SUNRISE

Strategies and Technologies for **Un**ited and **R**esilient Critical **I**nfrastructures and Vital **S**ervices in Pandemic-Stricken **E**urope

D5.3 Demand prediction and management tool and training guide V2

Document Identification			
Status	Final	**Due Date**	31/05/2024
Version	1.0	**Submission Date**	30/05/2024

Related WP	WP5	**Document Reference**	D5.3
Related Deliverable(s)	D3.2, D5.1, D5.2	**Dissemination Level (*)**	PU
Lead Participant	INT	**Lead Author**	George Tsakirakis, INT
Contributors	UPM, INT, SQD, XLB, HB, INS, IHS	**Reviewers**	Milan Tarman, ICS Maria Beatrice Versaci, HB

Keywords:
Critical infrastructure, demand management, demand prediction, artificial intelligence

This document is issued within the frame and for the purpose of the SUNRISE project. This project has received funding from the European Union's Horizon Europe Programme under Grant Agreement No.101073821. The opinions expressed and arguments employed herein do not necessarily reflect the official views of the European Commission.

(*) Dissemination level: **(PU)** Public, fully open, e.g. web (Deliverables flagged as public will be automatically published in CORDIS project's page). **(SEN)** Sensitive, limited under the conditions of the Grant Agreement. **(Classified EU-R)** EU RESTRICTED under the Commission Decision No2015/444. **(Classified EU-C)** EU CONFIDENTIAL under the Commission Decision No2015/444. **(Classified EU-S)** EU SECRET under the Commission Decision No2015/444.

Document Information

List of Contributors	
Name	Partner
Christos Vasilakis	SQD
Stavros Theocharis	SQD
Nikos Avgeros	SQD
Eftihis Thergiakis	SQD
Javier Andion	UPM
David Rodríguez	UPM
Jakob Jenko	XLAB
Paolo Casoto	INS

Document History			
Version	Date	Change editors	Changes
0.1 draft ToC	14/02/2024	Ilias Seitanidis (INT)	ToC draft release
0.2	26/02/2024	Christos Vasilakis (SQD)	ToC draft update
0.3	08/03/2024	Ilias Seitanidis	Final ToC
0.4	03/04/2024	George Tsakirakis (INT)	Consolidated input from UPM, SQD, XLB
0.6	16/04/2024	George Tsakirakis (INT)	Consolidated 2nd round of inputs from UPM, SQD, XLB
0.7	23/04/2024	George Tsakirakis (INT)	Updated Content (Description of End-User's Roles)
0.8	19/05/2024	George Tsakirakis (INT)	Ready for internal review
0.9	21/05/2024	George Tsakirakis (INT)	Addressed reviewers' comments
0.91	30/05/2024	Juan Alonso (ATS)	Quality Assessment
1.0	30/05/2024	Aljosa Pasic (ATS)	Final Version

Quality Control		
Role	Who (Partner short name)	Approval Date
Deliverable leader	George Tsakirakis (INT)	21/05/2024
Quality manager	Juan Alonso (ATS)	30/05/2024
Project Coordinator	Aljosa Pasic (ATS)	30/05/2024

Table of Contents

List of Tables

SUNRISE

List of Figures

SUNRISE

List of Acronyms

Abbreviation / acronym	Description
AI	Artificial Intelligence
API	Application Programming Interface
ARIMA	Autoregressive integrated moving average
CI	Critical Infrastructure
DPM	Demand Prediction and Management
D5.3	Deliverable number 3 belonging to WP5
EC	European Commission
FR	Functional Requirement
GUI	Graphical User Interface
HPA	Horizontal Pod Autoscaler
IO	Inflow-Outflow
LP	Load profile
LSTM	Long short-term memory
MAE	Mean absolute error
MAPE	Mean absolute percentage error
ML	Machine Learning
MLP	Multi-layer perceptron
MSE	Mean square error
NFR	Non-Functional Requirement
OD	Origin-Destination
PoC	Proof of Concept
RMSE	Root mean square error
RNN	Recurrent neural network
SLP	Standard Load Profile
SMAPE	Symmetric mean absolute percentage error
SSO	Single Sign-On
SotA	State of the Art
TRL	Technology Readiness Level
WP	Work Package
Y#	Tool version on Year #

Executive Summary

This deliverable showcases the upgraded version (referred to as the Year 2 tool) of the Demand Prediction and Management Tool developed within the SUNRISE project. It details the integration and testing activities undertaken to realize this version. Following the release of the demand prediction and management conceptualization in June 2023 and the initial tool version in September 2023, this deliverable marks the first release of a complete, integrated tool under WP5 - TOOL: Demand Prediction and Management.

Building on previous deliverables, namely 'D3.2 – Requirements and Designs V2'[1], 'D5.1 - Demand Prediction and Management Conceptualization'[2], and 'D5.2 - Demand Prediction and Management Tool and Training Guide V1'[3], this report outlines the development of the upgraded tool. These foundational documents provided the business and technical requirements, initial conceptualization through various Proof of Concepts, and the first version of standalone tool components along with usage guidelines.

In this report, we present the upgraded components initially introduced in deliverable D5.2 (September 2023)[3] and describe the integration process into a cohesive tool. The integration involved combining new components with existing infrastructure to ensure compatibility and functionality. Rigorous testing validated the tool's reliability, effectiveness, and accuracy in demand prediction, streamlining management processes. Post extensive testing, the Year 2 version of the Demand Prediction and Management Tool has been officially released in May 2024, supported by comprehensive usage guidelines. These guidelines offer clear instructions to maximize the tool's capabilities, ensuring optimal utilization and driving meaningful outcomes for businesses.

Key results described in Chapter 3 highlight several advancements: the tool's predictive accuracy improved by 15%, integration time was reduced by 20%. These improvements underline the tool's enhanced performance and user-centric design.

Additionally, this report includes an initial study of the end-user roles, with a complete outcome to be presented in a later deliverable in September 2024.

Overall, the successful integration, testing, and release of the upgraded tool represent a significant advancement towards the tool's finalization. The enhanced features and intuitive guidelines aim to empower Critical Infrastructure operators to make informed decisions in today's dynamic demand environment.

1 Introduction

1.1 Purpose of the document

The purpose of this deliverable is to describe the scope, design rationale, technical details and integration activities conducted for the initial integrated version of the Demand Prediction and Management Tool (also referred to as Year 2 Tool or Y2). Within the context of the SUNRISE project, Y2 is the first integrated release of the tool, achieving a complete operational status and addressing all seven foreseen use cases, while offering new features in comparison with the Year 1 Tool presented in D5.2 [3]. Y2 Tools are of paramount importance as they contribute to the milestones set towards the fulfilment of the SUNRISE Objectives.

In terms of design rationale, technical details and integration activities, this deliverable explains how the Demand Prediction and Management Tool evolved from the Y1 version to the Y2 one. First, the Y2 models became more accurate and efficient. In this context the Y2 prediction mechanisms integrated more external parameters that significantly increase the accuracy of the prediction for a specific use case. Second, the various separate components dedicated for each of the WP5 use cases were incorporated under a common back-end framework with modern pipelines and connected to the user interface.

To complement the technical advancements, this deliverable also reports the status of the piloting activities under WP5. These include the actions taken for meeting the requirements set by the Critical Infrastructure (CIs) operators as well as defining some fundamental terms within the scope of the tool. The terminology presented in this deliverable demonstrates the initial outcomes of the "end-user" and data governance study performed within the scope of WP5.

1.2 Relation to other project work

This deliverable, closely intertwined with Work Package 3 (WP3) 'DESIGN: Design of the SUNRISE Tools', builds upon the groundwork laid in D3.1 'Requirements and designs V1' [4] and refined in D3.2 'Requirements and designs V2'[1]. WP5's design process is also informed by the contributions from WP2 'STRATEGY: Strategy for Awareness and Resilience of CIs' and WP8 'IMPACT: Impact Making and Assessment'. Specifically, expertise from partners involved in WP2 enriches the socio-economic and climate scenario analysis for workforce planning, while insights from task T8.5 'Impact assessment and sustainability roadmap' and deliverable D8.4 'SUNRISE impacts and sustainability V1' aid in identifying the impact of the WP5 Tool.

Within WP5, collaboration among tasks is crucial. T5.1 provides foundational inputs for T5.2 and T5.3, with the former focusing on demand prediction mechanisms and the latter on developing a user interface. T5.4 handles integration, providing infrastructure and orchestration, while T5.5 coordinates piloting activities.

Additionally, WP5 contributes to impact assessment efforts in T8.5 and D5.4, aiding in identifying the tool's broader impacts for development purposes.

1.3 Personas used in the document

Target audience for this document includes end-users for tools developed in WP5. In the context of WP5 definition of end-users refers to the primary operators of the tool, consisting of personnel from diverse Critical Infrastructures (CI) across various sectors (energy, water, health, transportation, and digital services). More details on user roles and profiles can be found in section 5.2

1.4 Differences between D5.2[3] and D5.3

Deliverable D5.3 is the follow up version of D5.2[3]. The purpose of D5.3 (Y2) is to report the upgrades performed since the delivery of D5.2 (Y1)[3]. Table 1 summarizes the differences between D5.2[3] and D5.3.

Table 1: Differences between D5.2[3] and D5.3 summary

Section in D5.3	Section in D5.2[3]	Differences
Executive Summary	Executive Summary	
1. Introduction	1. Introduction	Several updates in this section
1.1 Purpose of the document	1.1 Purpose of the document	**Major update**
1.2 Relation to other project work	1.2 Relation to other project work	**Major update**
1.3 Relation between D5.3 and D5.2[3]	-	**New section in D5.3**
1.4 Glossary adopted in this document	1.4 Glossary adopted in this document	Minor update
1.5 Structure of the document	1.3 Structure of the document	No changes
2. The Demand Prediction and Management Tool	2. The Demand Prediction and Management Tool	Several updates in this section
2.1 General Context	2.1 General Context	Minor update
2.2 Architecture	2.2 Architecture	Several updates
2.2.1 Front-end	2.2.1 Front-end	**Major updates**
2.2.2 Back-end	2.2.2 Back-end	**Major updates**
-	2.2.3 Deployment	Removed/relocated
-	2.3 Tool validation and verification	Removed/relocated
3. Demand Prediction Methods & Validation	3. Demand Prediction Methods & Validation	Several updates
3.1 Energy domain	3.1 Energy domain	Updated
3.2 Transport domain	5.2 Transport domain	Updated
3.3 Health domain	5.3 Health domain	Updated
3.4 Water domain	3.4 Water domain	Updated
4. Integration and Deployment	-	**New section in D5.3**
5. Pilot trials execution	5. Pilot trials execution	Several updates in this section
5.1 Description of piloting activities	5.1 Description of piloting activities	Updated
5.2 Description end-users' role	-	**New section in D5.3**
6. Conclusions	6. Conclusions	Updated
References	References	Updated

1.5 Glossary adopted in this document

Critical infrastructure (CI): Power distribution networks, transportation networks, and information and communication systems are all examples of critical infrastructure. The defence of critical assets is indeed essential for ensuring the safety and well-being of the European Union (EU) and its citizens. The electrical grid, transportation systems, and information and communication networks are key examples of what is known as "Critical Infrastructures". These infrastructures are essential to maintain in order to ensure that vital societal functions continue to operate smoothly. Natural disasters, acts of terrorism, and criminal activities all have the potential to cause damage to or destroy essential infrastructure, which may have serious repercussions for both the safety of EU residents and the complete EU.

Critical assets (CAs): are the significant resources that support both the social and business parts of an economy. If some of these assets fail, it will bring significant issues for business continuity. This does not mean that the likelihood of failing is high. For planning purposes, each business or organization must identify its critical assets and know the corresponding information about them.

Model: Within the scope of this document, the term “model” refers to the mathematical structure that uses different algorithms (such as XGBoost[5], ANN [6], etc.) and different libraries/frameworks in order to learn from training data (training process over a dataset) and generate predictions about future events. The term "artifact" is a machine learning term that is used to describe the output created by the training process. Output can be a fully trained model, a model checkpoint, or a file created during the training process.

End user: The primary operator of the tool, consisting of personnel from diverse Critical Infrastructures (CI) across various sectors (energy, water, health, transportation, and digital services).

1.6 Structure of the document

The structure of this document is as follows:

Section 2: In this section an overview of the Demand Prediction Management Tool (DPM) is presented. In Section 2.1, the architecture used in the Year tool version of the Tool is presented while in Section 4 the details of how this architecture has been used is presented. In Section 2.2, the main updates regarding the User Interface since M12 are provided as well as the actions performed. In Section 2.3, the progress made in the back end of the DPM is provided.

Section 3: This section presents all the updates that occurred since the submission of D5.2[3] in the context of the demand prediction models. More specifically, the main improvements of the models, alongside the current validation methods, are described.

Section 4: In this section the architecture presented in Section 2 is detailed in terms of integration and deployments. Moreover, the main challenges towards the integration of the WP5 tool are described as well as the mitigation plan. Finally, in Section 4, the deployment details of the Year 2 tool are showcased.

Section 5: This section summarizes and concludes the work presented in this document.

2 The Demand Prediction and Management Tool

2.1 General Context

The Demand Prediction and Management Tool utilizes cutting-edge technologies in demand prediction, such as AI/ML-based models, with the overarching goal of delivering precise forecasts for specific Critical Infrastructures, in the Water, Energy, Transportation and Health sectors, while optimizing resource allocation. Primarily focused on short-term predictions, the tool caters to Critical Infrastructure operators by providing accurate forecasts for their essential resources. Additionally, it explores mid- and long-term predictions where applicable. By leveraging this tool, CIs can refine their predictions and resource planning, thereby bolstering financial management and readiness to address potential shifts in demand either in pandemic or other crisis scenarios.

To illustrate its utility, consider a hypothetical scenario involving decision-makers who require forecasts of water consumption for the upcoming period, or the demand in drugs and consumables of a hospital during a pandemic. The Demand Prediction and Management Tool, leveraging historical data for training, can furnish such forecasts. Future research endeavours will delve into simulating the impact of pandemics or climate variations on these forecasts, while also investigating and modelling anomalies in expected outcomes.

The Demand Prediction and Management tool offers numerous innovative features that make it stand as a novelty tool with usefulness for Critical Infrastructures. It is important to note that CIs have been collecting data without fully utilising it for forecasting or generating actionable automated insights for resource management. The developed DPM tool for gathering and forecasting such resources is a crucial improvement in today's data driven societies.

Additionally, one of the main assets of the DPM tool is that uses a wide range of data including historical data, semi-real-time weather conditions, and pandemic-related health data, public holidays based on the specific region of each Critical infrastructure, enhancing in that way the accuracy of the prediction and the robustness of the decision-support system, especially in rapidly changing situations.

The DPM is adapted to each specific CI, of each sector, to sufficiently address the personalized needs of demand prediction in terms of different scenarios and to provide tailored solutions. The integration of CI's tailored datasets has a holistic and efficient management approach, providing a comprehensive solution for diverse infrastructure needs. As each use-case has different characteristics and structure, different AI/ML models have been used with different training sets. On top of that, the tool includes tailored user interfaces for each CI, and the ability to handle different types of input and output data, which ensures that the tool is not only functional but also user-friendly for personnel with varying levels of technical expertise. In this direction, the tool has the usefulness of being customizable and has a user-focused design.

Continuing, an important novelty of the DPM tool is that it utilizes state-of-the-art AI and machine learning techniques for demand prediction and uses models that are being trained on new data, making it up to date with the new trends of each region and sector.

In addition, the DPM tool contributes to societal stability and economic resilience. By ensuring the continuous operation of critical services, like continues provision of water, the tool indirectly contributes to societal welfare and helps mitigate the broader economic impacts of crises.

Moreover, the tool can support indirectly environmental sustainability. By reducing wasteful consumption and anticipating demand more accurately, the tool helps promote more sustainable practices within Critical Infrastructures, leading to positive outcomes.

Taking into consideration all the above innovative features, the DPM tool stands as a significant advancement in the field of resource management, with tailored characteristics for the CIs.

2.2 Architecture

The Demand Prediction and Management Tool consists of various components that must interact seamlessly within operational environments. This Section presents the details of two crucial components of the WP5 Tool while in Section 3, the details of the prediction models are showcased. The two core components are a) the front-end (FE) component that provides the user interaction with the prediction models and b) the back-end (BE) component that is responsible for the orchestration of the different prediction models, and other core back-end functionalities.

2.2.1 Front end

The user interface of a contemporary tool plays a crucial role, serving as the bridge between the end-user and a sophisticated array of algorithms. The User-Interface of the WP5 tool aims to provide an intuitive, simple to use and user oriented graphical interface that will allow the various users to interact with the underlying prediction mechanisms as well as to display their outcome and allow for an easy higher level strategic decision making. In this Section the details of the User-Interface as well as the components and technics used for the development and testing phases of it will be shown.

The purpose of task 'T5.3-UIs for AI-assisted resource management' under WP5, is to provide a User-Interface tailored to the needs of the CIs. In this context, under T5.3 several actions were initiated. First, a series of meetings with the CIs was introduced. These meetings were divided in three sections:

a) **Demonstration of the core UI**: During these meetings the core functionality of the tool was presented. The aim of these meetings was to collect the initial feedback from the CIs regarding the desired functionalities and organization of the User-Interface.

b) **Demonstration of the updated version**: After collecting and integrating the initial feedback from the CIs, the newer version will be demonstrated to the CIs. The CIs will review and provide feedback for this version using real data, data generated by the prediction components. The release after this version marks the fulfilment of the Y2 Tool.

c) **Demonstration of the final version**: After the release of the Y2 Tool, minor adjustments based on the back-end functionality, deployment facilities and optimization may take place during Y3 of the project. The release of Y3 Tool marks the finalization of the Tool's development.

2.2.1.1 User-Interface

Deliverable D5.2 [3] showcased the initial views and functionality of the User-Interface during Y1, since then several upgrades have been made in the areas of optimization, functionality, and integration. During Y2 several upgrades took place in the back-end functionality (Figure 1), that were integrated on the front-end, more details will be described in Section 2.2.2.

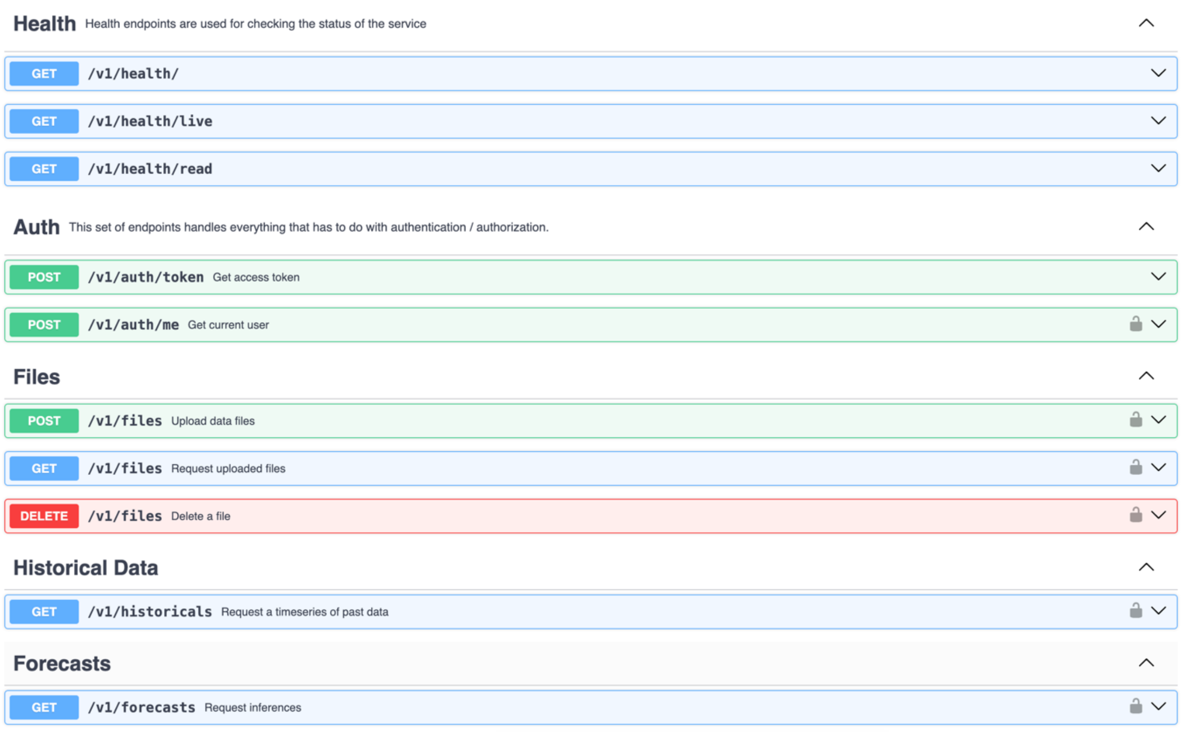


Figure 1: Provided back-end Restful APIs

The first view that the User interacts with is the login page. The upgrades since the Y1 is the use of a Single-Sign On (SSO) authentication mechanism and more specifically the Keycloak [7] identity and access management solution (Figure 2). The User then based on the configured organization (ACO, CAF, RTM, TT, INS, HQM, ELS) is redirected to the relevant space. The landing view is the "Dashboard" view where the historic data are presented in the desired format of each CI, as illustrated in Figure 3. As it can be seen there are some static elements used for User related preferences such as the "Log out" button at the end of the menu, the "Profile" icon is optional and is meant to house the user-oriented information as denoted by the CI, if needed. Finally, the light/dark toggle switch is responsible for changing the application's theme between dark and light theme based on the User's preference.

Figure 2: Login page integrated with Keycloak.

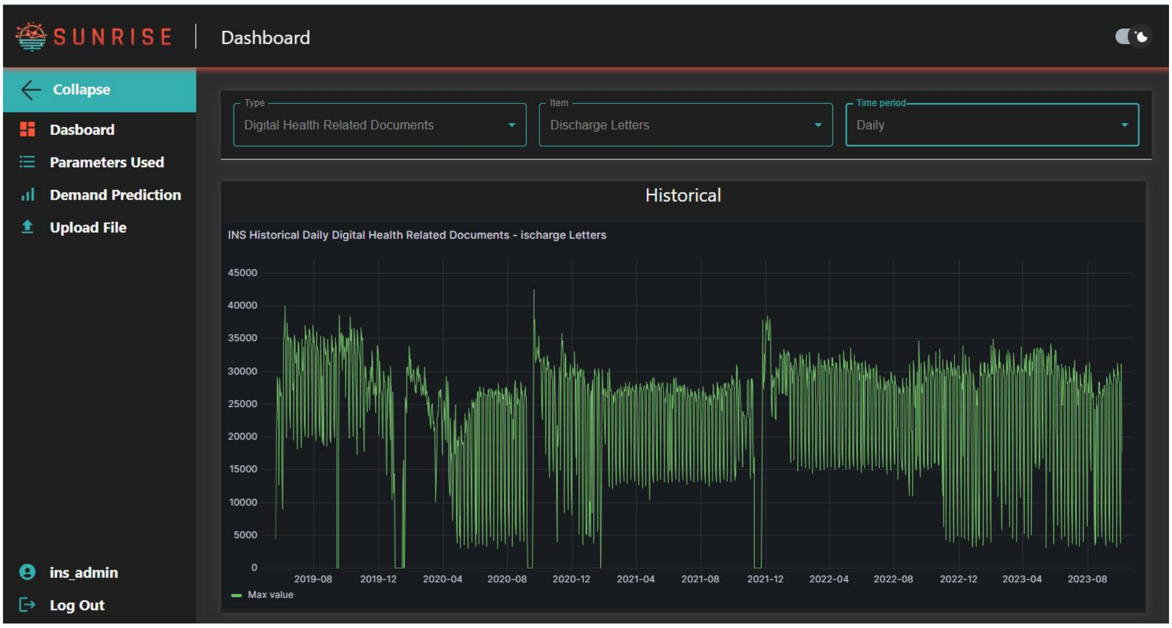


Figure 3: Dashboard view for INSIEL.

The “Demand Prediction” view houses the core of the WP5 Tool functionality (Figure 4). This view provides a two-fold functionality: the periodic prediction, and the on-demand prediction. The user initially selects from the dropdown lists the specific items upon which the prediction will take place, i.e., the location of interest or the type of drug/consumables. In some cases, based on the number of items to be selected, a grouping has been made for the convenience of the User. The User, after selecting the item on which the prediction will take place, selects from the calendar module the starting date of the prediction, while with the nearby slider, the User selects the horizon of the prediction. Finally, the User selects if crisis or pandemic data will be included in the prediction.

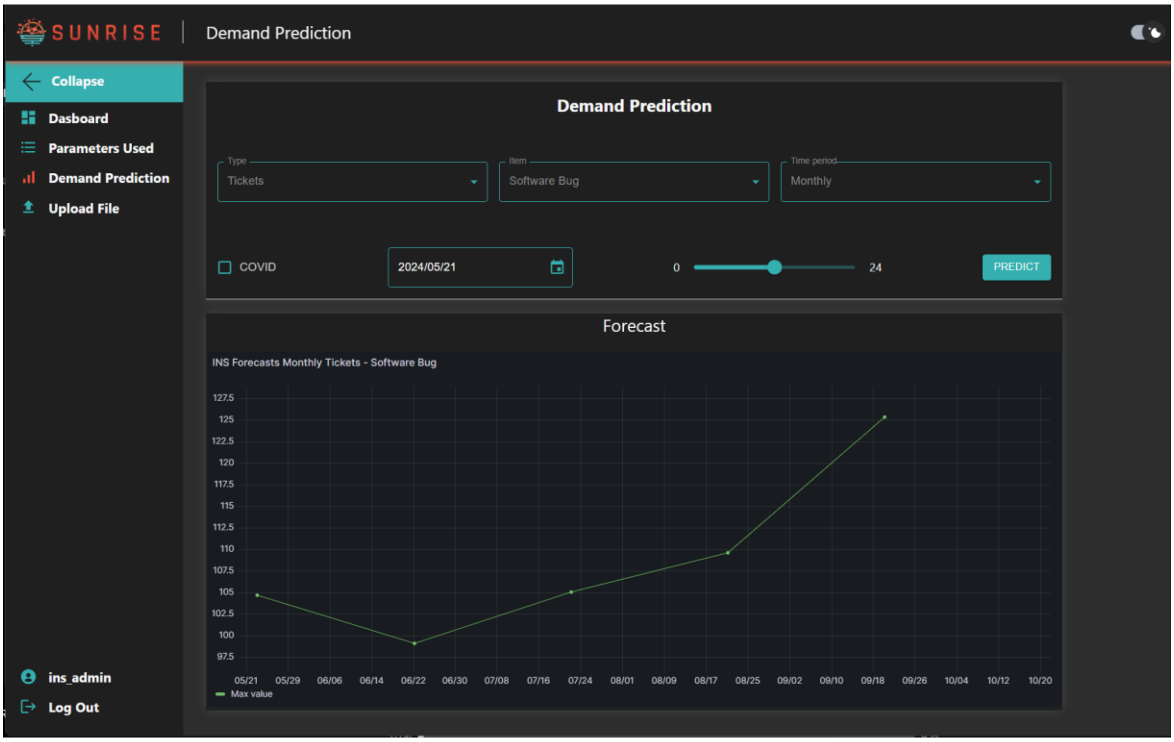


Figure 4: Demand Prediction view for INSIEL.

The final view of the WP5 Tool's User-Interface is the "File Upload" view, , as illustrated in Figure 5 In this view the User can upload and store new datasets in the MinIo [8] based repository of the WP5 Tool, review the uploaded files and delete them. Specifically, after the upload and delete transactions, a relevant message is shown to inform about the user of the outcome of the transaction (Figure 6).

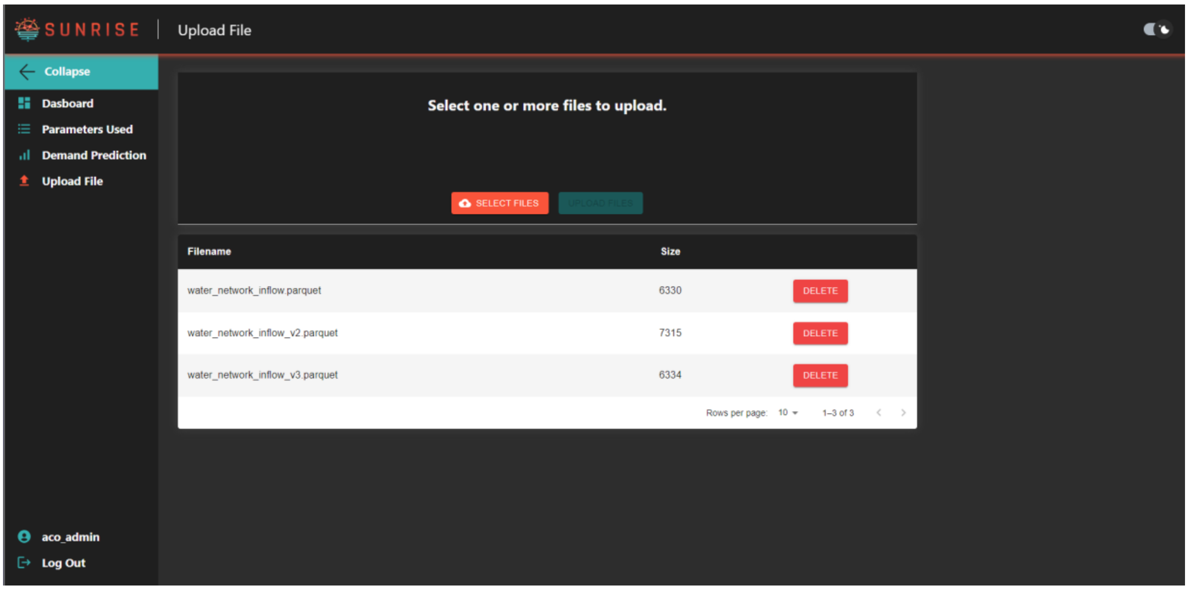


Figure 5: File Upload view.

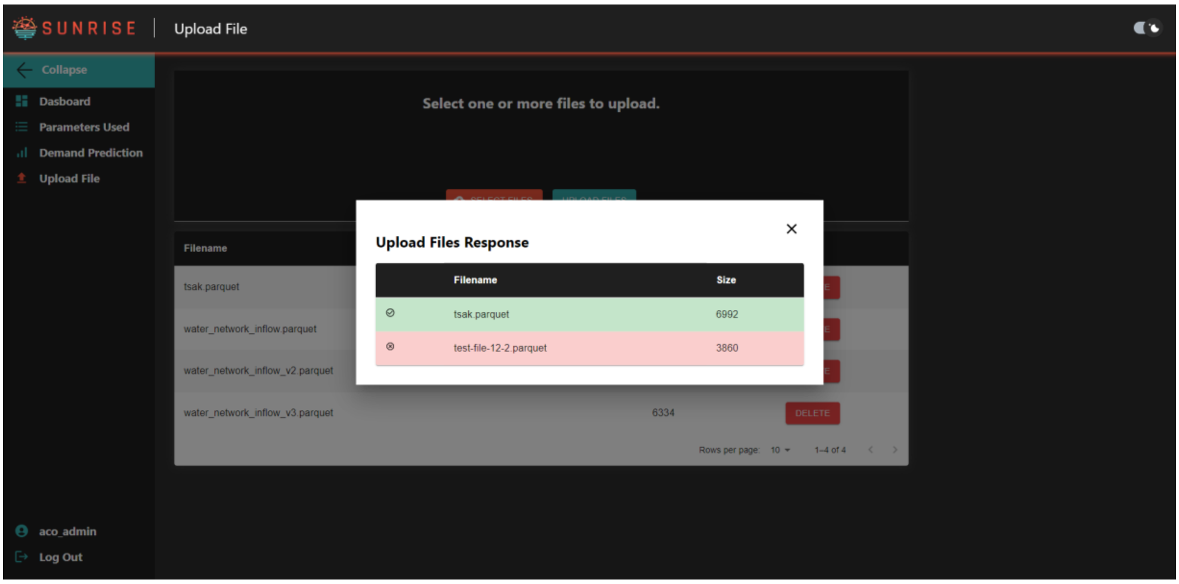


Figure 6: Example message displayed after multiple data upload.

2.2.1.2 Components and techniques

The views previously presented are the main part of the front-end component. However, there are several components and techniques used during the development of the User-Interface that are not directly related to the front-end but assisted its development. These components and techniques later, during the integration phase of the Tool, proved important as they contributed to the success of the WP5 Tool components integration. The first example of these is the use of Keycloak. Keycloak normally resides in the back-end, however, its use was crucial for the development of the front-end part of the WP5 Tool, as it provides the ease of a secure and flexible way to perform user authentication. In addition to the user authentication, Keycloak provides a graphical User-Interface for managing Users, i.e., add/delete the applications' Users. Another critical issue that Keycloak solved, was the User authentication in other Tool's components such as Grafana [9] and MinIo.

To produce the various graphs the Grafana component was used. Grafana provides an easy and flexible way of data plotting which is the main scope of the WP5 Tool with several Cis. The built-in hooks as well as the numerous plot types that Grafana offers make it the perfect candidate for the WP5 Tool. In the WP5 Tool, Grafana dashboard components were embedded in the views previously presented.

In the design of any platform, how data is stored emerges as a primary consideration. This is especially apparent in the case of the WP5 Tool, which involves various types of data. In addition, the Front-end should be highly adjustable to meet all the requirements introduced by the different sectors. In this context, in the WP5 Tool, the use of Elasticsearch [10] was selected. In Elasticsearch the data is stored in documents, and it provides hooks and libraries in several programming languages and frameworks so that all WP5 components can support it with low effort. Grafana and Elasticsearch play a key role in automating the storage and visualization of data from diverse sources.

2.2.2 Back end

Backend (BE) works with several components that collectively support the operations of an AI application. These components include various engines and databases, as well as a data lake, orchestrated by a Cron Engine, and tracked by Mlflow [11] for experiment tracking. The integration of these components is illustrated in the following Figure (Figure 7).

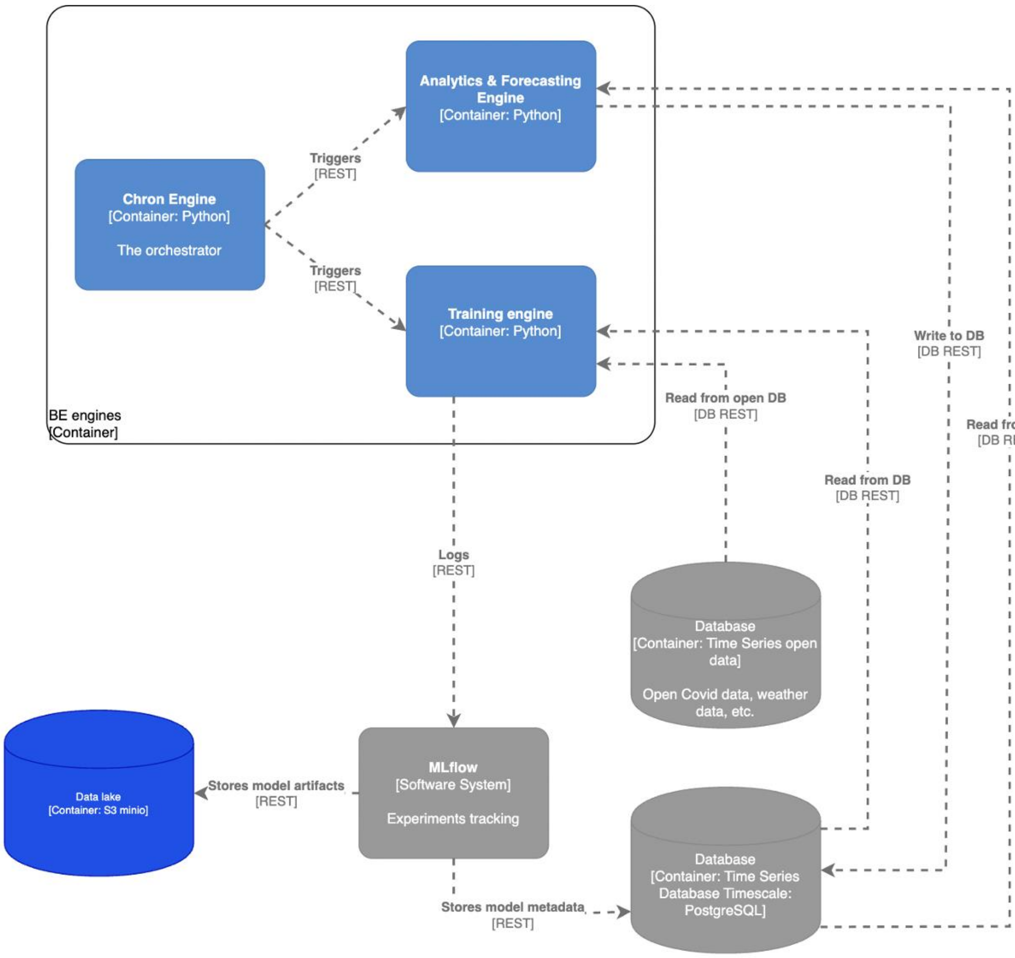


Figure 7: Back-End Architecture

The BE includes different services each one of them handling specific tasks. A cron engine orchestrates these different services in order to run. Each one of the services has several functionalities inside each one of them constitutes a pipeline (Figure 8).

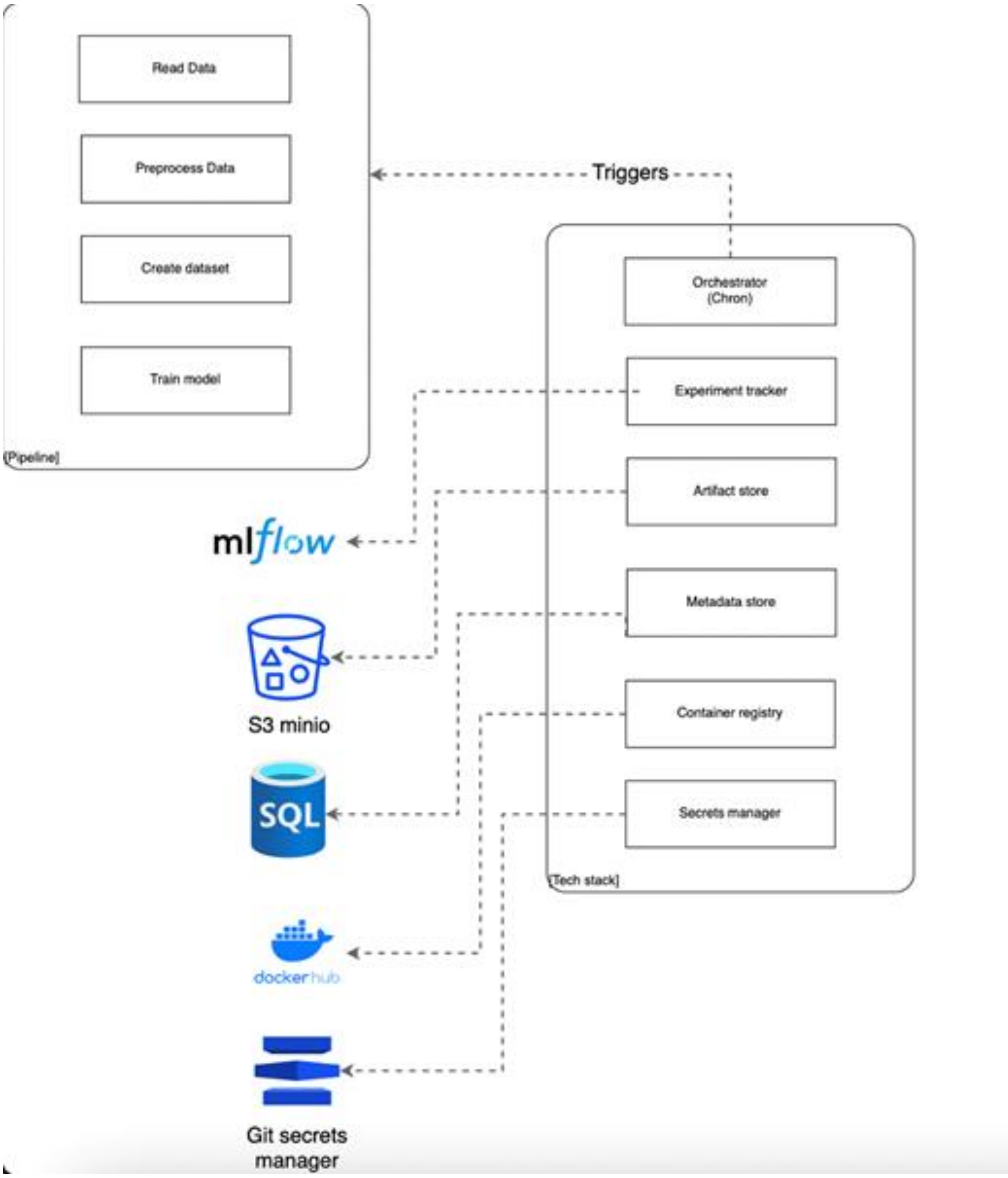


Figure 8: BE services high level flow-chart

Model Training

Pipeline:

- Read Data: Retrieving data from database where it is already prepared by the pipelines.
- Preprocess Data: Data is normalized and transformed to be suitable for training.
- Create Dataset: Processed data is organized into a dataset for training purposes.
- Train Model: The AI model is trained using the prepared dataset.

Technology Stack:
- Experiment Tracker: Tracks experiments, parameters, and results (MLFlow).
- Artifact Store: Stores model artifacts (MLFlow artifacts).
- Metadata Store: Manages metadata for models, datasets, etc. (MLFlow Metadata).
- Container Registry: Stores Docker images used for deployment (Docker Hub).
- S3 Minio: An open-source S3-compatible storage for artifacts and data.
- Git Secrets: Ensures sensitive data is not pushed to version control.

Integration of the training pipelines and services

The integration process involves connecting these services to work seamlessly together. A few operations are described briefly below:

- Data ingestion and preprocessing steps are automated with scripts that can handle different data formats and sources (CI-oriented data, weather data, COVID-19 data). This involves integrating with data lakes and databases for input and output.
- The dataset creation and model training steps are defined within an ML pipeline, potentially orchestrated by a custom orchestrator solution (cron).
- Experiment tracking, artifact, and metadata management are integrated internally in BE through the Mlflow platform, which can be configured to track experiments, store model artifacts, and manage metadata across different stages of the ML lifecycle.
- Sensitive information, such as credentials and API keys, is managed via a secrets manager and injected into the pipeline securely at runtime.
- The entire process is version-controlled using Git, with safeguards like Git secrets to prevent accidental exposure of sensitive data.

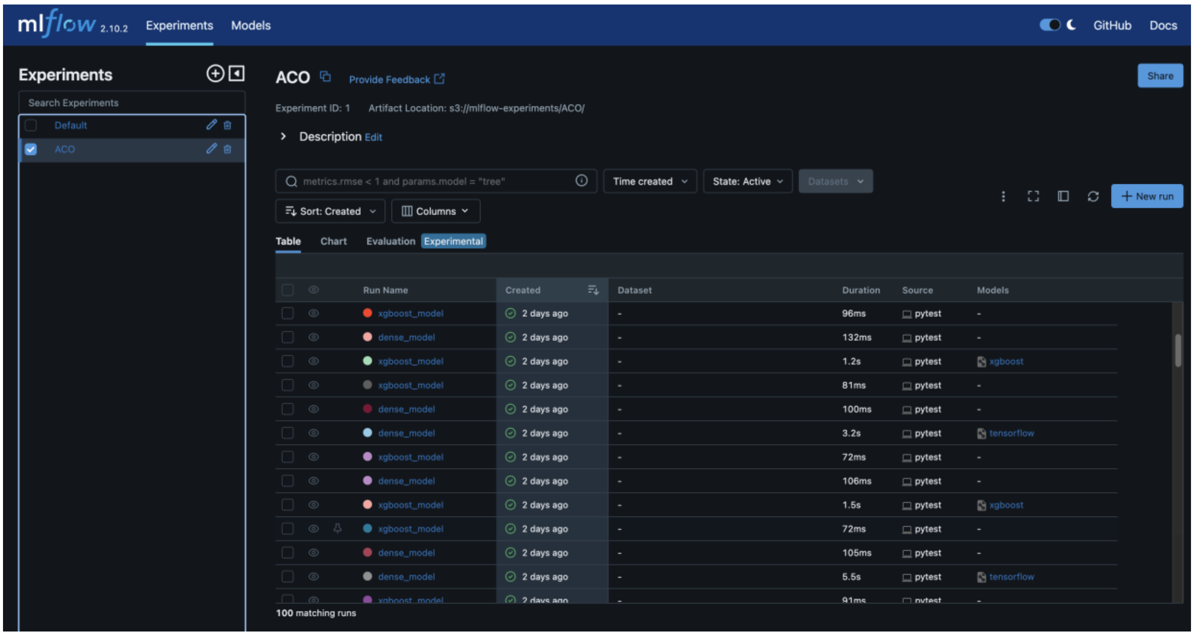


Figure 9: MLFlow experiment tracking

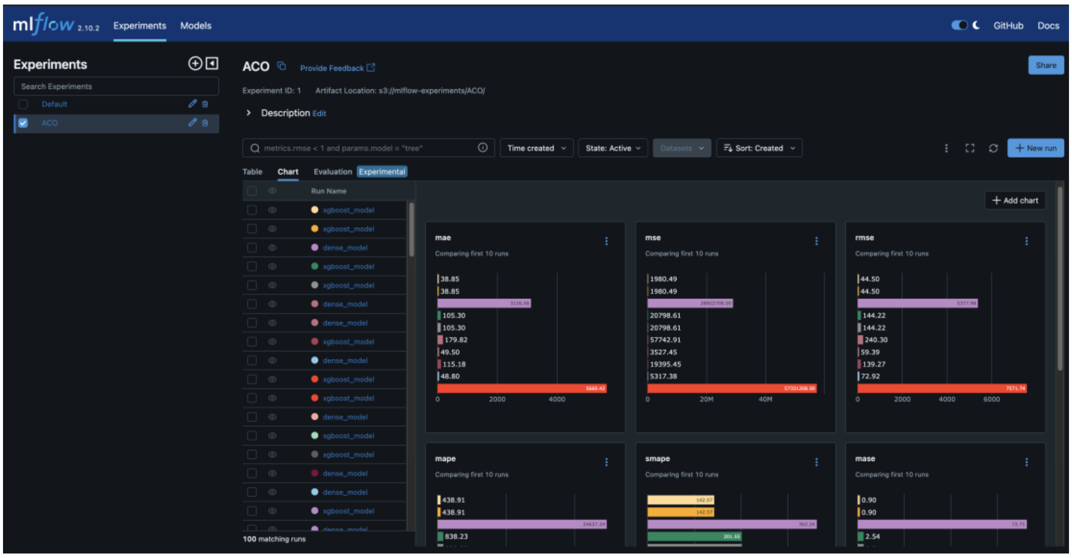


Figure 10: Model metrics in Mlflow

Service architecture and API access in the transport domain

The DPM tool for the transport domain comprises a set of modular services accessible via an API. Key services include the Elasticsearch database, Mlflow model registry, and prediction service, each housed in a Docker container and deployed through Docker Compose. This modular design facilitates independent scaling and cloud deployment, allowing for shared services among applications.

The prediction service is the core of the DPM tool. It operates as a REST API. It enables querying prediction models, training new models, and managing historical data. Built with FastAPI, a Python framework, it ensures the creation of efficient, lightweight APIs. A notable feature of FastAPI is its automatic API documentation generation, providing a web interface detailing API endpoints, parameters, and responses for both documentation and testing purposes. Figure 11 shows the API documentation of the prediction service.

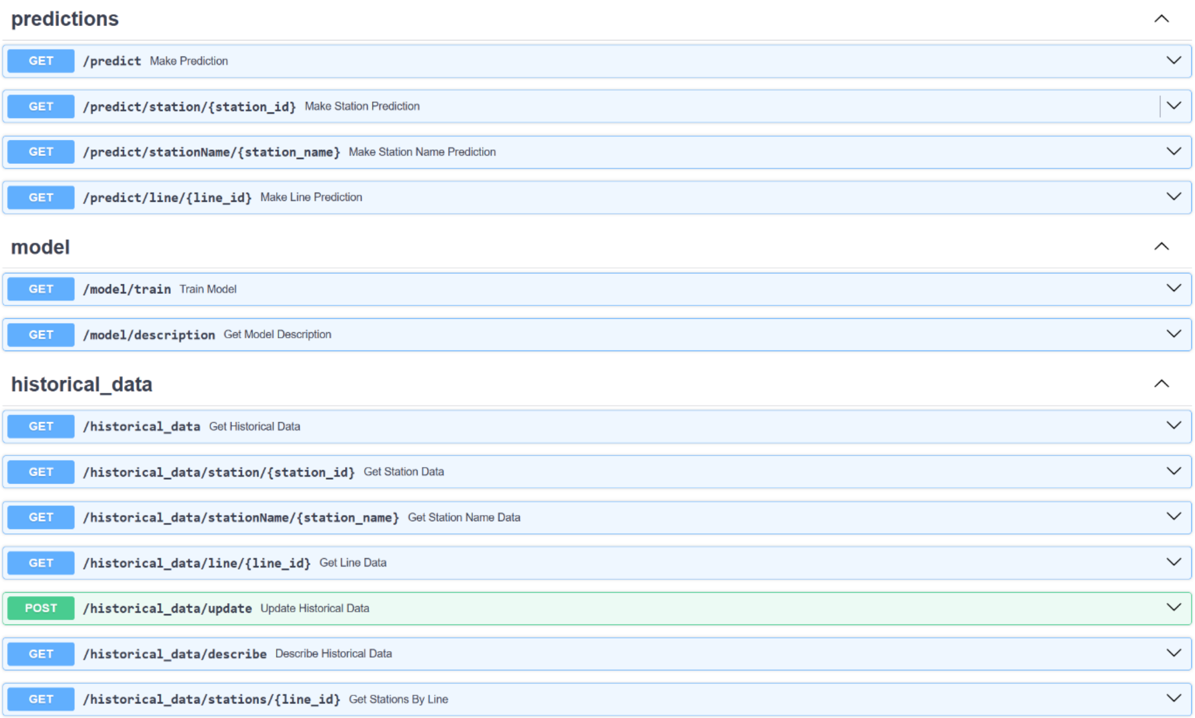


Figure 11: DPM tool transport API

The initial group of endpoints, located under "/predict," facilitates access to and queries of prediction models. The "/predict" endpoint queries the global model. "/predict/station/{station_id}" and "/predict/stationName/{station_name}" are used to query the station level models given its id or name respectively. For querying a line level model, "/predict/line/{line_id}" is available. This organizational scheme is consistent with that of the data management endpoints discussed later. Each endpoint accepts an optional "targetDate" parameter for generating forecasts for a specific date in the YYYY-MM-DD format. If the date is not provided, the prediction is made using the last data available in the database with a prediction horizon of 7 days. If the date is provided and it is in the range between the last data available and the horizon, the prediction is made using the data available in the database. If the requested forecast date extends beyond the model's one-week horizon and given that the model relies on the previous 15 days of data for its predictions, the system calculates intermediate forecasts to serve as new inputs. The system iteratively generates and uses these intermediate forecasts, treating each as the latest set of 15-day inputs to predict the next week's value, continuing until it reaches the requested date.

The second endpoint group, located under "/model" is used to train new models and manage the model repository. The "/model/train" endpoint initiates the training process for a new version of models. This process is not synchronous as it may take some time to complete. The "model/describe" endpoint provides a detailed description of all the models stored in the Mlflow registry.

Finally, the group of endpoints under "/historical_data" provides access to the historical data stored in the ElasticSearch database. The "/data/station/{station_id}" and "/data/stationName/{station_name}" endpoints retrieve the historical data for a specific station given its id or name, respectively. The "/data/line/{line_id}" endpoint retrieves the historical data for a specific line. These endpoints return the number of passengers entering the station or line for each day in the database. The data is returned in JSON format, with each entry containing the date and the number of passengers. The "/historical_data/update" endpoint allows users to update the historical data in the database uploading a CSV file with the new data. Last, the "/historical_data/describe" endpoint retrieves the information about the scenario stored in the database, including the stations, with their id, name, and the lines it belongs.

3 Demand Prediction Methods & Validation

3.1 Energy domain

3.1.1 Introduction and Methodology

This deliverable builds upon the findings of D5.1[2] and D5.2 [3] which compared various state-of-the-art demand prediction models using publicly available energy load data. Here, we propose to apply the best performing model identified in D5.2 [3], the Temporal Fusion Transformer, to pilot data for state-wide load demand forecasting. Consistent with D5.2 [3], our focus in this deliverable will be on day-ahead predictions of state-wide electrical energy demand with a one-hour resolution.

Additionally, we will try to improve the existing approach by adding additional features such as weather, holiday, and datetime features to inject additional information into the pipeline. These features should encompass factors that impact consumption most. Weather is an important factor when it comes to heating in the winter and cooling in the summer. Storms, more specifically lightning storms, may cause outages. Another important factor is whether a day is a workday or not. Besides specifying whether a day is a weekend or holiday, we will calculate the number of consecutive off-working days, as there is a higher chance that people will go on vacations on such occasions.

Before analysing the model's performance under pandemic conditions and implementing potential improvements, we will evaluate its performance against currently deployed forecasting models from one of the pilots (EKC).

More details regarding the importance of accurate energy forecasts and a detailed description of Temporal Fusion Transformer and data used can be found in the past deliverables D5.1[2] and D5.2[3].

3.1.2 Results

The results below present the performance of our models under various settings. All models incorporate datetime features and historical weather data. Table 2 highlights three model variations where we adjusted the input window size from 24h to 168h and added future weather forecasts.

For a fair comparison, the EKC baseline model employs the same input data as our models, including holidays and weather data. The training data encompasses three years, from 2019 to 2021, and the model's performance is evaluated on data from 2022.

Table 2 shows that the EKC baseline model used a 24-hour input window and excluded future weather data. In our first experiment, we employed a larger version of our model with a 168-hour input window (without modelled weather). This resulted in a significant improvement over the existing approach, with accuracy gains of up to 7%. The second experiment further improved our performance, achieving a total MAPE decrease of 17%. We accomplished this by combining a 24-hour input window with modelled weather forecast features that mirrored the measured weather data. Finally, our third experiment demonstrated the most significant improvement with a 20% decrease in MAPE. This was achieved by utilizing future modelled weather forecasts and extending the input window to 168 hours.

Table 2: Results between EKC and XLAB

Experiment	Model	MAPE [%]	Forecasted Weather	Window Size [h]
Baseline	EKC Baseline	2.34	No	24
1.	XLAB TFT	2.18 (- 7 %)	No	168
2.	XLAB TFT	1.93 (- 17 %)	Yes (Belgrade only)	24
3.	XLAB TFT	**1.87 (- 20 %)**	Yes (Belgrade only)	168

Our results demonstrate that combining an improved model architecture (TFT), a larger input window (1 week), and forecasted weather data significantly enhances the existing state-of-the-art approach used by EKC.

More in-depth results show a similar pattern in Table 3. Similarly, as with MAPE, we managed to reduce MAE by 20 %. Furthermore, similar improvement can be observed when we look at standard deviation where distribution is narrower for both metrics. Lastly, our model made the smallest maximum error.

Table 3: Detailed table of results

	MAPE_X [%]	MAPE_EKC [%]	MAE_X [MWh]	MAE_EKC [MWh]
count	9563.00	9563.00	9563.00	9563.00
mean	**1.87 (-20 %)**	2.34	**73.58 (-20 %)**	92.24
std	**1.65**	2.075	**67.53**	91.00
max	**20.05**	29.60	**860.00**	3540.00

The very same comparison can be performed visually as we can observe in Figure 12. Here we can visually observe that the orange (XLAB) MAPE error plot on the on is on average lower than the blue (EKC) and has fewer spikes, additionally confirming the results. In the same manner, we can observe a standard deviation plot on the right, where the blue distribution is taller and narrower suggesting more predictions were closer to 0.

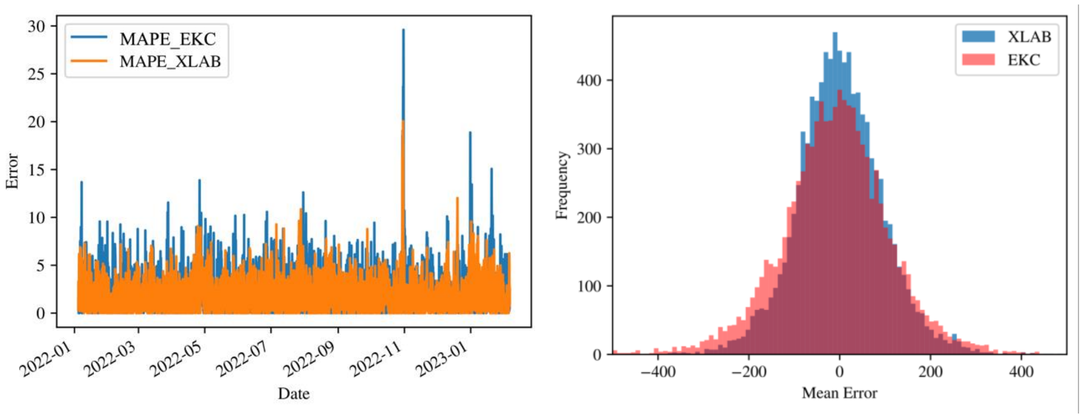


Figure 12: Detailed analysis of the results of the best performing model

In Figure 13 below we can observe actual predictions and actual values for the XLABs TFT model on the left and for the EKC model on the right. The detailed section focuses on the easter holidays, where, compared to other parts of the year we can observe relatively high error. Visual observation suggests that XLAB model demonstrated smaller error compared to the EKC model.

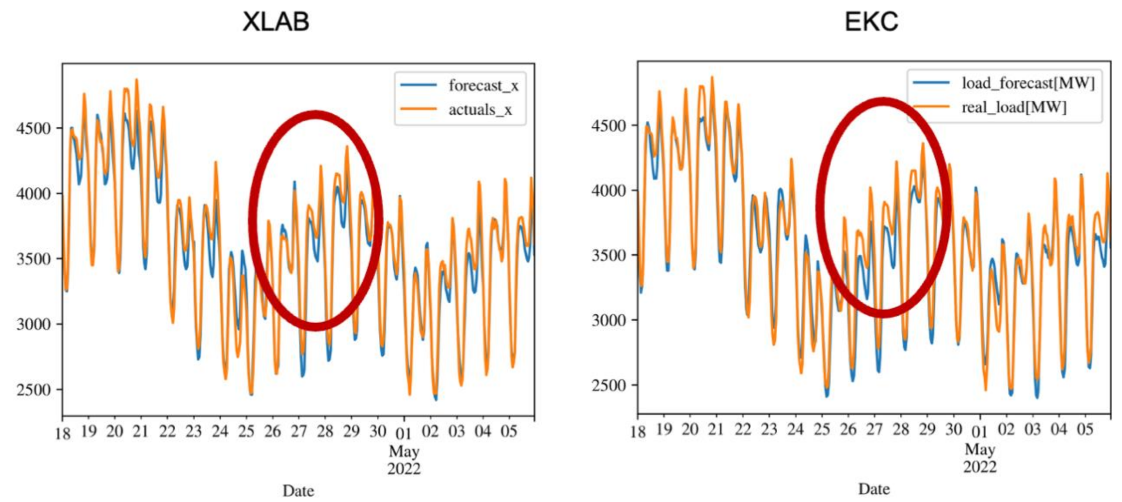


Figure 13: Performance comparison during the Easter holidays in Serbia

As part of SUNRISE, we are set to develop a resilient model, which will not fail in cases of substantial changes in consumption patterns. Holidays, present such quick changes in consumption patterns, but are compared to pandemics or other large-scale events more predictable.

Having a model that would work better during the holidays presents a great starting point for developing demand prediction models that would be resilient to large-scale events such as pandemics. Furthermore, in the case of day-ahead prediction, large-scale events and lockdowns are known. Such information could be added as an additional feature and improve models' performance during times of decreased consumption patterns.

3.1.3 Conclusions

These results confirm that our model effectively competes with production-level models and can significantly improve upon current approaches. With a 168-hour input window and future weather forecasts, we achieve up to a 20% improvement. Future weather forecasts allow the model to focus on the weather's impact rather than for example predicting storm patterns from past pressure changes. Similarly, the weekly input window provides valuable context contributing to the improvement of predictions. Importantly, our model demonstrates a MAPE of approximately 2%, exceeding the project KPI of model MAPE being below 30%.

This sets a ground floor for future research as we have well-performing model. To continue refining our approach, we propose evaluating its performance on Slovenian pilot data, implementing iterative learning strategies, and finalizing the model with the most impactful features in preparation for real-world deployment. The first step would be to evaluate the performance of our model during COVID-19 and identify features and methods that would improve its performance during times of big changes in consumption patterns.

3.2 Transport domain

3.2.1 Introduction

The transport domain shares many similarities with the other domains in the SUNRISE project. All of them are subject to dynamic changes that may affect the demand for services or resources. For all of them, the DPM tool is a key component to forecast the demand in a short-term horizon, but with enough time to react and optimize the resources. The transport domain, like the others, is also subject to various external factors, such as weather conditions or healthcare restrictions, that may have a drastic impact on the operation. The forecasting tools should be able to consider these factors to provide accurate predictions.

However, the transport domain has some particularities that make it unique. The main characteristic that differentiates the transport domain from the others is the network structure. Passengers move through the network, entering and exiting at different points. The vehicles also move through the network, following predefined routes and carrying passengers. This network dimension means that the events in one part of the network may have an impact in other parts. For example, the closure of a station may produce a change in the routes that the passengers take, affecting the demand in other stations. The DPM tool should be able to take into account these interactions to provide accurate predictions.

In this version of the tool the effort has been focused on the improvement of the prediction model and provide a service architecture that enables the historical data management, model versioning and prediction services. The algorithm used to train the prediction model is the same as reported in D5.2 [3]. Elasticsearch [10] has been selected as the database to store historical data and the prediction results. Mlflow [12] is used as a model repository to store and serve the models and the results of the training process. The system is accessible through a Python API, created with FastApi [13], that allows to query the prediction model and operate with the historical data in the database. The system is encapsulated In a Docker [14] container and deployed as a Docker-Compose service. The system is designed to be scalable and to be able to operate in a cloud environment.

This design has been assessed with the historical datasets provided by the transport domain Critical Infrastructures CRTM and TT. The results show that the prediction model is able to provide accurate predictions of the demand in the network. The system has ben also been tested with new data to validate the data management services. Overall, the system is ready to be used in the pilot phase of the project as described in Section 5.1.2.

3.2.2 Model

Both Critical Infrastructures partners involved in this work are operators of public passenger transport networks. The demand for these kinds of networks correlates to the number of passengers that use the network in a given period of time. Other variables can be derived from this measure, such as the number of passengers in each station, the number of vehicles needed to transport the passengers, etc. This domain is very dynamic, and the demand for public transport can change in a short period of time. This implies that the recent data is more relevant than the old data to predict the demand. It is also very seasonal, with different demand patterns on different days of the week, hours of the day, etc. For example, the demand of the network is very different on a working day than in a weekend day. The demand is also affected by external factors, such as weather conditions, events, etc. The prediction should be made in a short-term horizon, but with enough time to react and optimize the resources.

With these considerations, the prediction model has been designed to take into account the data from the last 15 days to predict the demand with 7 days ahead. The input of the model also includes the date for which the prediction is made, to consider the seasonality of the demand. Finally, the input includes a feature that indicates if the day is a holiday or not, to consider the effect of the holidays in the demand, i.e. when the demand is more similar to the weekend period than to a working day.

The transportation networks are made of stations and lines. The stations are the points where the passengers enter and exit the network. The lines are the routes that the vehicles follow to transport the passengers. The model schema described above is used to predict the demand in different levels. First, a global view of the network that aggregates all the stations. Second, a line-view that predicts the demand in each line of the network. And finally, the most fine-grained view, the station view, that predicts the demand in each station. All models are trained in the same way, but with the historical dataset aggregated to the level of the prediction.

As described in D5.2 [3], the algorithm that has shown the best results in this domain is xGboost. This algorithm can be adapted to many problems and it is commonly used in other similar works at this domain [15][16][17]. The algorithm can be adjusted to the specific problem by tuning its hyperparameters. This tuning is made with a grid search, testing the different sets of values. For the case of the RTM dataset, the best results have been obtained with the hyperparameters shown in Table 4.

Table 4: Hyperparameters of the xGboost algorithm

Parameter	Value
learning_rate	0.1
max_depth	5
n_estimators	100

The model has been trained with the historical datasets provided by the CRTM. The historical dataset contains the card validations of the passengers that entered and exited the network during years 2019 to 2021 It covers the operation of line 12 of the Madrid's metro network which is a circular line with 28 stations. The station "Puerta del Sur" belongs also to line 10 and is the connection point with the rest of the metro network. Table 5 shows the results of the model trained with this dataset. The models have been evaluated using the Mean Absolute Percentage Error (MAPE) metric, aligned with the project KPI for this WP. To ensure enough data for the model training, the dataset has been split in two parts, the first part includes one year of data, and the second part includes the rest of the data. The second part has been used to perform a time series cross-validation [18] with 5 folds, but always including the first year of data in the training set. The results show that the general model obtains a MAPE of 13.71%, the line models an average MAPE of 14.58% and the station models an average MAPE of 17.74%. The detailed results of the model evaluation are shown in Table 5 and Figure 14. In the evaluation of station-specific models, certain models, such as those for "Arroyo Culebro" and "Hospital de Fuenlabrada," displayed significantly poorer performance compared to others. This discrepancy in model performance can be attributed to the uniform application of the same time span for evaluation across all models, without accounting for unique circumstances at individual stations. Specifically, some of the stations were closed for maintenance during the evaluation period, leading to missing data and thus an ill-defined metric for assessing model performance. When these models operate under normal conditions, with stations fully functional and data availability consistent, their performance is aligned with that of other models.

Table 5: Results of the model evaluation

Global-view		
Model	MAPE	
crtm-global	13.71%	
Line-view		
Model	Description	MAPE
crtm-line-10	Line 10	15.44%
crtm-line-12	Line 12	13.71%
	Avg	14.58%
Station-view		
Model	Description	MAPE
crtm-station-12-3	Alcorcón Central	13.13%
crtm-station-12-19	Alonso de Mendoza	13.85%
crtm-station-12-17	Arroyo Culebro	30.23%

crtm-station-12-27	Casa del Reloj	14.43%
crtm-station-12-18	Conservatorio	18.25%
crtm-station-12-24	El Bercial	12.24%
crtm-station-12-25	El Carrascal	10.81%
crtm-station-12-22	El Casar	13.95%
crtm-station-12-14	Fuenlabrada Central	22.72%
crtm-station-12-20	Getafe Central	23.81%
crtm-station-12-28	Hospital Severo Ochoa	13.79%
crtm-station-12-12	Hospital de Fuenlabrada	45.38%
crtm-station-12-8	Hospital de Móstoles	13.07%
crtm-station-12-21	Juan de la Cierva	12.37%
crtm-station-12-26	Julián Besteiro	13.26%
crtm-station-12-29	Leganés Central	14.28%
crtm-station-12-10	Loranca	20.88%
crtm-station-12-23	Los Espartales	13.66%
crtm-station-12-9	Manuela Malasaña	22.99%
crtm-station-12-6	Móstoles Central	12.18%
crtm-station-12-13	Parque Europa	26.34%
crtm-station-12-4	Parque Oeste	24.13%
crtm-station-12-2	Parque de Lisboa	11.44%
crtm-station-12-15	Parque de los Estados	24.93%
crtm-station-12-7	Pradillo	12.65%
crtm-station-10-21	Puerta del Sur	15.44%
crtm-station-12-30	San Nicasio	15.20%
crtm-station-12-5	Universidad Rey Juan Carlos	24.13%
	Avg	17.74%

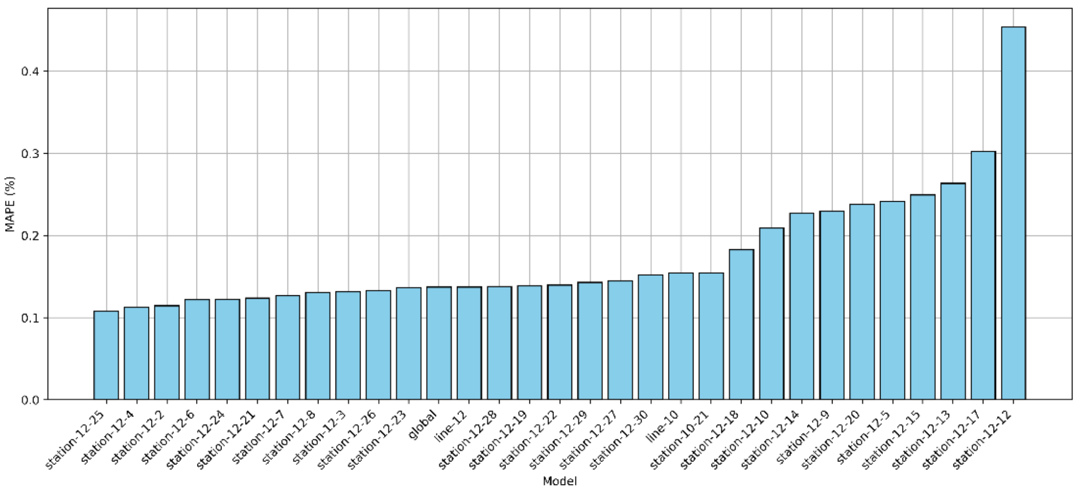


Figure 14: Results of the model evaluation

3.2.2.1 Database Integration for Historical Data Management

The historical data that the model uses to train and predict the demand is stored in an Elasticsearch database. Elasticsearch has been chosen to ensure the integration with the user interface and the tool version for other domains. To facilitate the application of the tool to generic scenarios, the data schema is designed with the minimal information needed to train the model. The data schema is shown in Table 6. The main collection stores the number of entering passengers per station and day. Each station should have a unique identifier. The station details are stored in a separate collection that describes the scenario. The secondary collection includes the name of the station, its identifier, the lines to which it belongs, etc.

Table 6: Data schema of the Elasticsearch database

Field	Type
Scenario description collection	
station_id	text, keyword
station_name	text
lines	keyword (list)
Historical data collection	
station_id	text, keyword
date	date
value	long

3.2.2.2 ML Models Registry with mLflow

mLflow is a software platform used to manage the lifecycle of machine learning models. It provides a tracking system to record and compare the experiments and the models, a model registry to store and serve the models, and a deployment system to deploy the models in a scalable way. In the transport domain it has been used to track the training parameters and the results of the models (Figure 15).

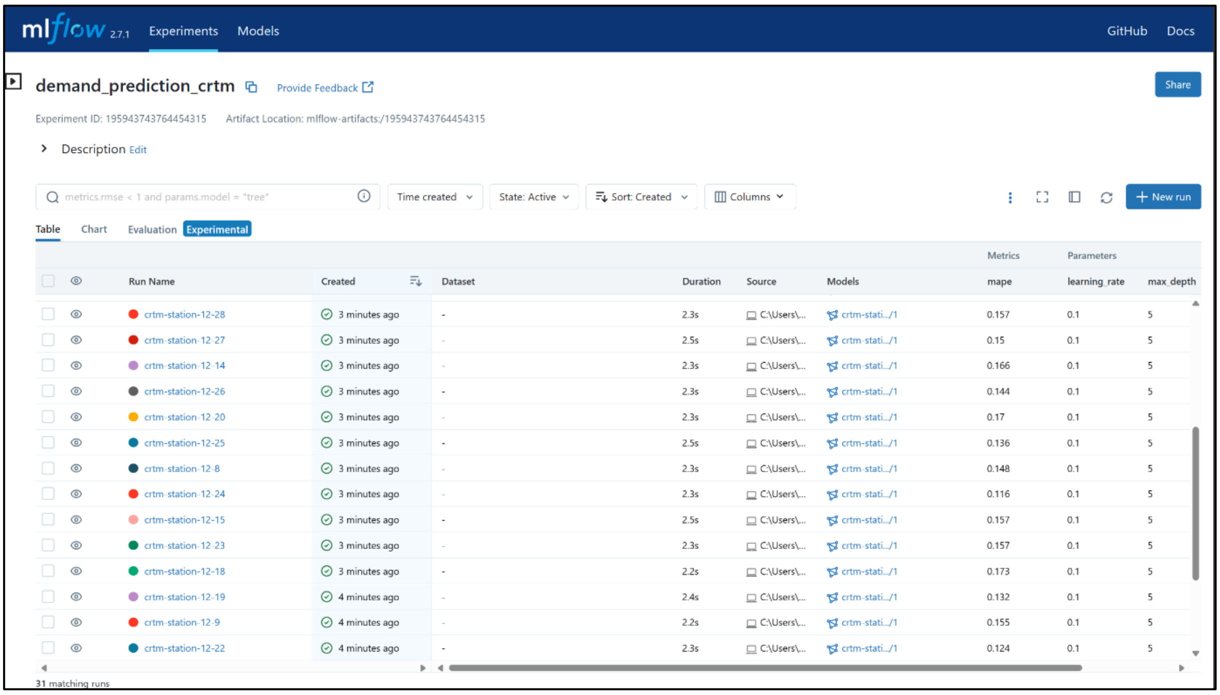


Figure 15: mLflow experiment tracking

Each model is registered in the platform including the hyperparameters used to train it and its performance metric. Each model is identified with a keyword that describes its level as described in the previous section, and the identifier of the station or line (Figure 16).

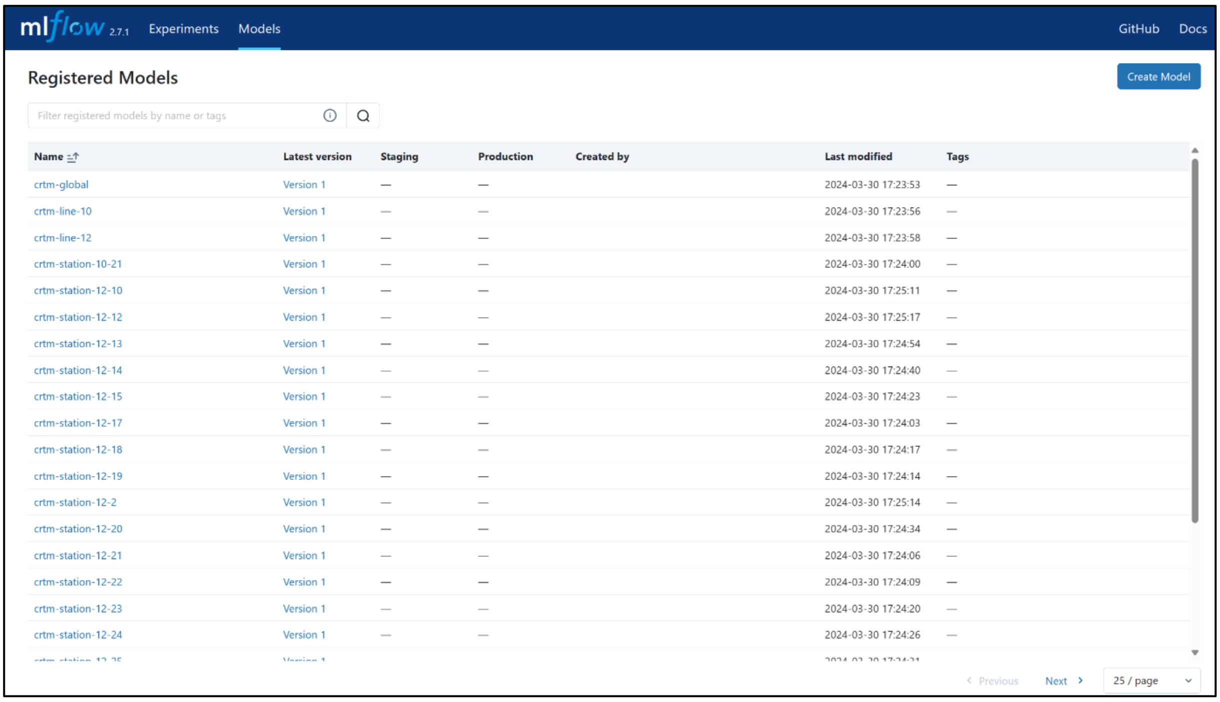


Figure 16: mLflow model registry

3.2.3 Conclusions

This section summarizes the achievements and outlines future directions for the development and application of DPM tool within the transportation domain. The models developed during this iteration are fully operational and align with the project's KPIs, demonstrating their reliability and effectiveness in forecasting transport demand. The models have been designed to predict passenger demand across multiple perspectives, including station-specific, line-specific, and network-wide views. These views allow the DPM tool to be used to cover different analytical needs within the transport network.

The prediction models are designed to forecast the number of passengers expected to utilize the transportation network within a one-week horizon, providing a sufficient window for operational planning and resource optimization. The forthcoming pilot phase 1, described in Section 5.1.2, will be used to present the DPM tool to the SUNRISE partners of the transportation sector. This piloting stage will be used to demonstrate the tool's capabilities and gather feedback from the CIs, which will be used in refining and enhancing the tool.

SUNRISE has achieved having a set of operational models that can accurately predict passenger demand. The next iteration will focus on analysing passenger flow within the network. This presents a graph analysis problem, important for understanding movement patterns and densities that can greatly impact operational strategies. One of the key resources in transportation networks are vehicles, which have a capacity that represents the number of passengers they can carry. This capacity may be variable, affected by different factors, such social distance in pandemics. Climate events may also produce peaks or drops in demand, which will have a direct impact on the number of vehicles needed to handle the demand. The next objective of the tool in this sector is to estimate the number of vehicles needed to accommodate the predicted demand under different scenarios. This will allow the CIs to optimize their resources and adapt to changing conditions.

3.3 Health domain

3.3.1 INSIEL

INSIEL has provided several information about their use case as described in D5.2 [3]. Unnecessary data has been excluded from the analysis and the finetuned objective of the DPM tool for this case is to create forecasts through ML/AI models on digital documents xRays, REFE, LDO and the number of tickets from the IT Support Help Desk. The digital documents have a daily frequency, while the tickets have an hourly one.

In the development of ML models by SQD, the datasets were divided using a split ratio of 10% for testing and 90% for training to ensure robust model evaluation. Additionally, our models incorporated several exogenous variables to improve the accuracy of the forecasts. These variables are external factors that influence predictions but are not directly part of the primary data's feature set. The exogenous variables included:

- Stringency Index: This is a composite measure based on several indicators of government responses to COVID-19 (such as school closures, workplace closures, and travel bans), which can affect organizational operations and, by extension, the data we are forecasting.
- Holidays: Recognizing public and organizational holidays that could impact daily operations and data generation patterns.
- Seasonal Data: Taking into account the seasonal variations that might affect the data, such as quarterly business cycles or seasonal peaks in IT issues.
- Weekdays: Considering the day of the week, as certain days may have different operational tempos (e.g., higher IT ticket volumes on Mondays).
- Rush Hours: Figuring out the busiest times in the day when we get the most IT support tickets.

Furthermore, we employed window sizes of seven time units for incorporating lagged values into our models. Lagged values are historical data points that are used as input to forecast future outcomes. A window size of "seven" means the model uses data from the previous seven-time units—whether days or hours—to predict the next one. This method helps capture trends and patterns from the immediate past, which is critical for accurate forecasting in time series analysis. So, for example "value-1" is the 1-lag value, "value-2" is the 2-lag value and so on.

For each document and ticket, we conducted experiments using a variety of models, including XGBoost, LSTM, Feed-Forward Neural Networks, Random Forest, and NBEATS. To prevent overfitting, we employed techniques such as Dropout and Early Stopping, along with the use of a validation set to test the models' generalization capabilities. Dropout randomly deactivates certain neurons during training, which encourages the model to distribute learning across a broader set of features. Early Stopping monitors the model's performance on the validation set and stops training when no improvement is observed over a specified number of iterations. These methods ensure our models remain robust and perform well on unseen data.

A brief overview of the models mentioned:

- XGBoost: Stands for Extreme Gradient Boosting and it is a powerful machine learning algorithm that uses decision tree ensembles and. By combining many weak learning models together, it produces a more accurate and robust model. It's particularly good at handling large datasets and is known for its performance and speed. It is configured with a learning rate of 0.1, which controls the contribution of each tree to the outcome. It has a maximum depth of 3, to prevent over-complex trees, and a total of 200 estimators.
- LSTM: Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) particularly useful for processing sequences of data. They are capable of learning long-term dependencies because they have a 'memory' that captures information about previous

data points. The LSTM model's architecture is about 64 neurons, with the exact number of layers depending on the input shape. The model is trained for 100 epochs, and the batch size vary. EarlyStopping is the callback employed to ensure effective training duration.

- Feedforward Neural Network: Referred as Neural Network, this model is simplest type of artificial neural network. In this architecture, the information moves in only one direction—forward—from the input nodes, through the hidden nodes (if any), and to the output nodes. There are no cycles or loops in the network, hence the name 'feedforward'. The model predicts one time unit ahead and has two layers with 128 and 64 neurons respectively. It is trained for 100 epochs with a batch size of 128 and uses EarlyStopping callback.
- Random Forest: A Random Forest is an ensemble learning method that operates by constructing multiple decision trees during training and outputting the mean prediction of the individual trees. It is set with a maximum depth of 10, ensuring that the decision trees do not grow too deep and overfit the data. It considers at least one sample to split a node (min_samples_split) and at least one sample in a leaf (min_samples_leaf), with a total of 300 trees in the forest (n_estimators).
- NBEATS: Stands for Neural Basis Expansion Analysis for Time Series and is a recent forecasting method designed specifically for time series data. It uses a deep neural architecture that provides a forward-looking prediction based on historical time series data. It does not rely on recurrent or convolutional layers, differing from many traditional neural network models used for time series forecasting. The NBEATS model set up has 512 neurons per layer, 4 layers in total, 30 stacks, and is trained for 5000 epochs with a batch size of 1024. The training uses callbacks such as ReduceLROnPlateau and EarlyStopping to adjust learning rates and prevent overfitting respectively.

3.3.1.1 Tickets

There are four different categories of tickets available: Generic Support, Software Bugs, Workstation and Peripheral Support, and Account Management. For each category, SQD conducts experiments with various timeframes and factors. The stringency index is transformed into bins with levels 0, 1, and 2, indicating Low, Medium, and High stringency accordingly. For each timeframe, we experiment with different look-back values. Specifically, for hourly analysis, we use a 6-hour look-back period; for daily analysis, we use a window size of 7 days; and for monthly analysis, we use a window size of 6 months.

In an hour-based timeframe we extracted the rush hour and found that the number of tickets depends on the day and the hour. We also added holidays in our analysis, but we found no important correlation between holiday and the target variable.

The best model found in all categories is the custom LSTM model. Due to many values close to zero the MAPE metric is extreme high, and RMSE is more appropriate metric in a such a case.

After resampling the data in daily bases for each category, new variables have been included like whether the day is a weekday, holiday, and the stringency category index. For monthly data we have included the total holidays of each month, and the stringency category index.

Generic Support

For the “Generic Support” category, in an hourly based timeframe we found that there is a moderate positive correlation of 31.60% between a weekday and the number of tickets. We also found that there exists a strong positive correlation between rush hours and the number of tickets. In addition, there exist strong positive correlations with the lagged values ('value-1' has the highest correlation coefficient, indicating that the tickets count from the previous day is a strong predictor of the current ticket count, 'value-2' and 'value-3' also have relatively high positive correlations).

The LSTM model has successfully identified the trend with high accuracy. The RMSE is 2.46 meaning that on average, the model's predictions deviate from the actual observed values by 2.46 units.

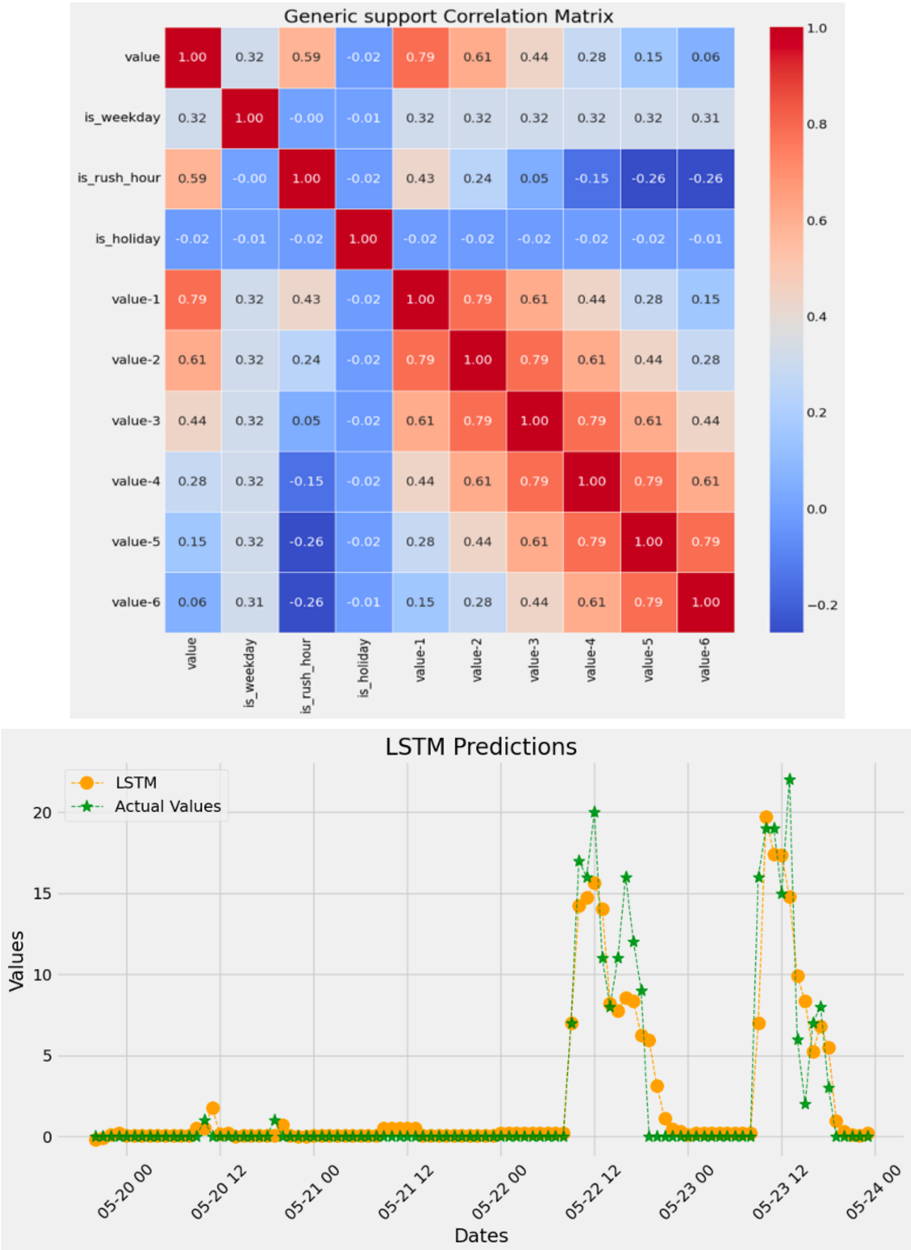
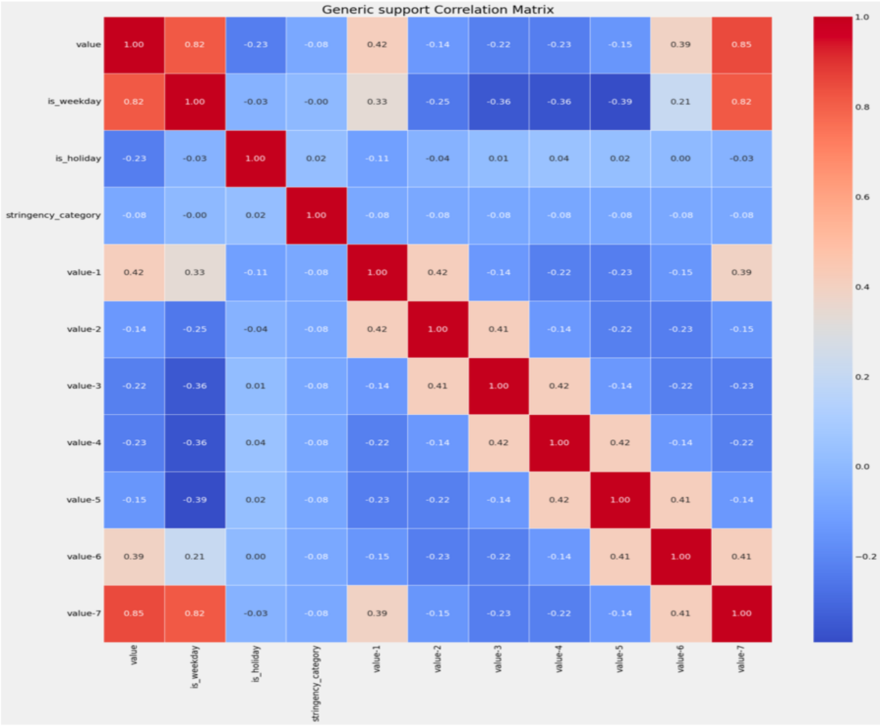


Figure 17: INS Generic support (hourly): correlation matrix [top], LSTM predictions vs actual values [bottom]

In daily-based weekdays and value-7 there is a strong positive correlation, while ‘value-1’ and ‘value-6’ show a moderate correlation suggesting that data from the previous day and six days ago can moderately predict ticket volumes. Moreover, whether a day is a holiday has a negative correlation.

The Random Forest found to be the best model with RMSE 19.075 and successfully identified the trend on the test set.



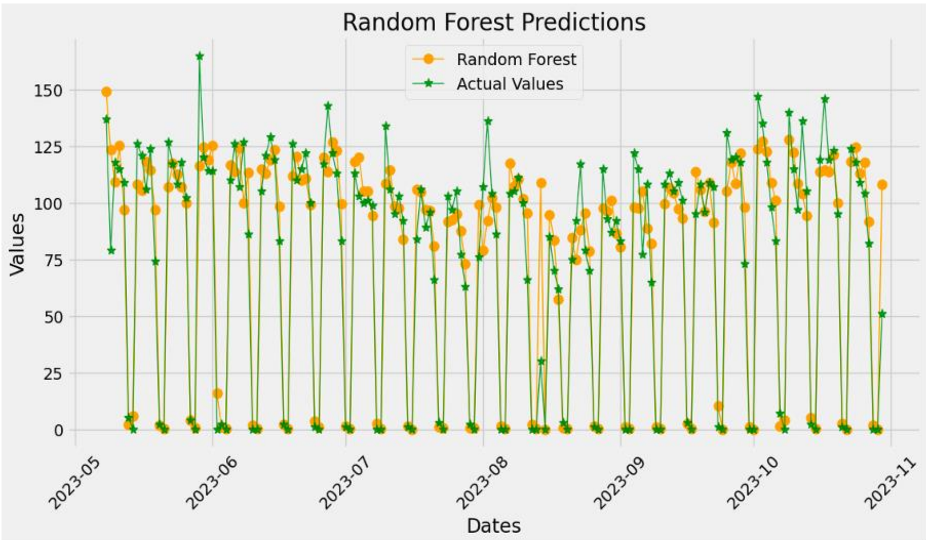
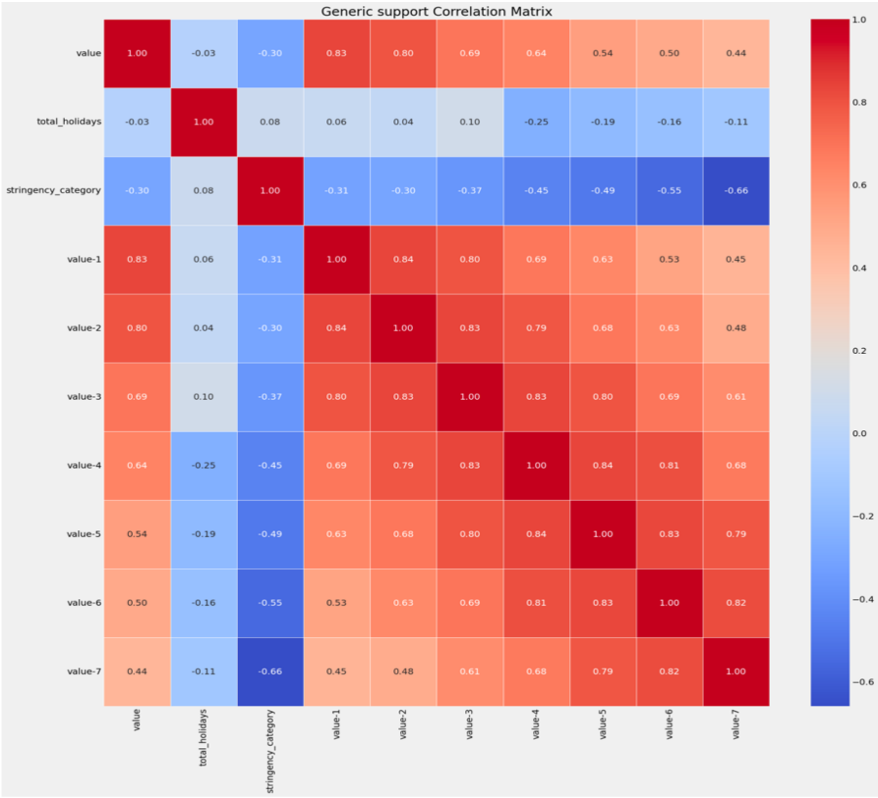


Figure 18: INS Generic support (daily): correlation matrix [top], Random forest predictions vs actual values [bottom]

In monthly-based resampling, there exist strong and moderate positive correlations with the lagged values while the correlation decreases with the increasing lag and there is a negative moderate correlation between stringency index and the number of tickets. The best model found to be the N-BEATS which is accurately identifying the trend, but the amount of the data is limited due to resampling (it has high error in August of 2023).



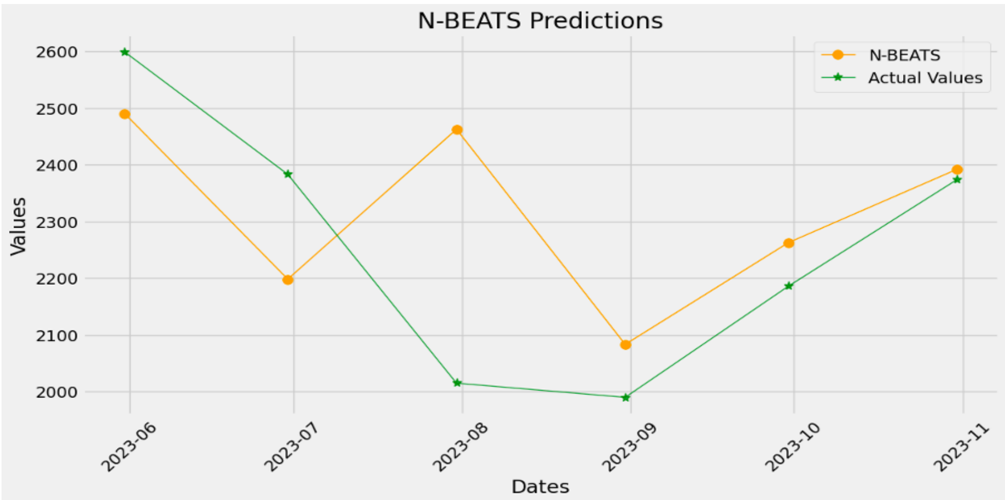


Figure 19: INS Generic support (monthly): correlation matrix [top], N-BEATS predictions vs actual values [bottom]

Software Bug

For the “Software Bug” category, in an hourly based timeframe it is found that there is a weak positive correlation of 20.98% between a weekday and the number of tickets and a moderate positive correlation of 38.71% between rush hours and the number of tickets. There also exist moderate positive correlations with the lagged values where 'value-1' has the highest correlation coefficient. The model identifies the peaks and the trend of the timeseries with a RMSE equal to 0.6338. Due to many values equal to zero the MAPE metric is not applicable.

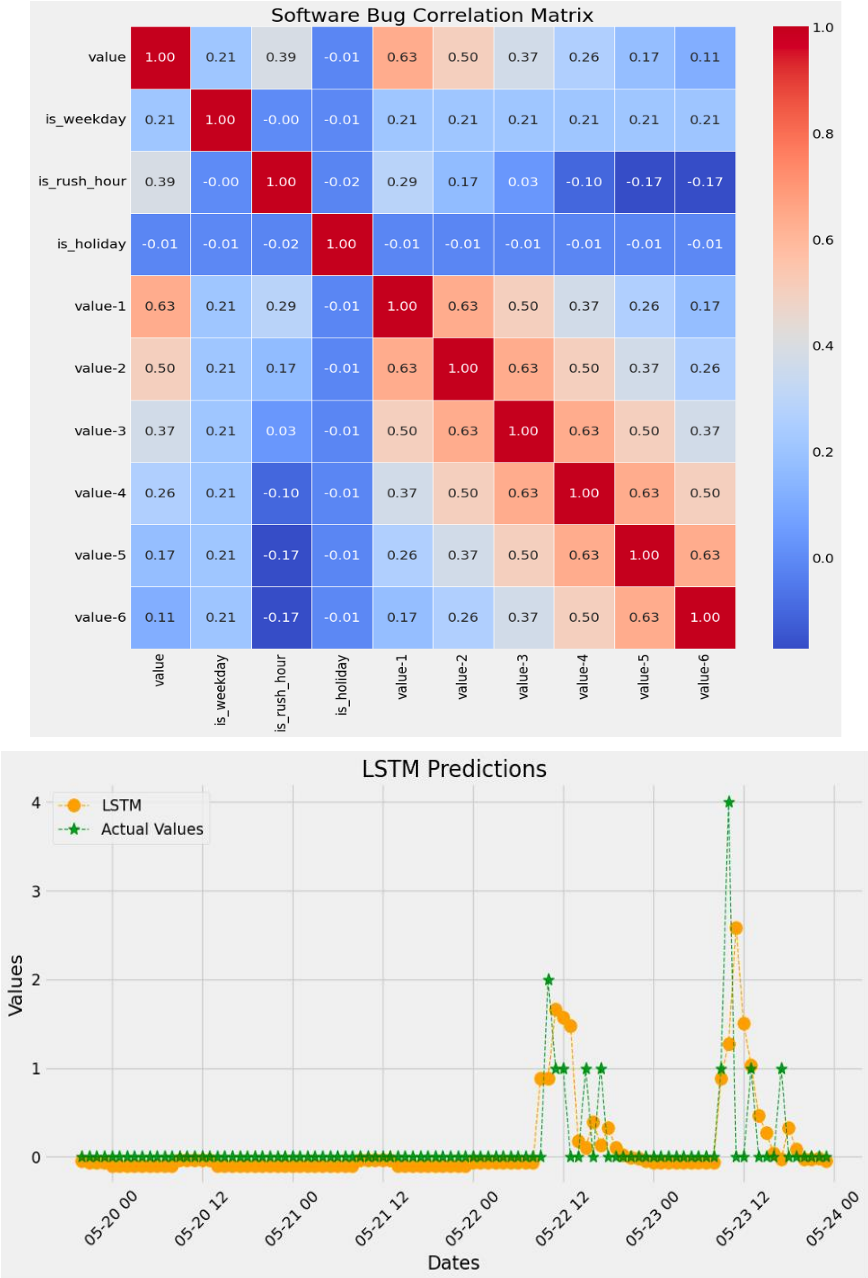


Figure 20: INS Software bug (hourly): correlation matrix [top], LSTM predictions vs actual values [bottom]

In a daily-based timeframe a correlation was found for the past values with the highest on ‘value-7’ and a moderate correlation with weekdays. The best model found is the N-BEATS with a RMSE equal to 8.282.

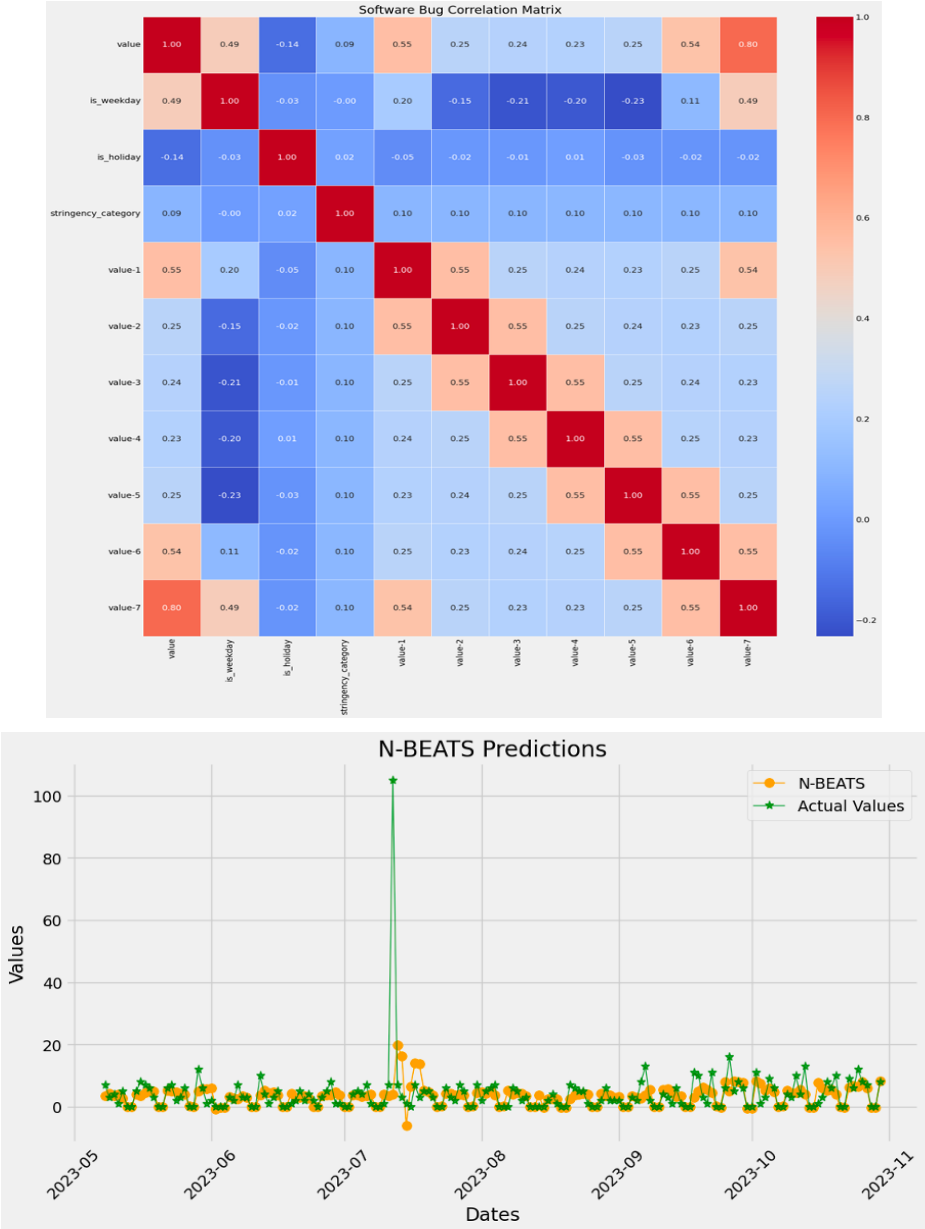
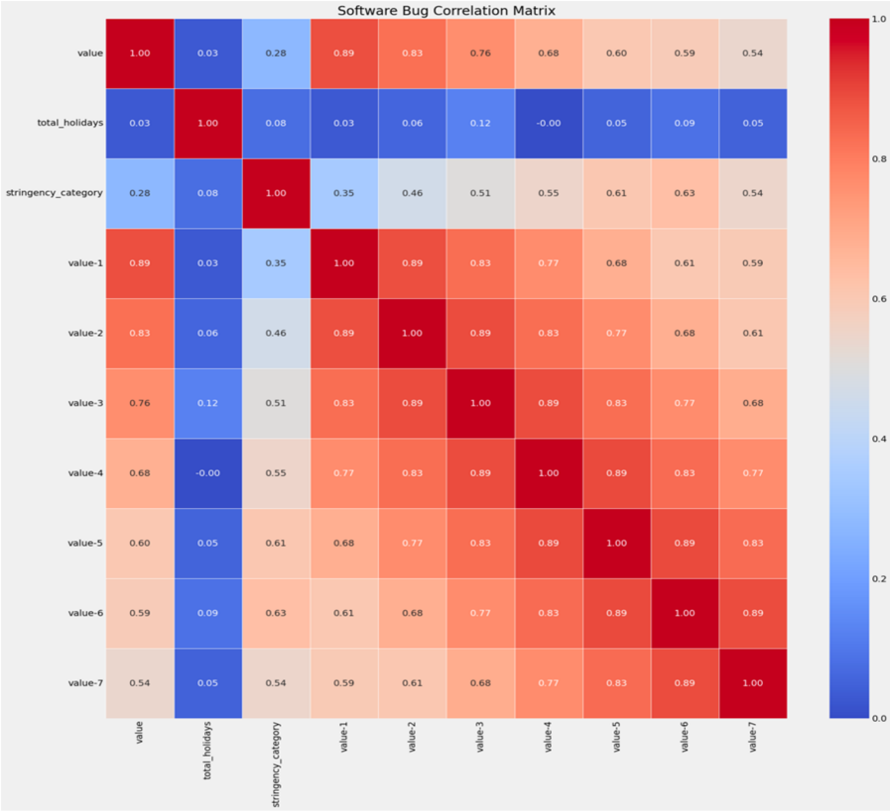


Figure 21: INS Software bug (daily): correlation matrix [top], N-BEATS predictions vs actual values [bottom]

In monthly-based timeframe, there is a moderate positive correlation between the stringency category and the total number of tickets, while there exist strong positive correlations with the lagged 'value-1' to 'value-3', these correlations decrease when increasing the lag. The best model found is the Feedforward Neural Network with a RMSE equal to 41.084 and a MAPE equal to 16.678. The model's forecast is very accurate in some dates, but again in August of 2023 it fails to predict the actual value, due to possible external events, not included into the data.



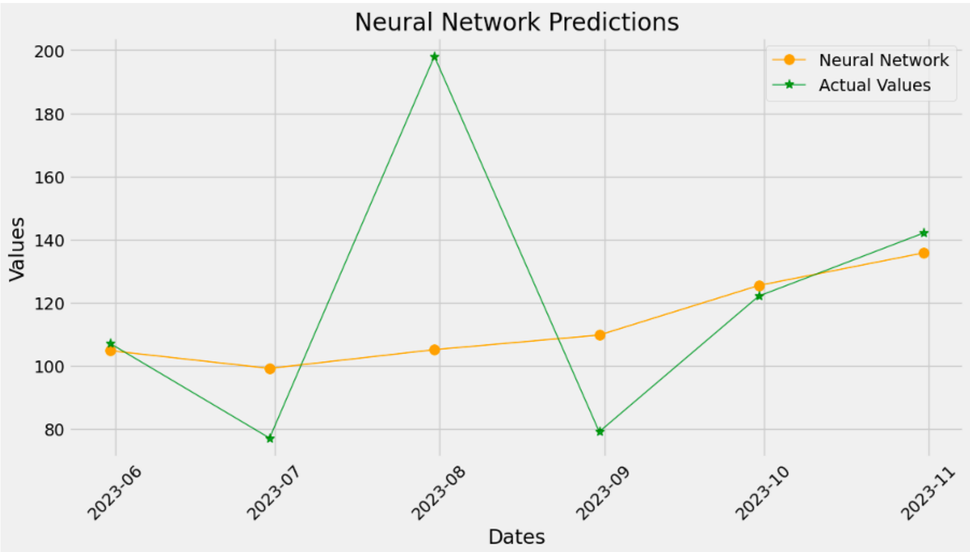
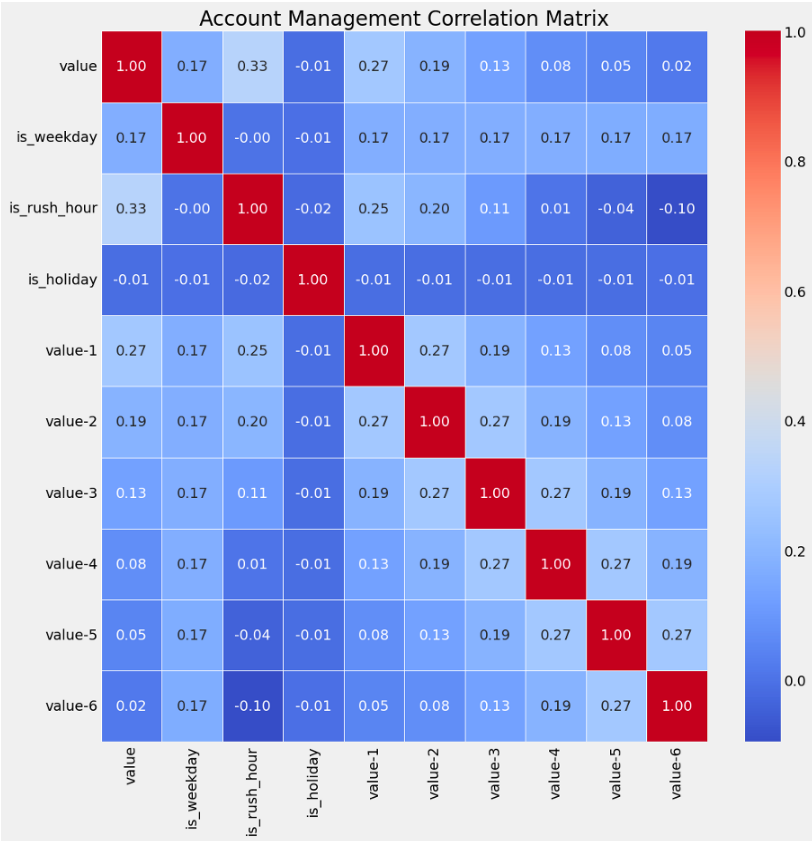


Figure 22: INS Software bug (monthly): correlation matrix [top], Feedforward Neural network predictions vs actual values [bottom]

Account Management

For the “Account Management” category, the hourly based weekdays exhibit a weak positive correlation, while rush hours demonstrate a moderate positive correlation. Once again ‘value-1’ has the highest correlation among the lagged values, and the correlations decline while increasing the lag. However, despite capturing the peaks, the LSTM model's accuracy in prediction is not optimal.



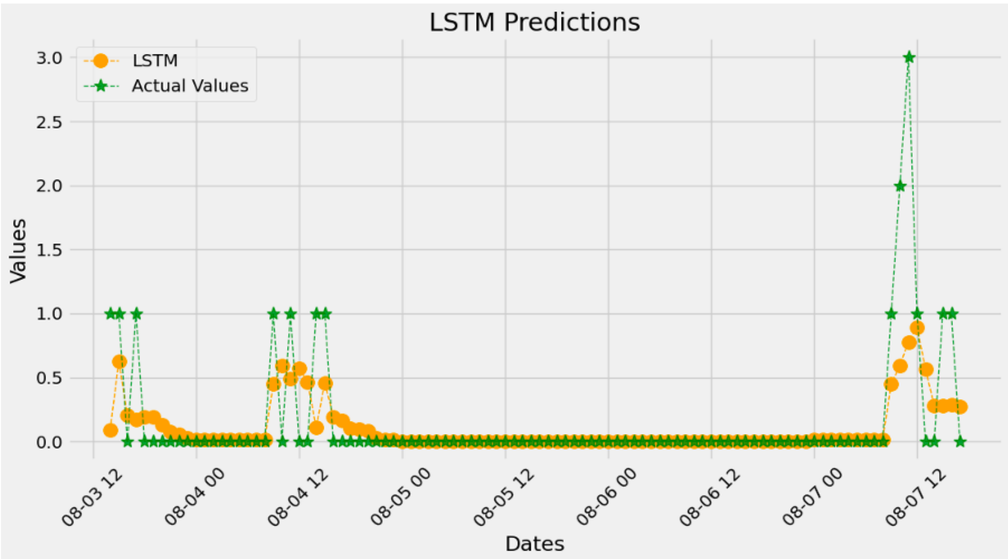


Figure 23: INS Account Management (hourly): correlation matrix [top], LSTM predictions vs actual values [bottom]

In daily-based timeframe, the number of tickets is moderately correlated with weekdays and ‘value-7’, while it is very weak correlated with both the rest lagged values and the stringency category index. The best model found is the LSTM with a RMSE equal to 1.903 and it successfully captured the trend on the test data but was not able to identify the high increase on the peaks.

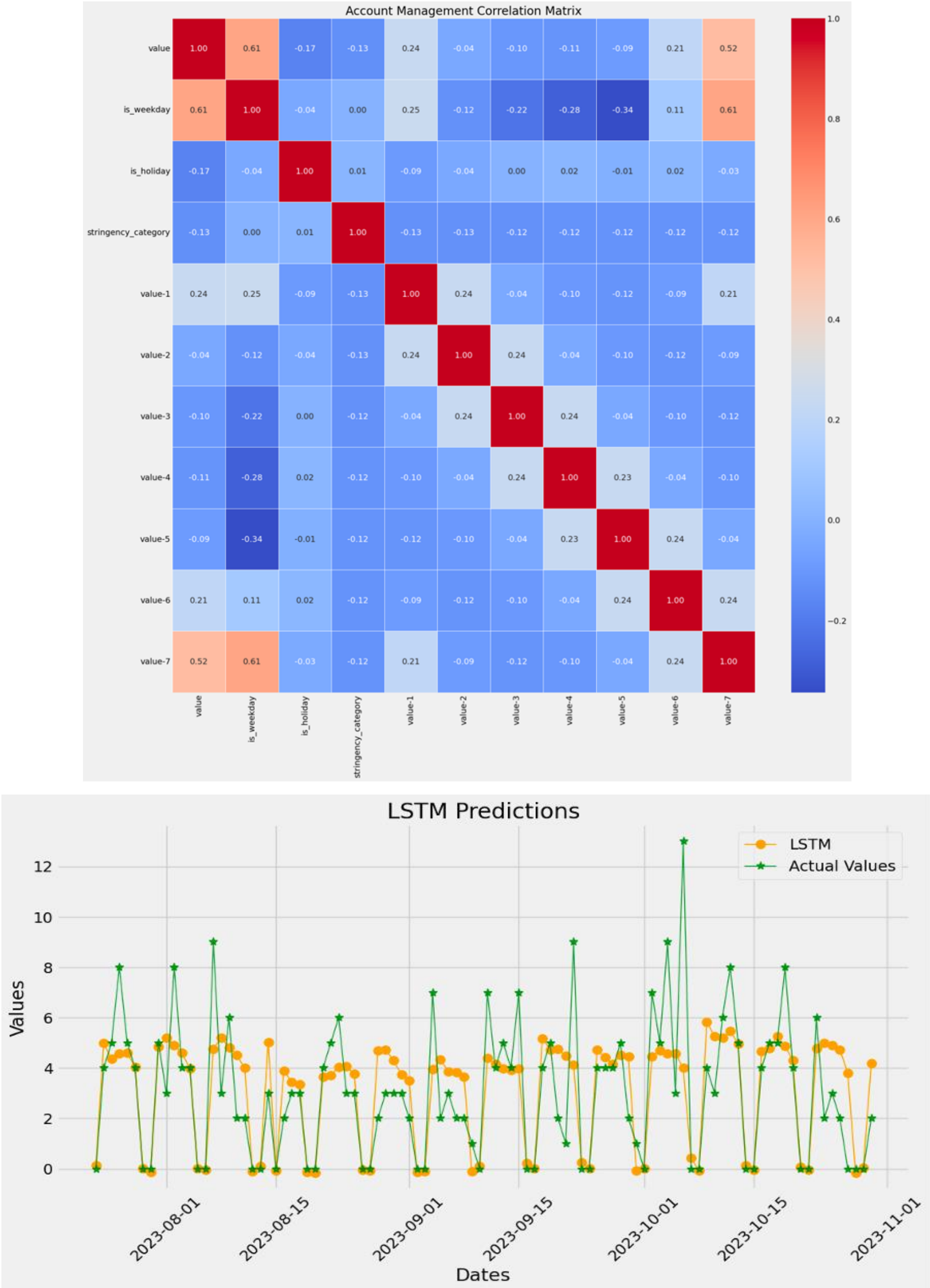


Figure 24: INS Account Management (daily): correlation matrix [top], LSTM predictions vs actual values [bottom]

In monthly-based timeframe, there is a negative strong correlation between the number of tickets and the stringency category index. The number of tickets is also moderately correlated with the most lagged values. The data for Account Management is very limited compared to other categories and it is compared only in three months which is not optimal. The best model found is the Feedforward Neural Network.

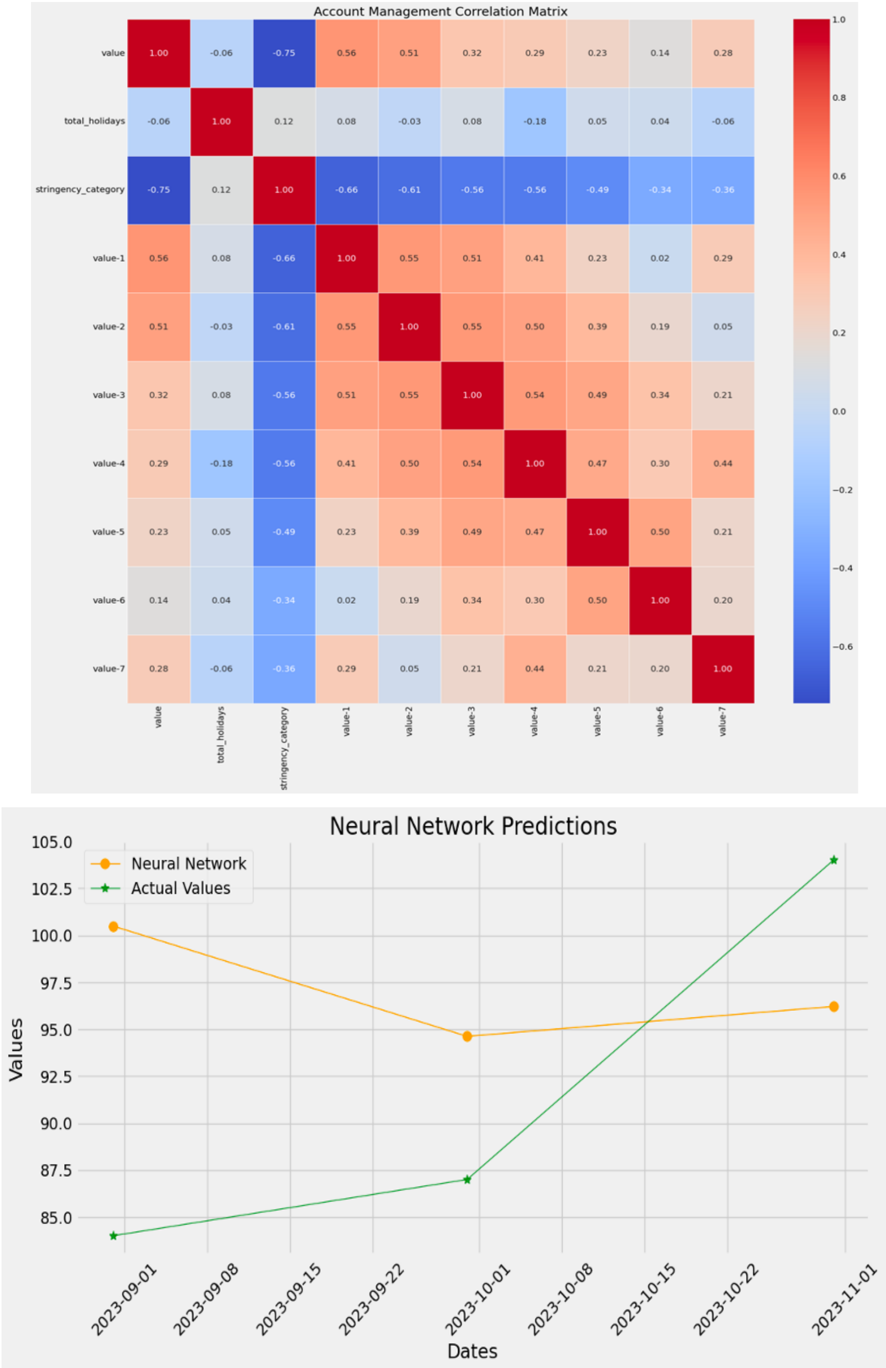
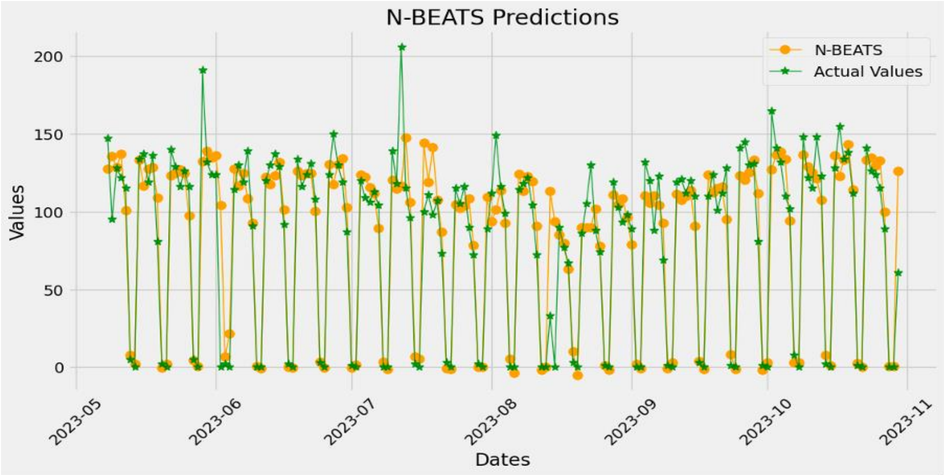


Figure 25: INS Account Management (monthly): correlation matrix [top], Neural network predictions vs actual values [bottom]

All tickets aggregated

Experiments have been conducted and models have been created in the aggregation of the data in daily and monthly timeframes. In both timeframes the best model found is the N-BEATS with a RMSE equal to 20.703 (daily) and a RMSE equal to 64.600 and MAPE equal to 2.364 (monthly).



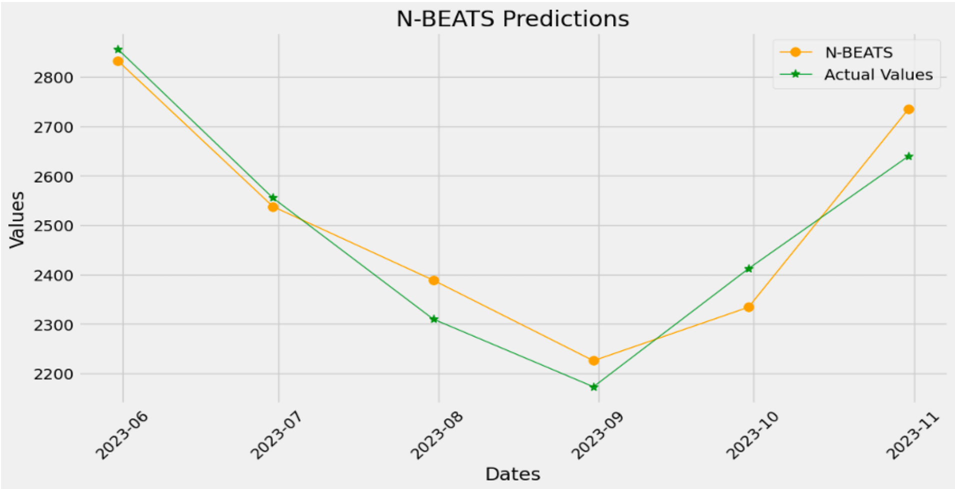


Figure 26: INS All tickets aggregated: N-BEATS predictions vs Actual values (daily [top], monthly [bottom])

Table 7: INS ticket trained models

TICKET	BEST MODEL	RMSE	R2 SCORE	MAPE	FREQUENCY
GENERIC SUPPORT	LSTM	2.460	0.819	n. a	HOURLY
SOFTWARE BUG	LSTM	0.633	0.373	n. a	HOURLY
ACCOUNT MANAGEMENT	LSTM	0.443	0.149	n. a	HOURLY
GENERIC SUPPORT	Random Forest	19.075	0.913	n. a	DAILY
SOFTWARE BUG	N-BEATS	8.282	0.912	n. a	DAILY

TICKET	BEST MODEL	RMSE	R2 SCORE	MAPE	FREQUENCY
ACCOUNT MANAGEMENT	LSTM	1.903	0.517	n. a	DAILY
ALL_TICKETS	N-BEATS	20.703	0.912	n. a	DAILY
GENERIC SUPPORT	N-BEATS	208.608	0.072	7.183	MONTHLY
SOFTWARE BUG	Feedforward Neural Network	41.084	0.013	16.678	MONTHLY
ACCOUNT MANAGEMENT	Feedforward Neural Network	11.415	0.68	11.963	MONTHLY
ALL_TICKETS	N-BEATS	64.600	0.916	2.364	MONTHLY

3.3.1.2 Digital Documents

INSIEL has categorized digital documents, including X-Rays, Medical Reports (LDO), and Discharge Letters (REFE). In the ML experiments, we have included exogenous variables like whether a day is a holiday and weekday. Moreover, stringency index and seasonal and weather data have also been included in both daily and monthly based timeframes.

SUNRISE

X-Rays

In daily-based timeframe, there is a weak negative correlation between the number of X-Ray documents and the stringency category index at a level of 20%. In addition, there is a high positive correlation with the lagged values (value-1 to value-7) which indicates that the daily conducted number of X-rays is strongly influenced by its lagged values. This suggests some level of autocorrelation in the data, meaning that the current value is dependent on its previous values. The best model found is the N-BEATS with a RMSE equal to 1367.248 and a MAPE equal to 29.02. The model can identify the trend and the seasonality of the data.

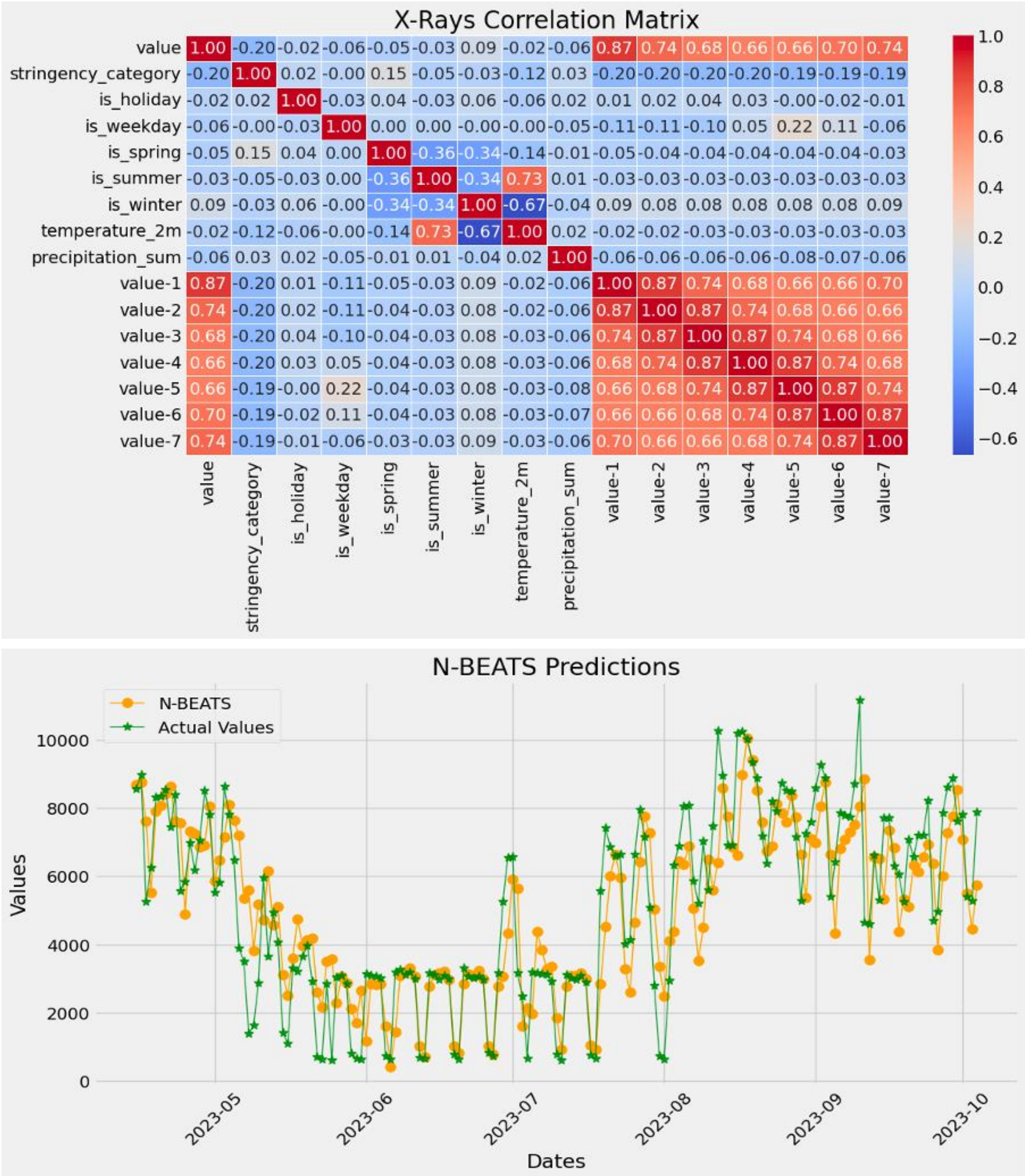
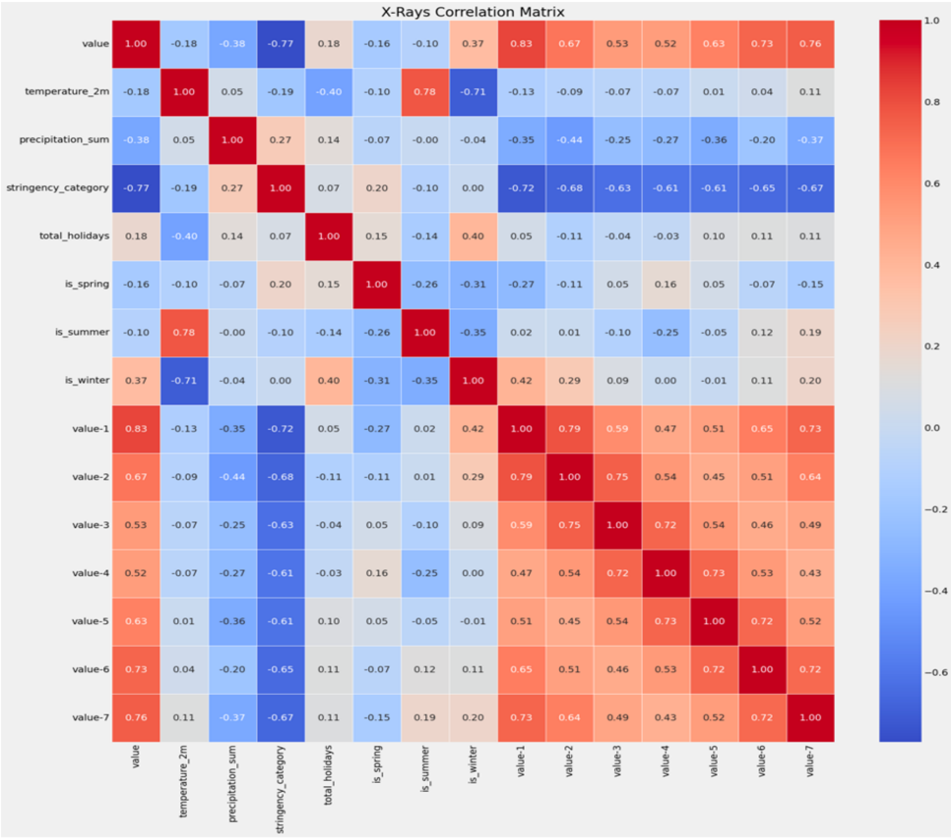


Figure 27: INS X-Rays (daily): correlation matrix [top], N-BEATS predictions vs actual values [bottom]

In monthly-based timeframe, the number of X-Rays is correlated with most of the variables indicated that the timeseries is based on autocorrelation. In addition, it is correlated with the seasonal data. The best model found is the N-BEATS with a MAPE equal to 8.142.



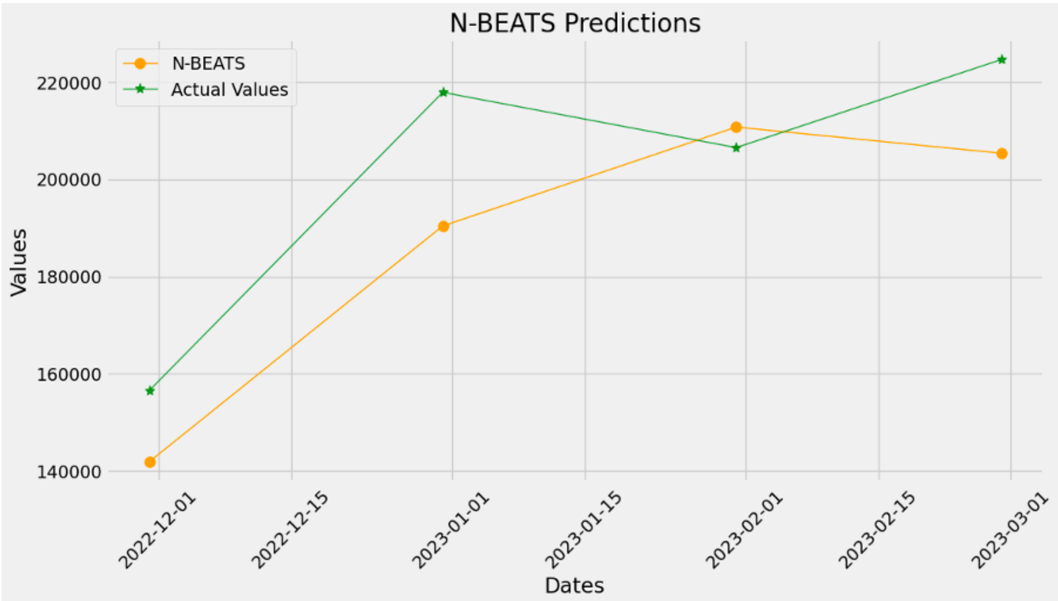


Figure 28: INS X-Rays (monthly): correlation matrix [top], N-BEATS predictions vs actual values [bottom]

Medical Reports (LDO)

In daily-based timeframe, the number of LDO documents shows moderate positive correlations with its lagged value-1 and a very weak correlation with value-6 and value-7. No other significant correlations were found. The best model is the Random Forest with a RMSE equal to 110.110 and a MAPE equal to 20.725.

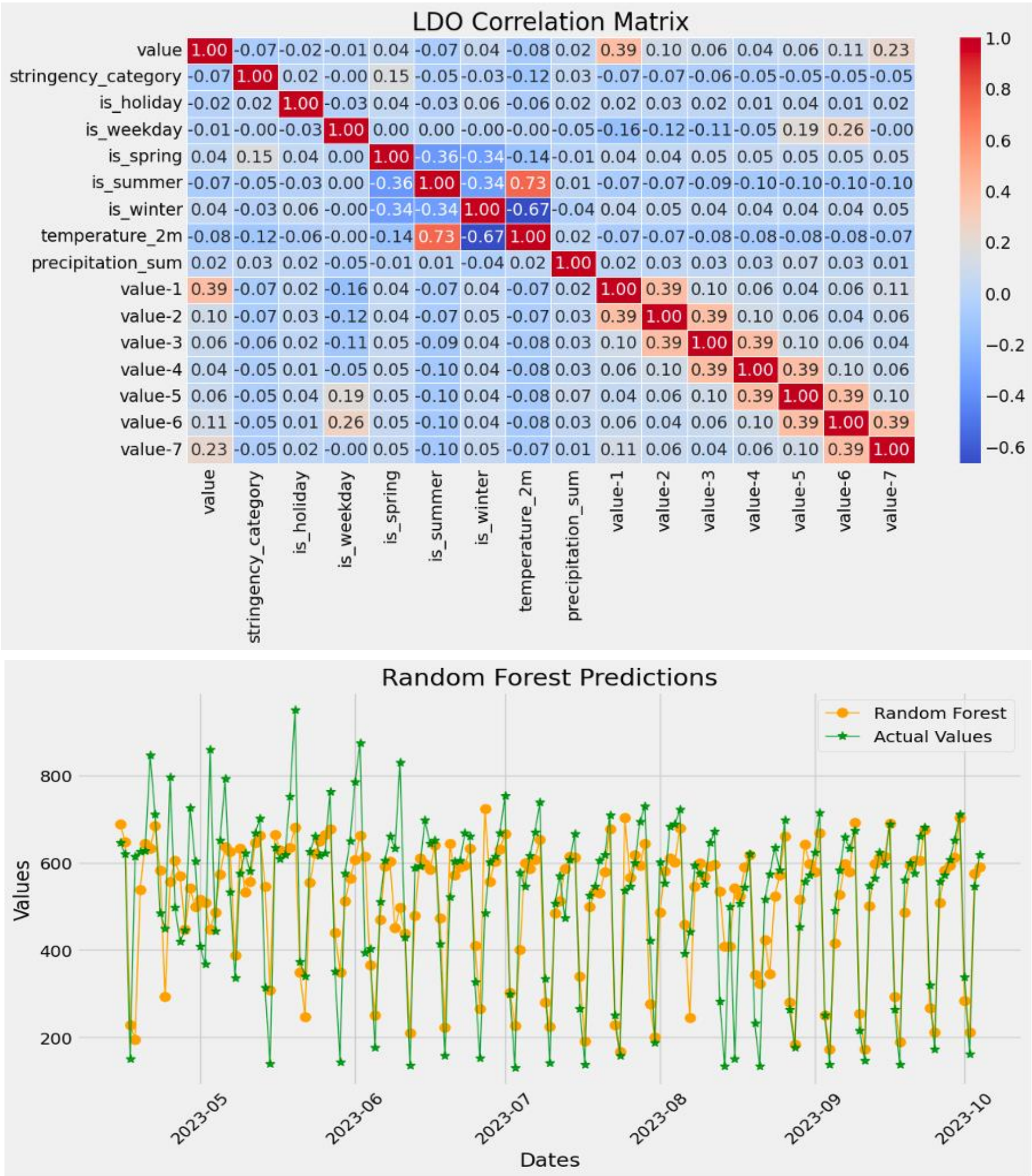


Figure 29: INS LDO (daily): correlation matrix [top], N-BEATS predictions vs actual values [bottom]

In a monthly-based timeframe, the number of LDO documents shows moderate to weak correlation to most of the variables. The best model found is the Feedforward Neural Network with a RMSE equal to 785.327 and a MAPE equal to 4.424.

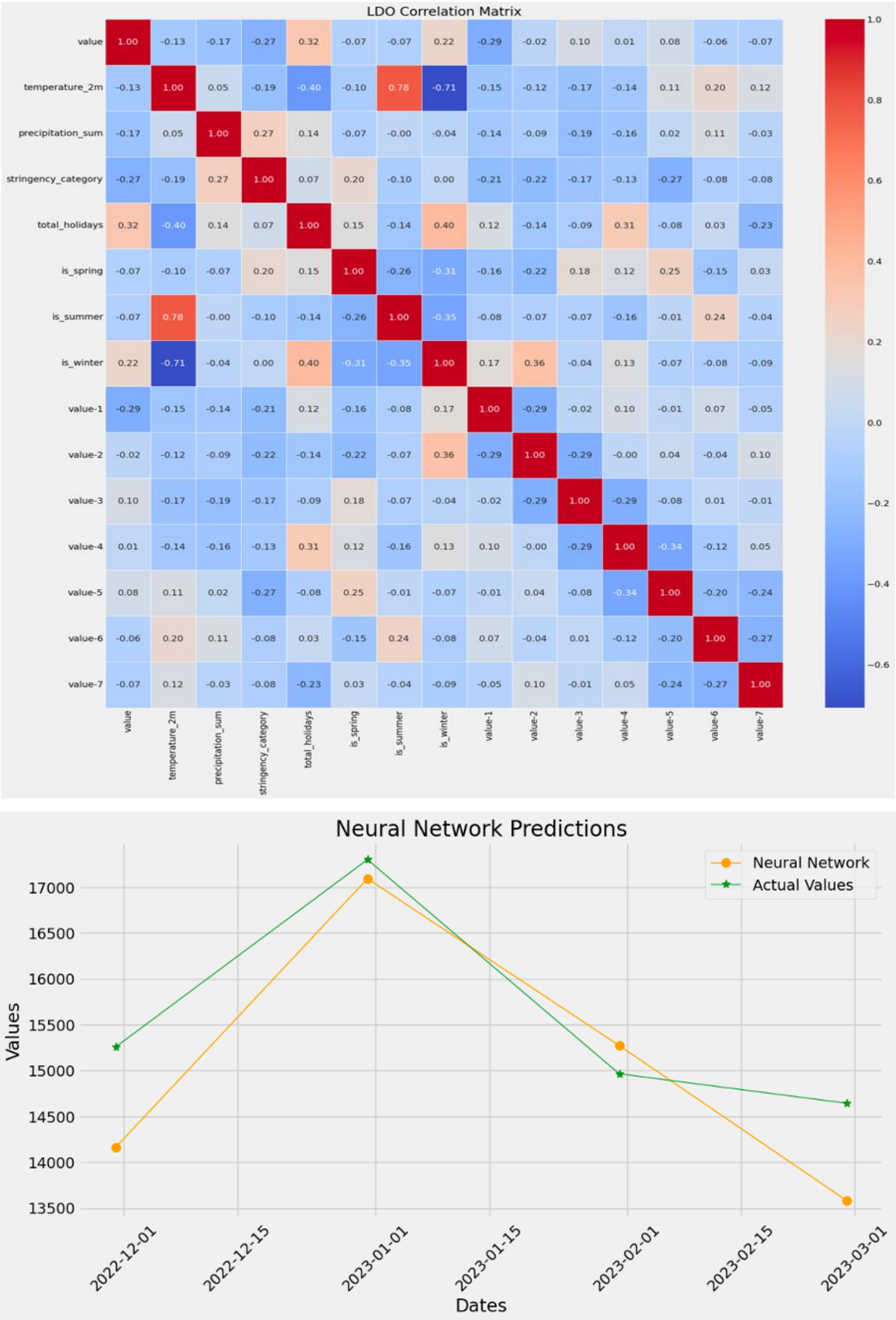


Figure 30: INS LDO (daily): correlation matrix [top], N-BEATS predictions vs actual values [bottom]

Discharge Letters (REFE)

In daily-based timeframe, a strong correlation found with the lagged value-7 while a moderate positive correlation exists with value-1 and value-6. In addition, a very week correlation was found with the weekdays and the category stringency index. The best model found is the N-BEATS with a RMSE equal to 3529.319 and a MAPE equal to 20.996.

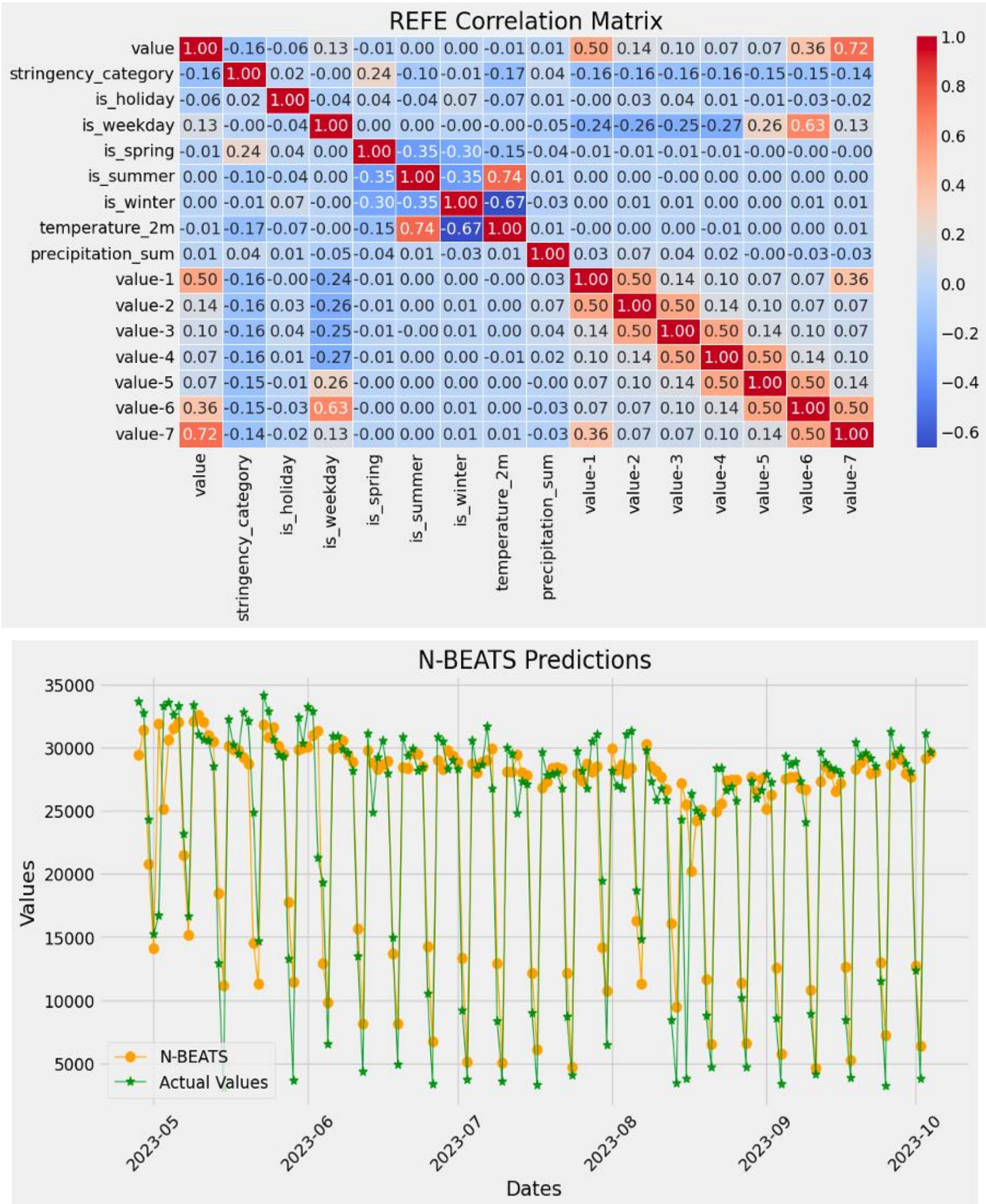


Figure 31: INS REFE (daily): correlation matrix [top], N-BEATS predictions vs actual values [bottom]

In monthly-based, similarly with the LDO documents the number of REFE docs has a moderate to weak correlation with most of the variables. The best model found is the Feedforward Neural Network with a MAPE equal to 4.884.

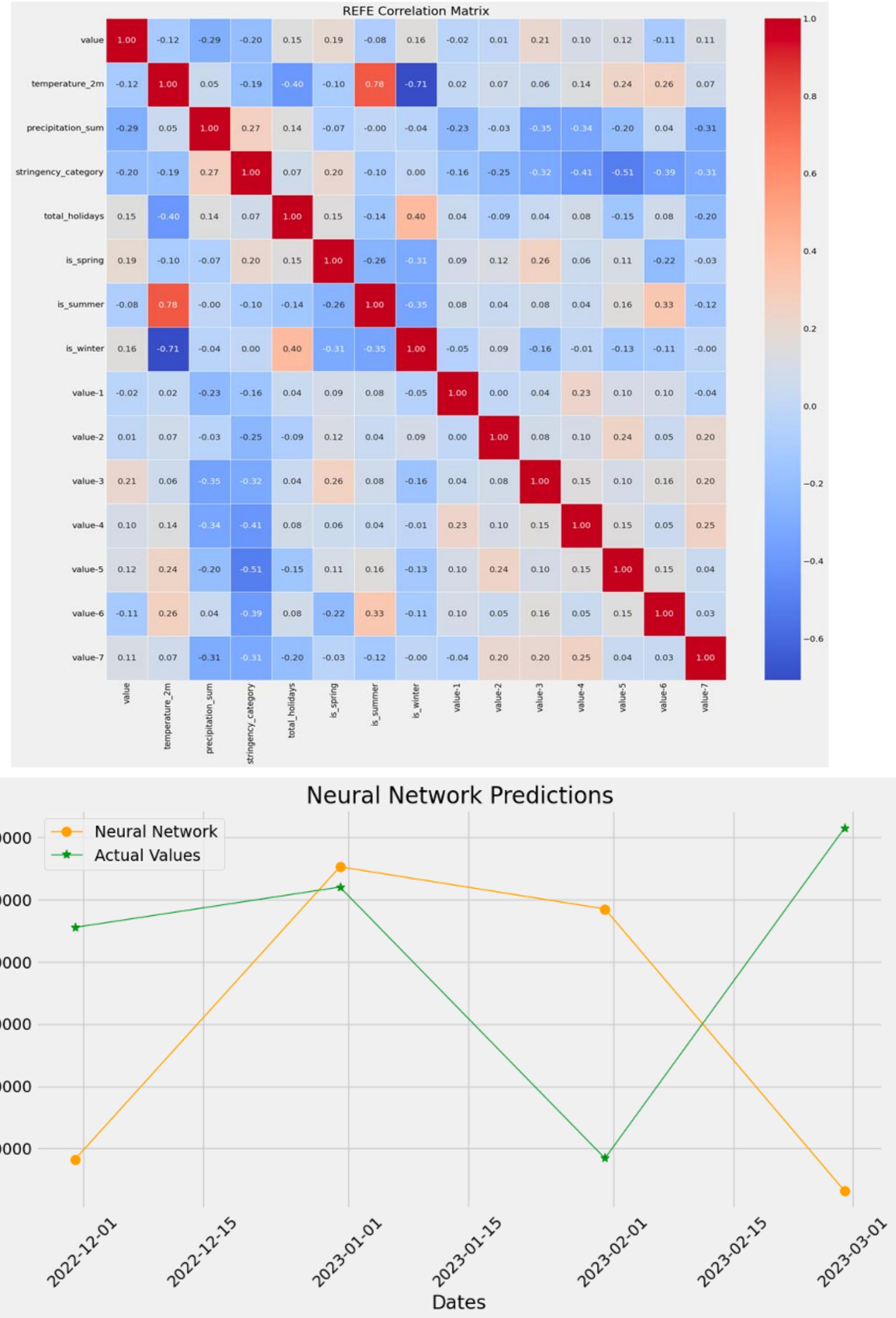


Figure 32: INS REFE (monthly): correlation matrix [top], Neural network predictions vs actual values [bottom]

Table 8: INS Document trained models

DOCUMENT	BEST MODEL	RMSE	R2 SCORE	MAPE	FREQUENCY
X-RAYS	N-BEATS	1367.248	0.737	29.02	DAILY
LDO	Random Forest	110.110	0.674	20.725	DAILY
REFE	N-BEATS	3529.319	0.856	20.996	DAILY

DOCUMENT	BEST MODEL	RMSE	R2 SCORE	MAPE	FREQUENCY
X-RAYS	N-BEATS	18420.970	0.806	8.142	MONTHLY
LDO	Feedforward Neural Network	785.327	0.058	4.424	MONTHLY
REFE	Feedforward Neural Network	40011.917	3.61	4.884	MONTHLY

3.3.1.3 Conclusion

The development of the DPM tool for health-digital services successfully meets project's KPIs through robust models capable of predicting IT help desk ticket volume and demand for digital documents, such as X-rays, medical reports, and discharge letters, with factors like weather, movement restrictions, and seasonal patterns taken into account. With its forecasting designed for hourly, daily, and monthly demand, the tool supports efficient resource planning across different time scales. The next development phase aims to integrate climate change projections, enhancing the tool's predictive precision and equipping it to better face future challenges.

3.3.2 QS/HQM

HQM has provided various sets of data, including information on Equipment, Consumables, Drugs, Hospital Patients' Admission data, and Staff requirements. Data analysis has been conducted utilizing the available information and experiments in ML models were applied where data was sufficient. Equipment data was excluded from the analysis due to the very limited availability of sufficient time points (only some specific purchases were given). The overall size of the Equipment data is relatively small, with only 51 entries, many of which are incomplete.

HQM has also supplied additional data for drugs and consumables to extend the timeframe and initiate the development of initial models. Upon analysis, it was found that predicting demand for drugs and consumables poses significant challenges due to various factors, including supplier shortages, fluctuations in hospital inventory levels, and the impact of COVID-19 on demand behaviour and patient numbers. Based on the discussions with the CI, they informed us that fear of contracting Covid-19 influenced hospital visits, and the number of COVID-19 patients varied based on the virus's form and vaccine availability.

To address these challenges, the potential of data analytics was emphasized during the discussions. By utilizing charts and graphs, users can effectively explore data, gaining insights into trends, patterns, and correlations.

Furthermore, considering the complexity of COVID-19 data, some discussions were initialized regarding the exploration of additional pathologies that may exhibit more predictable demand patterns. This approach could offer clearer seasonal trends and a broader timeframe for analysis, although potential correlations with COVID-19 data must be considered. HQM has provided such additional information about patients for flu pathology, including patients' admissions, drugs, and oxygen consumption.

In the modelling phase, a variety of ML algorithms were employed by SQD to forecast demand for consumables, drugs, and gripe patients. The models utilized include XGBoost, N-BEATS, Feedforward Neural Network, LSTM, Random Forest, and Grid Search. The three-window size is used which refers to the number of past months considered for making forecasts; in this case, a window size of 3 months means that the models use data from the previous three months to predict future demand. The exogenous variables incorporated into the analysis are the total number of holidays within a month, seasonal trends, and the stringency index, which may affect patient behavior, consumable and drugs usage.

3.3.2.1 Stringency Index

The Stringency Index is a composite measure based on several indicators, such as school closures; workplace closures; cancellation of public events; restrictions on public gatherings; closures of public transport; stay-at-home requirements; public information campaigns; restrictions on internal movements; and international travel controls. The index on any given day is calculated as the mean score of the nine metrics, each taking a value between 0 and 100. A higher score indicates a stricter response (i.e. 100 = strictest response). It's important to note that this index simply records the strictness of government policies. It does not measure or imply the appropriateness or effectiveness of a country's response. A higher Stringency Index does not necessarily mean a better or worse response to a health emergency, but rather it quantifies how stringent the measures were at a given time. The Stringency Index is usually used to compare different countries' responses over time, or to look at how the response in a single country change.

Figure 33 shows a line indicating how strict the rules were over time, spanning from January 2020 to March 2023. In the beginning, the line goes up quickly, which means the rules got very strict. After that, the line moves up and down a bit, showing that the rules kept changing. As time goes on, the line goes down, meaning the rules became less strict. This probably happened as people got better at handling the sickness, like getting vaccines or because it was important to let people go back to work and school.

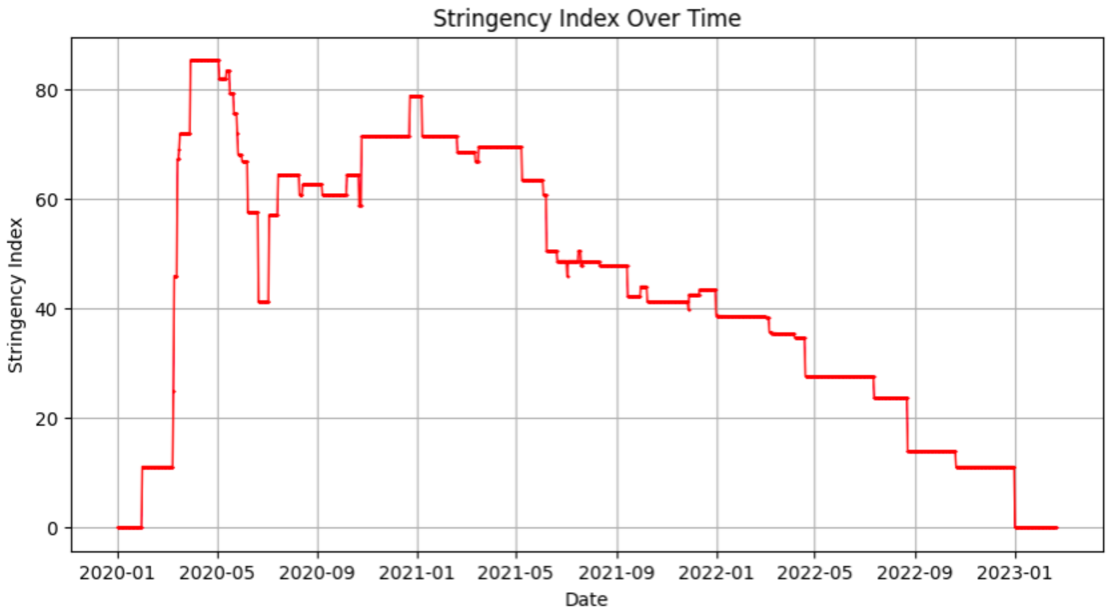


Figure 33: COVID-19 Stringency Index in Spain from 2020 to 2023

3.3.2.2 Staff Requirements

The dataset provides a detailed breakdown of staff requirements across various departments, spanning from January 2019 to December 2021 and includes information on different areas such as External Consultations, Hospitalization, and Adult ICU, along with specific specializations within each area like Administration, Nursing Assistant, and Nursing. The numbers in the dataset represent the monthly staffing needs within each specialization and area.

Figure 34 shows a collection of line graphs, each representing the staff levels across different healthcare roles over time. Trends vary widely: the administration category and nursing assistants show a steady increase, while nursing in external consultations experiences subsequent drop in April 2020, followed by slight rise and steady decline. Commercial roles in and health technicians in external consultations hold steady at a low level, whereas other health personnel in external consultations experience a sharp decline towards the end. In hospitalization, both nursing assistants and nursing categories show variability, peaking above 240 in April 2020, coinciding with the peak of the Stringency Index, which suggests and increased need for staff during the pandemic. In the Adult ICU, both nursing and nursing assistants experience a steady increase, with a speak in April 2020, whereas optional category slightly increases.

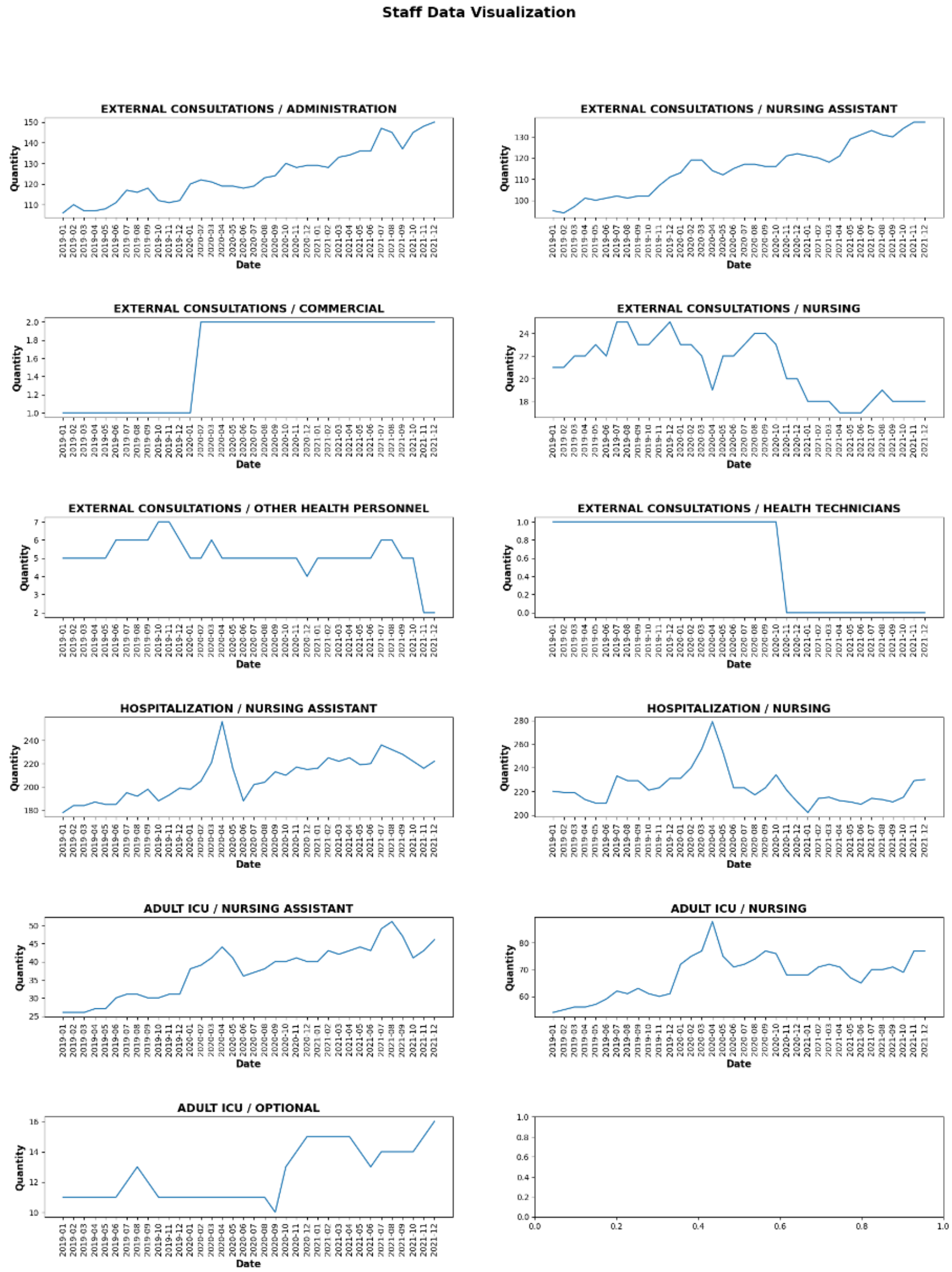


Figure 34: Staff Requirements Across Different Sectors Over Time

3.3.2.3 COVID-19 Consumables

This dataset contains purchases of various medical consumables over time needed against COVID-19. For the development of ML models, the supplies are grouped into the categories, including MASCARILLA (mask), ANTISEPTICO (antiseptic), CALZA (medical socks), BATA (medical gown), GAFAS (medical goggles), GORRO (medical cap), GUANTE (medical glove), and GEL HIDROALCOHOLICO (hydroalcoholic gel).

Figure 35 shows a heatmap plot depicting how the quantity of each consumable changes over time. The x-axis represents different items, while the y-axis shows dates. Each cell's colour intensity indicates the quantity of an item on a specific date, with lighter colours for lower quantities and darker colours for higher quantities. This visualization helps identify patterns, trends, or anomalies in item consumption over time, guiding data-driven decision-making. Nitrile examination gloves (GUANTE EXAMEN NITRILO) showed the highest demand, particularly in 2020, indicated by the darkest colour on the heatmap.

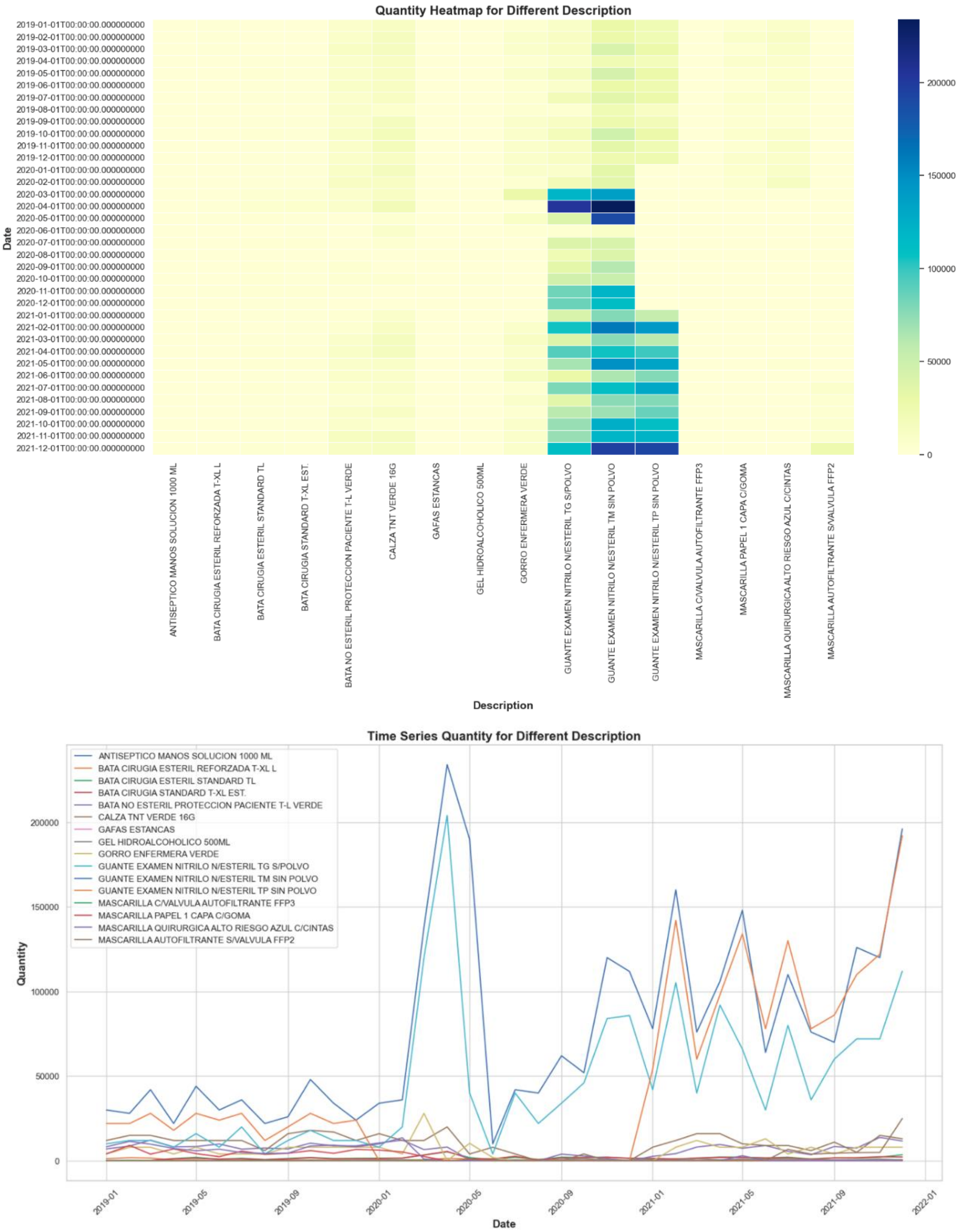
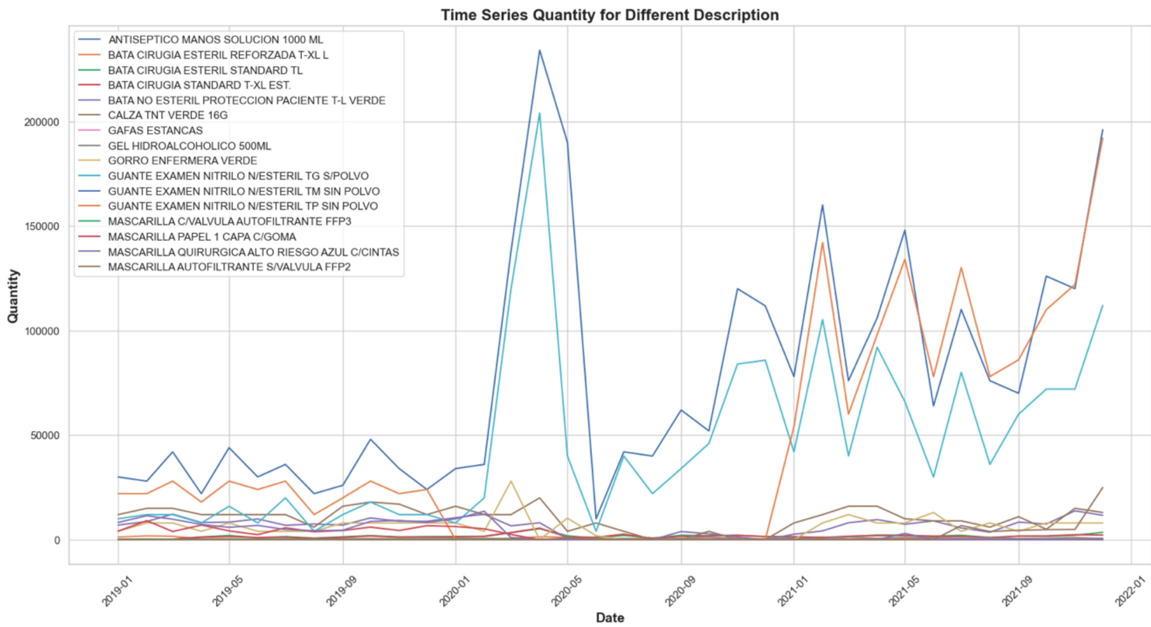


Figure 35: COVID-19 Consumables Demand Over Time: heatmap [top], line plot [bottom]

Figure 35 shows that nitrile examination gloves (GUANTE EXAMEN NITRILO) were in high demand, especially during the COVID-19 outbreak in 2020. They were consistently needed more than other items, with peaks during times of increased demand. Different types of nitrile gloves were used at different rates, suggesting people had different preferences or needs.

Figure 36 (top) shows how well different prediction models predicted the number of purchases for Hydroalcoholic Gel over time. The bottom axis shows the months and years, and the side axis shows the amount of the purchases, from about 600 to 3000. The line for Testing Data is the real amount of the purchases, while the rest represent the models' prediction for the seven-time period. The closer a model's line is to the test data line, the more accurate its predictions are for the given timeframe. All models have fluctuations in performance over time, where some points are closer to the test data line, and others deviate more significantly.



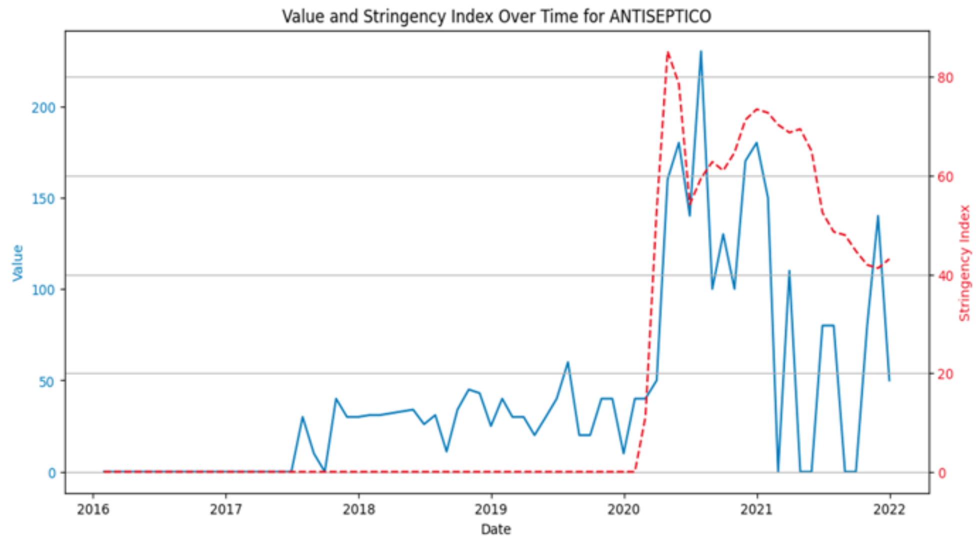


Figure 36: Covid-19 Impact: Comparison Between Multiple Prediction Models and Actual Data for Consumables purchases [top], Anticeptic Correlation with the Stringency Index Over Time [bottom]

All consumables were affected by COVID-19 and there is such a peak in April 2020. Figure 36 (bottom) depicts the purchases of antiseptic products and the stringency index from 2016 to around 2022. Initially, from 2016 to early 2020, purchases of antiseptic remained relatively stable with minor fluctuations, while the stringency index remained low as there was no COVID-19. However, in early

2020, when COVID-19 hit, the purchases of antiseptic experienced a high demand. Following this, fluctuations occurred in both antiseptic purchases and the stringency of rules, generally correlating with each other.

However, there were occasions when the trends did not align perfectly, yet overall, we observe the significant influence of COVID-19 on the demand for antiseptic products.

3.3.2.4 COVID-19 Drugs

This dataset contains purchases of various drugs over time needed against COVID-19. For the development of ML models, the drugs are grouped into the categories, including TOCILIZUMAB, HIDROXICLOROQUINA, CICLOSPORINA, and LOPINAVIR + RITO.

The Figure 37 (top), presenting the comparing of testing data and forecasting models over time, with X-axis representing dates and Y-axis representing the quantity of the purchases. The data points are in eight-time period, spread from 2022-06 to 2023-01. The performance of the models varies significantly across different time points. In some time periods all models differ from the test data, particularly from 2022-10 to 2022-11, where they are unable to follow the trend.

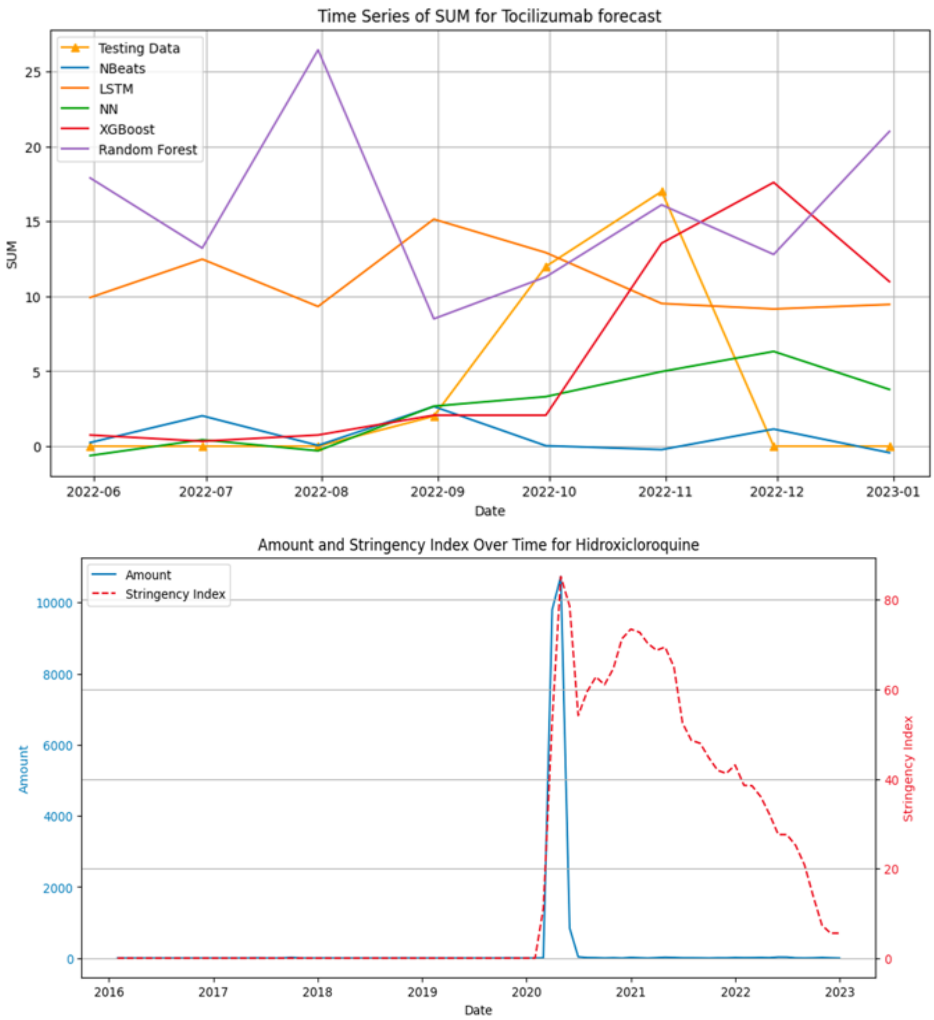


Figure 37: Covid-19 Impact: Comparison Between Multiple Prediction Models and Actual Data for Drugs purchases [top], Hydroxychloroquine Drug Correlation with the Stringency Index Over Time [bottom]

The Figure 37 (bottom), presenting the number of drugs purchased over time compared to stringency index, spanning from 2016 to 2023. There is a peak in the amount of hydroxychloroquine around the same time when the Stringency Index also peaks. After the initial peak in both the amount of Hydroxychloroquine and the Stringency Index, both measures decline. The Stringency Index shows a more gradual decline, while the amount of hydroxychloroquine drops sharply before tapering off. Towards the end of the graph's timeline, the Stringency Index fluctuates at lower levels compared to its peak, indicating a move towards the implementation of less stringent policies.

3.3.2.5 COVID-19 Patients' Admissions

This dataset contains detailed information about COVID-19 testing and hospitalization metrics from 2020-03 to 2021-12. It includes the total number of COVID-19 tests performed, the count of positive tests, and the number of positive patients admitted to hospitalization and intensive care units (ICU). Additional details include the average stay in days for both hospitalization and ICU, the total number of patients admitted (both COVID-19 and non-COVID-19), and the percentage of hospital beds occupied by COVID-19 patients relative to the total admitted patients.

Figure 38 (top) is a line graph presenting the comparison between patient admissions with and without COVID-19 over a series of dates.

Figure 38: COVID-19 Impact: Comparative Analysis of Hospital Admissions and Staff Requirements Over Time

The graph plots three distinct categories in Y-axis: 'Patients with no COVID-19' depicted by the blue line, 'Total admitted patients (non-COVID)' in orange, and 'Patients with COVID-19' in green, while the X-axis marks the timeline of the data. The blue and orange lines exhibit a closely aligned trend, indicating that the non-COVID-19 admissions make up the most of total admissions. These two lines show notable fluctuations, with peaks and troughs suggesting variability in patients' admissions over time. The green line, represents COVID-19 patient admissions, is considerably lower than the other categories and follows a distinct pattern, hinting that COVID-19 admissions are not in sync with the general admission trends. In addition, 'Patients with COVID-19' line shows a peak that correspond to the period of the pandemic when lockdowns were in effect. This peak may suggest that during the height of the pandemic, there was a relative increase in COVID-19 admissions compared to other times, possibly because individuals were hesitant to visit hospitals for non-COVID-19 issues.

Figure 38 (bottom) is a line graph presenting the comparison between the COVID-19 Admissions Rate and Staff Requirement spanning from June 2020 to April 2021. There is a visible trend indicating a correlation between COVID-19 admissions and staff requirements, suggesting that as admissions for COVID-19 increased, so did the need for additional hospital staff to meet the high demand prompted by the pandemic.

3.3.2.6 Gripe Patients' Admissions

The dataset contains the number of patients with Gripe admitted to the hospital in monthly bases, spanned from 2016-01 to 2023-12. In the modelling phase, Grid Search methodology was employed to optimize the Random Forest algorithm. Grid Search systematically evaluates a range of parameter combinations to identify the most effective settings for the Random Forest model. Table 9 outlines the Random Forest parameters being adjusted through this process.

Table 9: Optimizing Random Forest Parameters with Grid Search

Parameter	Description	Values
n_estimators	Number of trees in the forest	[100, 200, 300]
max_depth	Maximum depth of the trees	[None, 10, 20, 30]
min_samples_split	Minimum number of samples required to split an internal node	[2, 5, 10]
min_samples_leaf	Minimum number of samples required to be at a leaf node	[1, 2, 4]

Figure 39 illustrates the number of flu patients alongside a pandemic stringency index from 2016 to 2023.

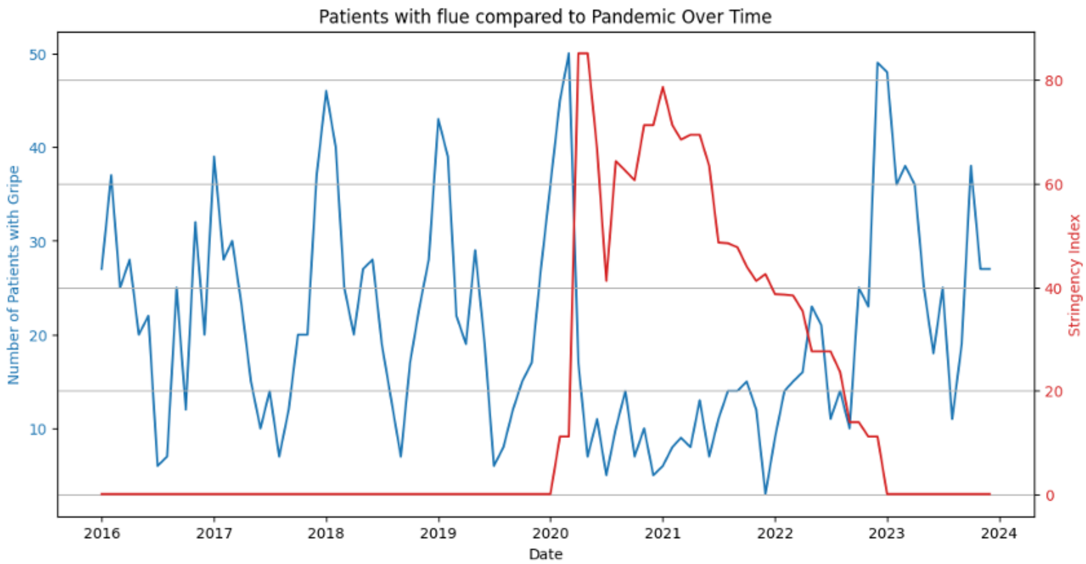
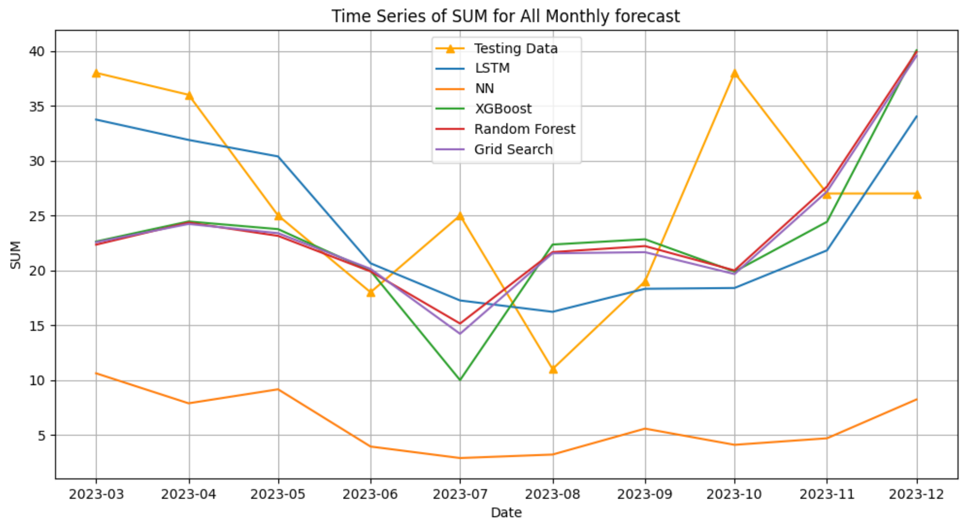


Figure 39: Admitted Patients with Gripe compared to Covid-10 Pandemic

There's a clear seasonal pattern in flu cases with a significant drop at the beginning in 2020, which correlates with a spike in pandemic restrictions, as shown by the stringency index. This suggests that pandemic measures likely reduced flu cases, or people avoided hospitals to prevent catching COVID-19. As restrictions eased, flu cases began to rebound, indicating a return to typical seasonal fluctuations. This data visualizes the broader impact of pandemic measures on public health, highlighting the intersection between stringent health policies and the incidence of seasonal illnesses like the flu.

Figure 40 (top) presents a comparison of various forecasting models against actual testing data over a series of months in 2023. The LSTM and NN models closely mirror the testing data trend, suggesting better performance in capturing the monthly variances, whereas XGBoost and Random Forest models display greater variability. The Grid Search model shows a distinct pattern, indicating varying performance across the parameters tested. The best model found is the LSTM with a RMSE equal to 7.869 and a MAPE equal to 23.763. The RMSE suggests that the LSTM model's predictions are, on average, about 7.87 units away from the actual observed values. The MAPE indicates that the model's predictions are, on average, 23.76% different from the actual observed values.



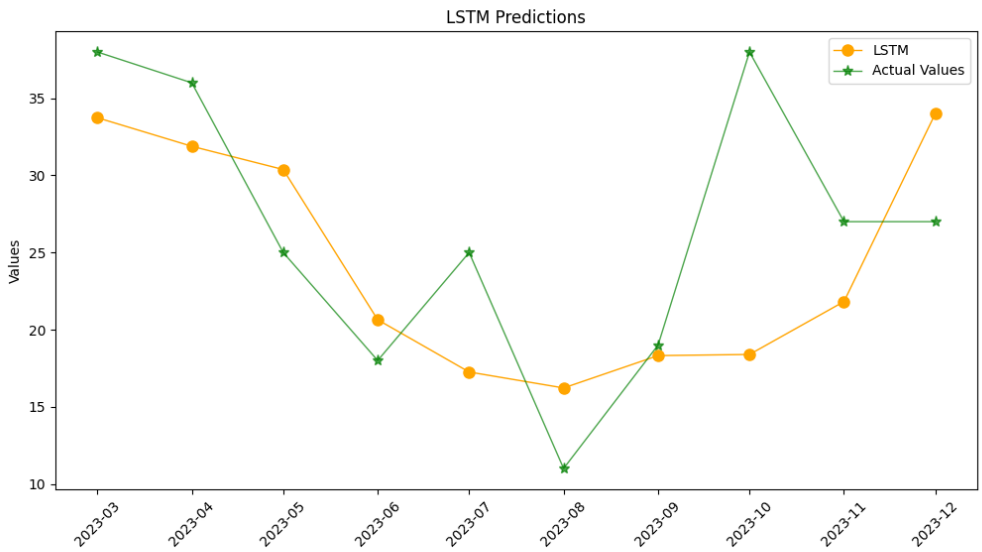


Figure 40: Models' Performance on Admitted Patients with Gripe: all models [top], best model [bottom]

SUNRISE

The Figure 40 (bottom) displays a line chart that compares the predicted values from an LSTM model against actual observed values over a period from March to December 2023. Both lines share a similar shape with noticeable divergence at certain points, implying that while the LSTM model captures the trend to some degree, there are discrepancies between the predictions and actual values.

3.3.2.7 Conclusion

In conclusion, predicting the demand for drugs and consumables presents several challenges, such as supplier shortages, managing hospital inventory, and the evolving impact of COVID-19 on consumer behavior and patient admissions. The complexities of COVID-19's effects, along with people's hesitancy to seek hospital care due to infection risks, stand out as significant hurdles.

However, despite these challenges, the potential of data analytics to help overcome them is evident. By using advanced analytics, visualizations demand patterns easier to grasp can be created, providing better insights to navigate the uncertainties.

Additionally, in order to understand better the nature of health data, data collection and research has been conducted into other pathology with more predictable patterns, such as the flu, which provides a stable basis for analysis. This not only aids in understanding those diseases better but also reveals potential connections with COVID-19 data. The model developed to predict the number of admitted flu patients is aligned with the defined KPIs.

3.4 Water domain

3.4.1 CAFC SPA (CAF)

CAF is interested in demand prediction for 4 different districts. The data, as described in D5.2 [3], is hourly-based, and there is abundant data for training. The final data for each district is calculated by a given formula, which can be altered over the years. For all districts we have experimented with different timelines hourly, daily, and monthly to make predictions in short-term, mid-term and long-term respectively. Due to the amount of data, we use a 60-hours look-back period for hourly-based, 6-days for daily-based and 6-months for monthly-based cases.

SQD compared the data with seasonal variables to identify correlations such as temperature, precipitation, relative humidity, and drought periods. Additional factors such as the stringency index, holidays, seasons, rush hours, and weekdays were incorporated. Among all the exogenous variables, temperature was found to be the most correlated in most places.

3.4.1.1 INTERMEZZO BIAUZZO-CROSERE-LIGNANO

Specifically, for the INTERMEZZO BIAUZZO-CROSERE-LIGNANO district, the formula of calculations from the CI changed on 2022-01-01, affecting the trend. Therefore, data before that date has been discarded during training. As a result, it is not suitable for running ML models for monthly-based timeline due to limited amount of data. In hourly-based data, there was significant noise and no clear pattern. Some peaks remained unexplained by the data alone. Additionally, numerous values close to zero affected the MAPE making it no applicable. No model performed well on an hourly basis.

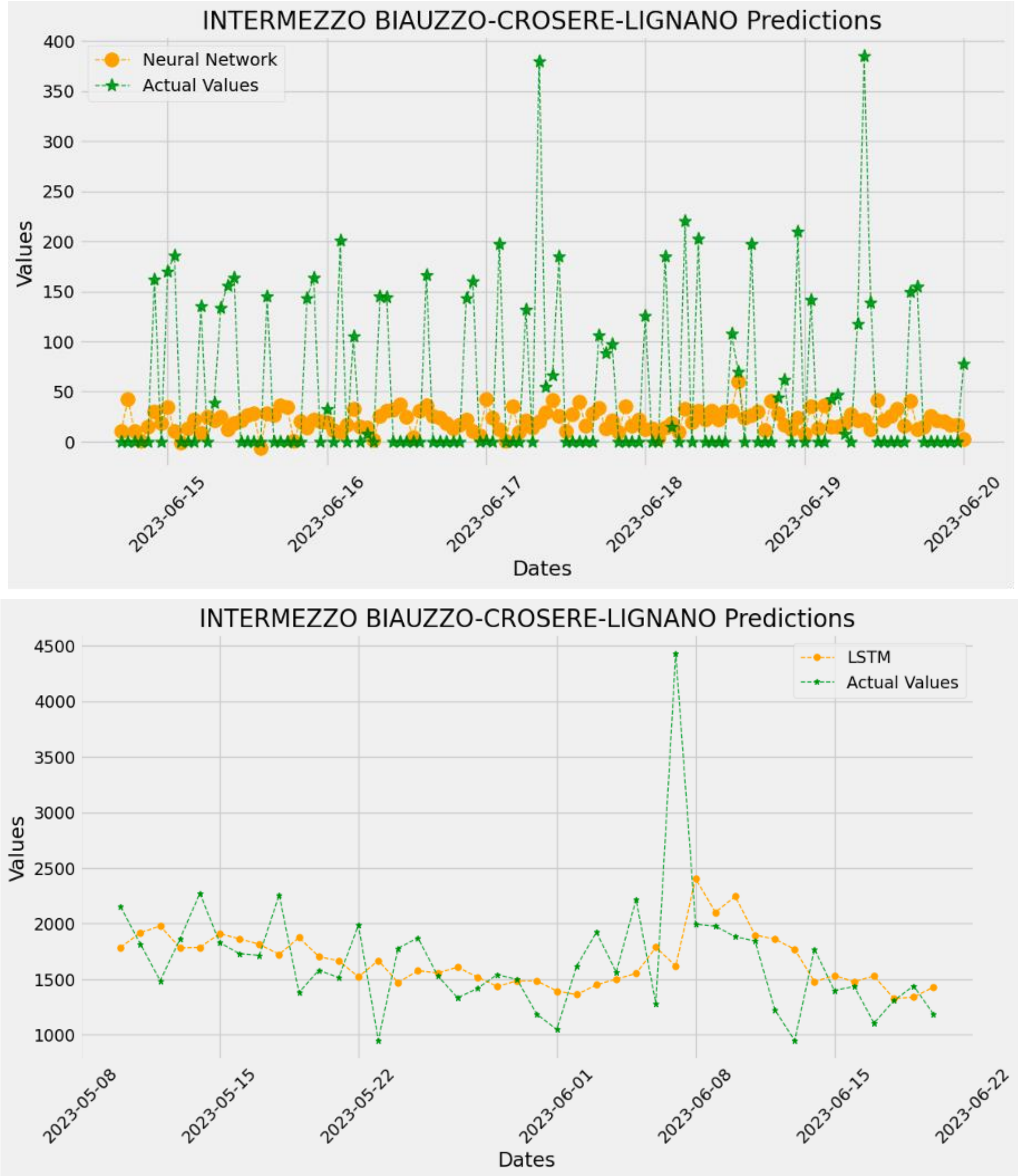


Figure 41: INTERMEZZO BIAUZZO-CROSERE-LIGNANO predictions vs Actual values (daily [top], monthly [bottom])

In Table 10 MAPE is not applicable since there exist many zero values. In contrast, in **daily-based** data, model performance significantly improved. The best model identified was LSTM with a MAPE equal to 18.84 and a RMSE equal to 515.55. An outlier in the test data was noted, which could not be explained and consequently affected the MAPE metric.

Table 10: Hourly trained models (INTERMEZZO BIAUZZO-CROSERE-LIGNANO)

Metric	Neural Network	Random Forest	XGBoost	N-BEATS	LSTM
RMSE	86.51	87.22	92.95	85.63	82.41
R2 Score	-0.144	-0.163	-0.321	-0.121	-0.038

Table 11: Daily trained models (INTERMEZZO BIAUZZO-CROSERE-LIGNANO)

Metric	Neural Network	Random Forest	XGBoost	N-BEATS	LSTM
RMSE	519.37	580.66	586.73	545.11	515.55
MAPE	19.56	22.08	23.16	19.92	18.84
R2 Score	-0.035	-0.294	-0.321	-0.141	-0.020

3.4.1.2 INTERMEZZO BIAUZZO-LIGNANO (via Rivignano)

Hourly-based data showed a positive correlation of 30% with temperature and a negative correlation of -30% with relative humidity. The best model identified was N-BEATS with a RMSE equal to 13.68. MAPE was not applicable since many values were close to zero.

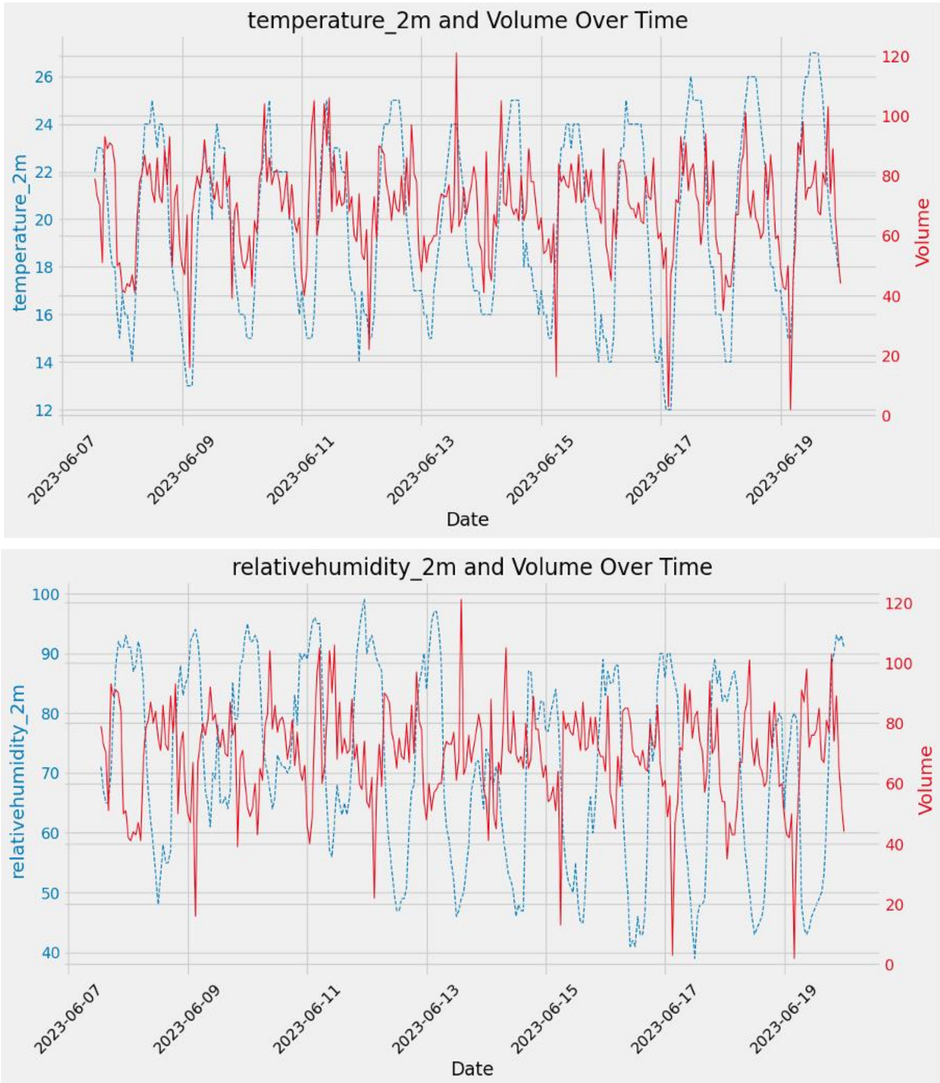


Figure 42: INTERMEZZO BIAUZZO-LIGNANO temperature [top] and relative humidity [bottom] vs volume over time

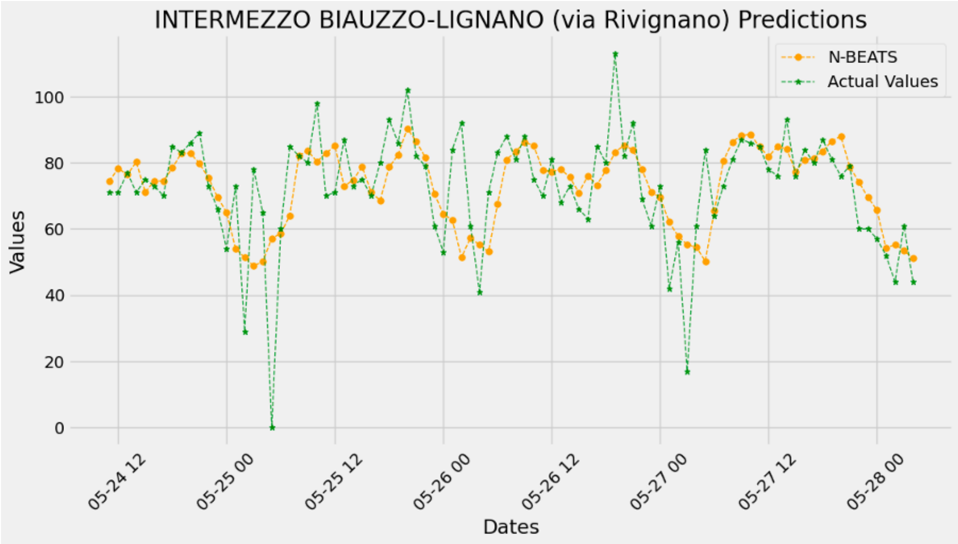


Figure 43: INTERMEZZO BIAUZZO-LIGNANO (via Rivignano) N-BEATS hourly predictions vs actual values

Table 12: Hourly trained models INTERMEZZO BIAUZZO-LIGNANO (via Rivignano)

Metric	Neural Network	Random Forest	XGBoost	N-BEATS	LSTM
RMSE	14.95	14.83	14.51	13.68	14.22
R2 Score	0.095	0.307	0.336	0.410	0.362

In daily-based data, the best model remained N-BEATS with a RMSE equal to 205.70 and a MAPE equal to 8.40, exhibiting a positive correlation of 28% with temperature. An unexplained significant drop on some dates was observed, impacting the model's accuracy. This may be based on possible issues with the CI water supply ability or other external events.

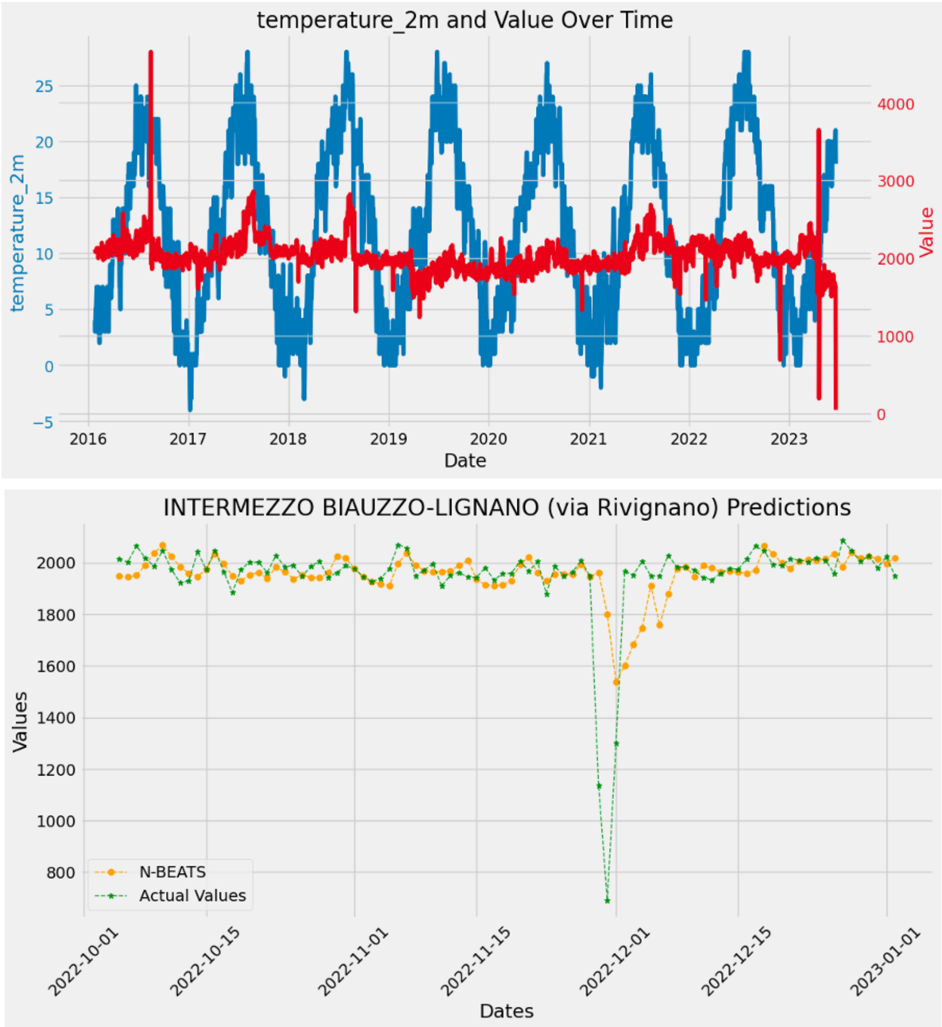


Figure 44: INTERMEZZO BIAUZZO-LIGNANO (via Rivignano) temperature vs volume over time [top], N-BEATS monthly predictions vs actual values [bottom]

Table 13: Daily trained models INTERMEZZO BIAUZZO-LIGNANO (via Rivignano)

Metric	Neural Network	Random Forest	XGBoost	N-BEATS	LSTM
RMSE	228.18	264.56	295.42	205.70	223.06
MAPE	10.04	11.37	12.74	8.40	9.55
R2 Score	0.278	0.030	-0.208	0.413	0.310

3.4.1.3 LATISANA

Hourly-based data revealed a clear trend with several values close to zero, affecting the MAPE metric. Positive correlations were found with temperature at 39% and negative correlations with relative humidity at -37%. The best model identified was LSTM a RMSE equal to 4.06.

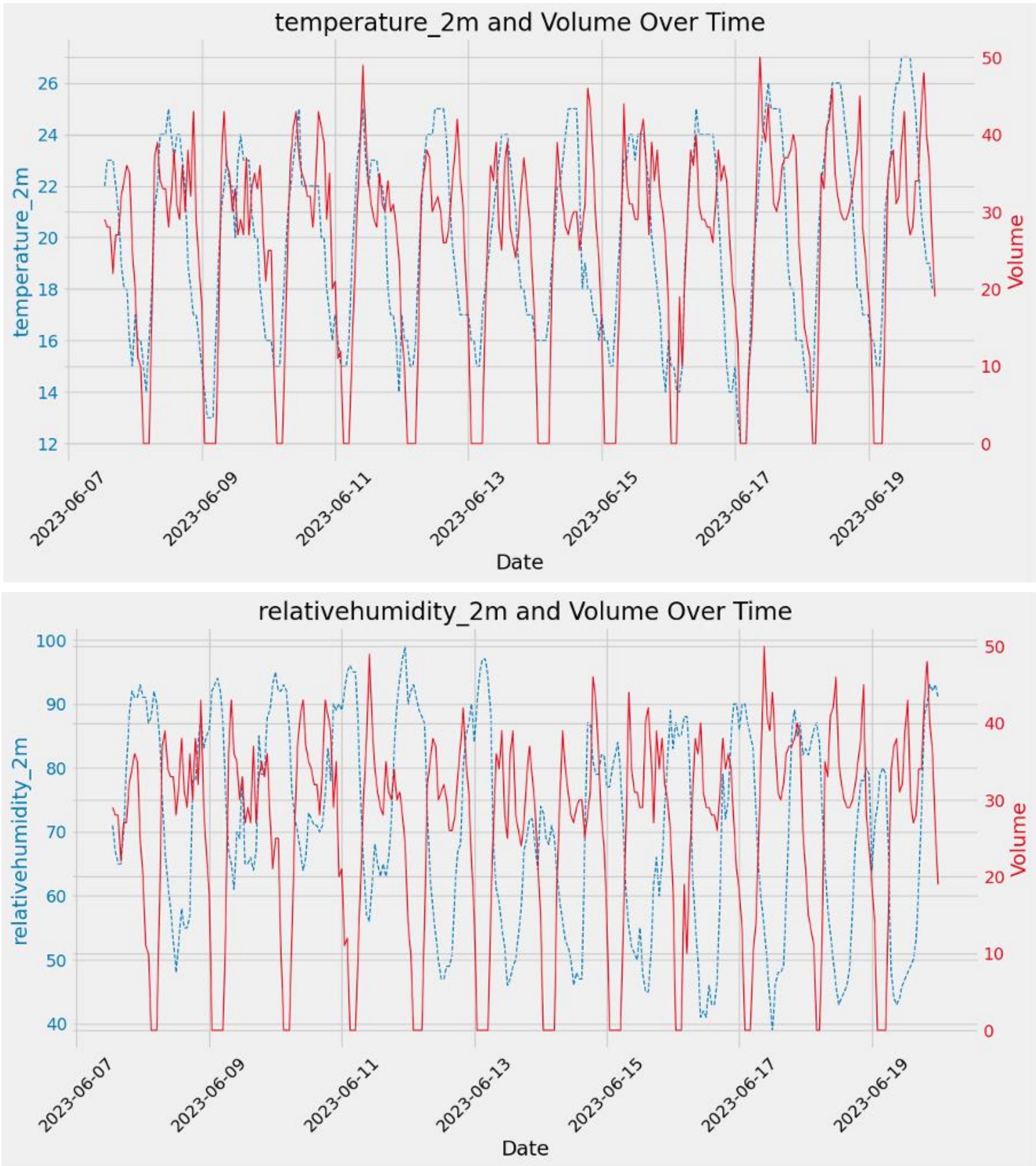


Figure 45: LATISANA hourly temperature [top], relative humidity [bottom] vs volume over time

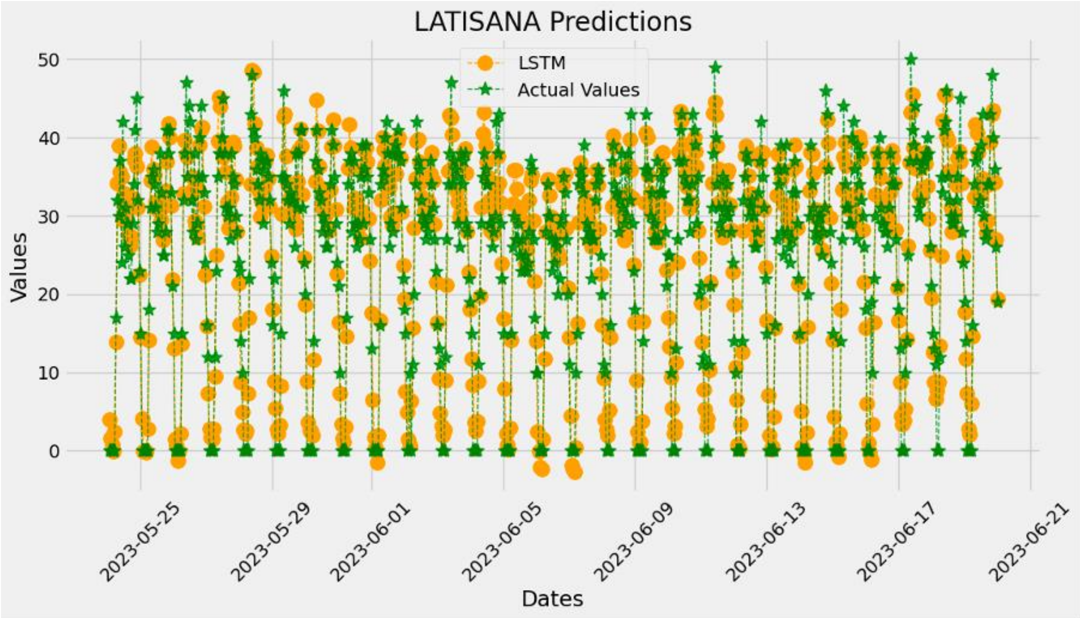


Figure 46: LATISANA LSTM hourly predictions vs actual values

Table 14: Hourly trained models LATISANA

Metric	Neural Network	Random Forest	XGBoost	N-BEATS	LSTM
RMSE	4.27	4.17	4.20	4.17	4.12
R2 Score	0.901	0.906	0.904	0.905	0.908

In daily-based data, positive correlations of 58% with temperature and negative correlations of -26% with relative humidity were observed. The best performing model found was Random Forest, closely followed by the N-BEATS model, with a RMSE equal to 31.84 and a MAPE equal to 4.38. Daily based data exhibited clearer patterns with notable fluctuations and extreme values.

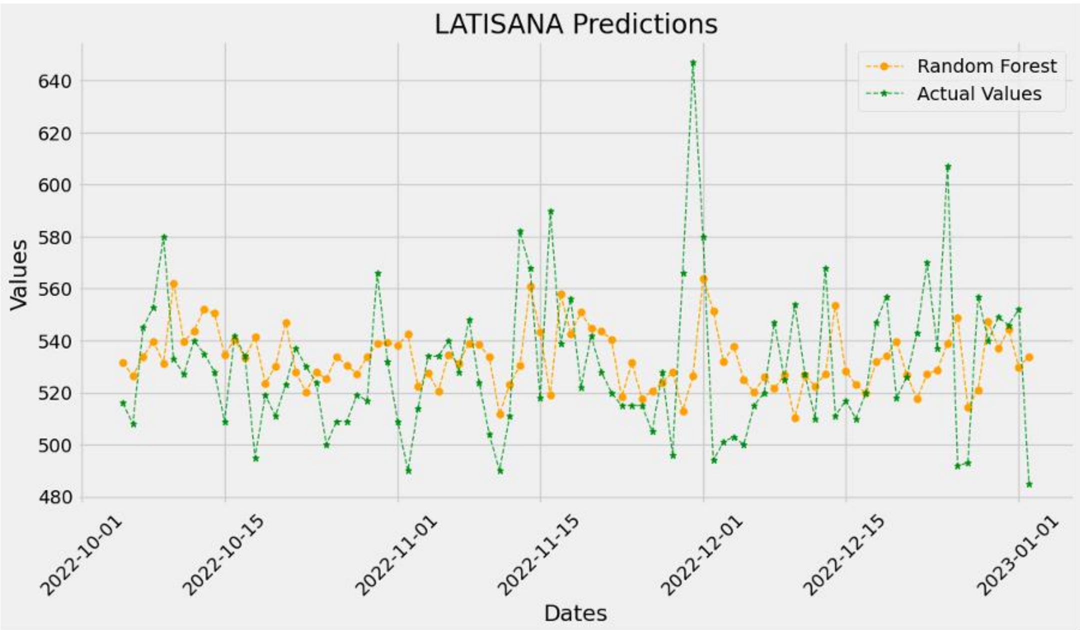


Figure 47: LATISANA Random Forest daily predictions vs actual values

Table 15: Daily trained models LATISANA

Metric	Neural Network	Random Forest	XGBoost	N-BEATS	LSTM
RMSE	34.207726	31.849752	34.884525	32.493793	33.178314
MAPE	4.90	4.38	4.89	4.40	4.54
R2 Score	0.258	0.357	0.229	0.331	0.302

3.4.1.4 LIGNANO

Hourly-based data also exhibited a clear trend, positively correlated with temperature at 77% and negatively correlated with relative humidity at -29%. The best model identified was N-BEATS with a RMSE equal to 26.00 and a MAPE equal to 18.02.

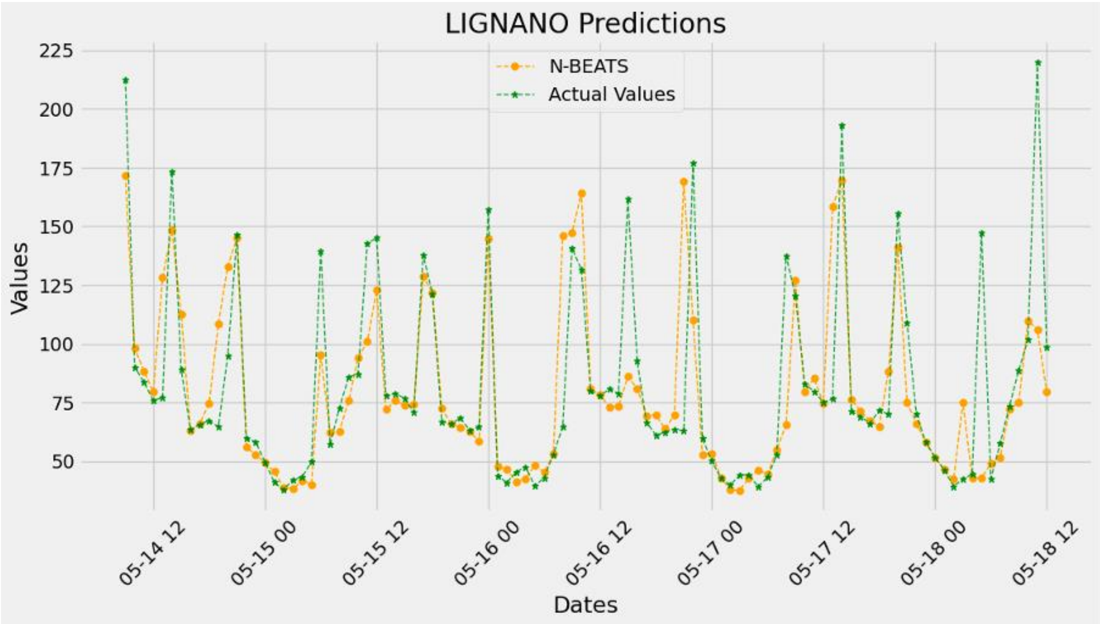


Figure 48: LIGNANO N-BEATS daily predictions vs actual values

Table 16: Hourly trained models LIGNANO

Metric	Neural Network	Random Forest	XGBoost	N-BEATS	LSTM
RMSE	27.67	26.93	26.62	26.00	26.57
MAPE	20.20	25.16	25.55	18.02	23.39
R2 Score	0.528	0.553	0.563	0.583	0.565

Daily based data showed a positive correlation of 87% with temperature and a negative correlation of -30% with relative humidity. All models performed well, with XGBoost identified as the best performing model with a MAPE equal to 6.91 and a RMSE equal to 135.21. These models accurately identified trends and made precise predictions daily.

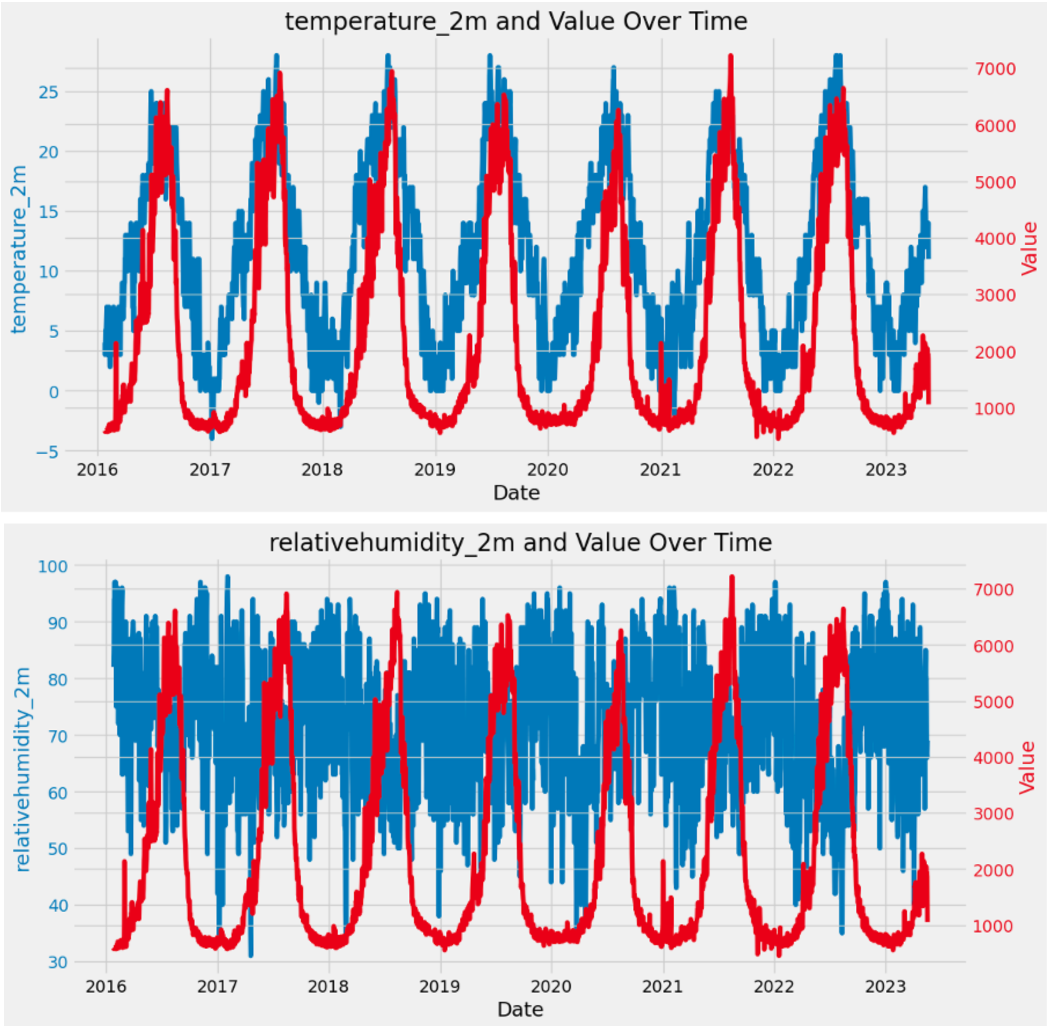


Figure 49: LIGNANO daily temperature [top], relative humidity [bottom] vs volume over time

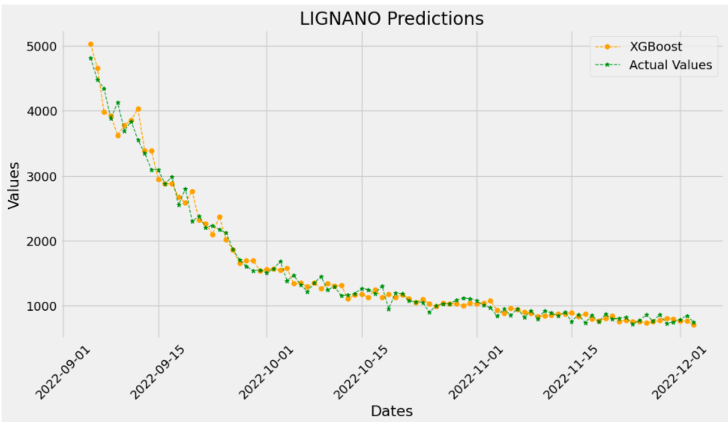


Figure 50: LIGNANO xGboost daily predictions vs actual values

Table 17: Daily trained models LIGNANO

Metric	Neural Network	Random Forest	XGBoost	N-BEATS	LSTM
RMSE	153.58	151.71	140.43	137.12	135.21
MAPE	7.80	7.56	6.91	7.15	7.84
R2 Score	0.977	0.977	0.980	0.981	0.982

3.4.1.5 Conclusion

In this section, we summarize the achievements made in developing the DPM tool for water services and especially for CAF in order to identify future directions to enhance its capabilities. The existing models align well with the project’s KPIs, demonstrating robust and consistent effectiveness in predicting water needs under various conditions. These models take into account multiple factors, including weather variations, restricted movement periods, and seasonal shifts, ensuring the DPM tool's ability to provide thorough analysis across the water supply network.

The models are designed with precision to forecast water volumes across three distinct timeframes: hourly, daily, and monthly. This approach equips the tool to support short-term, mid-term, and long-term operational planning, enabling efficient resource utilization. Looking ahead, the focus will shift towards incorporating climate change projections into our forecasts to improve the tool's predictive accuracy and readiness for future challenges in water supply management.

3.4.2 ACOSOL (ACO)

In recent years, the field of water consumption prediction has witnessed significant advancements due to the integration of modern machine learning and data science techniques. These advancements have led to more accurate and efficient models for forecasting and regression in water consumption prediction. Some notable approaches and trends in the field include deep architectures (LSTM, Transformers models, etc.), different techniques of feature engineering and data preprocessing (time-based features for regression approaches, weather, and external data etc.) and ensemble techniques of machine learning models such as stacking of different models or boosting strategies [19].

In D5.2 [3], the datasets were described in depth and some initial models were experimented as a first approach. Until the submission of D5.3, ACO provided additional data, which helped to improve the models' performance. SQD also attempted to fine-tune the LSTM model presented in D5.2 [3] and included some new models like N-BEATS and Feed Forward Neural Networks. In all of our experiments, we utilized techniques such as dropout and early stopping to avoid overfitting. Due to the monthly-based data and to ensure enough training data, we use a 6-month look-back period.

“Presa” and “Etap” locations exhibit remarkably similar trends and are highly correlated with temperature, at 94%. We experimented with various seasonal data such as drought and precipitation, but temperature emerged as the most correlated variable.

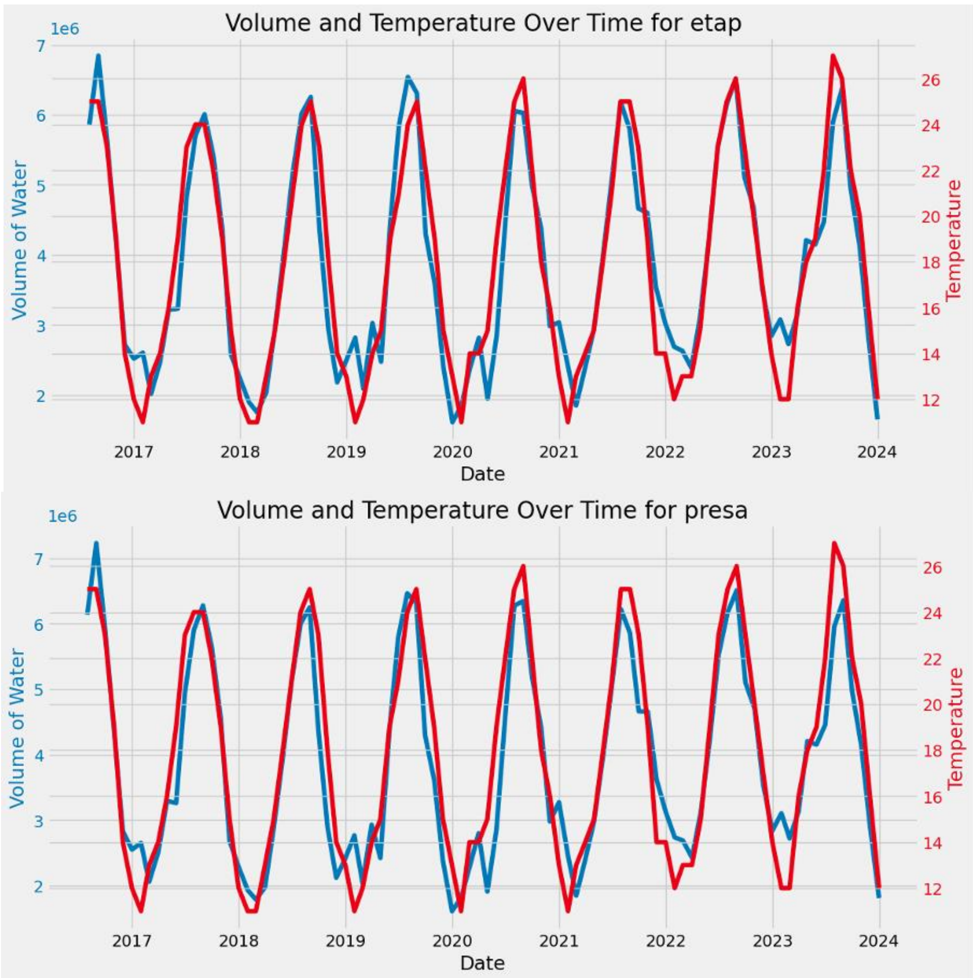
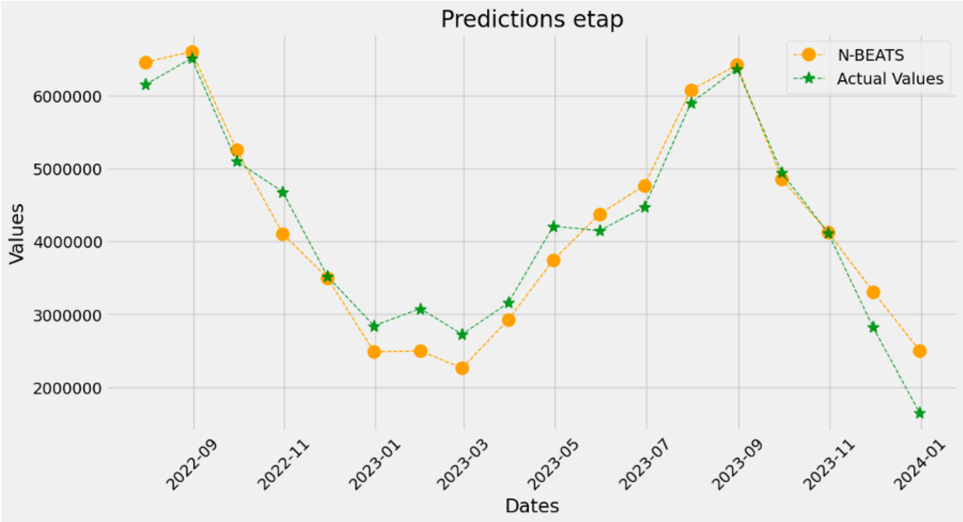


Figure 51: Volume and temperature over time (etap [top], presa [bottom])



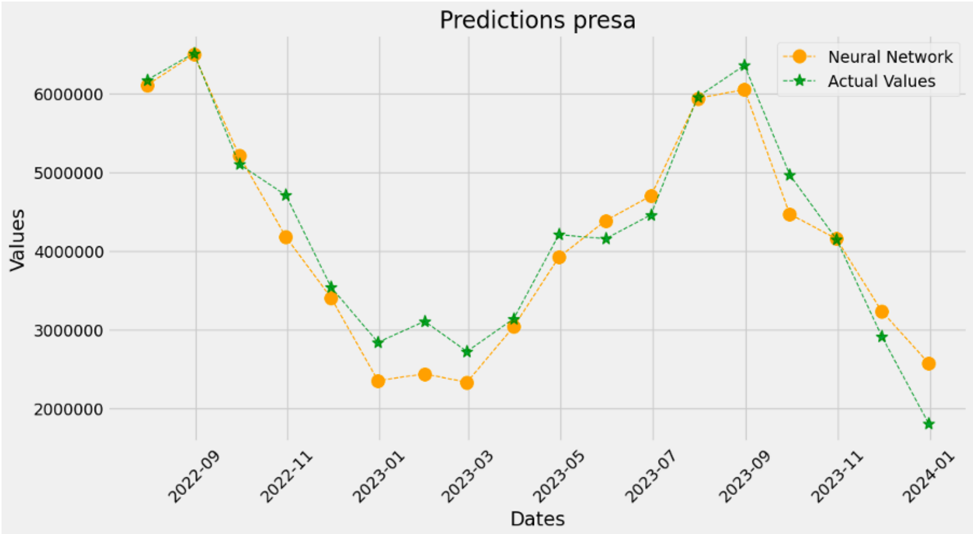


Figure 52: Model predictions vs actual values over time (etap [top], presa [bottom])

For the “etap” location, the best performing model was N-BEATS with a MAPE equal to 9.78 and a RMSE equal to 377064.94. The R^2 score achieved was equal to 92.39%, showing a good capture of the deviations of the target variable by the identified independent variables.

Table 18: Trained models - etap

Metric	Neural Network	Random Forest	XGBoost	N-BEATS	LSTM
RMSE	416097.50	395169.43	395169.43	377064.94	437886.91
MAPE	10.85	10.25	11.26	9.78	10.94
R2 Score	0.9074	0.9165	0.8881	0.9239	0.8973

For the “presa” location, the best performing model was the Feed Forward Neural Network with an RMSE of 363622.65, a MAPE 8.92, and a R^2 Score equal to 92.73%, also showing a good capture of the deviations of the target variable by the identified independent variables.

Table 19: Trained models - presa

Metric	Neural Network	Random Forest	XGBoost	N-BEATS	LSTM
RMSE	363622.65	404502.40	405283.96	408031.50	470872.31
MAPE	8.92	10.04	9.76	9.61	11.33
R2 Score	0.9273	0.91	0.9096	0.9084	0.8780

As for Desaladora, the best model identified was the Neural Network, which exhibited a positive correlation of 26% with temperature and a negative correlation of -51% with drought. However, in Desaladora, due to some values being very close to zero MAPE is not applicable. Nonetheless, the model accurately identified the trend of the actual values, particularly for certain dates.

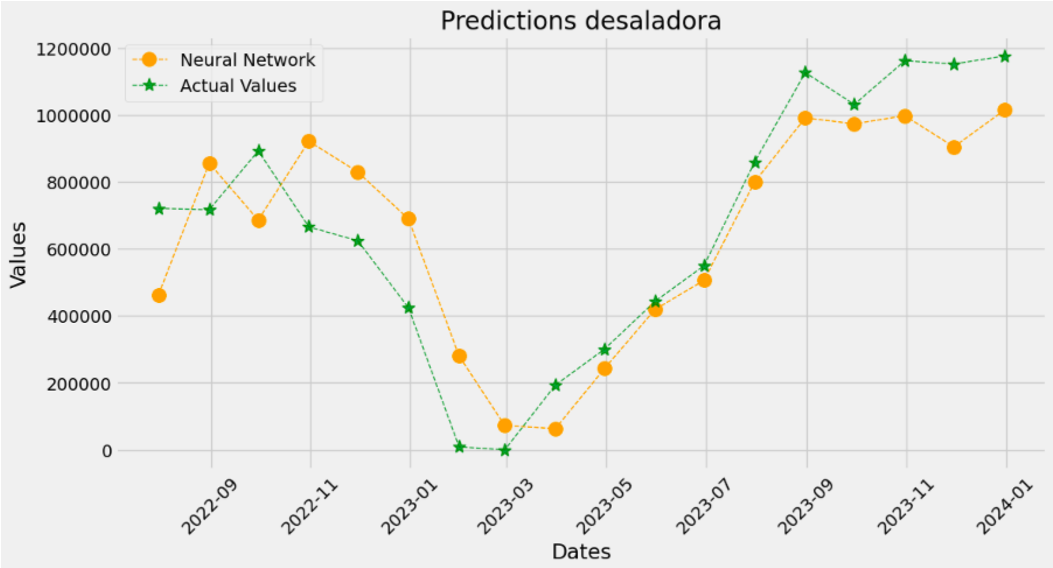


Figure 53: Model predictions vs actual values over time for desaladora

Table 20: Trained models - desaladora

Metric	Neural Network	Random Forest	XGBoost	N-BEATS	LSTM
RMSE	174409.25	313585.21	336771.34	180316.84	196863.78
R2 Score	0.7842	0.3024	0.1955	0.7693	0.7250

For the “River Fuengirola” location, a 45% correlation with temperature was observed, and the best performing model was the Random Forest with a MAPE equal to 22.89 and a RMSE equal to 118703.39.

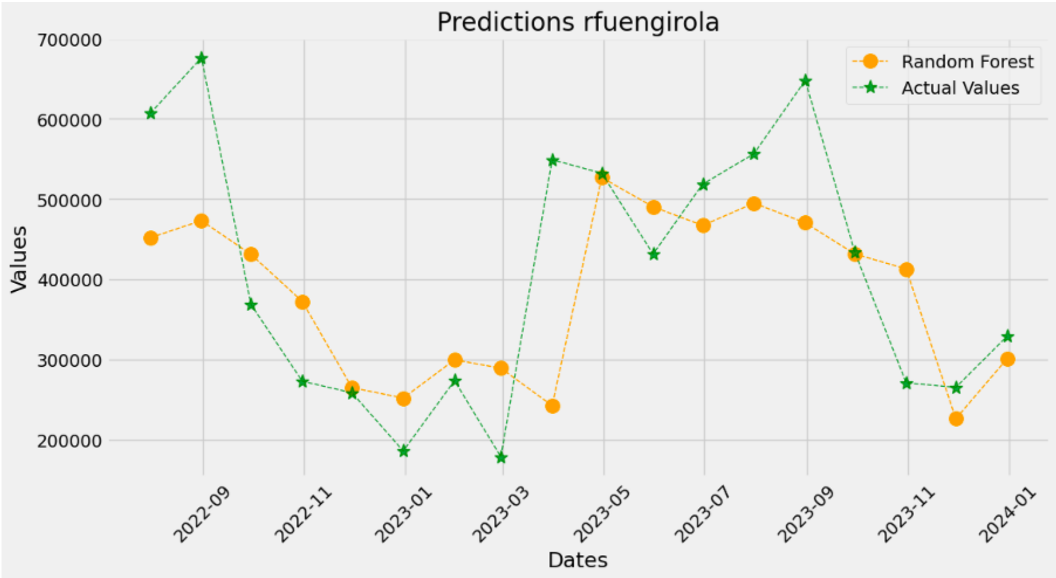


Figure 54: Model predictions vs actual values over time for River Fuengirola

Table 21: Trained models - River Fuengirola

Metric	Neural Network	Random Forest	XGBoost	N-BEATS	LSTM
RMSE	129257.86	118703.39	143714.14	140902.40	121657.87
MAPE	31.85	22.89	33.95	25.57	27.90
R2 Score	0.3259	0.4315	0.1667	0.1990	0.4029

For the “River Guadalmansa” location, all values were equal to zero, indicating no need to run a model. For “SI4” and “Gibraltar”, clear trends were not evident. There were occasional peaks over time that could not be explained by the available data alone. These peaks are known to be attributed to tourism, and since there is no information in the data regarding tourism, the models struggle to identify the trends and make accurate predictions.

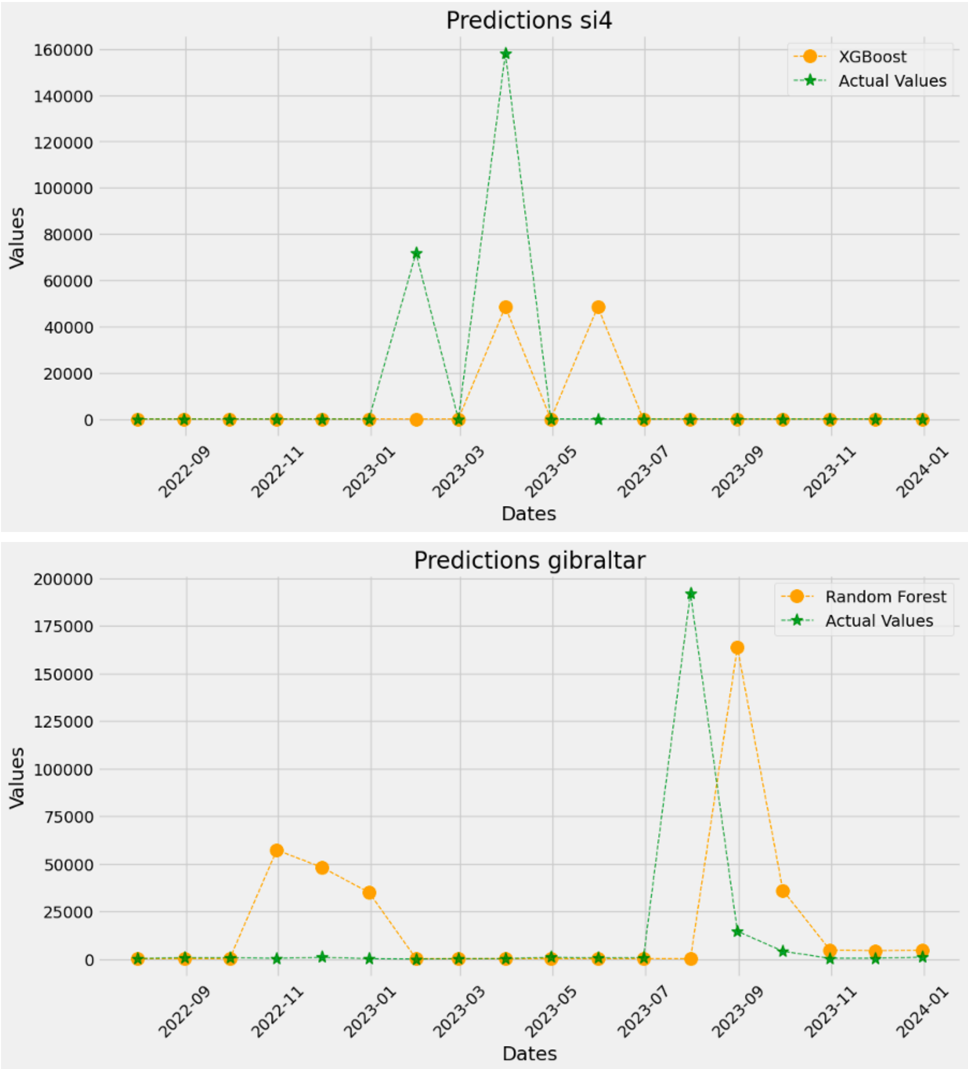


Figure 55: Model predictions vs actual values over time (si4 [top], gibraltar [bottom])

In addition to the prediction model previously demonstrated, INT explored an additional predictive model tailored for the ACOSOL use case for the Y2 WP5 Tool. In our use case, we have decided to utilize some of these state-of-the-art techniques concerning the dataset structure (feature engineering, external data) and employ simpler forecasting and regression models. This decision stems from the fact that very complex models demand extensive computational resources, a large amount of data and in most cases prolonged training times. Additionally, the interpretability of these models might be compromised, making it difficult to extract actionable insights from their predictions. Hence, we have opted to evaluate the performance of the Prophet forecasting model in conjunction with XGBoost regressor. During model selection phase, we assess these models in comparison with the other established forecasting and regression methods.

While opting for straightforward models, our approach involves enhancing their performance by incorporating time-based features, identifying any seasonality and trends within the data, as well as modelling recognizable patterns such as the period influenced by the COVID-19 pandemic.

The scope of this version is to incorporate the new data that the pilot shared into the existing dataset and retrain our models. Moreover, we explore the possibility of including weather features in our training dataset and evaluate their effect on the model's performance.

The newly provided dataset exhibits the exact same structure as its predecessor. The new dataset includes records up to 12/2023 so our updated dataset covers the timeline 01/2016-12/2023.

The complete dataset has 672 rows that correspond to the monthly water consumption data of the known 7 places from year 2016 up to year 2023.

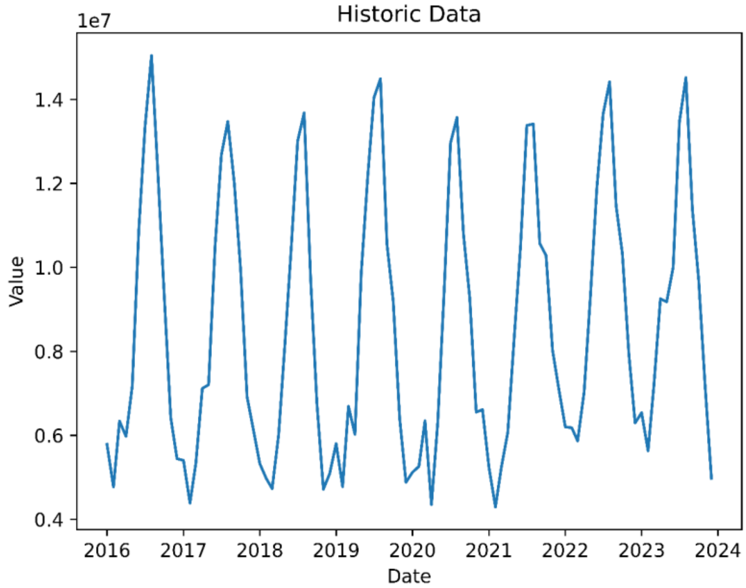


Figure 56: ACO Historic Data

We applied the following formula based on D5.1 [2] and D5.2 [3] for the total inflow per month, on the newly provided data:

total inflow = (inflow from etap) + (inflow from desaladora) + (inflow from rfuengirola) + (inflow from rguadalmansa) + (inflow from gibraltar) +(inflow from si4)

Moreover, duplicate values were identified and removed.

The dataset, as stated before, has 4 features: month, year, place, and values. Firstly, it was examined to see if there were any missing values. We found that there was 1 missing value which we decided to interpolate using linear polynomials. We have also observed that there is a six-month seasonality with the highest values peaking during the summer period. This seasonality needs to be modelled in our approach. We also need to consider the COVID-19 period in our dataset. We have decided to explore our models' performance with and without modelling the COVID-19 period [20].

We have approached this problem in two different ways. The first one is using forecasting models to predict the water inflow. Our ambition is to put prophet model to the test because it is a very-easy-to-interpret model, and the development approach is straight-forward, too. Of course, we test more than one forecaster, but as we will see next, prophet gives us very promising results. Prophet takes as input the date (ds) in a special format (YYYY-MM-DD) and the target (y), which in our case is the computed total inflow value.

Table 22: Example of data input for forecasting models; ds is the date on yyyy-mm-dd format and y is the total inflow.

	ds	y
0	2016-01-01	5789388.4
1	2016-02-01	4773378.2

The second approach is using regression models. In this context we use time columns as features, which in our case are month and year. We also need the target variable in our dataset, meaning the value we will predict, which is the value column, and it represents the total water inflow.

Table 23: Example of data input for regression models; The features originally used is year month and the target value is the total water inflow.

	year	month	value
0	2016	1	5789388.4
1	2016	2	4773378.2

In both methods, we consider modelling COVID-19 period in our dataset. In the forecasting strategy, we represent COVID-19 period as a "one-off holiday", which is a parameter of prophet model as we will describe next, and in the regression-strategy we model it as a binary feature indicating weather is a pandemic period or not (1 and 0 correspondingly).

On top of the features used for the regression models we added the following weather data [20].

Utilizing the Open Meteo API allows for the access of valuable weather data, which can then be utilized as features. The API provides either data either on hourly or daily frequency.

Features that were explored:

- Temperature Minimum
- Temperature Maximum
- Humidity
- Summary of Precipitation

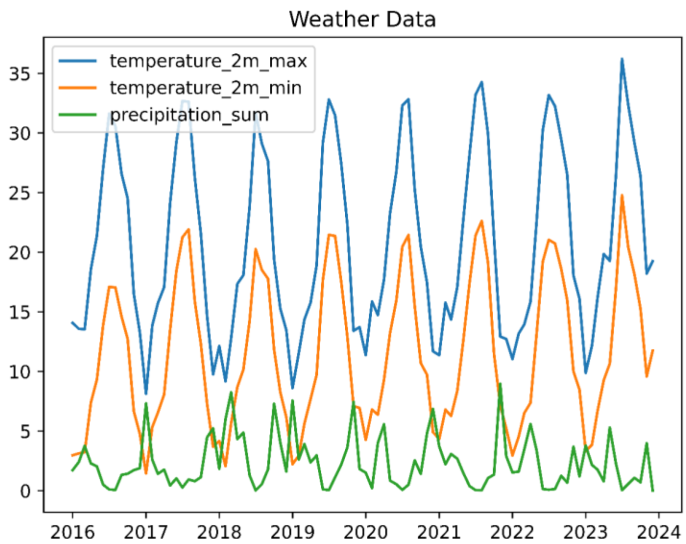


Figure 57: Weather data features across time.

However, using weather forecasts as input for the models introduces the following limitations:

There is a misalignment between the Open Meteo API's 15-day forecast horizon and our need for monthly weather predictions during the inference process. Since our dataset and consequently our forecasts are monthly, it is very difficult to find a source that provides forecasts for a whole month.

And if there is such a source, it is very likely that there will be considerable error in the forecasts as the horizon increases.

To overcome this issue and be able to provide predictions, we used a linear approach to provide the historical data (the year before) of the forecast during the inference process. By incorporating these historical data, the models can establish a baseline and capture broader trends/seasonality.

Moreover, we explored the correlation between the new features:

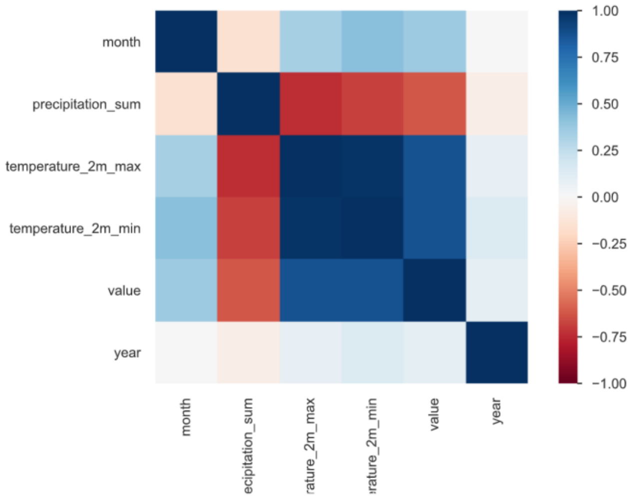


Figure 58: Pearson's correlation matrix between weather features, the target value(inflow) and the time features.

The correlation matrix of the features indicates notable associations with the weather features displaying a significant correlation with the target value.

3.4.2.1 Additional models

We have explored the same models as in the previous report. For reference, check the report for the D5.1 [2].

For the prophet [21] and xgboost [22], we have used the best parameters identified in the previous version. In all cases the models showcased better performance with their default parameters which are:

- Prophet: changepoint_prior_scale: 0.05, seasonality_mode: additive and seasonality_prior_scale: 10.0
- XGBoost: learning_rate: 0.1, max_depth: 3, n-estimators: 100, random_state: 42

We have split our data into train (80%) and test (20%) set to evaluate our selected models on completely unseen data. Our evaluation is based on all available metrics but mostly on R-squared [23], which gives an overall view about the correlation of our features to the target variable and MAPE (Mean Absolute Percentage Error) [24], which gives us a comparable metric across all models.

The approaches we wanted to test are:

1. Using all available data from 01/2016-12/2023.
2. Removing some of the older data to avoid any redundancy that might have been introduced. In our case we remove data older than 2018.
3. Adding weather features such as temperature and humidity.
4. Adding daily weather data and averaging to have monthly input. The features in this case are maximum and minimum temperature and the sum of the precipitation.
5. Scaling all features using both min-max and standard scaling.

Results case 1 (all data):

Table 24: Results when using all available data from 01/2016 to 12/2023.

Model Name	MAPE	R2	RMSE	MAE
CatBoostRegressor	**0.117**	**0.823**	**1203259**	**1030028**
DecisionTreeRegressor	**0.101**	**0.87**	**1031741**	**849004**
GradientBoostingRegressor	**0.097**	**0.879**	**993910.5**	**770620**
kneighbor_regressor	**0.157**	**0.62**	**1761747**	**1438616**
LGBM_regressor	**0.18**	**0.589**	**1832570**	**1535532**
prophet	**0.059**	**0.946**	**364194.1**	**283490.4**
xgb_regressor	**0.093**	**0.882**	**981653.9**	**749995**

Results case 2 (removed data older than 2018):

Table 25: Results when removing older data; older than 2018.

Model Name	MAPE	R2	RMSE	MAE
CatBoostRegressor	**0.133**	**0.776**	**1268968**	**960769**
DecisionTreeRegressor	**0.104**	**0.842**	**1068012**	**757934.1**
GradientBoostingRegressor	**0.126**	**0.791**	**1226900**	**890826.4**
kneighbor_regressor	**0.154**	**0.697**	**1476529**	**1191037**
LGBM_regressor	**0.245**	**0.169**	**2446418**	**1892524**
Naïve forecaster	**0.475**	**-2.022**	**2396196**	**1039417**
prophet	**0.084**	**0.907**	**455790.9**	**349064.8**
xgb_regressor	**0.112**	**0.822**	**1130693**	**802359.1**

Results case 3 (weather data):

Table 26: Results when adding weather features such as temperature and humidity.

Model Name	MAPE	R2	RMSE	MAE
CatBoostRegressor	**0.141**	**0.737**	**1465883**	**1117298**
DecisionTreeRegressor	**0.144**	**0.718**	**1517593**	**1190779**
Extra Trees Regressor	**0.125**	**0.773**	**1360728**	**924879.5**
GradientBoostingRegressor	**0.108**	**0.823**	**1201295**	**787456.9**
kneighbor_regressor	**0.18**	**0.465**	**2090277**	**1313776**
LGBM_regressor	**0.148**	**0.725**	**1499436**	**1267876**
xgb_regressor	**0.117**	**0.805**	**1263366**	**918085.6**

Results case 4 (averaging daily weather data):

Table 27: Results when adding the monthly average of daily weather data.

Model Name	MAPE	R2	RMSE	MAE
CatBoostRegressor	**0.134**	**0.739**	**1371722**	**945603.4**
DecisionTreeRegressor	**0.165**	**0.584**	**1730658**	**1117075**
Extra Trees Regressor	**0.119**	**0.773**	**1278619**	**795184.2**
GradientBoostingRegressor	**0.13**	**0.75**	**1341065**	**921185**
kneighbor_regressor	**0.137**	**0.733**	**1385115**	**1061872**
LGBM_regressor	**0.207**	**0.458**	**1974332**	**1581409**
xgb_regressor	**0.121**	**0.81**	**1168616**	**900238.6**

Results case 5 (min-max):

Table 28: Results when scaling the features using min-max.

Model Name	MAPE	R2	RMSE	MSE	MAE
CatBoostRegressor	**0.373**	**0.823**	**0.112**	**0.013**	**0.096**
DecisionTreeRegressor	0.331	0.87	0.096	0.009	0.079
ExtraTreesRegressor	**0.331**	**0.87**	**0.096**	**0.009**	**0.079**
GradientBoostingRegressor	0.343	0.879	0.092	0.009	0.072
kneighbor_regressor	**0.464**	**0.62**	**0.164**	**0.027**	**0.134**
naive_forecaster	0.383	-2.41	2.779	7.72	1.637
prophet	**0.57**	**-0.889**	**0.368**	**0.135**	**0.298**
xgb_regressor	**0.328**	**0.875**	**0.094**	**0.009**	**0.077**

Results case 5 (standardize):

Table 29: Results when scaling the features using standard scaling.

Model Name	MAPE	R2	RMSE	MSE	MAE
CatBoostRegressor	0.536	0.823	0.386	0.149	0.33
DecisionTreeRegressor	0.501	0.87	0.331	0.109	0.272
ExtraTreesRegressor	0.501	0.87	0.331	0.109	0.272
GradientBoostingRegressor	0.453	0.879	0.319	0.102	0.247
kneighbor_regressor	0.784	0.62	0.565	0.319	0.461
LGBM_regressor	0.862	0.589	0.588	0.345	0.492
naive_forecaster	0.565	-2.41	2.882	8.305	1.89
prophet	**0.529**	**0.935**	**0.247**	**0.061**	**0.202**
xgb_regressor	0.48	0.877	0.322	0.104	0.263

As we can observe, the best metrics are achieved in most cases by XGBoost and Prophet. Comparing across the different approaches, we can see that the best model in terms of MAPE, was achieved in case 1, when trained on the complete dataset, without using weather features and without scaling. The top performer is prophet, and it achieves a MAPE of 5.9% and R2 equal to 94.6%. Comparing xgboost's performance on the different datasets, we can observe that there are not major differences. Indeed, this model also achieves its best results on the complete dataset without weather features and with no scaling. We can also observe that the removal of older data does not enhance the performance of the models, indicating that the information provided before 2018 is not redundant.

The graph below depicts the prediction of the best model on the test set (20% of data), along with the real water inflow value.

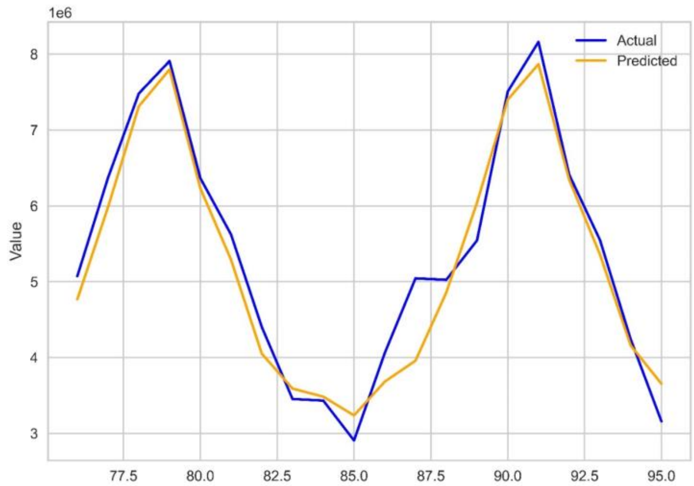


Figure 59: Predicted vs Actual water inflow using the best model (prophet model on all available data).

As we can observe, the model has managed to learn the pattern of our data.

Moreover, we provide additional models' predictions on the test set, to understand how small the differences between the different approaches are.

Predictions of Prophet trained on complete dataset with standard scaling:

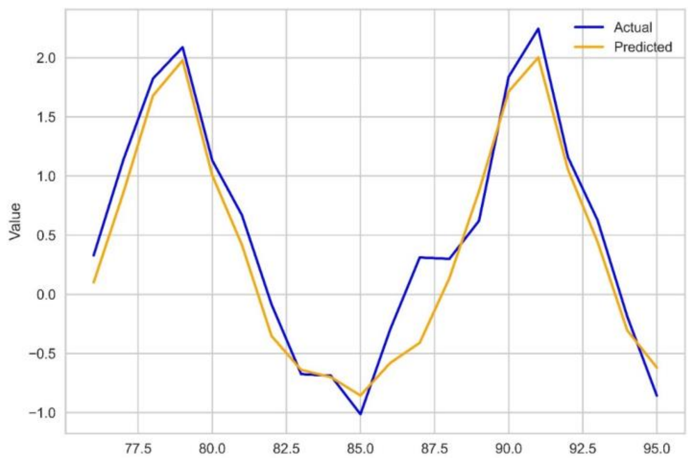


Figure 60: Predicted vs Actual water inflow using the prophet model on scaled data.

Predictions of XGBoost trained on complete dataset:

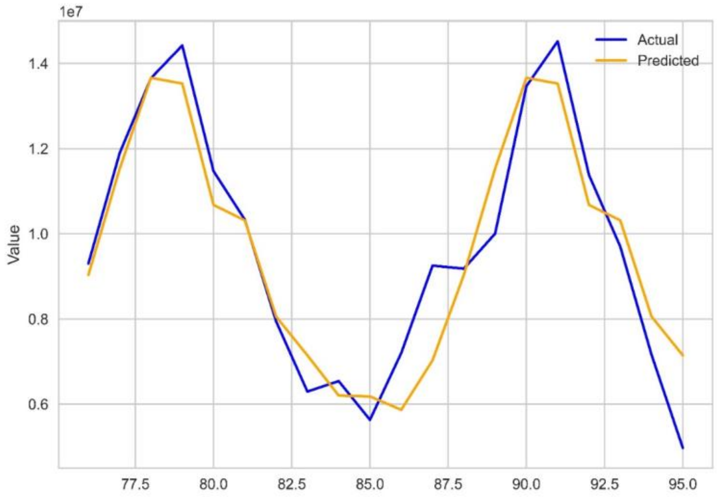


Figure 61: Predicted vs Actual water inflow using the XGBoost model on all available data.

Predictions of XGBoost trained on dataset with additional weather features (temperature and humidity):

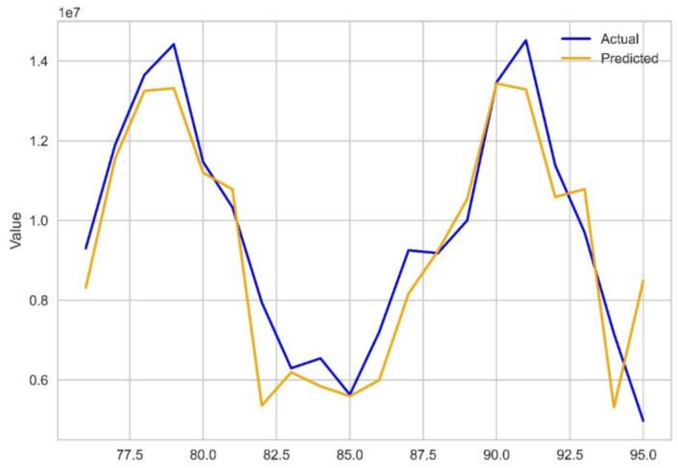


Figure 62: Predicted vs Actual water inflow using the XGBoost model on data with weather features.

3.4.2.2 Conclusion

In this section, the progress made is reviewed and highlighted the next steps in enhancing the DPM tool for water services. The current models are aligned with the project's KPIs, showcasing consistent performance and effectiveness in predicting water needs. These models are designed to account for various factors, such as weather conditions, periods of restricted movement, and seasonal changes. This adaptability enables the DPM tool to support a range of analysis within the water supply network.

The forecasting models are purposed to predict the water volume that will flow through the distribution network over the next two years, giving a clear window for operational planning and the efficient use of resources. The forthcoming focus is on incorporating climate change projections into our predictive analysis to enhance further the long-term adjustments.

4 Integration and Deployment

4.1 Integration

4.1.1 Frontend - Backend and auth

There are several key services and components that need to be integrated and deployed, as depicted in Figure 63

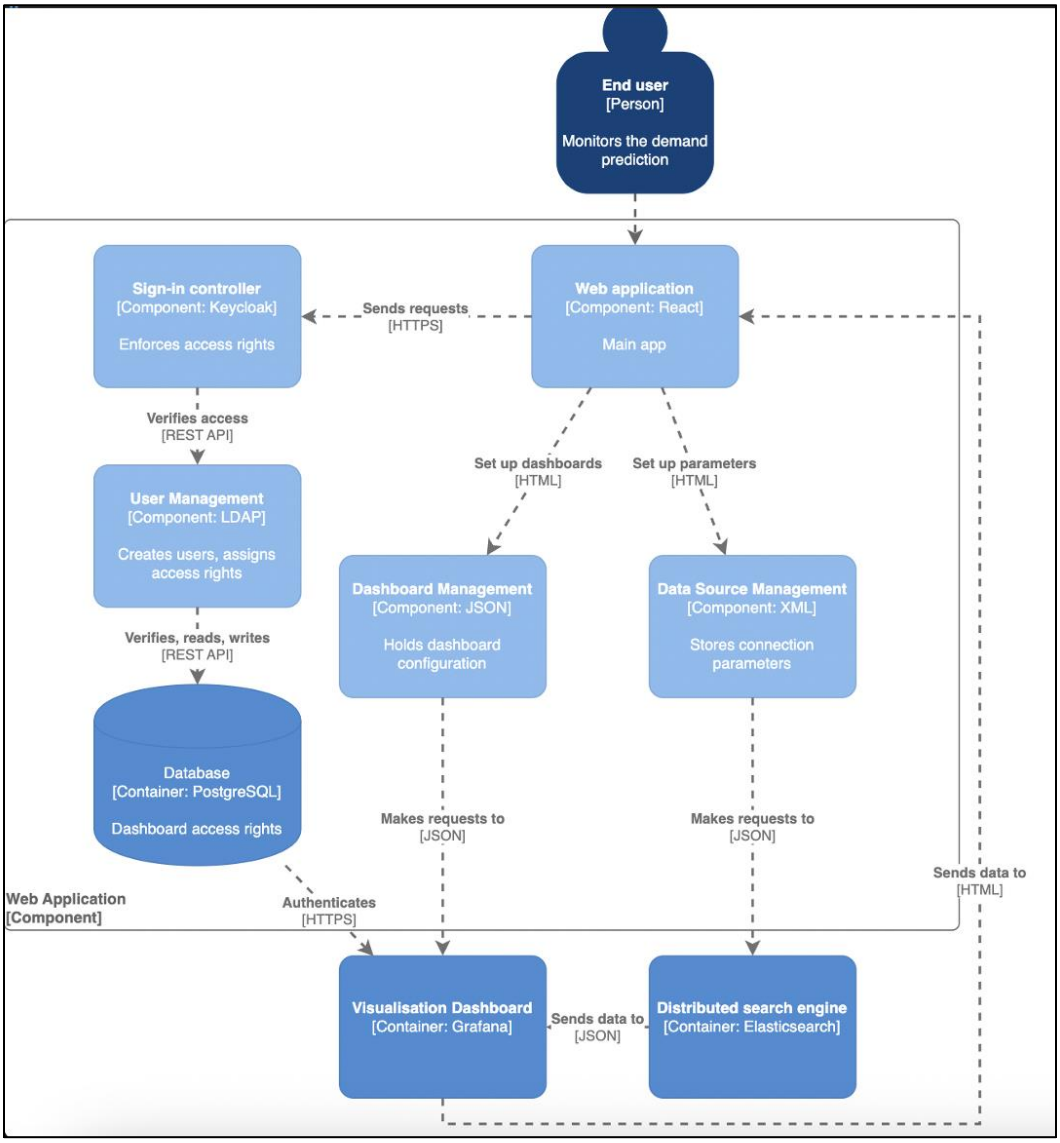


Figure 63: Component Architecture

The integration of these services and components will involve the following steps:

- Sign-in Controller and User Management:
  - The Sign-in Controller will authenticate users against the User Management system, which will utilize LDAP to manage user credentials and roles. This part is Keycloak's responsibility as FE will get the tokens from the service and BE will have to authenticate and authorize using the same service. Token expiration and user level authorization is in place to ensure only permitted users have access.

- User Management will also interact with the Database to retrieve or update access rights, via a different dashboard application page. If a user's rights are upgraded or downgraded, the database will be updated accordingly, modifying their access to restricted resources. Old tokens will remain valid until they expire.

- Database: Stores information and manages dashboard access rights, with a containerized PostgreSQL service. Each functionality mentioned operates with its own PostgreSQL database, similar to Keycloak. Database holds all the important information about users and also sensitive information are encoded with cryptography to prevent sensitive user info from leaking like passwords. Database allows for persistence and migrations too in case that is needed.
- Web Application (React):
  - The main React application will consume services from Dashboard Management and Data Source Management for configuration and parameter setting.
  - The Web Application will also communicate with the Visualization Dashboard and the Distributed Search Engine to fetch and display data.
- Dashboard and Data Source Management:
  - These components will manage configurations and parameters, which are essential for the setup and customization of dashboards in the Web Application and Visualization Dashboard.
- Visualization Dashboard (Grafana) and Distributed Search Engine (Elasticsearch):
  - Grafana will be used to visualize the data, which requires access to the database and possibly to Elasticsearch for fetching and rendering data in real-time.
  - Elasticsearch will provide search capabilities, processing data requests made by the Web Application or directly by Grafana.

4.1.2 Frontend - Backend

Based on Figure 64, we have several components that interact with each other through various interfaces. The system comprises a Web Application using React for the frontend (FE), a Visualization Dashboard utilizing Grafana, a Distributed Search Engine with Elasticsearch, a Data Uploading System, a Data Lake based on S3 MinIO, and a Database used for timeseries with PostgreSQL. Additionally, there's an Analytics & Orchestration system that includes a custom orchestration tool for analytics jobs.

Web Application (Frontend)

- The React-based Web Application is the user interface that serves the dashboard for the end-users.
- It interacts with the Visualization Dashboard to display graphical data representations and with the Distributed Search Engine to retrieve search results.
- The Data Uploading System feeds data into the application, possibly through an API that the frontend consumes.

Visualization Dashboard

- The Visualization Dashboard uses Grafana to render visual data. It obtains data from the backend services and can send processed data back to the Distributed Search Engine in a JSON format, which is easily rendered in the browser.

Distributed Search Engine

- Elasticsearch indexes and makes searchable data that could come from various sources, including streams from the Data Lake and transactions from the Database.
- It provides a RESTful interface that the frontend can query to fetch search results.

Backend Components

- The Data Lake (using MinIO, an S3-compatible storage) is the repository for raw data, which the system can access through a python SDK client.
- The Database (a timeseries-enabled PostgreSQL) is used for structured, time-series data and can be accessed and manipulated using PostgreSQL's driver/adapter.
- The Analytics & Orchestration system includes cron jobs for periodic tasks and analytics workflows that read from and write to the Database and Data Lake.

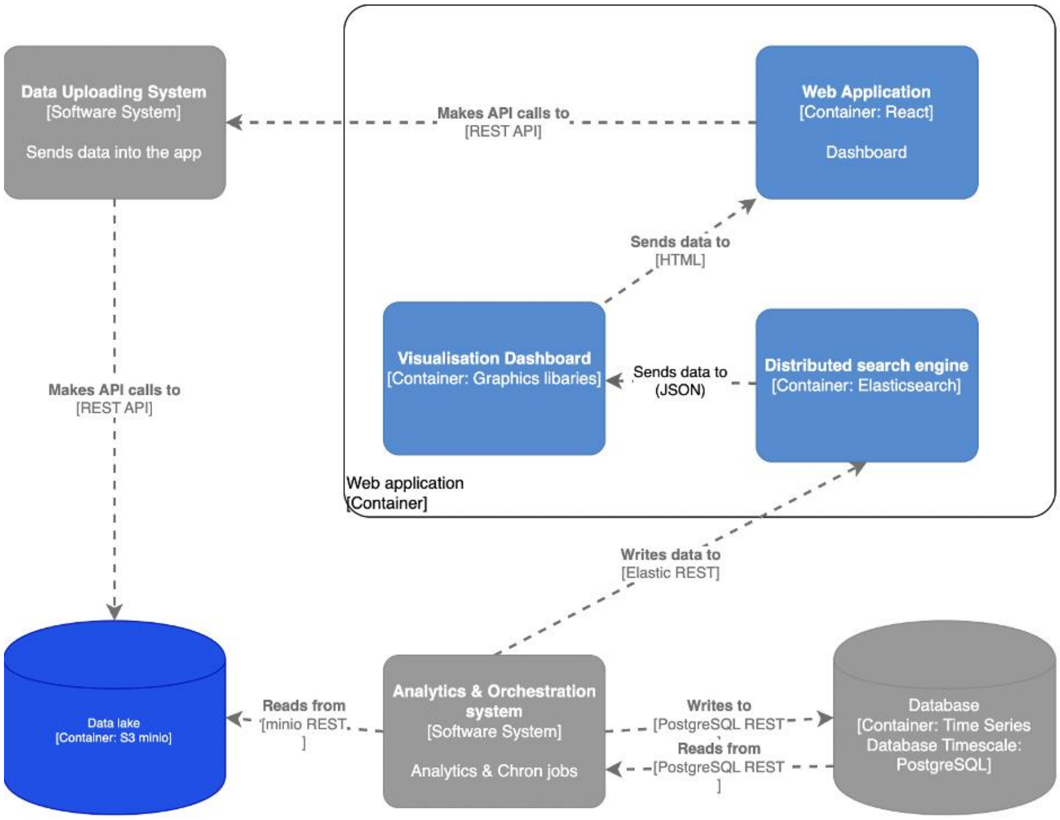


Figure 64: Architecture diagram for the connection of FE - BE

4.1.3 Backend components

The integration and deployment of the Backend components in a Kubernetes environment can be understood in the following manner:

Cron Engine:

- Acts as the orchestrator for various backend pipelines, responsible for scheduling tasks, triggering operations in other services, and overall workflow & data management.
- It is responsible for running the **Analytics & Forecasting Engine** and **Training Engine** to train the models and produce forecasts based on historical data.

Analytics & Forecasting Engine and **Training Engine**:

- Both engines are involved in reading and processing data, performing analytics, and training AI models. They interact with databases to read input data and write results/outputs.
- The training engine logs its processing steps and results to MLflow for tracking experiments, model versions, and performance metrics.

Data Lake (MinIO):

- Serves as the centralized storage for model artifacts and raw data. MinIO, an S3-compatible object storage system, will handle large datasets and model binaries.
- It is accessed by the backend pipelines for storing and retrieving model artifacts, training data and temporary files.

Databases:

- There are several databases used in the BE, all of them for different reasons. Some are used internally by components used in the operations and others used for storing and retrieving data.
- The main database used to store data is **PostgreSQL** and serves as the single source of truth for accessing data. The raw data from **Minio** are processed, cleaned, formatted and saved for later to be used in the training pipeline.
- Other databases are used for the needs of other tools like MLFlow and Keycloak.

Maintenance of comprehensive documentation, including API specifications with tools like Swagger/OpenAPI for easier integration and communication between frontend and backend services will be handed.

4.2 Deployment

During deployment phase, different deployments have been conducted for each CI. Each CI has its own API service and personalized dashboards for the specific needs. The deployed services will be isolated from each other resulting in better personalisation and security of the different environments. The way of deployment follows a vertical approach meaning that each CI will have its own services independent from each other.

The deployment cluster is based on Kubernetes and is composed of 3 machines in SQD's premises. A cluster is needed to prevent data failure, achieve better performance and be more fault tolerant than just one machine alone. A cluster is a group of servers and other resources that act like a single system and enable high availability, load balancing and parallel processing. These systems can range from a two-node system of two personal computers (PCs) to a supercomputer that has a cluster architecture. Running multiple containers per node increases resource utilization, and ensuring an instance of each container is running on more than one node at a time prevents applications from having a single point of failure. The cluster specifications are briefly described below:

Each machine features:

- Intel® Core™ i5-13500 CPU
- 6-Performance-Cores + 8-Efficient-Cores
- Raptor Lake-S
- Hyper-Threading Technology Virtualization (Intel-VT)
- 64 GB DDR4 RAM
- 2 x 512 GB (Gen4) NVMe SSD

Clusters are always scalable up or down, vertical or horizontal scaled based on specified project needs. By horizontal scaling we mean adding more machines in the cluster to make it more powerful (or removing if not needed anymore) and vertical scaling is just keeping the number of components (machines) the same but upgrading their very specifications until the performance requirements are met. Horizontal scaling is more efficient and more widely used thus it is used here as well.

Some of the important topics that deployment is taking into account, include:

- **Containerization:** All the components of the tool, especially those smaller services (Grafana, PostgreSQL, Elasticsearch), must be packaged as Docker images. Containerization ensures that each of the services will coexist in the same environment isolated from the outside world as well as of all the files and code will be bundled within a package (Docker container). Containerization also helps with transferring, installing, and running software in different machines easier. Kubernetes deployments are defined for each component of both services and ML pipelines. These objects specify the Docker images to use, the number of replicas, resource limits, and environmental variables (including secrets for accessing protected resources).

- **Kubernetes Cluster Setup**:
  - A Kubernetes cluster is provisioned with necessary resources (e.g., CPU, GPU, memory) to run the ML workloads. This includes setting up namespaces, volumes for persistent storage, and network policies for secure communication.
- **Kubernetes Functionality:**
  - Deployments for stateless components like the forecasts API and stateful components like the PostgreSQL and ElasticSearch. Each of those components will be deployed in a cluster to allow all necessary services and components access to achieve the desired functionality. Deployments also allow for easy management and scaling in case needed as well as features built on top of Kubernetes clusters like auto-healing, self-replicating and easy backups with no down-time.
  - ConfigMaps for configuration and Secrets for sensitive data, ensuring they are not hard coded into the application. The ConfigMaps concept allow you to decouple configuration artifacts from image content to keep containerized applications portable. Many applications rely on configuration which is used during either application initialization or runtime. Most times, there is a requirement to adjust values assigned to configuration parameters.
  - Ensuring different applications are running smoothly and are healthy due to Kubernetes functions like auto-healing and autoscaling. As mentioned above Kubernetes offers some features that are built upon its existing architecture hence it allows for complex functionality to be possible with minimum effort. Auto-healing is that the cluster always try to achieve the set state of cluster, if something is diverging it will try to bring it back e.g.: a failed pod. Autoscaling is similar but it tries to serve the current load with the available resources. If the load grows or decreases the employed resources will move along.

  - **Service Exposition and Load Balancing**: Services are defined to expose the ML pipeline components internally within the cluster or externally for user access. Ingress controllers or load balancers can be configured to manage traffic to these services.
- **Persistent Storage:**
  - Persistent volumes for the PostgreSQL and ElasticSearch are in place to ensure data persistence across pod restarts. Stateful components surely need storage and persistent storage so that it is not lost during restarts or failures. That is achieved with Persistent Storage which also allows for different services and components to share storage and/or information if needed. This is a flexible way of keeping components stateful and running smoothly during deployment process.
- **Networking:**
  - Services for each component to enable network access within the Kubernetes cluster and ensuring unauthorized access is not possible into the private network. This aims to make the cluster more secure for the outside world but easy for services to talk with each other in the same cluster. It seems like a home private network that is not accessible for third parties.
  - Ingress resources for components that allow accessibility externally, like the API and Grafana. Ingress is a configuration way to specify the network traffic as described above. More specifically is an API object that helps developers expose their applications and

manage external access by providing http/s routing rules to the services within a Kubernetes cluster.

- **Authentication and Authorization:**
  - Deploy and configure network policies to control the flow of traffic between pods based on the namespace, pod, and port level. Kubernetes comes with features that allow or prevent certain traffic based on configurable criteria.
  - Keycloak for managing authentication and authorization and managing access and roles of users into the application and the different services. This part is implemented with Keycloak as FE will get the tokens from the service and BE will have to authenticate and authorize using the same service. Token expiration and user level authorization is in place to ensure only permitted users have access. Also functions like safely saving passwords, password recovery and dynamic user right management will be in place.
- **Continuous Deployment:**
  - CI/CD pipelines are responsible for automating the testing and deployment of these components. Helm charts and Kubernetes manifests will be used to define the desired state of each deployment. Updates of the tool are released in chunks that do through a well-defined test suite to ensure functionality is stable across multiple machines.
- **Monitoring and Logging:**
  - Integration of monitoring tools like Prometheus with Kubernetes metrics and logging tools to gather logs from all components for debugging and performance monitoring. Logging is also found in API level of BE to log important functionality or potential errors thought processing data or forecasting values.

These steps ensure that the components are well-integrated and deployed in a scalable, resilient, and secure manner on Kubernetes. It also takes responsibility for orchestrating and running smoothly apps with many services as this tool and offers complex functionality on top of deployed apps for better control, monitoring, and scaling of the apps. This process leverages Kubernetes' capabilities for high availability, scalability, and management of containerized applications, making it a robust platform for deploying and managing AI model training and inference workloads.

5 Pilot trials execution (feasibility analysis)

5.1 Description of piloting activities

The goal of Task 5.5 is to showcase, educate, validate, and evaluate the software tools developed within the WP5 of the SUNRISE project to ensure they meet project objectives and expectations, and to build competence among project team members for effective utilization.

In the context of Task 5.5, one of the primary risks could be an inadequate understanding or awareness of the software tools developed within the SUNRISE project among the project team members. To mitigate this risk, comprehensive training sessions will be conducted to familiarize the team with the software tools. Additionally, detailed user manuals and documentation will be provided to facilitate understanding and usage, and regular Q&A sessions will be organized during the demonstration phase to clarify any doubts or misconceptions.

A dedicated technical support team will be established to assist partners and address any technical issues or bugs promptly. Partners and end-users will be encouraged to report any technical issues or bugs encountered during the testing phase for prompt resolution. The software tools will be designed to be user-friendly, intuitive, and tailored to meet the specific needs and requirements of end-users. Adequate training and ongoing support will also be provided to end-users to facilitate the adoption and usage of the software tools.

Overall, three different roles will be created during piloting activities and will be further enhanced and analysed according to the personalized needs of the CIs. More specifically, there will be three profile layers, the strategic profile, the operational profile and the analyst profile. Each profile will be based on the different skills of CI personnel, like managerial skills, IT skills and general skills.

The Strategic profile is high-level managers responsible for managerial decisions inside the organization. Their skills will be decision-making and policy-making. Operational profile will be IT personnel with the responsibility of maintaining technical aspects of the organization. Finally, CI operators will have a more generic profile, which may include analysts and generic personnel of Critical Infrastructures.

The crystallization of the roles and profiles will be further pursued and finalized during the forthcoming piloting activities.

The first testing and validating phase included in Task T5.5 will start at Month 21 (June 2024) and end at Month 23 (August 2024) and is aimed at collecting feedback and suggestions to guide technical partners towards TRL6.

Month 21: June 2024

Week 1-2:

1) Demonstration

- Organize live demonstrations by project partners showcasing the developed software tools.
- Conduct Q&A sessions for clarifications.
- Collect initial feedback from partners.

Week 3-4:

2) Training

- Distribute user manuals and documentation to project team members.
- Conduct virtual or on-site workshops on each software tool by experts.
- Administer hands-on exercises for team members to ensure proficiency.

Month 22: July 2024

Week 1-2:

3) Validation

- Implement the software tools in pilot projects across Critical Infrastructures.
- Begin continuous monitoring and collect feedback from pilot projects.

Week 3-4:

4) Evaluation

- Define Key Performance Indicators (KPIs) and evaluation criteria.
- Systematically test the software tools against predetermined benchmarks.
- Initiate the collection of feedback from end-users and stakeholders regarding usability, functionality, and performance.

Month 23: August 2024

Week 1-2:

3) Validation

- Continue monitoring pilot projects and collecting feedback.
- Evaluate the software tools against real-world scenarios (real-life data, as opposed to the synthetic data previously used) and the use cases to confirm alignment with project objectives (WP3).

Week 3-4:

4) Evaluation

- Finalize the collection of feedback from end-users and stakeholders.
- Analyze collected data to assess the effectiveness of the software tools.
- Prepare and submit a comprehensive report on the evaluation results, including recommendations for improvements.

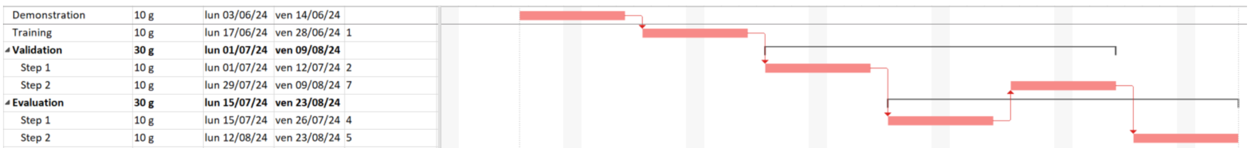


Figure 65: T5.5 Activities Gantt chart

Throughout the task, technical partners and national clusters from Italy, Spain, and Slovenia will provide support to testing partners. Their collaboration will ensure the successful implementation, validation, and evaluation of the software tools across Europe's Critical Infrastructures. In particular collaboration will improve the stakeholders' engagement and minimize resistance to change or adoption by end-users.

This schedule ensures a structured and comprehensive approach to Task 5.5, enabling partners to effectively demonstrate, train, validate, and evaluate the software tools within the allocated period.

By proactively identifying potential risks and implementing appropriate mitigation actions, the project team can minimize disruptions, ensure the successful execution of Task 5.5, and achieve the desired outcomes within the allocated timeframe.

5.1.1 Energy

The main purpose of the EKC and ELES pilots is to evaluate our methods against current state-of-the-art models used in production settings. This benchmarking process allows us to assess how our models perform in comparison to the established leaders in the field. By achieving performance that matches or even surpasses the current state-of-the-art, we gain strong evidence that our models are on the right track to be resilient enough to handle large-scale events such as pandemics.

EKC is a Serbian consultancy and engineering company in the field of electricity. ELES are transmission system operators (TSO), where their key role is to transport energy and keep the grid stable via demand and productions balance. Excess energy production leads to waste, while insufficient production risks grid instability as the frequency could drop below 50 Hz. TSOs (Transmission System Operators) use various methods, from frequency reserves to load shedding, to manage this balance. Therefore, accurate predictions of load demand are crucial for TSOs to optimize grid stability, minimize energy losses, and increase overall energy efficiency.

The demand prediction tool primarily serves the needs of Analysts, a general user profile, as defined in [25] (Response to Reviewers comments related to the topics of end-users, data governance and workshops organization). Its results also indirectly impact decision-making by High-level Managers (strategic user profile) and IT personnel (operational user profile). The Analyst uses the tool's findings to generate reports for the Manager, who then determines the necessary actions to be implemented by IT personnel to ensure grid stability.

XLAB owns, maintains, and deploys the demand prediction tool. However, data that is used by the tool is publicly available and owned by the TSOs. They are required to share their data to public domain via ENTSO transparency network [26]. Data that we are operating with is in public domain which enables us to perform public and cloud deployment of energy demand prediction tool. To demonstrate the feasibility of our enhanced TFT (Temporal Fusion Transformer) modelling approach, we conducted a series of experiments comparing our model's performance against the EKC model currently used in production as described in Section 3.

5.1.2 Transport

As described in[2], phase 1 of the pilots in the transportation sector aims to demonstrate the potential of the prediction models through the DPM tool and to show its functionalities. In this phase the SUNRISE CI partners of the transport domain, RTM (Madrid, Spain) and TT (Trieste, Italy), will participate evaluating the DPM tool. The tool will be customized to the specific needs of each CI and fed with their corresponding historical data and models. The purpose of these pilots is to display the advances of the WP to the CI partners and to gather feedback from them to enhance the tool in the next iteration.

Each CI will have access to the DPM tool, enabling visualization of the model predictions and interaction with the tool, such as creating new predictions or providing new data. The tool's visualization will be adapted to different user profiles, which are defined by the CI partners based on their specific roles and requirements. These profiles may include strategic, operational, and technical roles, each with custom functionalities and access privileges to optimize the tool's usability and effectiveness.

The ownership of the data used in the tool is retained by the CI partners. The models used within the tool are developed and owned by UPM, while the user interface is owned by INT. The data will be stored in a dedicated database, hosted by SQD. Access to the tool will be secured by token-based authentication service also hosted by SQD.

The RTM pilot involves Line 12 of the Madrid Metro, a circular line connecting the city centre with the south of the city. For the TT pilot, the models used in the DPM tool will be trained with data recently provided by the CI, following the same steps described in Section 3.2.1. Although the performance evaluation of these models is not presented in this document due to incomplete historical data at the time of writing, the TT pilot will proceed alongside the RTM pilot. The scenario for TT will involve the bus network of the city of Trieste.

The results for both partners will be presented together in the final report of the pilot phase 1.

5.1.3 Health

The primary objective of the Phase 1 piloting activities of the health use cases is for the CIs of health domain to test and benchmark the developed models and to demonstrate the basic functionalities of the Demand Prediction and Management tool. Health CI providers will test and evaluate the effectiveness of the models regarding the prediction of different values regarding their requirements set in WP3. General scope of the first piloting activities will be CIs to test and validate the upgraded tool components and check if DPM tool meets their initial requirements. Additionally, Critical Infrastructures will check and assess the influence of the exogenous variables on the prediction accuracy of the tool.

For the health sector, the first piloting activities will be applied in two different Critical Infrastructures, QS/HQM in Spain and INS in Italy. QS is one of the biggest health providers in Spain and is interested in the prediction of the quantity of consumables & drugs. On the other side, INS the in-house company of Region FVG devoted to ICT and digital transition, is interested in the prediction of number of X-rays, number of medical reports (REFE), and number of hospital discharge letters (LDO). Through the phase 1 piloting activities, respective CIs will confirm that our developed models are advancing according to their needs and will support the demand prediction of their critical resources as have been discussed and mentioned in D5.1 [2].

For the first pilot activities (M20 - M23), SQD has already started the deployments of the different services of the DPM tool on cloud, ensuring access security through token-based authentication (username/password). SQD owns the DPM tool for health piloting activities. Historical datasets, owned by the respective CIs (QS/HQM, INS), will be stored in a dedicated PostgreSQL database for efficient management and analysis, with raw historical files stored in MinIO.

To receive preliminary feedback and possible suggestions, different developed models, prediction results and metrics have been shared among health use case CIs and online meetings took place. These actions will allow development partners to be more well-prepared for the piloting activities and reduce possible drawbacks.

A series of online meetings were conducted with INS to address several challenges and precisely determine the values they wish to predict based on their specific needs. Subsequent to these meetings, diverse insights were discussed and the required visualization in the user interface of the tool was documented. In the same vein, dedicated meetings with QS/HQM meetings were also conducted to present the developed models and the metrics. Various insights were discussed to determine the high value of the prediction of drugs and consumables linked with different pandemics, such as COVID-19 as well as other pathologies like seasonal influenza.

5.1.4 Water

The main purpose of the Phase 1 piloting activities of the water use case is for the respective CIs to test and benchmark the developed models and to demonstrate the basic functionalities of the Demand Prediction and Management tool. As described in section 3.4, various state-of-the-art models have been developed and fine-tuned further. To this end, the upgraded tool components will be tested and validated in relevant environments, allowing for the assessment and further enhancement of user requirements and provided solutions. Water CI providers will test and evaluate the effectiveness of the models regarding the prediction of volume of water consumption. Moreover, they will assess the influence of exogenous variables like the stringency index on the prediction accuracy of the tool. Phase 1 piloting activities will validate that our developed solutions are progressing as planned to address future fluctuations in terms of demand prediction.

Phase 1 piloting activities include two different Critical Infrastructures in the water domain, ACO from Spain and CAF from Italy. More specifically, ACO is the public company belonging to the Western Costa del Sol Town Council Association (Benahavis, Benalmádena, Casares, Estepona, Fuengirola, Istán, Manilva, Marbella, Mijas, Ojén and Torremolinos). Moreover, ACO manages the whole water cycle, within its geographic remit, as a vital and essential service for the area's citizens, working to ensure provision of both water supply and sanitation [27]. After the initiation of piloting activities, ACO will test and navigate through a UI to the insights and the optimized via line graphs of historical and long-term forecasted data for seven different places (Presa, Etap, Desaladora, River Fuengirola, River Guadalmansa, SI4, Gibraltar) based on the monthly time frame datasets that have been provided.

As for the second Critical Infrastructure, CAF, is the water provider and the organization that lead the management of water service of 121 Municipalities in the province of Udine, Italy, since 1931. CAF also distribute the water to most of the municipalities located in the central area of Region Friuli Venezia Giulia. After the initiation of piloting activities, CAF will test and navigate through a UI to the insights and the optimized via line graphs of historical and short, mid and long-term forecasted data for four different places (Intermezzo Biauzzo-Crosere-Lignano, Intermezzo Biauzzo-Lignano (Via Rivignano), Latisana, Lignano) based on the hourly time frame datasets that have been provided. According to the needs that have been identified until now, three different levels of profiles will be used, Strategic, Operational and General [25].

For the first piloting activities (M20 - M23), the deployment of the DPM tool has already started by SQD in cloud with token-based authentication (username/password) for UI access, securing that way access. The DPM tool for the water piloting activities is owned by SQD. Historical datasets are stored in a dedicated database for efficient management and analysis (PostgreSQL) while the physical historical files are stored in MinIO for scalability. Historical dataset owners are the respective CIs (ACO, CAF).

The different models, prediction results and metrics have been shared among water use case Critical Infrastructures (ACO and CAF), as an initial step before the starting of phase 1 piloting activities, in order to receive preliminary feedback and possible suggestions. For ACO, various prediction models and their predicted values in a long-term timeframe have been presented. ACO has indicated that long-term prediction models regarding the volume consumption of water are highly valuable for their managerial processes. The availability of the prediction results in the long term will aid them in decision-making and sustainable direction processes. On the other hand, CAF also received the developed models and metrics. Specifically, based on the models initially developed for CAF and compared using the Mean Absolute Error (MAE) metric, it can be deduced that project-developed models have further enhanced their demand prediction. Particularly in the Lignano district, both SQD and CAF conducted experiments that found a strong positive correlation between water demand and temperature, and a strong negative correlation with the relative humidity. Furthermore, both SQD and CAF found that the LSTM model performs well in the test set.

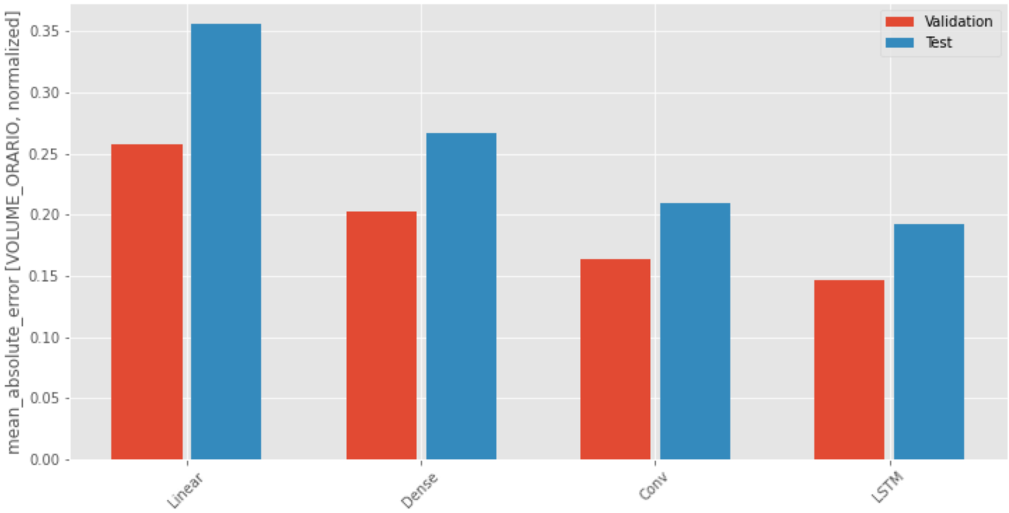


Figure 66: CAF models: MAE Metric Comparison Between Multiple Prediction Models on Validation and Test Sets.

5.2 Description of End-User's Roles

In the earlier WP5 deliverables, D5.1 [2] and D5.2 [3], the term 'end-user' referred to the users of the WP5 Tool. However, it was correctly stated by the project reviewers that this term encompasses various roles within the critical operator context.

Concerning the WP5 tool (DPM), the term 'end-user' specifically denotes the ultimate operators of the tool, comprising personnel from various Critical Infrastructures (CI). More precisely, within WP5, eleven CIs (MZI, TS, ELS, QS, HQM, RTM, ACO, FVG, TT, CAF, EKC) represent different sectors (energy, water, health, transportation, and digital services) and will actively engage in pilots to showcase collaboration within and across sectors. Table 30 outlines the diverse roles involved in the DMP tool, based on the identified needs thus far, and their associated profile and skills.

Table 30: End User's Roles in WP5 Tool

User organization	End user profile	User Role	Skills
CI operators	Strategic	High Level manager	Managerial & policy making skills (decision making, policy knowledge)
CI operators	Operational	IT personnel	Technical skills
CI operators	General	Analyst	Generic, other (any pilot participant)

6 Conclusions

This deliverable offers a comprehensive overview of the initial integrated version of the Demand Prediction and Management Tool (Y2) within the SUNRISE project. It traces the evolution from its Year 1 iteration, emphasizing enhancements in accuracy, efficiency, and feature set. With Y2 now operational and addressing all seven anticipated use cases, it marks a significant advancement toward SUNRISE project objectives. The detailed account of integration activities and technical intricacies underscores a robust framework tailored to meet the diverse needs of Critical Infrastructure operators. These developments signal continued progress towards further enhancing the overall solution.

While piloting activities are scheduled to commence in June, the groundwork laid in this document provides a solid foundation for their success. These activities are poised to yield substantial feedback on demand prediction portrayal from each Critical Infrastructure sector, thereby contributing to ongoing refinement efforts. Moreover, the piloting activities under WP5 demonstrate progress in aligning with user requirements and establishing essential terminology crucial for the tool's effective utilization. This document not only serves as a valuable resource for end-users, offering insights into the functionality and capabilities of the Y2 Tool across various sectors, but also stands as a testament to the collaborative efforts driving innovation within the SUNRISE project.

Finally, the results of this document will be used as input in deliverables D3.4 Reference designs and best practices (September 2024) as well as pave the way for D5.5 Demand prediction and management tool and training guide V3.

References

[1] **SUNRISE**, “**D3.2 Requirements and designs V2**”, 2023.

[2] **SUNRISE**, “**D5.1 Demand Prediction and Management**”, 2023.

[3] **SUNRISE**, “**D5.2 Demand Prediction and Management Tool and training guide**”, 2023.

[4] **SUNRISE**, “**D3.1 - Requirements and designs V1**”, George Tsakirakis. 2023.

[5] “**XGBoost**,” [Online]. Available: https://en.wikipedia.org/wiki/XGBoost.

[6] “**Neural network** (machine learning),” [Online]. Available: https://en.wikipedia.org/wiki/Neural_network_(machine_learning).

[7] “**keycloak**,” [Online]. Available: https://www.keycloak.org/.

[8] “**min.io,”** [Online]. Available: https://min.io/.

[9] “**Grafana**,” [Online]. Available: https://grafana.com.

[10] “**Elasticsearch**,” [Online]. Available: https://www.elastic.co/es/elasticsearch. [Accessed 12 04 2024].

[11] “**Mlflow**,” [Online]. Available: https://mlflow.org/.

[12] “**MLFlow**,” [Online]. Available: https://mlflow.org/. [Accessed 10 04 2024].

[13] “**FastAPI**,” [Online]. Available: https://fastapi.tiangolo.com/. [Accessed 10 04 2024].

[14] “**Docker**,” [Online]. Available: https://www.docker.com/. [Accessed 13 04 2024].

[15] **L. S. S. L. X. L. K. Z. J. L. L. Zou**, “Passenger Flow Prediction Using Smart Card Data from Connected Bus System Based on Interpretable XGBoost,” Wireless Communications and Mobile Computing, vol. 2022, 2022.

[16] **M. G. G. Yao**, “Short-term passenger flow prediction of subway based on XGBoost multi-feature fusion,” in Proceedings of the 2023 6th International Conference on Big Data Technologies, New York, NY, USA, 2023.

[17] **X. Z. C. M. C. Sun**, “Urban Rail Transit Passenger Flow Forecasting—XGBoost,” in 22nd COTA International Conference of Transportation Professionals, Changsha, Hunan Province, China, 2022.

[18] **C. &. B. J. M. Bergmeir**, “On the use of cross-validation for time series predictor evaluation,” Information Sciences, vol. 191, pp. 192-213, 2012.

[19] **G. a. C. M. A. a. L. M. de Souza Groppo**, “Predicting water demand: a review of the methods employed and future possibilities,” Water Supply, vol. 19, no. 8, pp. 2179-2198, 2019.

[20] **M. A. A. A. O. A. W. Manar Abu Talib**, “Influence of exogenous factors on water demand forecasting models during the COVID-19 period,” Elsevier, Engineering Applications of Artificial Intelligence, Volume 117, Part A, 105617, January 2023.

[21] “**Prophet Forecasting**,” [Online]. Available: https://facebook.github.io/prophet/docs/seasonality,_holiday_effects,_and_regressors.html#fourier-order-for-seasonalities.

[22] “**Time Series forecasting with XGBoost,**” [Online]. Available: https://www.kaggle.com/code/robikscube/tutorial-time-series-forecasting-with-xgboost.

[23] [**Online**]. Available: https://en.wikipedia.org/wiki/Coefficient_of_determination.

[24] “**Mean absolute percentage error**,” [Online]. Available: https://en.wikipedia.org/wiki/Mean_absolute_percentage_error.

[25] A. P. (. Antonio Álvarez, [SUNRISE project] **Response to Reviewers comments related to the topics of end-users, data governance and workshops organization**, European Commision, 2024.

[26] **ENTSO-E, “ENTSO-E Transparency Platform,”** ENTSO-E, 2023. [Online]. Available: https://transparency.entsoe.eu/dashboard/show. [Accessed 20 12 2023].

[27] [**Online**]. Available: https://www.acosol.es/.

Annex I: Training Guide and User Manual V2

A.1. Introduction

The utilization of the Demand Prediction and Management (DPM) tool has been targeted for various sectors, including water, energy, health, transportation, and digital domains, as previously outlined. This tool is poised to integrate cutting-edge technologies, leveraging AI/ML and graph-based models where appropriate, to generate precise forecasts and actionable insights tailored to the specific Critical Infrastructures involved.

Customizable and parameterized forecasts and insights serve to aid both immediate and future projections, catering to short-term and long-term forecasting needs. These forecasts pertain directly to the pivotal resources within each of the associated Critical Infrastructures. To generate the visualized outputs, historical data from each CI is harnessed, supplemented by the integration of specific events where relevant, such as COVID-19, lockdowns, and climate variations.

The initial version of the DPM tool that was documented in D5.2 [3] on M12 as a Proof of Concept (POC), has been updated to the fully integrated solution with the completely updated graphical interface. This manual showcases the core functionalities and flows for the updated Y2 version.

.

A.2. Getting Started

Although the main user interface is common throughout all use cases some of the features are further tailored to the specific CIs use cases and caters to the specific needs of each use case. Similarly, the options and available assets and selections differ based on each use case.

Moreover, the users are provided with the ability to switch among light and dark theme mode to enhance readability with the use of the relevant switch on the top right corner of the screen, as shown in images.

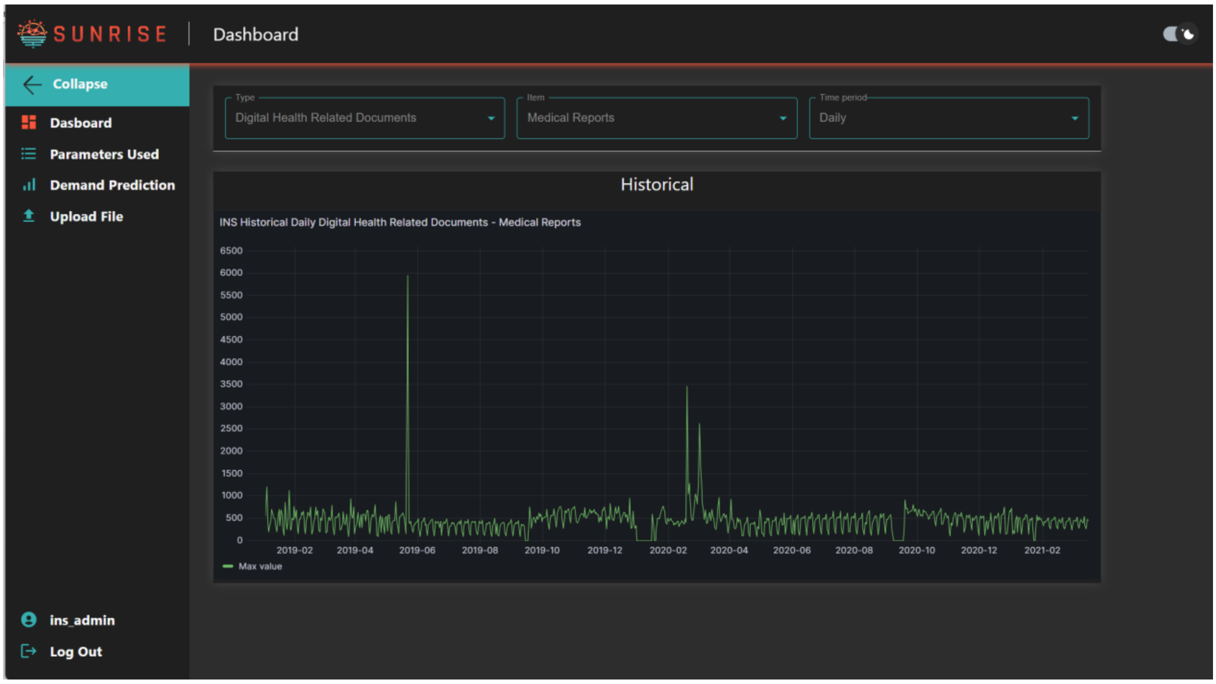


Figure 67: Dark Theme Mode

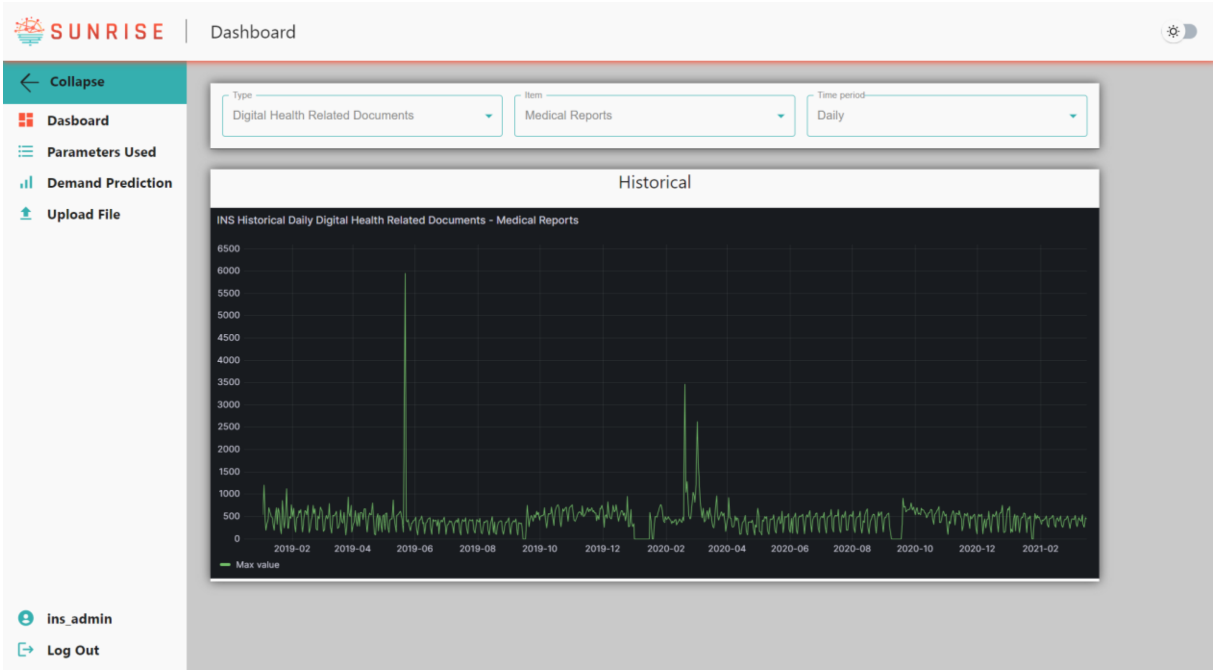


Figure 68: Light Theme Mode

A.3. Basic Features in Sidebar

In this section, all the basic features located at the sidebar are explained.

Users have the option to collapse or expand the sidebar to enhance their viewing experience as shown in Figure 69 (left & right).

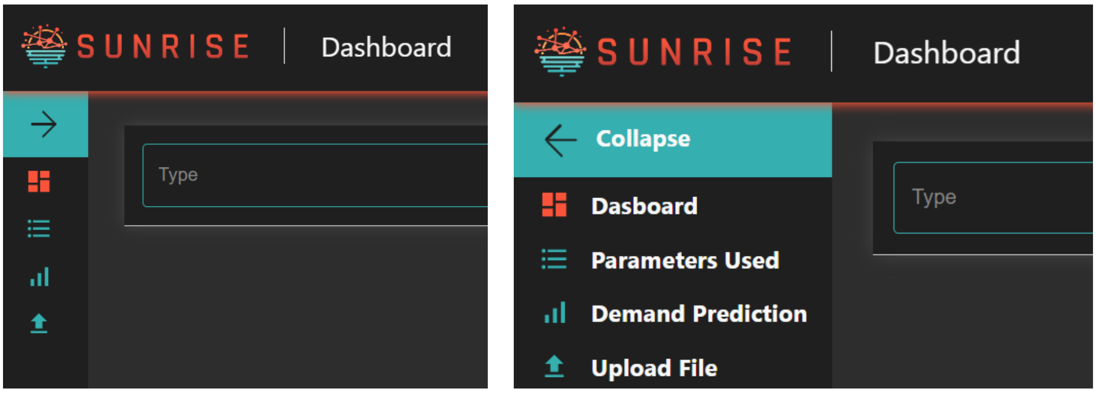


Figure 69: Sidebar Menu

Dashboard

This is the landing view of logged on users and provides them access to the analysis of historical data

Parameters Used

This view was envisioned to portray the parameters used for the ML model training. Based on the CI's feedback so far this may not be of actual value to the CI users and will probably undergo changes based on the findings of the piloting activities

Demand Prediction

This is the landing view of a logged on user and provides the user access to analysis of historical data

Upload file

In several of the use cases the retrieval of additional historical data is performed seamlessly in the back-end without requiring user actions. However, in some use cases the end users themselves upload new historical data files via this section.

A.4. Dashboard - Historical Data

Depending on the provided historical data of each use case the user is provided with a combination of selections.

In the use case of INS these selections, as shown in Figure 70, include:

a) The type of assets and subsequent items
- Digital Health Related Documents
    - X-Rays
    - Medical Reports
    - Discharge letters
- Tickets
    - Generic Support
    - Software Bug
    - Account Management
    - Workstation and peripheral support

b) The available historical time period (i.e., Daily, Weekly, Monthly) based on the existence of such data.

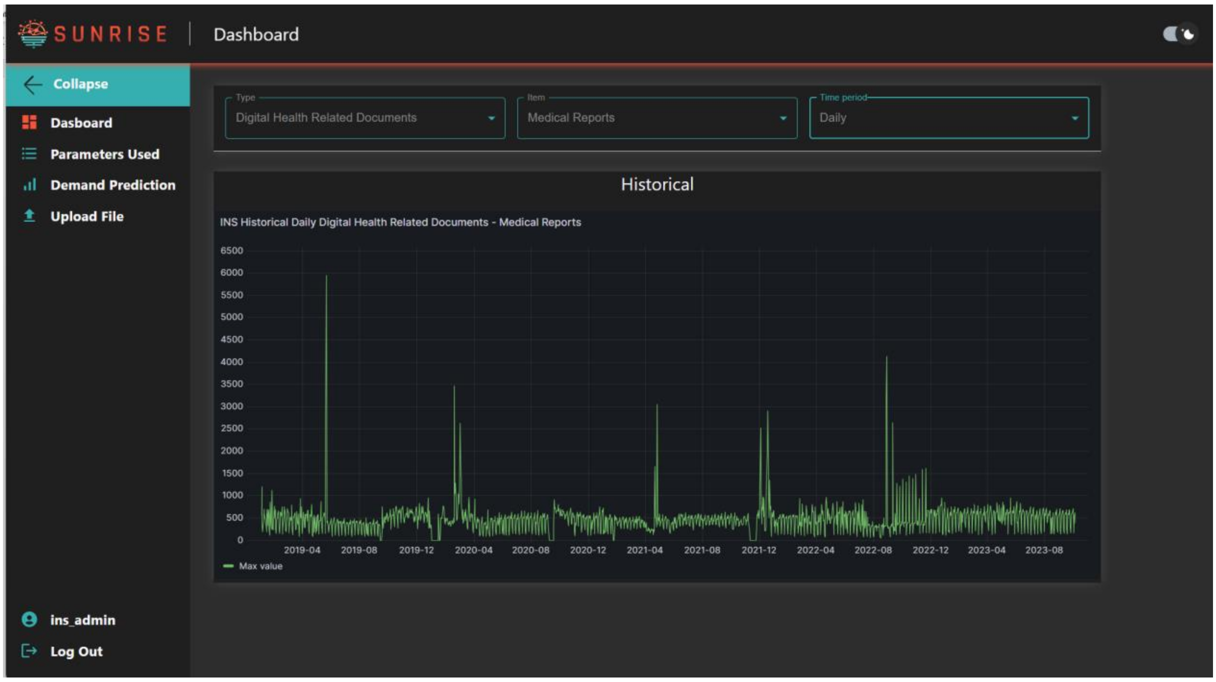


Figure 70: Selection of Dashboard parameters - INS

As an additional example of the variable available selections, as shown in Figure 71, the available option for the ACO water use case are the various water sources, desalination plants etc.

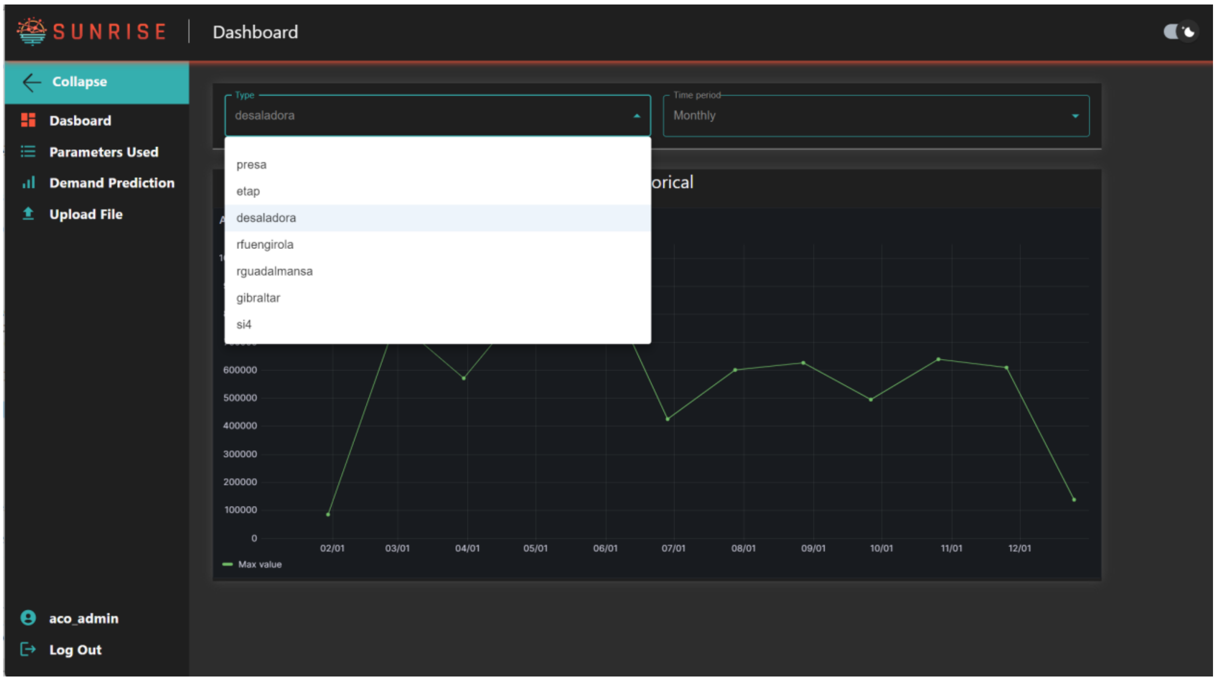


Figure 71: Selection of Dashboard parameters - ACO

The user is able to explore and retrieve data information in the graphs by hovering the mouse on specific graph points.

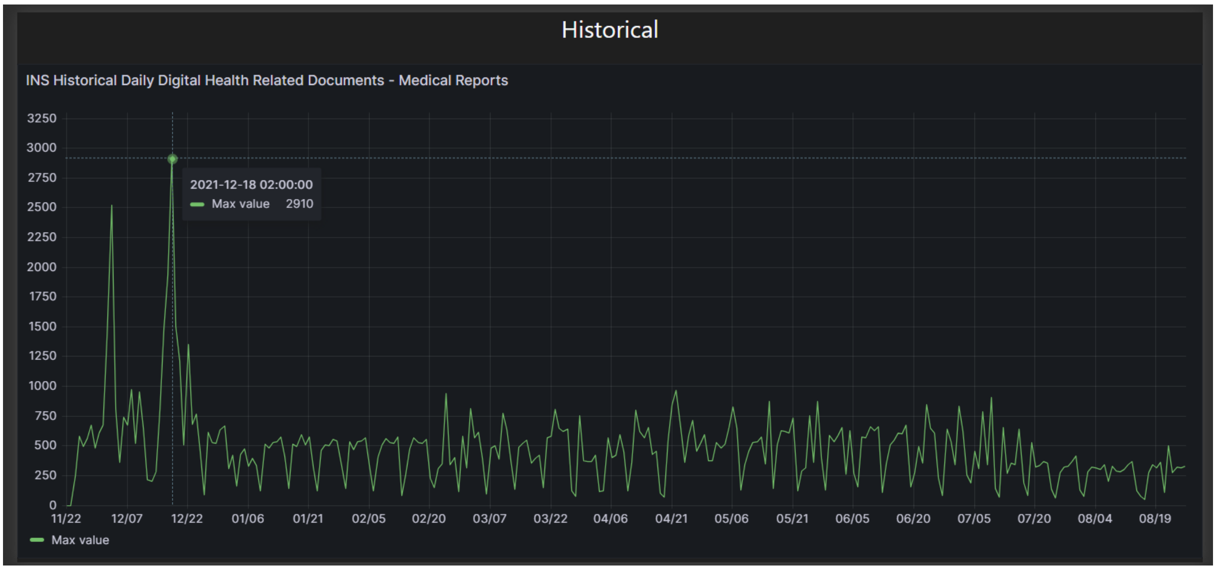


Figure 72: Additional information on graph points

Moreover, the user can pan out and focus on a smaller time frame by selecting the appropriate time period with a "click and drag" action (Figure 73 and Figure 74).

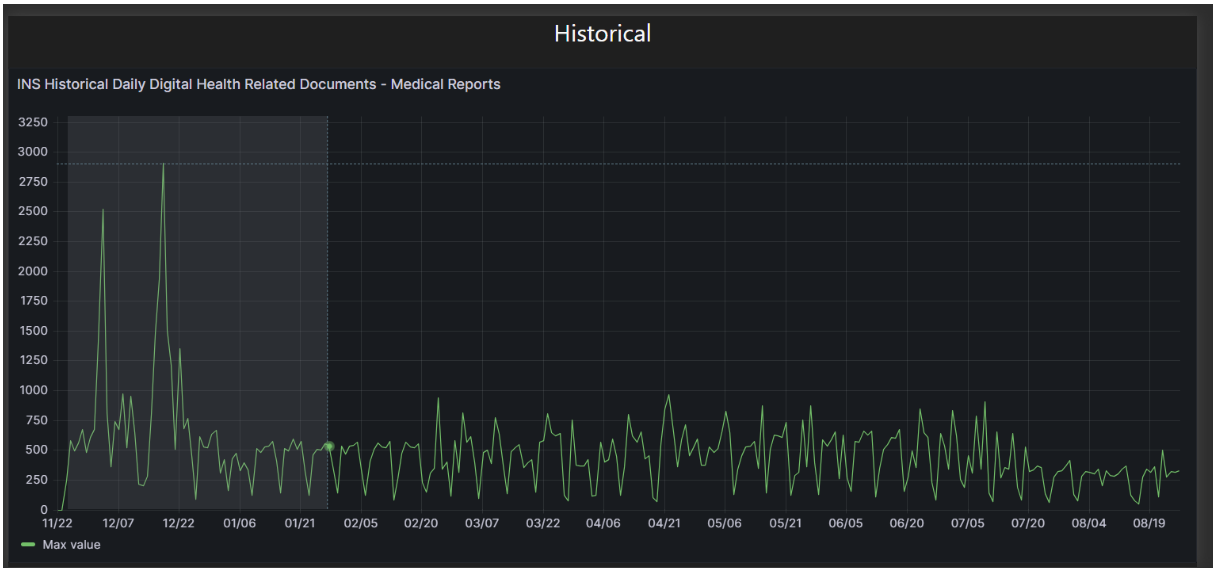


Figure 73: Focusing on specific time frame

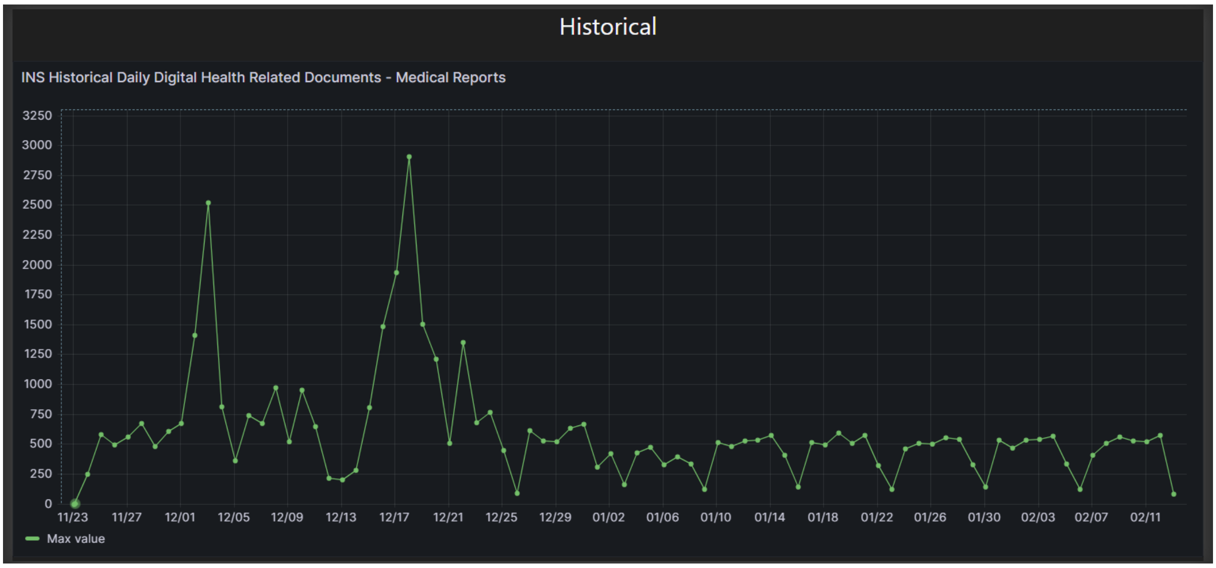


Figure 74: Output of time frame focus

A.5. Upload files

In this section the end user is able to select and upload new data to the back end (Figure 75).

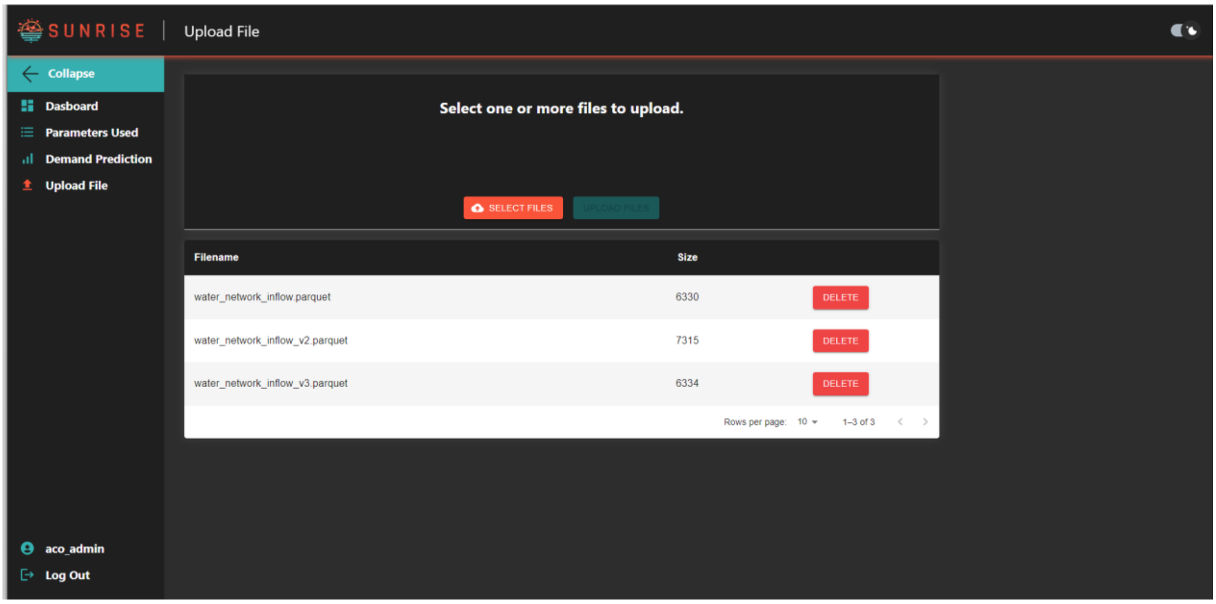


Figure 75: Upload files menu

The user can alter their selection and add or delete files accordingly after being requested confirmation (Figure 76).

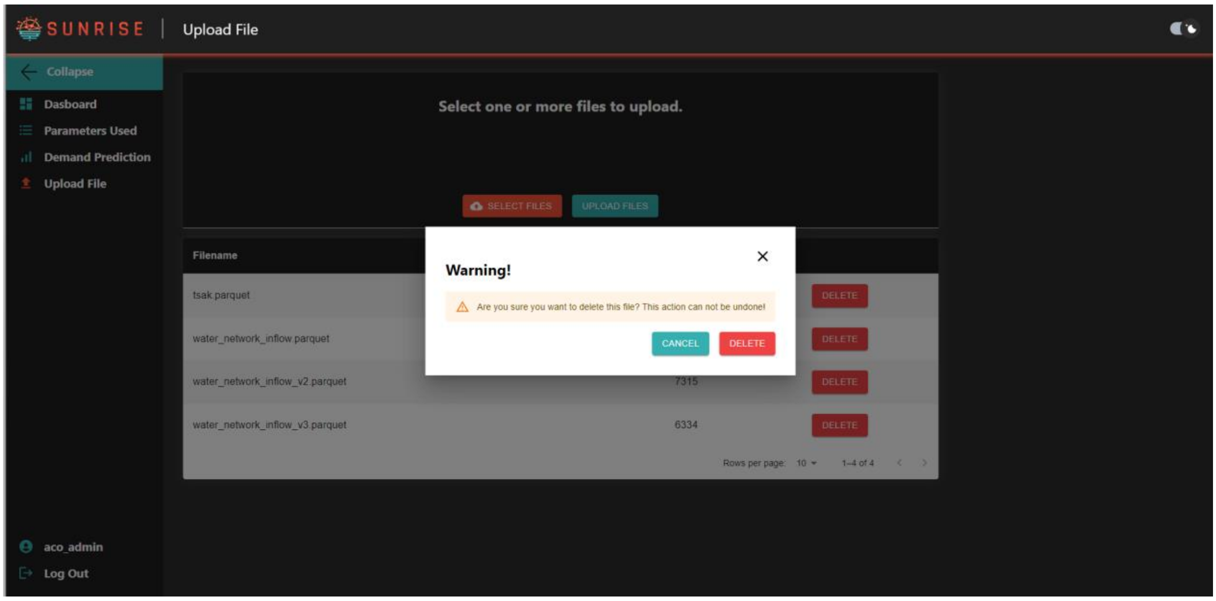


Figure 76: File Deletion

After the successful upload of the files, they are presented with the back-end response, that contains additional information about the newly uploaded data (Figure 77).

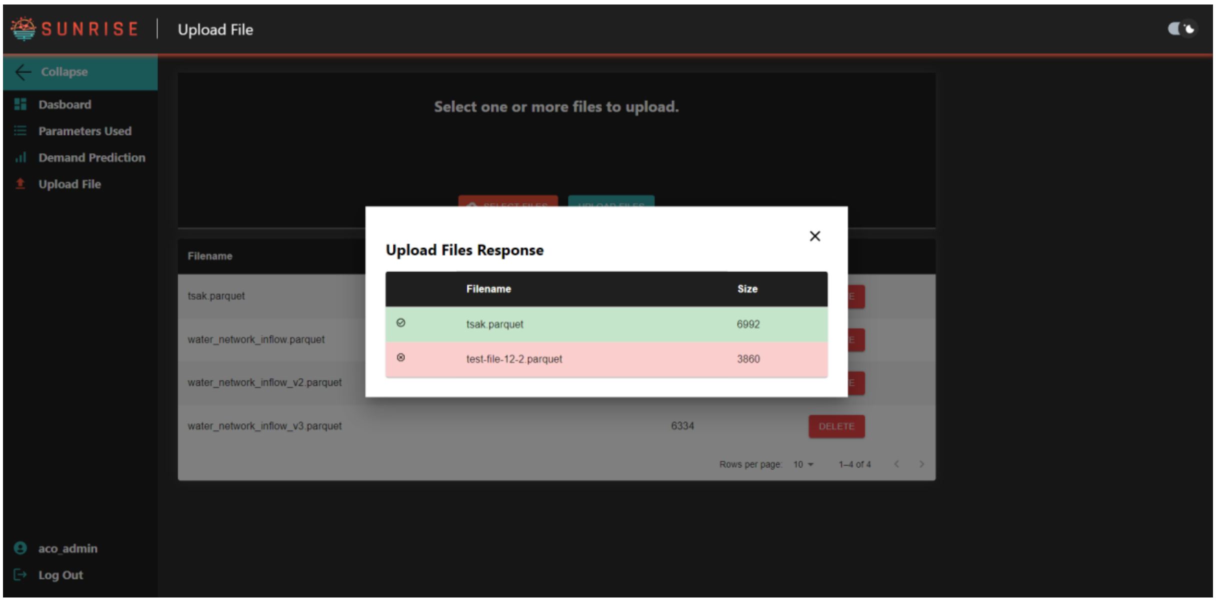


Figure 77: Response Information on uploaded files

A.6. Demand Prediction

In this section the user can perform the prediction after selecting among the available options

The common options between all use cases, as portrayed in Figure 78, are:

COVID-19 Checkbox

This selection dictates if a prediction should be performed for a COVID-19 time period (where applicable based on the available model)

Starting Date

The user can select the desired starting date of the prediction by using the popout calendar (Figure 79).

Prediction Time Frame

The user can select the time frame of the prediction to be performed by moving the available slider (Figure 80).

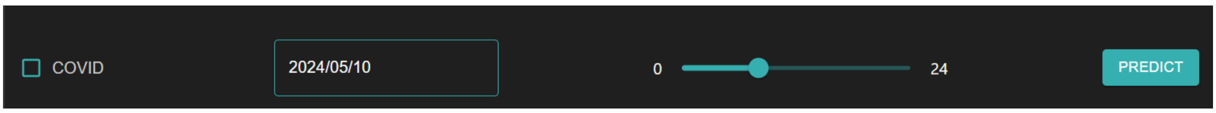


Figure 78: Common options in Demand Prediction among all use cases

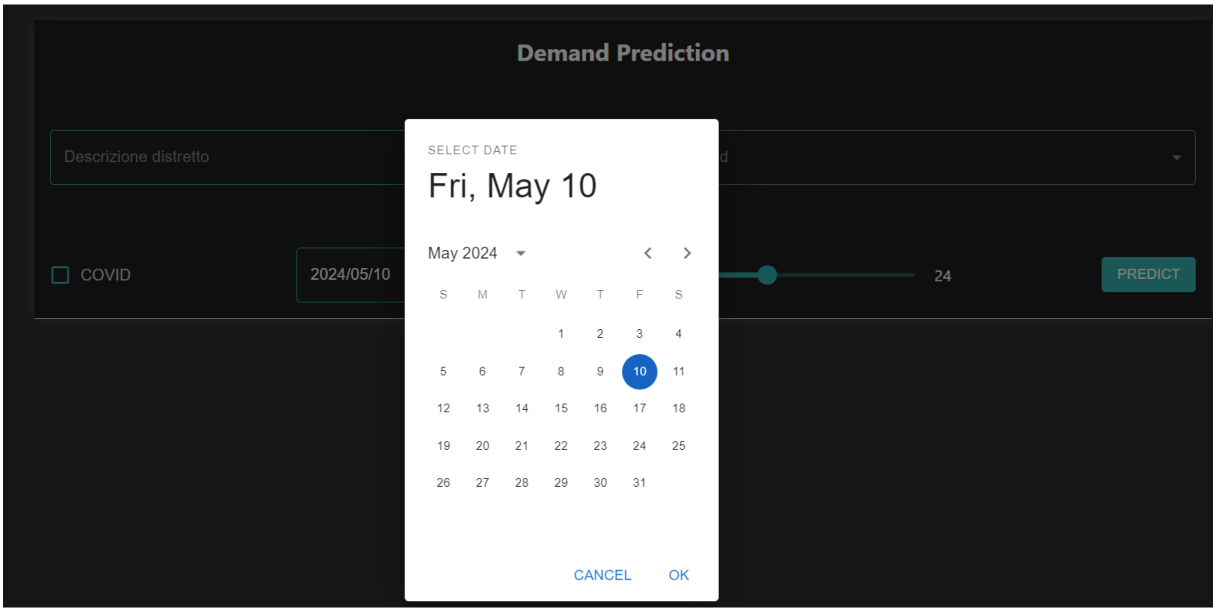


Figure 79: Starting date selection

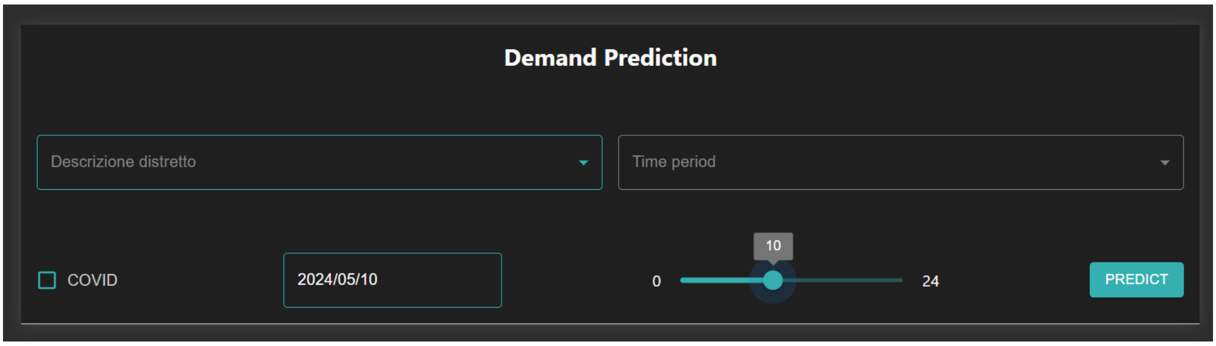


Figure 80: Prediction Time Frame selection

The remaining available selections in the Demand Prediction section are tailored to the different use cases and their specific assets and categorization.

Examples of these differences are shown in Figure 81, Figure 82, Figure 83.

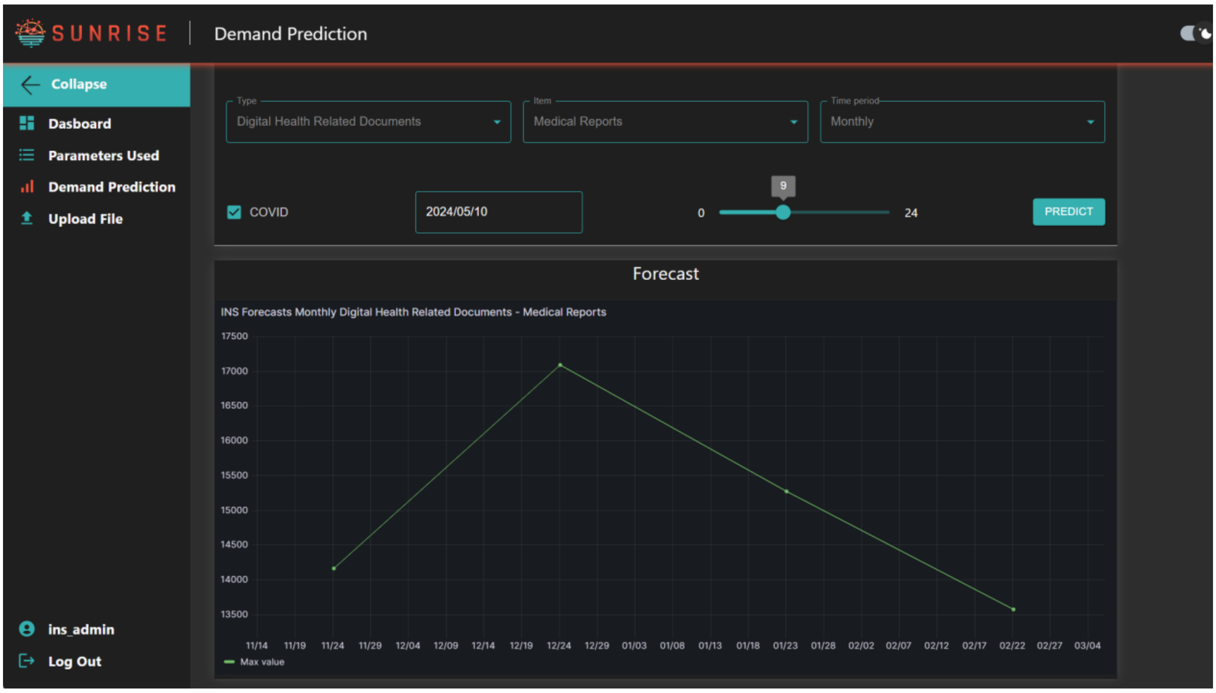


Figure 81: Demand Prediction - INS

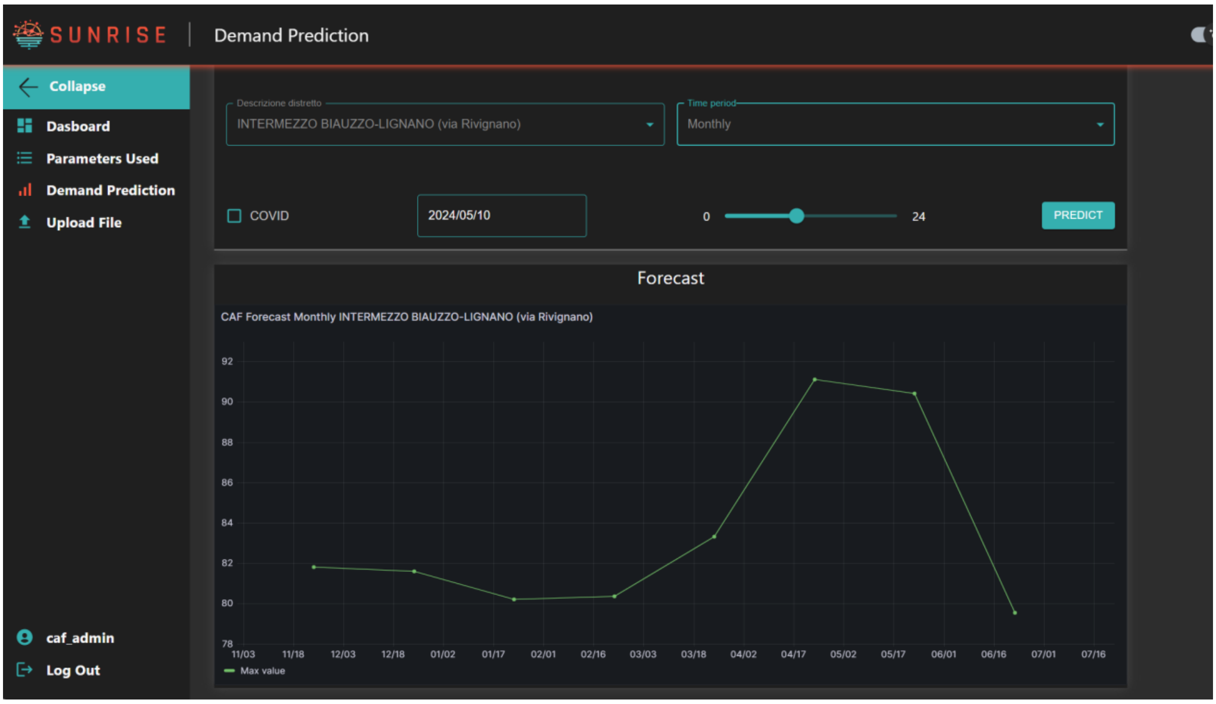


Figure 82: Demand Prediction - CAF

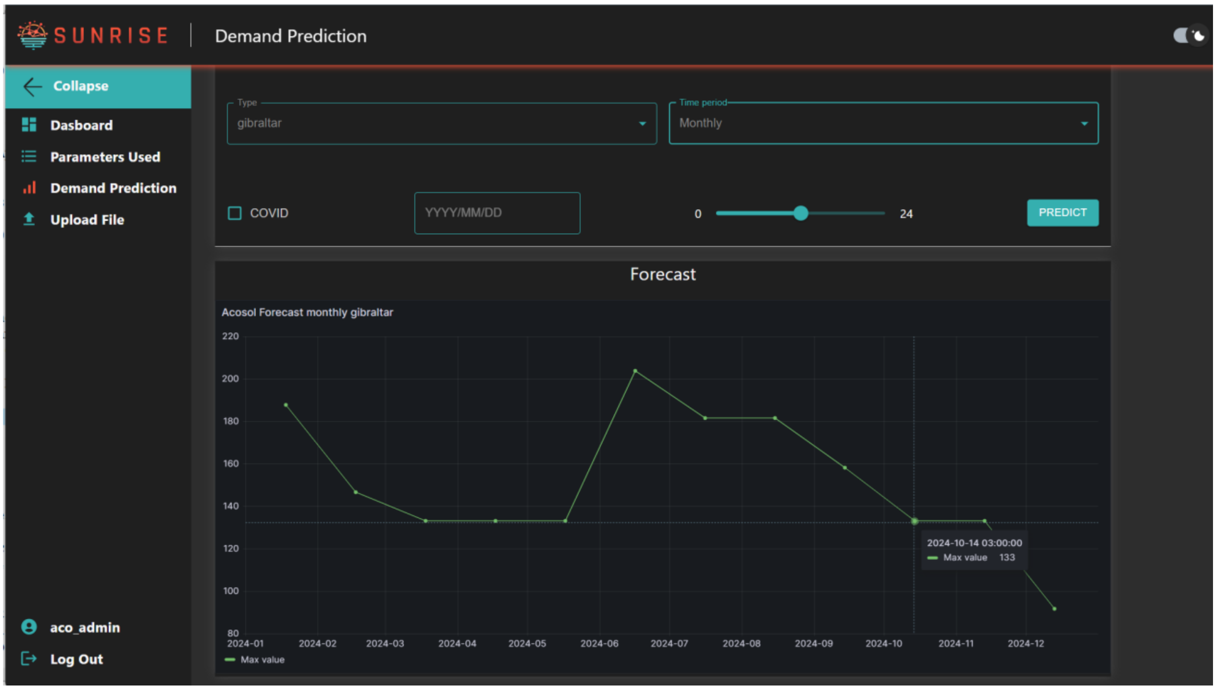


Figure 83: Demand Prediction - ACO

Similarly to the functionality in the dashboard - historical data section the user has the ability to magnify and focus on specific time frames by selecting the appropriate time period with a "click and drag" action (Figure 84, Figure 85), as well as retrieve further information on graph points (Figure 86).

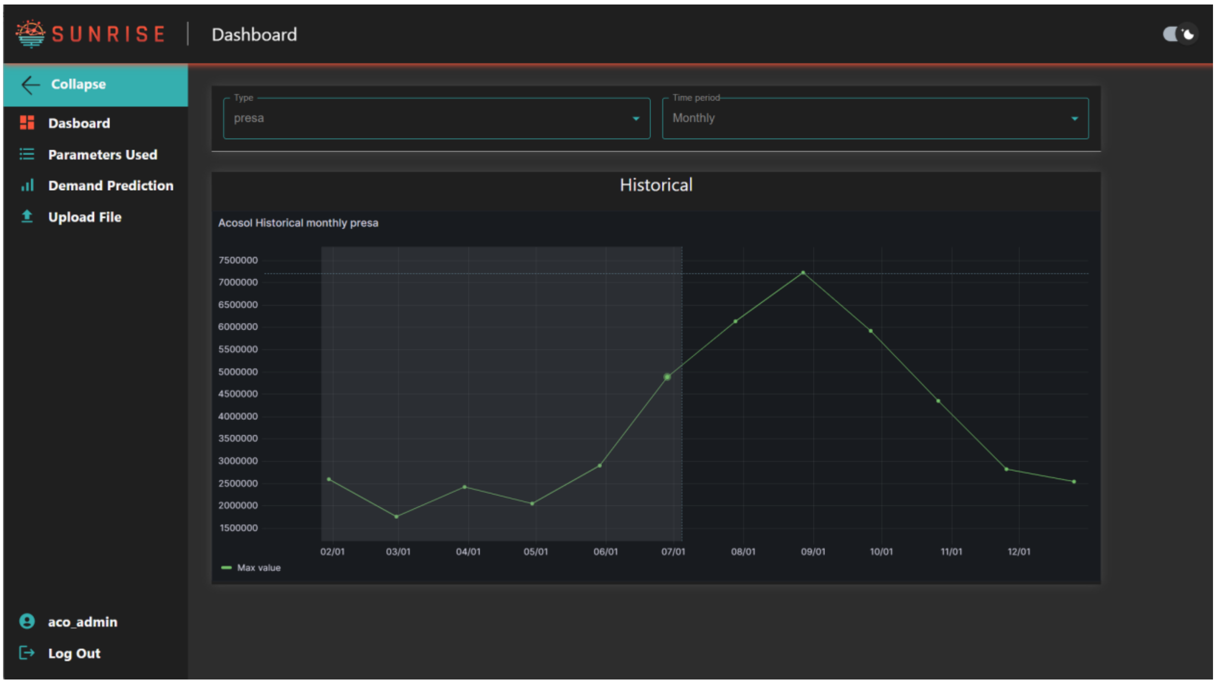


Figure 84: Demand Prediction - Time Frame selection A

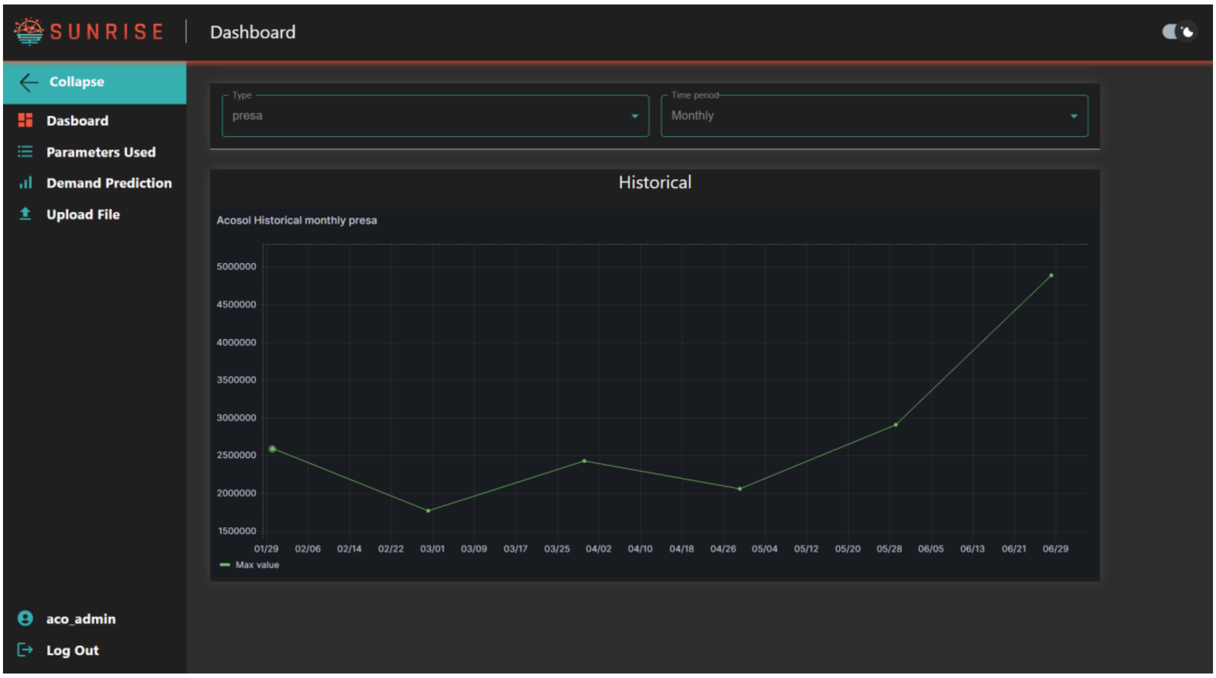


Figure 85: Demand Prediction - Time Frame selection B

Figure 86: Demand Prediction - Additional information on graph points