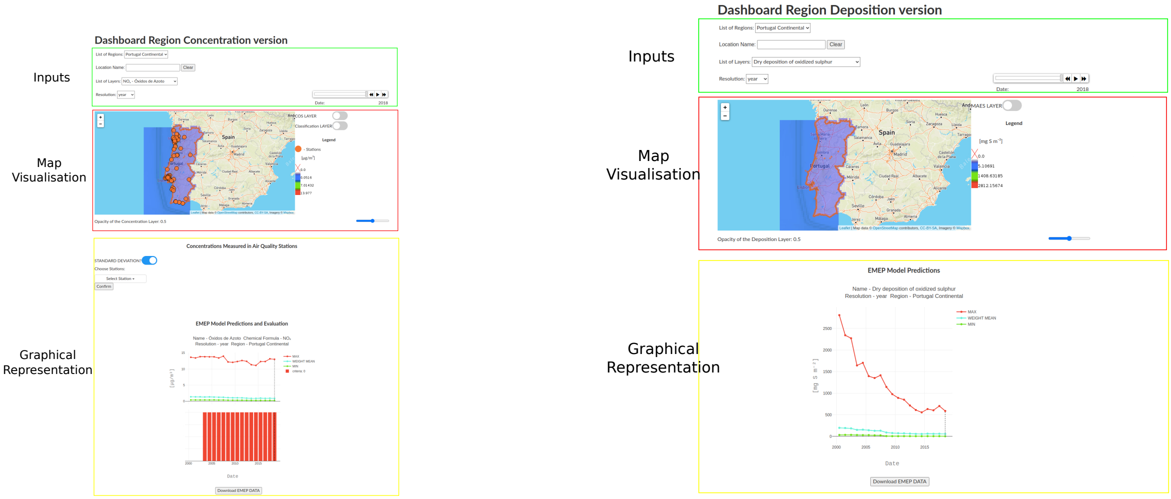
Figure 5.3: Example from the Concentration Dashboard showing how model evaluation is presented.

This application meets some requirements described in section 5.1. EMEP model predictions and observations are integrated and to evaluate the model the data is overlapped. The evaluation is done with criteria defined by Chang & Hanna and the information is displayed through a bar chart.

5.4 Dashboards

The Dashboards are an interactive web pages that gather and show official air pollution information for mainland Portugal.

The web pages are divided in three sections: inputs, map visualisation and graphical representation (figure 5.4) and was optimised to use with the Google Chrome browser.


Dashboard Region Concentration version
Inputs
Map Visualisation
Graphical Representation
Concentrations Measured in Air Quality Stations
EMEP Model Predictions and Evaluation
Download EMEP DATA
Dashboard Region Deposition version
Inputs
Map Visualisation
Graphical Representation
EMEP Model Predictions
Date
Download EMEP DATA

(a) Concentration Dashboard.

(b) Deposition Dashboard.

Figure 5.4: Dashboards Layout.

Two Dashboards were developed, differing in the data presented, the Concentration Dashboard shows data concerning modelled and measured concentrations of pollutants in the air, the Deposition Dashboard shows only data concerning only modelled nitrogen and sulphur deposition.

It is only possible to access graphical representation of data when when the user selects parameters in the input section, that is located at the top of the Dashboards. Figure 5.5 shows the layout of the 5 inputs in the Dashboards.

1. List of Regions - Comprises a *select list*, allowing to select between mainland Portugal or one of the main 5 regions in Portugal (*Norte*, *Centro*, *Lisboa e Vale do Tejo*, *Alentejo*, and *Algarve*);

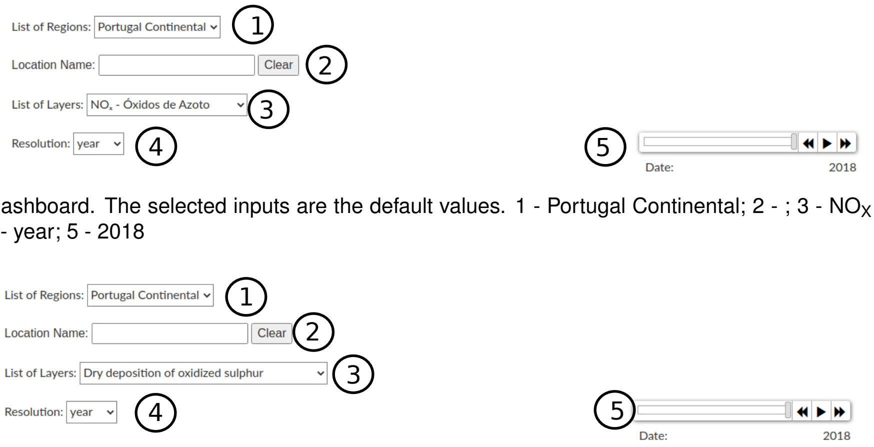
2. Location Name - An input text that allows to type a location, or toponym;

3. List of Layers - Comprises a *select list*, where the user can choose the layer of interest. In the Concentration Dashboard, the layers include only concentrations of pollutants that exist in both APA and EMEP databases. In the Deposition Dashboard, only predicted deposition layers extracted from the EMEP model and spatial representation of nitrogen critical load exceedances, are shown;

4. Resolution - Comprises a *select list* where the user can choose the temporal resolution. In another words, the user can choose how the data is presented, aggregated in years or in months;

5. Date Selection - Comprises a slider, where the user can choose the date of interest. This slider only works when the concentration/deposition layer appears in the map visualisation section, meaning when the user chooses valid values for the ①, ③, ④ inputs in figure 5.5;

These inputs are always visible in the web page even when scrolling down the page. With this feature, when the user wants to check which inputs chosen when viewing the graphical representation section, there is no need to scroll up.



List of Regions: Portugal Continental

Location Name: Clear

List of Layers: NO_x - Óxidos de Azoto

Resolution: year

Date: 2018

(a) Concentration Dashboard. The selected inputs are the default values. 1 - Portugal Continental; 2 - ; 3 - NO_X *Óxidos de Azoto*; 4 - year; 5 - 2018

List of Regions: Portugal Continental

Location Name: Clear

List of Layers: Dry deposition of oxidized sulphur

Resolution: year

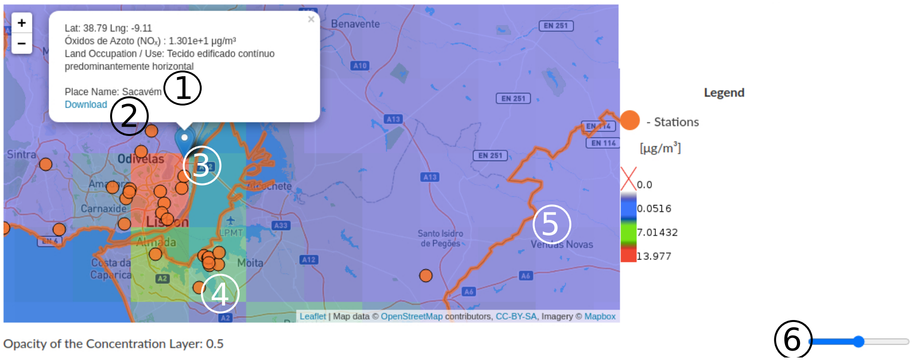
Date: 2018

(b) Deposition Dashboard. The selected inputs are the default values. 1 - Portugal Continental; 2 - ; 3 - Dry deposition of oxidised sulphur; 4 - year; 5 - 2018

Figure 5.5: Layout of the inputs in the Dashboards.

The next section in the Dashboards is the map visualisation. The Mapbox API [69] is default cartographic base that uses OpenStreetMap [70] as data source and shows physical elements in the landscape, such as terrain (using hill-shade) and water bodies, it identifies natural parks, and also represents man-made structures such as roads, subway stations, etc. The detail of the cartographic information shown increases with map zoom. The map visualisation section has several other features, depending on the dashboard.

EMEP model predictions are presented as a layer, allowing the representation of the model predictions across the region of interest in an intuitive way, using a colour scheme, with red representing higher values and blue lower values, as shown in the legend (figure 5.6).



Lat: 38.79 Lng: -9.11
Óxidos de Azoto (NO_X) : 1.301e+1 µg/m³
Land Occupation / Use: Tecido edificado contínuo predominantemente horizontal

Place Name: Sacavém
Download

Legend

● - Stations

[µg/m³]

0.0
0.0516
7.01432
13.977

Opacity of the Concentration Layer: 0.5

Figure 5.6: Layout of the map section in the web page Dashboard. Opacity level is 0.5 (default level).

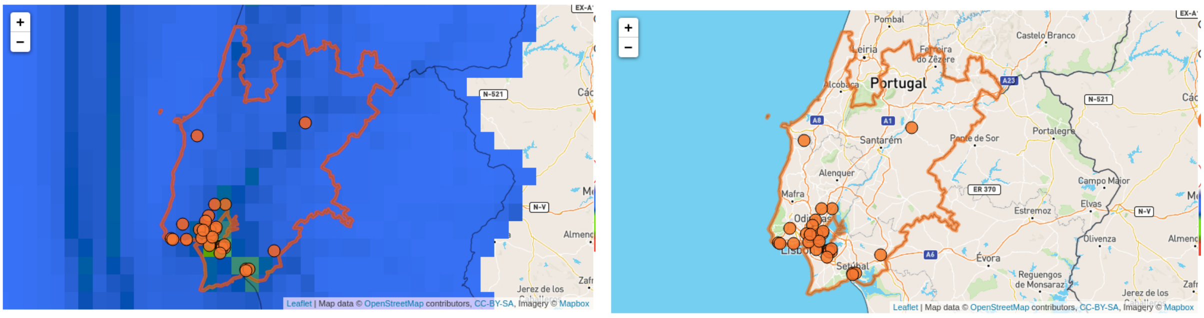
If a specific region is selected (instead of the default value-mainland Portugal), an orange line appears delimiting the selected region, ⑤ in figure 5.6.

The orange circles in the map (④ in figure 5.6) represent the air quality stations. When a region is chosen in the input section, only stations within that region appear. This information only appears on the Concentration Dashboard.

By clicking on the map, or, alternatively, by typing a location in the input section (② in figure 5.5), a blue marker (③ in figure 5.6) appears, associated with a white popup showing the following information:

- The coordinates of the marker, Latitude and Longitude in decimal degrees;
- The name of the pollutant, its chemical formula, and the value of predicted concentration/deposition on that specific location (selected in the input section);
- Relevant cartographic information related with the content of the dashboard, further described in the specific dashboard sections;
- Location name ① in figure 5.6, only appears when typed in the input section;
- Link to download the temporal variation of the pollutant in the chosen location, extracted from the EMEP model prediction data set, that appears in the bottom of the popup, (② in figure 5.6);

In the map visualisation section, there is a slider (⑥ figure 5.6), that allows the user to change the opacity of the EMEP prediction layer. It varies between 0 and 1 or 0% and 100%. Figure 5.7 shows the minimum opacity (figure 5.7b), where it is not possible to see predicted concentrations/depositions, and the maximum opacity (figure 5.7a) that hides all the the cartographic information beneath. The default opacity level is 0.5, or 50%.



(a) The opacity level is at maximum, value = 1.

(b) The opacity level is at minimum, value = 0.

Figure 5.7: Demonstration of the opacity level.

In addition to the map visualisation section, the dashboard offers data processing results in graphical format, that vary with selected inputs. The information contained in the sections map visualisation and graphical representation depends on the dashboard and is described in the following sections.

The Dashboards only present data limited to mainland Portugal, provide a search based on location names and provide a map where EMEP model predictions can be viewed as a layer. The popup, that appears after clicking on the map, displays information regarding: the coordinates; and EMEP model predictions (pollutant information). This popup information can be downloaded. It is only possible to view information on the Dashboards after the user has chosen the input parameters. From the input parameters, it is possible to choose a date of interest and a resolution type, annual or monthly.

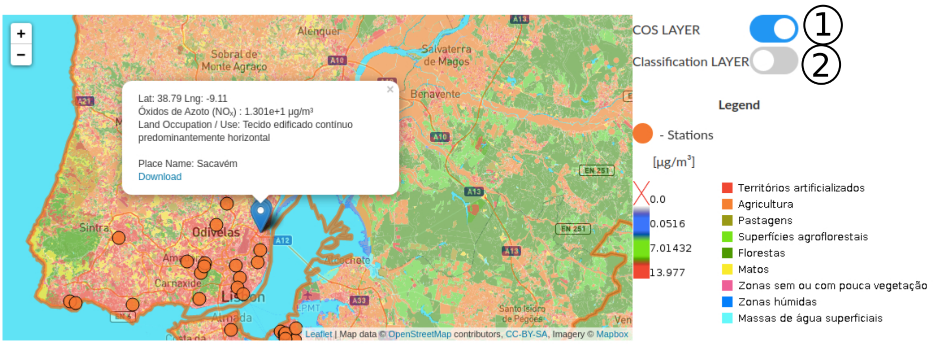
5.4.1 Concentration Dashboard

In the Concentration Dashboard, the user can choose the following pollutants: Ammonia - NH_3; Nitrogen Dioxide - NO_2; Sulphur Dioxide - SO_2; Nitrogen Oxide - NO_X; Ozone - O_3; Particulate Matter 10μm or less in diameter - PM10; Particulate matter 2.5μm or less in diameter - PM25; Sulphate Ion - SO_4^{2-}. These pollutants can be selected because there is information in both data sources, APA and EMEP.

In the Concentration Dashboard, the land occupation/use classification is shown as an alternative cartographic base, which offers relevant information concerning possible pollutant emission sources. For example, higher NO_X concentrations are frequently related with urban and industrial land occupation, and higher NH_3 is frequently related with agriculture and pastures [4].

Land occupation/use information can be enabled or disabled by clicking the COS LAYER button in the dashboard (① in figure 5.8). When enabled, land occupation/use polygons appear in the map, with different classifications differentiated by colour and identified in the map legend.

Land occupation/use data is a product from the *Direção-Geral do Território* [71], and is provided in shapefile format, available for download [8]. The information is stored locally in the GeoServer, and is accessed trough WMS standards.



Lat: 38.79 Lng: -9.11
Óxidos de Azoto (NO_X) : 1.301e+1 μg/m³
Land Occupation / Use: Tecido edificado contínuo predominantemente horizontal

Place Name: Sacavém
Download

COS LAYER ①
Classification LAYER ②

Legend

● - Stations

[μg/m³]
0.0
0.0516
7.01432
13.977

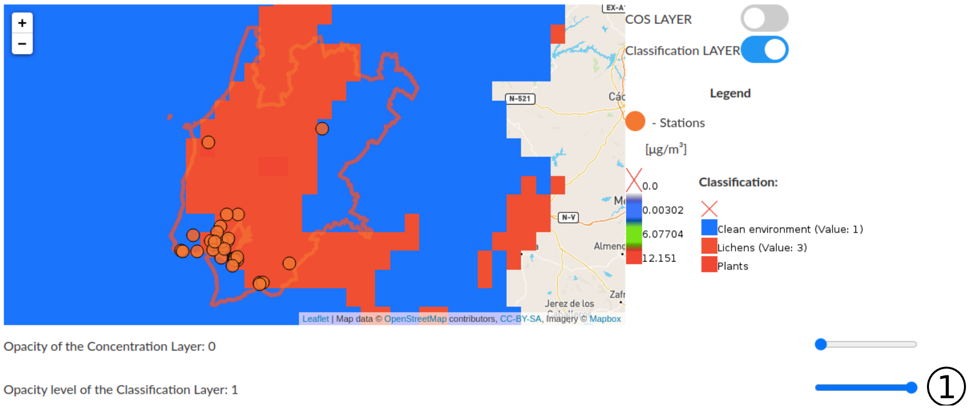
🟥 Territórios artificializados
🟧 Agricultura
🟩 Pastagens
🟩 Superfícies agroflorestais
🟩 Florestas
🟨 Matos
🟪 Zonas sem ou com pouca vegetação
🟦 Zonas húmidas
🟦 Massas de água superficiais

Figure 5.8: Representation of land occupation/use in the map visualisation section of the Concentration Dashboard.

Information regarding land occupation/use is also provided in the popup (figure 5.8) that appears when a location is selected in the input section, or when the user clicks on the map.

Critical levels for NO_X, SO_2, and NH_3 can also be viewed in the map visualisation section, by enabling the Classification Layer button (② in figure 5.8). This information is based on annual predicted concentrations and by considering threshold concentrations beyond which effects have been reported in either plants and lichens. Threshold values vary according to pollutant and were extracted from the air quality guidelines for Europe [17], and Cape et al [72, 73]. These values define limits for each class that is represented by a colour. The blue colour means that the concentration is below any threshold value, while the red colour means that the concentration is above the maximum threshold value. Yellow and orange are the colours for the intermediate classes.

This layer has also a opacity leveller (① figure 5.9), making it possible to combine the transparency of this layer with that of the pollutant. Figure 5.9 shows an example of ammonia (NH_3) critical level exceedances. In this case, there are two concentration thresholds, 1 μg/m^3 for lichens and 3 μg/m^3 for plants. Beyond these values, both lichen and plant communities' less tolerant species are replaced by species more tolerant to ammonia, leading ultimately to a reduction in biodiversity [74]. This information is also added in the Concentrations Measured in Air Quality Stations Graph and EMEP Model Predictions and Evaluation Graph as a horizontal dashed lines, as shown in figure 5.10 and 5.12.


+
−
N-521
N-V
Almend
Jerez de los
Leaflet | Map data © OpenStreetMap contributors, CC-BY-SA, Imagery © Mapbox
Legend
- Stations
[µg/m³]
0.0
0.00302
6.07704
12.151
①

COS LAYER
Classification LAYER

Classification:
🟦 Clean environment (Value: 1)
🟥 Lichens (Value: 3)
🟥 Plants

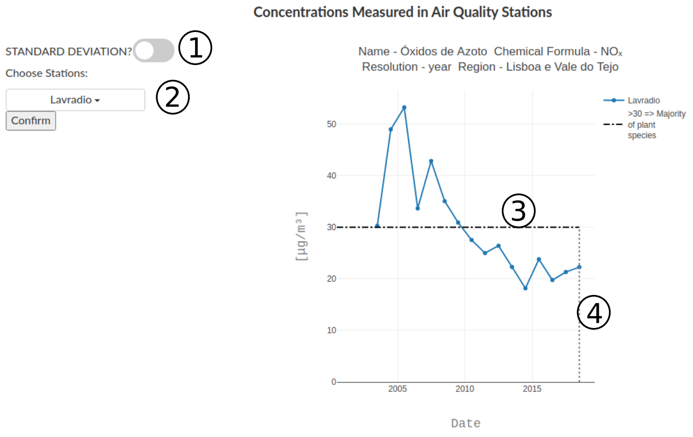
Opacity of the Concentration Layer: 0

Opacity level of the Classification Layer: 1

Figure 5.9: Critical levels of ammonia (NH_3) concentration in 2018 in the *Lisboa and Vale do Tejo* region.

When the user selects a station on the map, the Concentrations Measured in Air Quality Stations Graph appears on the graphical representation section. This Graph represents the temporal variation of the measurements captured in the selected air quality station(s), where x axis represents the timeline and y axis the pollutant concentration.

Another way to access this graph is to select the desired air quality stations from a list (② figure 5.10) that only contains stations from the selected region. By using this method, it is possible to select and represent multiple stations in the same graph, (figure 5.11).


Concentrations Measured in Air Quality Stations

STANDARD DEVIATION?

Choose Stations:

Lavradio ▾

Confirm

Name - Óxidos de Azoto Chemical Formula - NO_x
Resolution - year Region - Lisboa e Vale do Tejo

Figure 5.10: Layout of the Concentrations Measured in Air Quality Stations Graph in the Concentration Dashboard. The selected station is *Lavradio*. The selected input data is listed above the graph.

A vertical dashed line, (④ figure 5.10), identifies the year selected in the input (⑤ in the figure 5.5). When the classification layer is enabled (button ② figure 5.8) horizontal dashed lines will appear on the Concentrations Measured in Air Quality Stations Graph (③ figure 5.10), that indicates the critical level threshold(s).

As it was mentioned before in section 2.3.1, air quality stations capture samples each hour. So, to represent monthly and annual concentrations, the data must be aggregated using averages. Furthermore, to provide a measure of dispersion around the average, the standard deviation is also determined. The determined standard deviation is represented by error bars starting at the mean value point. The button ① in figure 5.10 enables or disables error bars in the graph. As it can be seen in the figure 5.11, it is easier to read a graph with multiple stations without representing the standard deviation. The mean value and standard deviation value appear when the user hovers over the points on the graph.

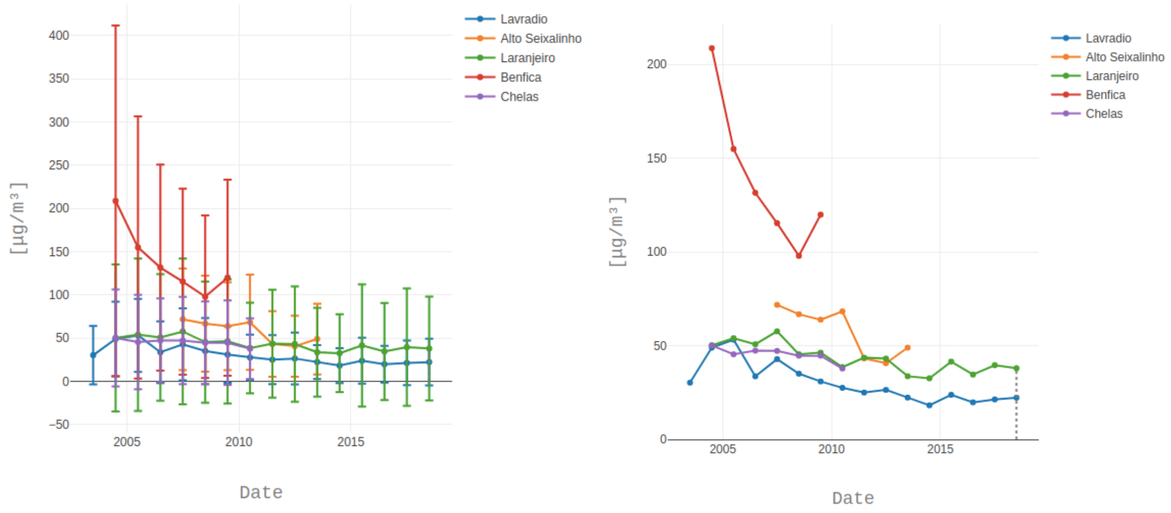


Figure 5.11: Standard deviation enabled vs disabled.

The comparison between the observations and EMEP predictions is shown in the EMEP Model Predictions and Evaluation Graph. This Graph is divided in 3 parts (figure 5.12). The top part represents the temporal variation of spatially averaged, minimum and maximum EMEP predicted concentration for the selected region of interest. The middle part of the graph shows the evaluation of model results. The bottom part of the graph is a button that allows the user to download the EMEP model predictions data plotted and its evaluation.

Since EMEP model predictions are in raster format, the mean value is the average value of all pixels within the chosen region. However, along region boundaries, most pixels are represented in two regions. Therefore, a weighted mean is determined, where the value of the pixel is associated to the area that the pixel occupies in the region's polygon. The max and min values correspond to the maximum and minimum values that exist inside of the region's polygon. The vertical and horizontal dashed lines have the same meaning as in the Concentrations Measured in Air Quality Stations Graph.

The bar graph (② figure 5.12) represents model evaluation results explained in section 5.3. Briefly, this graph shows the evaluation of the EMEP model predictions.

The dashboard also provides the possibility to download the data used to produced the graphs by

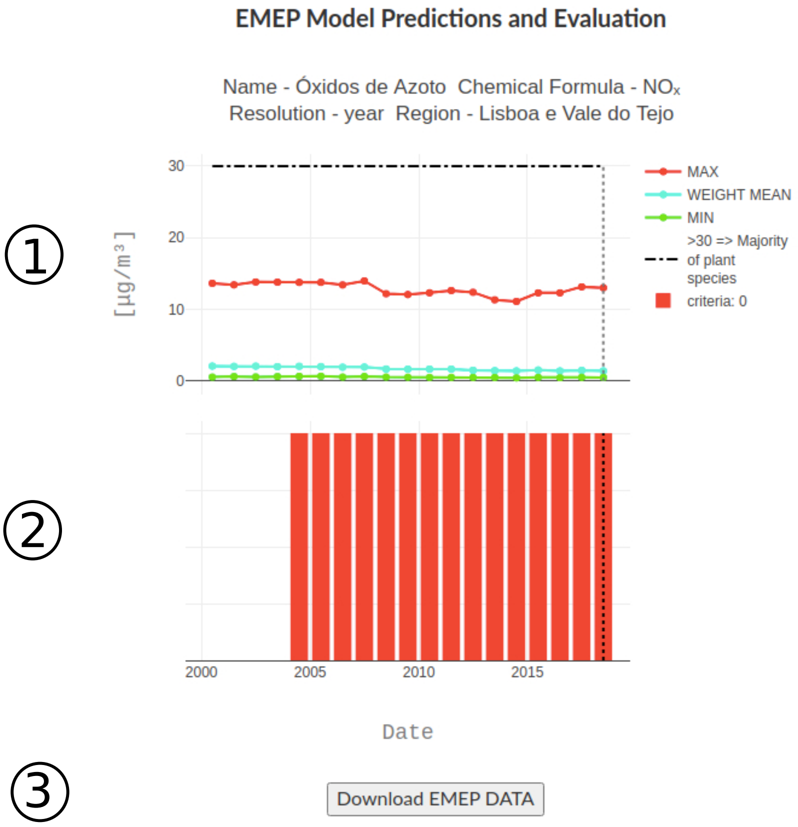

EMEP Model Predictions and Evaluation
Name - Óxidos de Azoto Chemical Formula - NOx
Resolution - year Region - Lisboa e Vale do Tejo
①
②
③
[µg/m³]
30
20
10
0
MAX
WEIGHT MEAN
MIN
>30 => Majority of plant species
criteria: 0
2000
2005
2010
2015
Date
Download EMEP DATA

Figure 5.12: EMEP Model Predictions and Evaluation Graph Layout in the Concentration Dashboard. The selected input data is listed above the graph.

clicking in the button ③ in figure 5.12. The downloaded data corresponds to an excel file containing the temporal variation of the minimum, maximum and averaged model predictions, as well as of model evaluation statistics (FAC2, FB, and NMSE).

Another way to compare the observations with EMEP Model Predictions is using Predictions / Observations Scatter Plot which shows paired observations (APA) and predictions (EMEP) plotted against each other. The abscissas contains the observations corresponding to average concentrations from air quality station measurements, and the ordinates contain the model prediction values in pixels that overlap each station. The graph in the figure 5.13 is an example of the scatter plot, where each point represents a station. As stated before, the dashboard only shows the background stations with a more than 75% data capture for the selected time resolution.

These scatter plots are useful for an initial visual evaluation of model performance, representing the first level of comparative data description and analysis [75]. As it can be seen in figure 5.13, in addition to having the points, the graph has three more lines, $x = y$,$x = 2y$ and $x = \frac{1}{2}y$. These lines offer a better perception of FAC2 and of bias (predominant over or underestimation). If the majority of the points are aligned with $x = y$ line, then the model shows a good performance. The model underestimates and is biased if the majority of the points are below the line $x = \frac{1}{2}y$ and if it is between $x = y$ and $x = \frac{1}{2}y$ the model slightly underestimates. The same reasoning can be applied to the overestimation, which can be detected when points fall in the upper-left half of the graph.

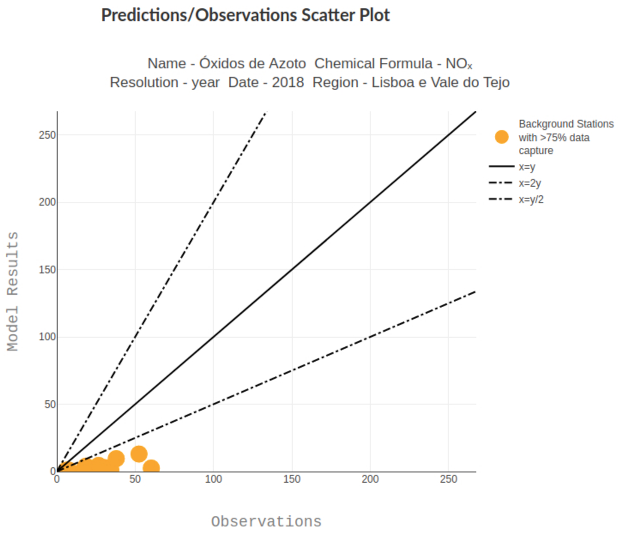


Figure 5.13: Predictions/Observations Scatter Plot Layout in the Concentration Dashboard. The selected input data is listed above the graph.

The Concentration Dashboard provides information on pollutant concentration, land use, critical levels, and air quality stations. In addition, the pollutant information derives from the APA and EMEP server and the pollutants available are those that contain information in both data sources. Furthermore, the Concentration Dashboard shows the temporal variation of observations and EMEP model predictions, and the comparison between observations and model predictions is shown in a scatter plot. From this Dashboard, it is possible to download information about the EMEP model predictions and the statistics information calculated to evaluate the EMEP model.

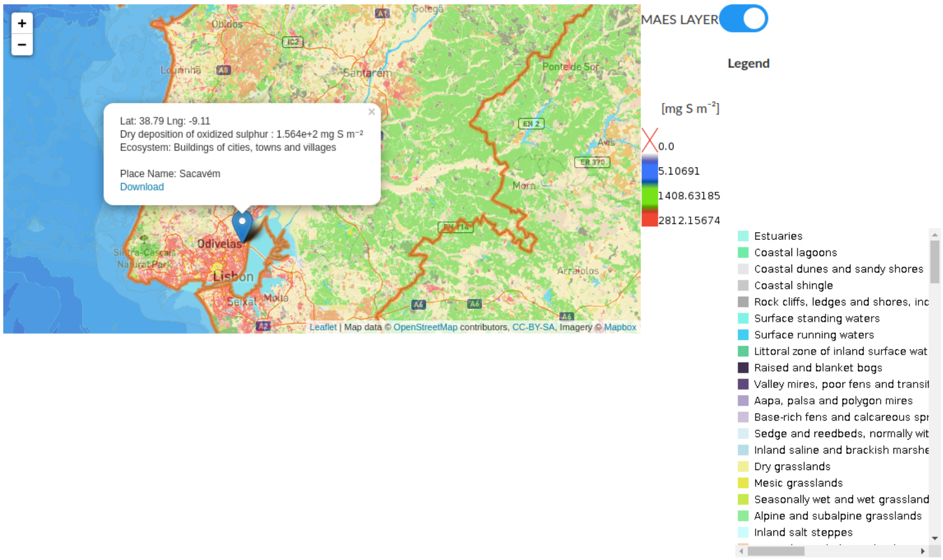
5.4.2 Deposition Dashboard

The Deposition Dashboard has information concerning the deposition of nitrogen and sulphur components (dry and wet deposition, oxidised and reduced forms) in mainland Portugal. Given that only observations for concentration were collected, model evaluation is shown solely in the Concentration Dashboard. For the same reason, temporal aggregated observations are also exclusive to the Concentration Dashboard. On the other hand, in the Deposition Dashboard, only EMEP Model Predictions are shown.

The alternative cartographic base most relevant in the Deposition Dashboard, is the ecosystem type, as different ecosystems show distinct vulnerabilities to nitrogen deposition [76]. Ecosystem type classification corresponds to a tiff raster file downloaded from link [10], locally stored in the GeoServer. The information of ecosystem type, similarly to the land occupation classification in the Concentration Dashboard, can be accessed in the dashboard in one of two ways.

The ecosystem type classification can be visualised in the form of a new layer, obtained trough WMS standards, by enabling the MAES layer button (figure 5.14).

Information regarding the ecosystem type is also provided in the white popup (figure 5.14) that appears when a location is selected in the input section, or when the user clicks on the map. This infor-



Lat: 38.79 Lng: -9.11
Dry deposition of oxidized sulphur : 1.564e+2 mg S m⁻²
Ecosystem: Buildings of cities, towns and villages

Place Name: Sacavém
Download

MAES LAYER
Legend
[mg S m^{-2}]
0.0
5.10691
1408.63185
2812.15674
Estuaries
Coastal lagoons
Coastal dunes and sandy shores
Coastal shingle
Rock cliffs, ledges and shores, inc
Surface standing waters
Surface running waters
Littoral zone of inland surface wat
Raised and blanket bogs
Valley mires, poor fens and transit
Aapa, palsa and polygon mires
Base-rich fens and calcareous spr
Sedge and reedbeds, normally wit
Inland saline and brackish marshe
Dry grasslands
Mesic grasslands
Seasonally wet and wet grassland
Alpine and subalpine grasslands
Inland salt steppes
Óbidos
Golegã
Lourinhã
Santarém
Ponte de Sor
Avis
Mora
Odivelas
Lisbon
Seixal
Moita
Arraiolos
Sintra-Cascais Natural Park
Leaflet | Map data © OpenStreetMap contributors, CC-BY-SA, Imagery © Mapbox

Figure 5.14: Ecosystem type classification visualised by enabling the MAES Layer button.

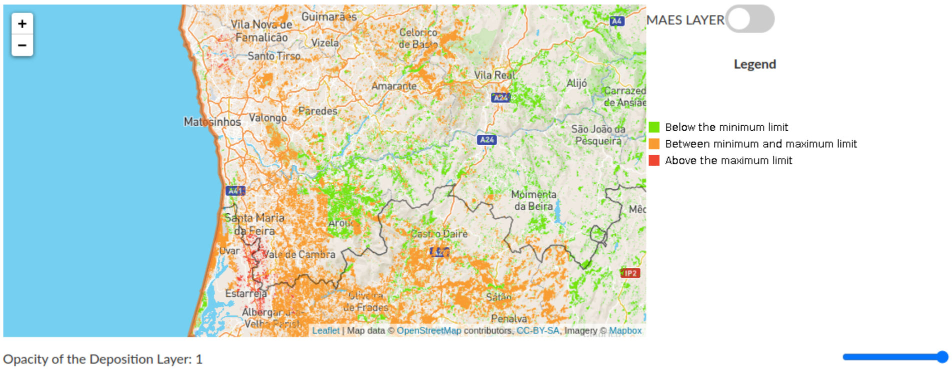
mation is accessed with GetFeatureInfo from the WMS standard, based on a spatial query using the latitude and longitude coordinates of the selected location. However, it is necessary to have a dictionary that matches the ecosystem with the value of the index obtained from GetFeatureInfo and the location marker.

Critical loads, which correspond to empirical threshold deposition values beyond which an ecosystem is compromised, are available for only 5 of the 41 ecosystems mapped in mainland Portugal. Furthermore, empirical critical loads are also only available for nitrogen deposition. Given these limitations, is only possible to identify some of the regions at risk. Empirical critical loads were extracted from Bobbink and Hettelingh (2010) [76], in the form of a range (minimum and maximum value). By overlapping total nitrogen deposition, ecosystem type, minimum and maximum critical load for that ecosystem (when existing), it was possible to map nitrogen critical load exceedances and classify that information in 3 classes: green colour represents a value below minimum critical load (no exceedance); orange colour represents a value between minimum and maximum critical load; and orange colour represents a value above the maximum critical load.

This information is treated as another deposition layer, and can be accessed in the list of layers. Unlike the other layers, this information has no graphic representation, being presented only as a map layer (figure 5.15).

All the data processing to compute the critical load exceedance requires a spatial intersection to overlap total nitrogen deposition, ecosystem type and critical load thresholds, which is only possible with vector data. In addition, the coordinate system of all data must be the same.

The MAES ecosystem type is a raster with a projected coordinate system ETRS89 / LAEA Europe. This coordinate system differs from the remaining information used in these Dashboards, as it represents metric coordinates, instead of degrees. So it is necessary to convert the coordinates to decimal degrees in WGS84. Moreover, the MAES ecosystem type data must be converted from raster to vector format. The procedure used to determine the critical load exceedances was:



Opacity of the Deposition Layer: 1

Figure 5.15: Example of critical level exceedances in the Deposition Dashboard.

1. Convert MAES ecosystem type raster data to vector and then convert the coordinates system from ETRS89 / LAEA Europe to geographical coordinates in WGS84. This must be done in this order, as projection of a raster file implies resampling (interpolation of pixels) and the final projected information will be different.

2. For each ecosystem, critical load (minimum and maximum values) were compared with the total nitrogen deposition value for each pixel. This was achieved by classifying each pixel of the deposition layer according to the critical load (0 - below the minimum limit; 1 - between the minimum and maximum limit; 2 - above the maximum limit). The final result is raster converted to a vector.

3. The final step is to classify each polygon/ecosystem of the MAES data with 0, 1, 2 or -1 (no critical loads are known for the ecosystem). This step is simple because all required data are already in vector format. The computation time depends on the number of ecosystems in which the limits of the critical loads are known. It is possible to assign a colour to a classification number, using a style. In this work, the attribution was: 0 green; 1 orange; 2 red; and -1 no colour (figure 5.15).

When the user selects a region, a pollutant, a resolution and a date, the EMEP Model Predictions Graph appears on the graphical representation section. This graph (figure 5.16) shows the temporal variation of spatially averaged, minimum and maximum EMEP predicted deposition for the selected region of interest, working the same way as the corresponding graph in the Concentration Dashboard. The only difference is that there are no horizontal dashed lines. The user can download the information of this graph by clicking in the button ③ in figure 5.16.

The Deposition Dashboard does not include observation data yet, but it is already prepared to include this information, given that the template is the same as the Concentration Dashboard.

The Deposition Dashboard provides information on pollutant deposition, type of ecosystem, and nitrogen critical loads. Furthermore, the Deposition Dashboard shows the temporal variation of EMEP model predictions. From this Dashboard, it is possible to download information about the EMEP model predictions for nitrogen and sulphur deposition.

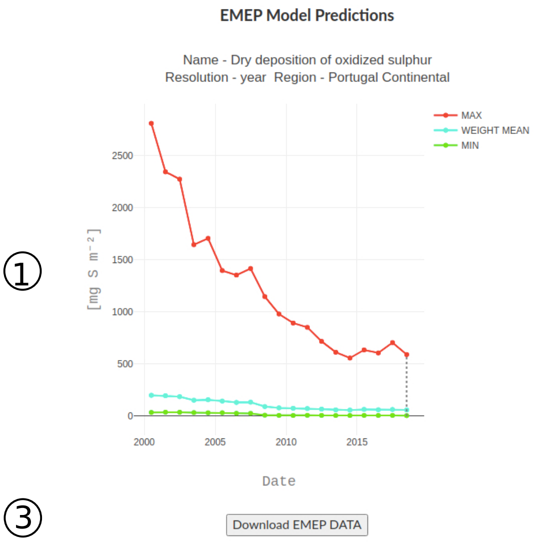


Figure 5.16: EMEP Model Predictions and Evaluation Graph Layout in the Deposition Dashboard. The selected input data is listed above the graph.

5.5 Management of Dashboard Data

The Dashboards inherit data structures from the integrated libraries. From the WebAPA, they inherit the tables: `DataObservations`, `stations_location` and `rede_info`. From the WebEMEP, they inherit the data structures: `Clip`, `raster` and `Year`. From the solution of the management of the equivalences between observations and EMEP model predictions, the Dashboards inherit the table `EMEP_APA_INFO`. From the GeoNames, the Dashboards inherit the `geonames` table.Figure 5.17 shows the Unified Modelling Language of the Dashboards.

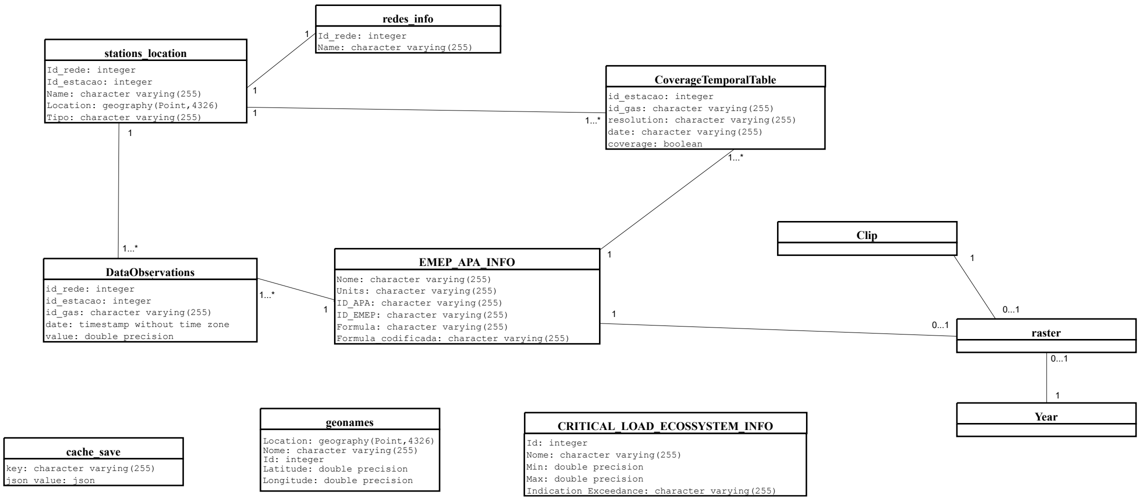

redes_info
Id_rede: integer
Name: character varying(255)
stations_location
Id_rede: integer
Id_estacao: integer
Name: character varying(255)
Location: geography(Point,4326)
Tipo: character varying(255)
CoverageTemporalTable
id_estacao: integer
id_gas: character varying(255)
resolution: character varying(255)
date: character varying(255)
coverage: boolean
DataObservations
id_rede: integer
id_estacao: integer
id_gas: character varying(255)
date: timestamp without time zone
value: double precision
EMEP_APA_INFO
Nome: character varying(255)
Units: character varying(255)
ID_APA: character varying(255)
ID_EMEP: character varying(255)
Formula: character varying(255)
Formula codificada: character varying(255)
Clip
raster
Year
cache_save
key: character varying(255)
json value: json
geonames
Location: geography(Point,4326)
Nome: character varying(255)
Id: integer
Latitude: double precision
Longitude: double precision
CRITICAL_LOAD_ECOSSYSTEM_INFO
Id: integer
Nome: character varying(255)
Min: double precision
Max: double precision
Indication Exceedance: character varying(255)
1
1
1
1...*
1
1...*
1...*
1
1...*
1
1
1
0...1
0...1
0...1
1

Figure 5.17: Unified Modelling Language of the data structures of the Dashboards.

The table `rede_info` stores information about the network of the air quality stations. The table `DataObservations` stores the observations. The table `stations_location` stores information about the stations linking with the observations to give spatial attributes. The table `EMEP_APA_INFO` links information of observations with EMEP model predictions.

The table CRITICAL_LOAD_ECOSYSTEM_INFO links information of the type of ecosystem with with the critical loads. The table CoverageTemporalTable stores the data capture of the observations, indicating which stations can be used to evaluate the model, meaning that by linking to the station information it indicates the geographical points for evaluation of the model.

The table cache_save stores all the results that are visualised on the graphical station of the Dashboards.

Some of the data presented in the Dashboard takes too much time to be generated. So the implemented solution is to generate the data before it is needed. The generated data is stored in the database, table cache_save. The data is associated with a generated key that identifies the type of data, the location and the year (if necessary). The database is a persistent data storage, giving security in storing data even when the server goes down. However, using a remote database adds some latency. Thus, to speed up the process, memory cache is used, that is linked with the database using the same key. To summarise the process, the user asks the server for the data to a specific graphic, the server generates the key and consults the cache. If the cache doesn't have the data, then it asks the database. If the database also doesn't have the data, then the server starts the computation. At the end of the computation, the data is stored in the database and in the cache, and then, the server responds to the user with the requested data. So, when the server responds, the data is stored in the database and in the cache, even if the generated data is null.

To facilitate the process of graphic data generation, a web page was created where the admin can request the server to delete or generate data. Figure 5.18 shows the layout of this web page. For each type of data used in the graphs, the admin has three buttons available. One of the buttons is to delete the data (③ in figure 5.18). The other two buttons are related with data processing, to either verify if all data is processed (① in figure 5.18) or to process all the available data (② in figure 5.18) in the storage, overwriting previous information stored in the system. In addition, there is an option to clean the cache (④ in figure 5.18) and an option (⑤ in figure 5.18) to remove all the data from the system, including cache and the information that is in table cache_save.

The *Admin Model Prediction Region* manages the temporal variation of EMEP model predictions. That is, it either erases or calculates all temporal variations of EMEP model predictions by region for the annual and monthly temporal resolutions. The *Admin Model Prediction County* works similarly as the *Admin Model Prediction Region*, but instead of calculating by region, it calculates by county. The *Admin Observations* also works similarly as the *Admin Model Prediction Region*, but instead of calculating for EMEP model predictions, it calculates for observations, meaning that instead using data from GeoServer, the data is from the table DataObservations.

The *Admin Model Predictions Vs Observations* manages all the information for the scatter plot that compares observations with EMEP model predictions. The information is calculated by region and for annual and monthly resolutions. To find out the identifiers of each type of data, table EMEP_APA_INFO is accessed as it links the two types of data.

The *Admin Evaluation Model Predictions* manages all the information on EMEP model evaluation.

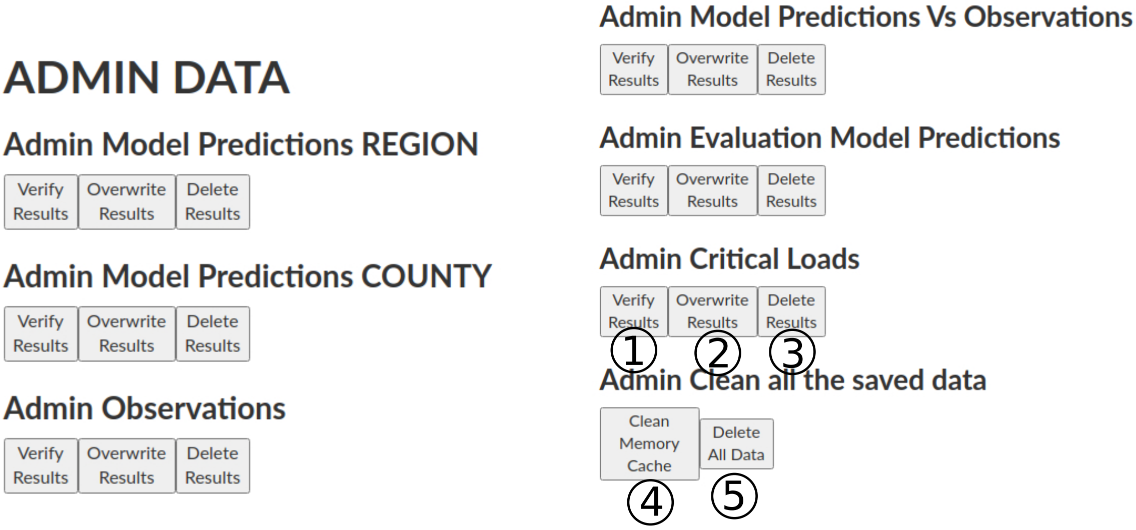


Figure 5.18: Layout of the buttons in the graphics admin web page.

The statistical information used to evaluate the model is calculated by region and for annual and monthly resolutions. The table `EMEP_APA_INFO` is also accessed to link observations with EMEP model predictions.

The *Admin Critical Loads* manages the information of the nitrogen critical loads for annual resolution. The *Admin Clean all the saved data* cleans all the stored data, meaning that it is clean the cache and all the entries of table `cache_save`.

Chapter 6

Evaluation

The evaluation of the work is presented based on a survey and requirements evaluation. The survey was used to evaluate the design of the dashboards, the implemented functionalities, and the data that is presented in the dashboards. With the results, it was possible to evaluate usability, functionality and decide future implementation according to the needs of environmental researchers.

6.1 Survey

In order to evaluate the developed dashboards a survey was presented to users that will be more likely using the dashboards in the future: environmental/ecology researchers.

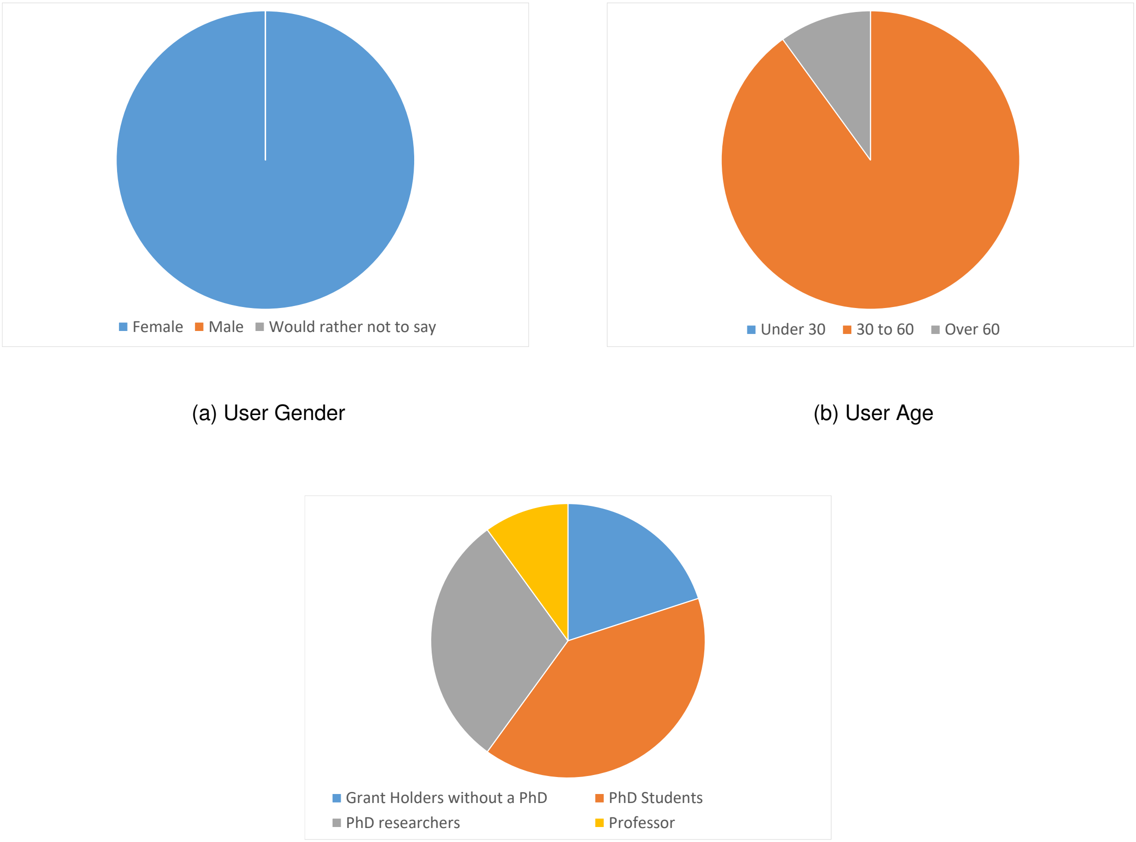
The questionnaire is divided in 3 sections: User Characterisation, System Usability and, Utility. The questions is available in Annex A.1.

The questions are presented in English, but the questionnaire that was given to the researchers was in Portuguese.

6.1.1 User Characterisation

The questions about the user characterisation provide information regarding the user profile. The survey asks user gender, age and profession. The professions are focused on the area of research, being the options:

- Master Student;
- Scholarship Holder without a PhD;
- PhD Student;
- PhD researcher;
- Professor;



(a) User Gender

(b) User Age

(c) User Profession

Figure 6.1: Survey answers on User Characterisation

The survey was answered by 10 people and figure 6.1 shows the the distribution of users' responses. All respondents were women aged between 30 and 60, only one respondent was over 60 years old. Their profession was related to ecology, the most common professions were researchers and students of PhD .

6.1.2 System Usability

Usability refers to the quality of a user's experiences when interacting with a product. Usability is about effectiveness, efficiency and the overall satisfaction of the user. Therefore, usability evaluation focuses on how well users can use a product to achieve their goals.

A variety of methods can be used to gather feedback from users. These methods can be SUS (System Usability Scale), QUIS (Questionnaire for User Interface Satisfaction), CSUQ (Computer System Usability Questionnaire), etc. The chosen method was SUS based on results of Tullis & Stetson [77]. The authors compared different methods for measuring usability, and noted that SUS was one of the simplest questionnaires that produced the most reliable results in all sample sizes, even for small samples. In addition, SUS was the only questionnaire whose questions addressed all the different aspects of user reaction to the system as a whole.

This questionnaire consists of 10 questions with five response options; from Strongly agree to Strongly disagree. The questions were taken from [78] and the word "system" was replaced by "dashboards".

Furthermore, an extra question was added that usually complements the results from the SUS questionnaire: "Overall, I would rate the user-friendliness of the dashboards". The answer is a seven-point, adjective-anchored Likert scale (Worst Imaginable, Awful, Poor, OK, Good, Excellent, Best Imaginable) [79].

The SUS value calculated from the answers was 80.5 (0-100). To complement this result the utilisation of the Dashboards was mostly classified as excellent. In addition, Dashboards were found to be easy to use. The less positive result was that features were not fully integrated making them not feel 100% confident in using the Dashboards.The detailed results of the SUS questionnaire are available in Annex A.2.

6.1.3 System Utility

The questions of this part of the survey serve to assess the usefulness of the developed dashboards. The questions are about the implemented features, the data used, and the results presented. The answers of the questions are Yes/No, or text. When the answer is open, it means that the answer to the previous question was yes. The questions are presented in Annex A.1.

9 out of 10 respondents think that the Dashboards would make their job easier. Out of 10 people, only one expressed difficulty to access the data that is available in the Dashboards. The data, that the respondent had contact with, was the EMEP modelled predictions. In addition, 50% of the respondents had previously accessed the data presented on the Dashboards by other means. All 50% of respondents believe that the data was presented in a simpler way on the Dashboards.

The survey had open answers to give respondents the opportunity to suggest new features/data they may need / be useful in their research. These suggestions were about types of data, how the data is presented, or what features are missing in the dashboards. The answers of the respondents, that are discussed later in section 7.2, are:

- Missing data / results:
  - Information about habitats;
  - Information about climate;
  - Spacial data on precipitation, temperature, wind, emissions, exceedance of levels (ozone, NO_X e SO_2) and critical loads for nitrogen and acidifiers. Evaluation of EMEP with satellite data. Evaluation of the model results about the deposition (wet deposition);
  - Background information, so that it can be possible to assess the confidence of the data. It was easy for the respondent to obtain the data, that he/she wanted, but he/she would like it to be associated with information such as the source of basic data (how many and which stations, for example), degree of confidence in EMEP modelling. The respondent also suggests that

when the user downloads data to a particular location, it should also come with ecosystem information. This information appears on the interactive map, when the user clicks on a point, but it does not appear in the downloaded data.

- Functionalities in the Concentration dashboard:
  - Download station data
- Functionalities in the Deposition dashboard:
  - Define polygon of interest

Further details of the survey responses can be found in Annex A.3.

6.2 Requirements Validation

The implemented requirements are:

1. **The library allows the development of applications that use georeferenced data stored in excel**

   The GeoExcel library implements all these requirements and was validated in the dashboards with observations

2. **The library allows the collection and processing of pollution data from the Portuguese network of air pollution stations**

   The WebAPA library implements these features and was validated in the Concentration Dashboards.

3. **The library allows EMEP model predictions data management**

   The WebAPA library implements these features and was validated in the Concentration Dashboard and Deposition Dashboard.

4. **The library allows an easy access to Geonames service**

   The GeoNames library implements these features and was validated in the Concentration Dashboard and Deposition Dashboard.

5. **Libraries allow the management of equivalences between EMEP model predictions and observations data**

   The Dashboards show in an integrated way data from both sources.

6. **Libraries must run as part of a Middleware for web/georeferenced application development**

   The libraries run on GeoNode and Django.

7. **Libraries must store data in geospatial database**

   PostgreSQL was used in this project but GeoNode allows other geospatial database.

8. **Libraries must store data on a geospatial data sharing server**

   GeoServer was used in this project.

9. **Libraries must use a standard API for database access**

   To access geospatial data managed by the libraries and dashboards, any other programmer can use standard SQL and OGC standards for accessing PostgreSQL and GeoServer, respectively.

6.3 Performance Evaluation

Although it has not been possible to measure the presentation times of the graphical interfaces of the developed applications, it can be considered that the presentation times are not very high, due to the usability results.

The second possible evaluation is related to the time the system takes to process the available data. Most of the results presented in the dashboards are previously computed in the background, to speed the data visualisation.

Nonetheless, the table 6.1 indicates the execution times to calculate the temporal variation of EMEP model predictions, the temporal variation of observations, and to evaluate the model. The time period under analysis was between 2000 and 2018. The applications were run in the background on a server with 16GB of RAM and 64-bit processor. The applications ran in the background to ensure that the results were available as quickly as possible.

	resolution			
	Annual		Monthly	
Type of Results	Time	Size	Time	Size
Observations	$\approx$ 6h	$\approx$ 600 MB	$\approx$ 2 days	$\approx$ 600 MB
EMEP Model Predictions	$\approx$ 37,5h	$\approx$ 1.5 GB	$\approx$ 19 days	$\approx$ 20 GB
EMEP Model Evaluation	$\approx$ 20h	$\approx$ 1.5 GB	$\approx$ 9 days	$\approx$ 20 GB

Table 6.1: Data sizes and execution times to calculate the temporal variation of the EMEP Model Predictions, the temporal variation of the observations, and to evaluate the model. The time period was 19 years.

The execution times for the observations were the shortest because the calculation used the average, maximum and minimum operations offered by the database, and these operations are efficient when applied to a single table.

To evaluate the model, it was necessary to calculate the data capture of all stations, this being the slowest operation in this process. The overlapping is a relatively fast operation because it uses the properties of the raster structure (matrix) to discover the pixel index to which the station to compare belongs.

The process to determine weighted averages of EMEP model predictions is the slowest, because it needs to determine which pixels belong to the region of interest. To determine which pixels to consider is a slow process, because it is always necessary to verify if the region's polygon intersects the vectored pixel polygon.

Even if processes EMEP Model Evaluation and EMEP Model Predictions initially have the same volume of data to process, they do not have the same execution time because the process EMEP Model Evaluation overlaps observations with EMEP model predictions, making it unnecessary to go through the whole raster.

In addition, the time differences between annual and monthly resolution directly depends on the amount of data to be processed. The data with monthly resolution is 12 times larger than with annual resolution, and execution times differ approximately 12 times. For example, the execution times for processing EMEP model predictions are 37.5h for annual resolution, and 19 days (456h) for monthly resolution, the ratio being approximately 12.

By adding a new year of data, the system needs to process this data in both time resolutions (annual and monthly), taking approximately: 2h for Observations; 1 day for EMEP Model Predictions; and 12h for EMEP Model Evaluation. The new data is processed in the background, not affecting the functioning of the system, i.e. the system continues to provide results for the other years.

To process 19 years of data it took approximately 20 GB of information. The results of the data processing stored in memory do not take up much space because for each year the result of the data processing is a value that is saved in a list. Therefore, on average there is 1 GB of information for each year.

Normally a machine has more than 50GB free, meaning that it is possible to allocate a system with more than 50 years of data processing with annual and monthly resolution.

Chapter 7

Conclusions

7.1 Achievements

The developed applications were presented at a scientific congress, XIX National Ecology Meeting organised by SPECO (Portuguese Ecology Society). The flash presentation was about 3 minutes, where it was possible to present the main characteristics of the platform. The presentation received good feedback, and the questions asked were related to the types of data available. The certificate of participation in the ecology congress is in Annex A.4.

The architecture of the libraries worked according of the requirements of gathering, processing and representing pollution data.

The Dashboards were created because there is a need to automate processes related to air pollution data. To help the development of the Dashboards, four libraries have been developed, where two of them are responsible for collecting data from the APA (observations) and EMEP (model results). The other two libraries were created because of the need to facilitate the loading of excel files into the database and to facilitate translation between the toponym and its coordinates using the Geonames database.

There has been a positive assessment of the dashboards by researchers who have evaluated it. Through the surveys, the dashboards were evaluated as 85 out of 100 in the system usability scale (SUS), showing that they are relatively easy to use and show results that can be useful in air pollution studies.

For data with annual resolution, it took about 1.5 days to process all data automatically, which, through the cache, becomes available to users in a matter of seconds. If researchers had to process this data using their commonly used commercial and open source applications (e.g., R, GIS software), it would take much longer and human intervention (prone to errors) would be required between each step. For monthly resolution, where the data is much larger, the whole processing took about two weeks. It should be noted that the times mentioned before corresponds to a process of 20 years of data, in the case of one year, it takes no more than one day to process all resolutions.

With these applications, it has been possible to solve collection and processing problems for researchers. In the future, depending on their needs, it is possible to develop new libraries for new types of data (e.g., air pollution derived from satellite data), using the same architecture, and to create new applications to present new results in a simple way.

7.2 Future Work

In the future, it would be interesting to implement suggestions given by specialists through the survey (section 6.1).

By clicking on the dashboard map, there is information about the ecosystem or about land-use. But the downloaded data of that point does not include this information at the moment. To implement this feature, it would be only necessary to know the coordinates that are given by the request parameters and ask through the WFS standards to the geospatial data server. After getting the response, it would be necessary to check which polygon the point belongs to. Having the polygon, the final step would be to add the information to the download data. Another improvement of the graphical interface would be the possibility of drawing a polygon on the map to replace the regions available. Due to the available plugins in the Leaflet library, it is possible to draw polygons on the map. Having a drawn polygon, it is simple to extract its coordinates and send them to the server. Once having the coordinates limiting the polygon, the server can then start to process the data, following the same procedure as when calculating for a predefined polygon from a region in mainland Portugal. Therefore, it is possible to implement this functionality. The drawback is the time that the user has to wait for all the process to be complete, because the server does not have the capability to store all results, for all possible polygons, on a given map. The time consumption increases with the size of the drawn polygon.

One of the improvements to be made in the developed libraries is to have available on the Dashboards multiple versions of the downloaded EMEP prediction model. Therefore, there would be no need to replace the data when there is a new update. As seen in section 5.4.2, there is no evaluation of model data on the Deposition Dashboard. Therefore, it is necessary to evaluate this data, to give some confidence to the results presented in the dashboard.

It would be interesting to integrate new data such as: natural habitat information; climate information; and satellite information. Natural habitat information would be useful to researchers because it identifies the area where a particular species lives. This information has the same type as ecosystem information, so the only thing to do is to download and store this information. After the data is stored, the use of this data is the same as for MAES (ecosystem) data, already developed. The pollution is related with climate, because the weather influences how pollution is transported in atmosphere and/or it is deposited. So, a future implementation would be to provide this information to the user. Climate information (spatial and temporal data of precipitation, temperature, wind) is available as raster just like EMEP data, so the data procedure is similar. The difficult is on the representation of the wind on the map, because the wind direction is not instinctive to read with colour graduation, but should be represented with arrows (e.g. one arrow indicating the wind direction per pixel). The precipitation and temperature data are scalar, and should be represented on the map with a colour scale just like the EMEP model predictions. Satellite data can be used to evaluate EMEP model results. The procedure is the same using satellite data, but instead of calculating the valid stations, the valid regions would be calculated.

Although there was a large number of suggestion that are relevant and that could provide relevant results, we think that only a set of dashboards and backend infrastructures like ours could allow the successful implementation of such suggestions, thus demonstrating the need and innovation of the work described in this thesis.

Bibliography

[1] United Nations. Sustainable Development Goals, (accessed: 27.12.2020). URL https://www.un.org/sustainabledevelopment/sustainable-development-goals/.

[2] Directive, EU. Directive (EU) 2016/2284 of the European Parliament and of the Council of 14 December 2016 on the reduction of national emissions of certain atmospheric pollutants, amending Directive 2003/35/EC and repealing Directive 2001/81, 2016.

[3] S. Yatkin, M. Gerboles, C. Belis, F. Karagulian, F. Lagler, M. Barbiere, and A. Borowiak. Representativeness of an air quality monitoring station for pm2.5 and source apportionment over a small urban domain. *Atmospheric Pollution Research*, 11(2):225 – 233, 2020. ISSN 1309-1042. doi: https://doi.org/10.1016/j.apr.2019.10.004. URL http://www.sciencedirect.com/science/article/pii/S1309104219304726.

[4] M. Oliveira, S. Tomlinson, E. Carnell, A. Dore, H. Serrano, M. Vieno, C. Cordovil, U. Dragosits, M. Sutton, C. Branquinho, and P. Pinho. Nitrogen and sulfur deposition over a region in sw europe based on a regional atmospheric chemical transport model. *Atmospheric Environment*, 223:117290, 2020. ISSN 1352-2310. doi: https://doi.org/10.1016/j.atmosenv.2020.117290. URL http://www.sciencedirect.com/science/article/pii/S1352231020300327.

[5] D. Simpson, A. Benedictow, H. Berge, R. Bergström, L. D. Emberson, H. Fagerli, C. R. Flechard, G. D. Hayman, M. Gauss, J. E. Jonson, M. E. Jenkin, A. Nyíri, C. Richter, V. S. Semeena, S. Tsyro, J.-P. Tuovinen, A. Valdebenito, and P. Wind. The emep msc-w chemical transport model &ndash; technical description. *Atmospheric Chemistry and Physics*, 12(16):7825–7865, 2012. doi: 10.5194/acp-12-7825-2012. URL https://acp.copernicus.org/articles/12/7825/2012/.

[6] J. C. Chang and S. R. Hanna. Air quality model performance evaluation. *Meteorology and Atmospheric Physics*, 87(1-3):167–196, 2004.

[7] Direcção Geral do Território. *Especificações Técnicas da Carta de Uso e Ocupação do Solo (COS) de Portugal Continental para 2018*, (accessed: 22.11.2020). URL https://www.dgterritorio.gov.pt/sites/default/files/documentos-publicos/2019-12-26-11-47-32-0__ET-COS-2018_v1.pdf.

[8] Sistema Nacional de Informação Geográfica (SNIG). Download COS data, (accessed:

18.07.2020). URL http://mapas.dgterritorio.pt/DGT-ATOM-download/COS_Final/COS2018_v1/COS2018_v1.zip.

[9] European Commission. Mapping and Assessment of Ecosystems and their Services - MAES (Website), (accessed: 22.11.2020). URL https://ec.europa.eu/environment/nature/knowledge/ecosystem_assessment/index_en.htm.

[10] European Environment Agency. Download MAES data, (accessed: 24.11.2020). URL http://cmshare.eea.europa.eu/s/KscZR3EcKrGmPbK/download.

[11] C. Dore and S. Vidič. Considerations of changing the EMEP grid. Technical report, note 3, www. unece. org, 2012. URL https://unece.org/fileadmin/DAM/env/documents/2012/air/EMEP_36th/n_3_EMEP_note_on_grid_scale__projection_and_reporting.pdf.

[12] T. D. Schowalter. Chapter 11 - ecosystem structure and function. In T. D. Schowalter, editor, *Insect Ecology (Fourth Edition)*, pages 367 – 404. Academic Press, fourth edition edition, 2016. ISBN 978-0-12-803033-2. doi: https://doi.org/10.1016/B978-0-12-803033-2.00011-X. URL http://www.sciencedirect.com/science/article/pii/B978012803033200011X.

[13] European Environment Agency (EEA). Glossary - List of environmental terms used by EEA, (accessed: 18.12.2020). URL https://www.eea.europa.eu/help/glossary.

[14] World Meteorological Organisation (WMO). Atmospheric Deposition, (accessed: 18.12.2020). URL https://public.wmo.int/en/our-mandate/focus-areas/environment/atmospheric-deposition.

[15] R. Bobbink, M. Hornung, and J. G. Roelofs. The effects of air-borne nitrogen pollutants on species diversity in natural and semi-natural european vegetation. *Journal of ecology*, 86(5):717–738, 1998.

[16] A. Bouwman, D. Van Vuuren, R. Derwent, and M. Posch. A global analysis of acidification and eutrophication of terrestrial ecosystems. *Water, Air, and Soil Pollution*, 141(1-4):349–382, 2002.

[17] W. H. Organization et al. Air quality guidelines for europe. 2000. URL https://www.euro.who.int/en/publications/abstracts/air-quality-guidelines-for-europe.

[18] ESRI. GIS Dictionary, (accessed: 27.12.2020). URL https://support.esri.com/en/other-resources/gis-dictionary.

[19] ArcGis. What is raster data, (accessed: 27.12.2020). URL https://desktop.arcgis.com/en/arcmap/10.3/manage-data/raster-and-images/what-is-raster-data.htm.

[20] ArcGis. What is NetCDF data, (accessed: 27.12.2020). URL https://pro.arcgis.com/en/pro-app/latest/help/data/multidimensional/what-is-netcdf-data.htm.

[21] UniData. User Guide, (accessed: 27.12.2020). URL https://www.unidata.ucar.edu/software/netcdf/docs/user_guide.html.

[22] Techopedia. Tagged Image File Format (TIFF), (accessed: 27.12.2020). URL https://www.techopedia.com/definition/2093/tagged-image-file-format-tiff.

[23] NASA. GeoTIFF, (accessed: 27.12.2020). URL https://earthdata.nasa.gov/esdis/eso/standards-and-references/geotiff.

[24] ESRI. What is GIS?, (accessed: 08.07.2020). URL https://www.esri.com/en-us/what-is-gis/overview.

[25] Computer Science Department at Universitat Politècnica de Catalunya. What is a Geographic Information System?, (accessed: 08.07.2020). URL https://www.cs.upc.edu/~lpv/general.dir/whatgis.html.

[26] Caitlin Dempsey. What is GIS?, (accessed: 08.07.2020). URL https://www.gislounge.com/what-is-gis/.

[27] ESRI. What is ArcGIS?, (accessed: 08.07.2020). URL https://developers.arcgis.com/labs/what-is-arcgis/.

[28] ESRI. Architecting the ArcGIS Platform: Best Practices, (accessed: 08.07.2020). URL https://www.esri.com/content/dam/esrisites/en-us/media/pdf/architecting-the-arcgis-platform.pdf.

[29] ESRI. What is ArcGIS Enterprise on Amazon Web Services?, (accessed: 08.07.2020). URL https://enterprise.arcgis.com/en/server/latest/cloud/amazon/what-is-arcgis-server-on-aws.htm.

[30] ESRI. What is ArcMap?, (accessed: 16.12.2020). URL https://desktop.arcgis.com/en/arcmap/10.3/main/map/what-is-arcmap-.htm.

[31] S. Steiniger and A. J. Hunter. Free and open source gis software for building a spatial data infrastructure. In *Geospatial free and open source software in the 21st century*, pages 247–261. Springer, 2012.

[32] PostgreSQL. PostgreSQL 11.8 Documentation, (accessed: 10.07.2020). URL https://www.postgresql.org/docs/11/index.html.

[33] PostGIS. PostGIS 2.5.5dev Manual, (accessed: 10.07.2020). URL https://postgis.net/docs/manual-2.5/.

[34] S. A. Mehdi, M. Ali, G. Nima, R. Zahra, S. Reyhaneh, and B. Peyman. How to implement a governmental open source geoportal. *Journal of Geographic Information System*, 2014, 2014.

[35] GeoServer. What is Geoserver?, (accessed: 10.07.2020). URL http://geoserver.org/about/.

[36] Powered by eAtlas. What is GeoServer? Why would I use it?, (accessed: 10.07.2020). URL https://eatlas.org.au/node/300.

[37] GeoServer. User Manual about WMS Service, (accessed: 12.08.2020). URL https://docs.geoserver.org/stable/en/user/services/wms/reference.html.

[38] ESRI. Spatial Data Infrastructure: A Collaborative Network, (accessed: 27.12.2020). URL https://www.esri.com/library/brochures/pdfs/spatial-data-infrastructure.pdf.

[39] S. Buonanno, G. Zeni, A. Fusco, M. Manunta, M. Marsella, P. Carrara, and R. Lanari. A geonode-based platform for an effective exploitation of advanced dinsar measurements. *Remote Sensing*, 11(18):2133, 2019.

[40] P. Corti, F. Bartoli, A. Fabiani, C. Giovando, A. T. Kralidis, and A. Tzotsos. Geonode: an open source framework to build spatial data infrastructures. Technical report, PeerJ Preprints, 2019.

[41] GeoNode. What is GeoNode, (accessed: 10.07.2020). URL https://docs.geonode.org/en/3.0/about/index.html.

[42] Unidata. Documentation about NetCDF4 module, (accessed: 08.08.2020). URL https://unidata.github.io/netcdf4-python/netCDF4/index.html.

[43] Unidata. Web page about NetCDF, (accessed: 08.08.2020). URL https://www.unidata.ucar.edu/software/netcdf/.

[44] Unidata. Documentation about NetCDF, (accessed: 08.08.2020). URL https://www.unidata.ucar.edu/software/netcdf/docs/netcdf_introduction.html#netcdf_format.

[45] Pandas. About page of Pandas website, (accessed: 08.08.2020). URL https://pandas.pydata.org/about/.

[46] NumPy. Web Page, (accessed: 08.08.2020). URL https://numpy.org/.

[47] Rasterio. Documentation, (accessed: 08.08.2020). URL https://rasterio.readthedocs.io/.

[48] Shapely. The Shapely User Manual, (accessed: 08.08.2020). URL https://shapely.readthedocs.io/en/latest/manual.html.

[49] Fiona. Repository, (accessed: 08.08.2020). URL https://github.com/Toblerity/Fiona.

[50] Leaflet. Web Page, (accessed: 08.08.2020). URL https://leafletjs.com/.

[51] F. Warmerdam. The geospatial data abstraction library. In *Open source approaches in spatial data handling*, pages 87–104. Springer, 2008.

[52] Frank Warmerdam, Even Rouault, and others. GDAL Documentation, (accessed: 14.01.2021). URL https://gdal.org/gdal.pdf.

[53] Agência Portuguesa do Ambiente. Data Download Web age, (accessed: 09.08.2020). URL https://qualar1.apambiente.pt/qualar/index.php?page=6&subpage=.

[54] Agência Portuguesa do Ambiente. CSV with information about stations, (accessed: 22.11.2020). URL https://qualar.apambiente.pt/qualar/estacoes?_export_type=csv_with_hidden_cols&keywords=&order=.

[55] Agência Portuguesa do Ambiente. Information about deactivated stations, (accessed: 22.11.2020). URL https://qualar1.apambiente.pt/qualar/index.php?page=4&subpage=2.

[56] N. Aste, R. Adhikari, J. Compostella, and C. D. Pero. Energy and environmental impact of domestic heating in italy: Evaluation of national nox emissions. *Energy Policy*, 53:353 – 360, 2013. ISSN 0301-4215. doi: https://doi.org/10.1016/j.enpol.2012.10.064. URL http://www.sciencedirect.com/science/article/pii/S0301421512009494.

[57] Norwegian Meteorological Institute. Website of EMEP MSC-W Model, (accessed: 15.07.2020). URL https://emep.int/mscw/.

[58] European Environment Agency. Introduction, (accessed: 15.07.2020). URL https://www.eea.europa.eu/publications/EMEPCORINAIR/page005.html.

[59] Norwegian Meteorological Institute. Links to catalogues of the EMEP data, (accessed: 22.11.2020). URL https://emep.int/mscw/mscw_moddata.html.

[60] European Commission. Mapping and Assessment of Ecosystems and their Services (Paper), (accessed: 22.11.2020). URL https://ec.europa.eu/environment/nature/knowledge/ecosystem_assessment/pdf/MAESWorkingPaper2013.pdf.

[61] Agência Portuguesa do Ambiente. Home Page, (accessed: 30.12.2020). URL https://qualar1.apambiente.pt/qualar/.

[62] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017. URL https://www.R-project.org/.

[63] ArcGIS. What is a geodatabase?, (accessed: 22.09.2020). URL https://desktop.arcgis.com/en/arcmap/10.3/manage-data/geodatabases/what-is-a-geodatabase.htm.

[64] Agência Portuguesa do Ambiente. Download information about active stations, (accessed: 27.12.2020). URL https://qualar.apambiente.pt/qualar/estacoes?_export_type=csv_with_hidden_cols&keywords=&order=.

[65] Agência Portuguesa do Ambiente. Url to download an excel, (accessed: 09.08.2020). URL https://qualar1.apambiente.pt/qualar/excel_new.php?excel=1.

[66] GeoNames. README to extract files form GeoNames Server, (accessed: 18.07.2020). URL http://download.geonames.org/export/dump/readme.txt.

[67] J. C. Chang and S. R. Hanna. Technical descriptions and user's guide for the boot statistical model evaluation software package, version 2.0. *George Mason University*, 4400:22030–4444, 2005. URL http://www.harmo.org/Kit/Download/BOOT_UG.pdf.

[68] S. R. Hanna and J. Chang. Setting acceptance criteria for air quality models. In *Air Pollution Modeling and its Application XXI*, pages 479–484. Springer, 2011.

[69] Mapbox. Web Services APIs, (accessed: 27.12.2020). URL https://docs.mapbox.com/api/overview/.

[70] Mapbox. Glossary - OpenStreetMap, (accessed: 27.12.2020). URL https://docs.mapbox.com/help/glossary/osm/.

[71] Direcção Geral do Território. Quem somos, (accessed: 18.07.2020). URL https://www.dgterritorio.gov.pt/dgt/quem-somos.

[72] J. N. Cape, L. van der Eerden, A. Fangmeier, J. Ayres, S. Bareham, R. Bobbink, C. Branquinho, P. Crittenden, C. Cruz, T. Dias, et al. Critical levels for ammonia. In *Atmospheric Ammonia*, pages 375–382. Springer, 2009.

[73] J. N. Cape, L. J. van der Eerden, L. J. Sheppard, I. D. Leith, and M. A. Sutton. Reassessment of critical levels for atmospheric ammonia. In *Atmospheric ammonia*, pages 15–40. Springer, 2009.

[74] P. Pinho, T. Dias, C. Cruz, Y. Sim Tang, M. A. Sutton, M.-A. Martins-Loução, C. Maguas, and C. Branquinho. Using lichen functional diversity to assess the effects of atmospheric ammonia in mediterranean woodlands. *Journal of Applied Ecology*, 48(5):1107–1116, 2011.

[75] C. J. Willmott. On the validation of models. *Physical geography*, 2(2):184–194, 1981.

[76] R. Bobbink, J.-P. Hettelingh, et al. Review and revision of empirical critical loads and dose-response relationships. In *Proceedings of an expert workshop, Noordwijkerhout*, volume 2325, 2010.

[77] T. S. Tullis and J. N. Stetson. A comparison of questionnaires for assessing website usability. In *Usability professional association conference*, volume 1. Minneapolis, USA, 2004.

[78] U.S. General Services Administration Technology Transformation Services. System Usability Scale (SUS), (accessed: 11.11.2020). URL https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html.

[79] A. Bangor, P. Kortum, and J. Miller. Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of usability studies*, 4(3):114–123, 2009. URL https://uxpajournal.org/wp-content/uploads/sites/8/pdf/JUS_Bangor_May2009.pdf.

Appendix A

Extra Information

A.1 Questionnaire

The Questionnaire is:

1. Do you think that Dashboards can be used in your work, making it easier to access the data presented? (Answer: Yes or No)
2. In the past, in your work, have you ever tried to access data available on Dashboards, but failed? (Answer: Yes or No)
3. Describe the data you have tried to access in the past and were unable to do so. (Answer: open answer)
4. In the past, in your work, have you ever accessed data that is now available on Dashboards through other means? (Answer: Yes or No)
5. Is access to this data easier on Dashboards? (Answer: Yes or No)
6. Would you like Dashboards to make other data or results available? (Answer: Yes or No)
7. What data/results would you like to see on Dashboards? (Answer: open answer)
8. About the Concentration Dashboard: Do you think there is another feature that should be included? (Answer: Yes or No)
9. About the Concentration Dashboard: Describe the features you think are missing. (Answer: open answer)
10. About the Deposition Dashboard: Do you think there is another feature that should be included? (Answer: Yes or No)
11. About the Deposition Dashboard: Describe the features you think are missing. (Answer: open answer)

A.2 SUS responses

4. Acho que gostaria de utilizar as dashboards frequentemente.

10	4.2
Responses	Average Number

5. Acho que as dashboards são desnecessariamente complexas

10	1.8
Responses	Average Number

6. Acho que as dashboards são fáceis de usar

10	4.3
Responses	Average Number

7. Acho que é necessário suporte de um técnico especialista para conseguir usar as dashboards.

10	1.5
Responses	Average Number

8. Acho que as várias funcionalidades das dashboards estão bem integradas.

10	3.8
Responses	Average Number

9. Acho que há demasiadas inconsistências nas dashboards.

10	2
Responses	Average Number

10. Acho que a maioria das pessoas aprenderia a utilizar as dashboards muito rapidamente

10
Responses

4.7
Average Number

11. Acho que as dashboards foram muito complicadas de usar.

10
Responses

1.6
Average Number

12. Senti-me muito confiante ao utilizar as dashboards

10
Responses

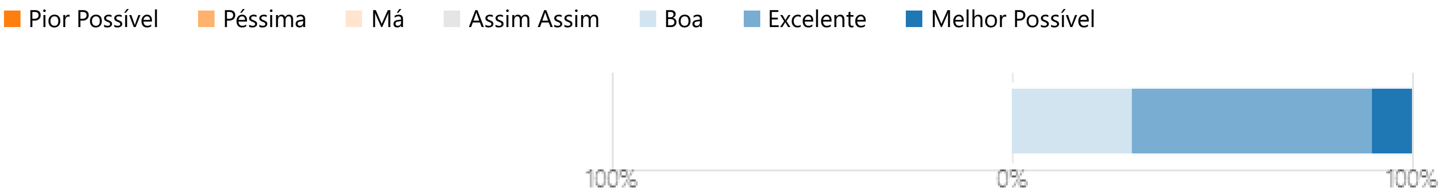
3.8
Average Number

13. Foi necessário aprender muitos conceitos antes de conseguir usar as dashboards.

10
Responses

1.7
Average Number

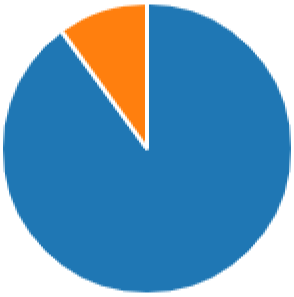
14. Globalmente, como é que classificaria a facilidade de utilização das dashboards?



A.3 Utility Answers

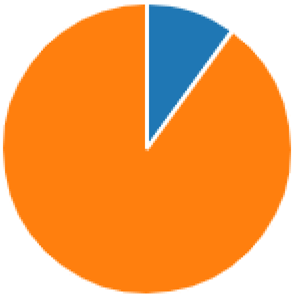
15. Acha que as Dashboards poderão ser usadas no seu trabalho, facilitanto o acesso aos dados apresentados?

● Sim	9
● Não	1



16. No passado, no seu trabalho, alguma vez tentou aceder a dados disponíveis nas Dashboards, mas não conseguiu?

● Sim	1
● Não	9



17. Descreva os dados que no passado tentou aceder e não conseguiu.

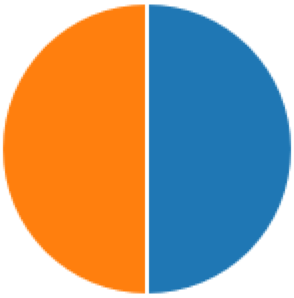
1

Responses

Latest Responses

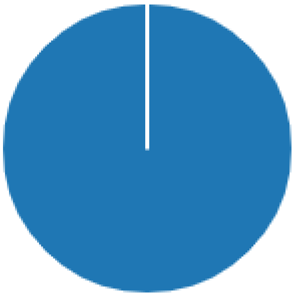
18. No passado, no seu trabalho, alguma vez acedeu através de outros meios a dados que agora estão disponíveis nas Dashboards?

● Sim	5
● Não	5



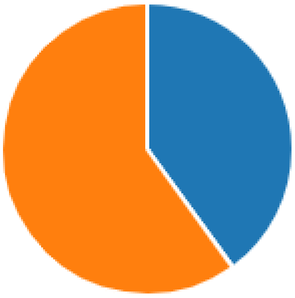
19. O acesso a esses dados apresenta-se mais simples nas Dashboards?

● Sim	5
● Não	0



20. Gostaria que as Dashboards disponibilizassem outros dados ou resultados?

● Sim	4
● Não	6



21. Quais são os dados/resultados que gostaria de ver nas Dashboards?

4

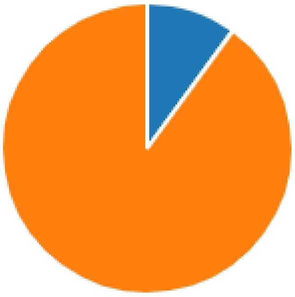
Responses

Latest Responses

"Gostaria de ver informação de base, para poder avaliar a confiança q...

22. Sobre a Dashboard das Concentrações: Acha que há alguma outra funcionalidade que deveria ser incluída?

● Sim	1
● Não	9



23. Sobre a Dashboard das Concentrações: Descreva as funcionalidades que acha que estão em falta.

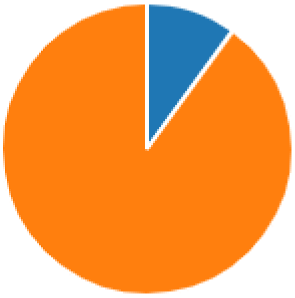
1

Responses

Latest Responses

24. Sobre a Dashboard das Depoisções: Acha que há alguma outra funcionalidade que deveria ser incluída?

● Sim	1
● Não	9



25. Sobre a Dashboard das Deposições: Descreva as funcionalidades que acha que estão em falta.

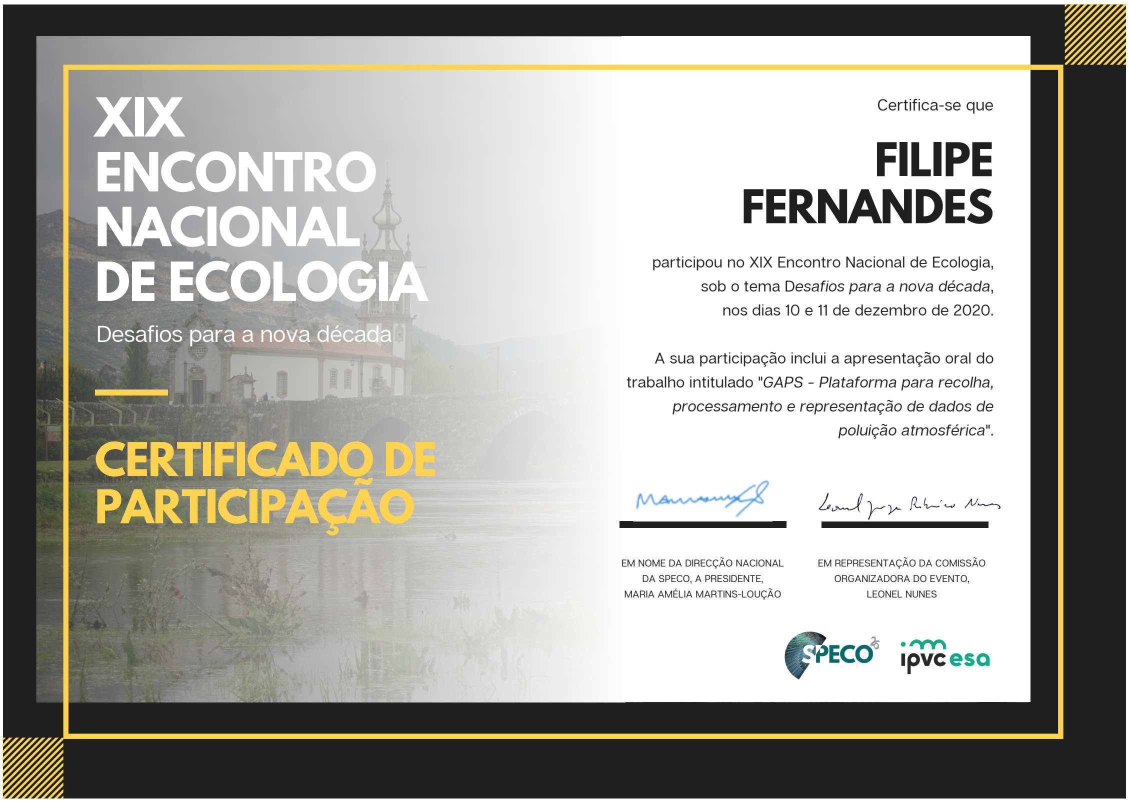
1

Responses

Latest Responses

A.4 Certificate

Figure A.1 shows the certificate of participation at a congress of ecologists, XIX National Ecology Meeting organised by SPECO (Portuguese Ecology Society).



XIX ENCONTRO NACIONAL DE ECOLOGIA

Desafios para a nova década

CERTIFICADO DE PARTICIPAÇÃO

Certifica-se que

FILIPE FERNANDES

participou no XIX Encontro Nacional de Ecologia, sob o tema *Desafios para a nova década*, nos dias 10 e 11 de dezembro de 2020.

A sua participação inclui a apresentação oral do trabalho intitulado "*GAPS - Plataforma para recolha, processamento e representação de dados de poluição atmosférica*".

EM NOME DA DIRECÇÃO NACIONAL
DA SPECO, A PRESIDENTE,
MARIA AMÉLIA MARTINS-LOUÇÃO

EM REPRESENTAÇÃO DA COMISSÃO
ORGANIZADORA DO EVENTO,
LEONEL NUNES

Figure A.1: Certificate of participation at a congress of ecologists