ENHAnCE

Featuring Engineering

D5.3. Analytics O&M Platform (HAPMS) and results of O&M simulations

Project Acronym: ENHAnCE	
Project full title: European training Network in intelligent prognostics and Health mAnagement in Composite structurEs	
Call: H2020-MSCA-ITN-2019	
Grant Number: 859957	
Deliverable nº: D5.3	
Deliverable Nature: Demonstrator	
Dissemination Level: Public	
Authored: Ali Saleh, Manuel Chiachio & Everyware Technologies	**Date:** 10/06/2024
Internally Revised: Manuel Chiachío, María Megía	**Date:** 25/06/2024
Internally Approved: Coordinator	**Date:** 25/06/2024

Document index

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 859957.

1. Introduction

The object of this deliverable is the development of the demonstrator *Holistic Assessment Platform of Management of Structure*s (HAPMS), a pivotal component within the Horizon 2020 European project ENHAnCE, belonging to the Innovative Training Networks (ITN) of the Marie Skłodowska-Curie Actions (MSCA). HAPMS serves as a comprehensive decision support system for the structural health management of composite structures, integrating decision-making processes. This deliverable has been focused on the maintenance of wind turbine blade composite structure, which is the structural type where the maintenance is very critical, and also because it has been extensively studied in the ENHAnCE project through some of the ESRs.

HAPMS is designed to facilitate a decision-making framework through an expert system which gives the optimal range of maintenance strategies for a given state of the turbine blade and considers the temporal aspect of the maintenance and the degradation. In other words, the systems consider that maintenance actions are not typically applied just when a degradation or failure occurs, but after a certain time has passed for preparation of maintenance (resource allocation, travel to site, availability of technicians and spares, etc.). During that time, the blade might continue the degradation process, thus a decision on when to maintain is key based on the state of the structural component.

The development of HAPMS has incorporated the findings of some research work previously published openly by the researchers of the project, and can be found in the following list of references:

- Saleh, A., Chiachio, M., & Chiachio, J. (2022, October). Optimized Petri Net Model for Condition-Based Maintenance of a Turbine Blade. In *World Congress on Engineering Asset Management* (pp. 657-664). Cham: Springer International Publishing.
- Lopez, J. C., Kolios, A., Wang, L., Chiachio, M., & Dimitrov, N. (2024). Reliability-based leading edge erosion maintenance strategy selection framework. *Applied Energy*, *358*, 122612.
- Saleh, A., Remenyte-Prescott, R., Prescott, D., & Chiachío, M. (2024). Intelligent and adaptive asset management model for railway sections using the iPN method. *Reliability Engineering & System Safety*, *241*, 109687.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 859957.

- Saleh, A., Chiachío, M., Salas, J. F., & Kolios, A. (2023). Self-adaptive optimized maintenance of offshore wind turbines by intelligent Petri nets. *Reliability Engineering & System Safety*, *231*, 109013.
- Wu, W., Prescott, D., Remenyte-Prescott, R., Saleh, A., & Ruano, M. C. (2023). An asset management framework for wind turbine blades considering reliability of monitoring system.

A significant aspect of this deliverable is that it has implied the collaborative efforts of all Early Stage Researchers (ESRs) involved in the project. Unless technically the demonstrator represents part of the PhD work of ERS7, 8, 9, & 10, the demonstrator has been discussed jointly with all ESRs during several special sessions activities, being of them in person, on the occasion of the CENAERO's Training Week, October, 2022.

In summary, the Analytics O&M Platform (HAPMS) is a demonstration of collaborative PhD research that move further the management of structural health in composite structures, ensuring enhanced reliability and operational efficiency for a specific case study on turbine blades, taken as example. The tool has been entirely designed by the ESRs of the project, with the supervision of their PhD advisors, however, to provide a more professional layout, the front-end has been assisted by a local software company (Spin-off of the University of Granada) named Everyware Technologies (https://www.everyware.es).

2. Description of the tool

In this development, the Petri net model (PN) is combined with the Monte Carlo Reinforcement Learning (MCRL) method to determine the optimal maintenance strategy and inspection intervals for wind turbine blades. These strategies are based on factors such as the quality of the condition monitoring system (CMS), the health of the blade, and the remaining useful life of the wind turbine. To facilitate the demonstration of these results, an API web (an Application Programming Interface web-based) has been developed. This website allows users to input data via a form, execute the script, and obtain the corresponding graphs after a few minutes.

The implementation of this platform employs technologies such as Node.js and Flask, which will be further explained in subsequent sections.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 859957.

3. API functionality and technologies employed

3.1 Technologies

The Analytics O&M Platform (HAPMS) leverages a variety of technologies to ensure efficient and reliable functionality. The core of the server-side implementation has been built using Flask, a robust Python framework for creating and managing web applications.

- **Flask:** Used on the server side to handle incoming requests and manage the web application.
- **Node.js:** Utilized for handling asynchronous operations and managing server-side logic, ensuring smooth and efficient processing of user requests.
- **Python Scripts**: Core analytical operations and simulations are executed using Python scripts, which integrate advanced models such as the Petri net model (PN) and the Monte Carlo Reinforcement Learning (MCRL) method.

3.2 API functionalities

- **Data Input and Initialization:**

  When a form is submitted from the web interface, the API receives the input data.

  The API then initializes the necessary variables and triggers the execution of the script.

- **Script Execution and Processing:**

  The script runs on the server, typically taking about 20-30 minutes to complete.

During execution, the script generates multiple files that are post-processed to create the graphs and results.

- **Email Notifications:**

If the user has provided an email address, the application sends two notifications: one at the start and one at the end of the script execution. Upon completion, the user receives a link to view the results, assuming no errors occurred during processing.

- **Result Visualization:**

The generated graphs and results are displayed on a results page accessible via the provided link. Users can review the visualized data to understand the optimal maintenance strategies and inspection intervals for wind turbine blades.

The execution flow of the API has the following sequence of steps, illustrated in Figure 1:

- The user submits input data via the web form.
- The API receives and validates the input data.
- The API initializes variables and executes the Python script.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 859957.

- The script processes the data and generates output files.
- The output files are post-processed to create visual graphs.
- Email notifications are sent to the user (if an email address is provided).
- The user accesses the results through the link provided in the final email notification.

This structured approach ensures that users can efficiently interact with the HAPMS platform, providing them with timely and accurate results for their structural health management needs.

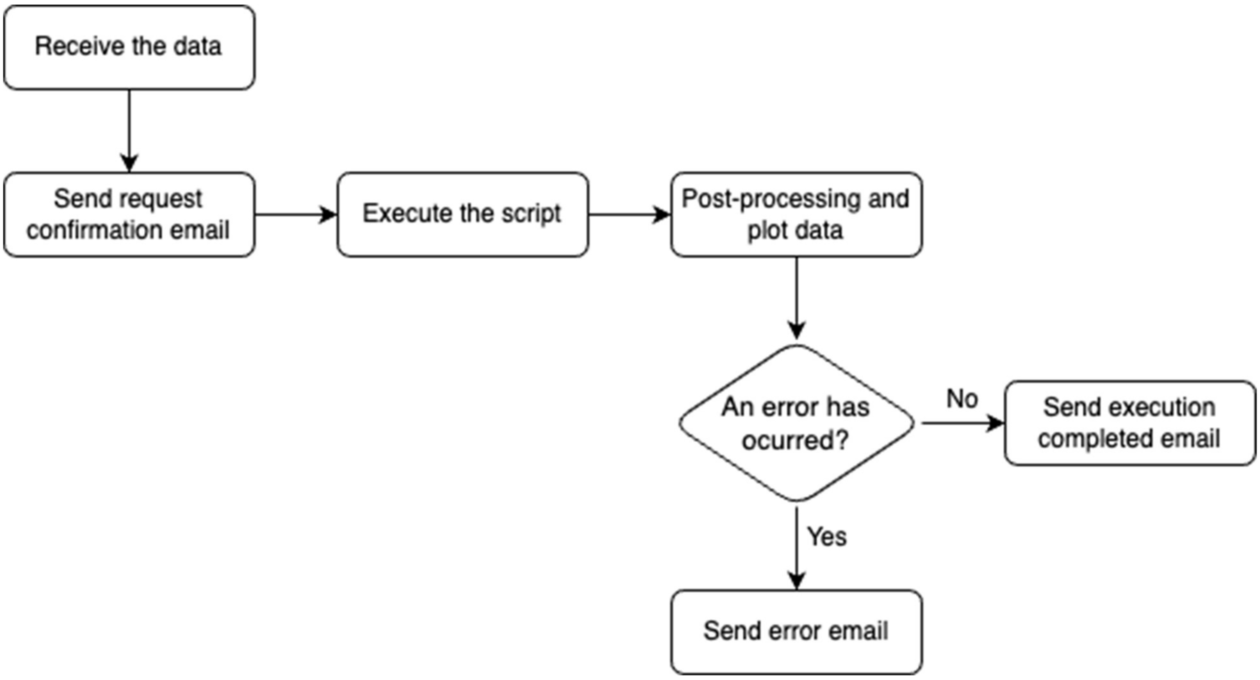

Figure 1. Flow of the tool

4. Web functionality and technologies employed

4.1 Technologies

To demonstrate the research project, we have chosen to develop a simple web page that, roughly speaking, collects the user's data and sends it to the server (API) to execute the script with said parameters.

The web page, when sending the data and due to the long time that the script requires, shows a loading interface with the option to copy the link so that the user can consult it later.

Once the results are available on the server, the web page automatically reloads to display these results. Similarly, if any error occurs in the execution of the script, the page will automatically display an error text related to the execution of the script.

The following technologies and libraries have been used to implement this website:

- **Nodejs:** Web deployment. Using libraries such as express, ejs and body-parser, the server serves the HTML web page rendered on the server to facilitate SEO.
- **Axios:** For requests to the API to send input parameters.
- **HTML, CSS and Javascript:** to render the web page.
- **Materializecss:** Frontend framework used to improve the style and functionality of the page.
- **jQuery:** Javascript framework to speed up front-end development.

For the deployment on the server, the following has been used:

- **Pm2:** Process manager for node and Python applications. To facilitate deployment and improve error handling. Used for both web and API.
- **Apache2.** As a web server. All requests made to the demonstrator domain are redirected to the previously launched and configured Nodejs server.

4.2. Web functionality

To run the script you will first need to complete the form provided. At the top right, there is a button that allows you to fill it with default data. At the end of the form, the email is requested in case you want to be notified that the script has finished executing.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 859957.

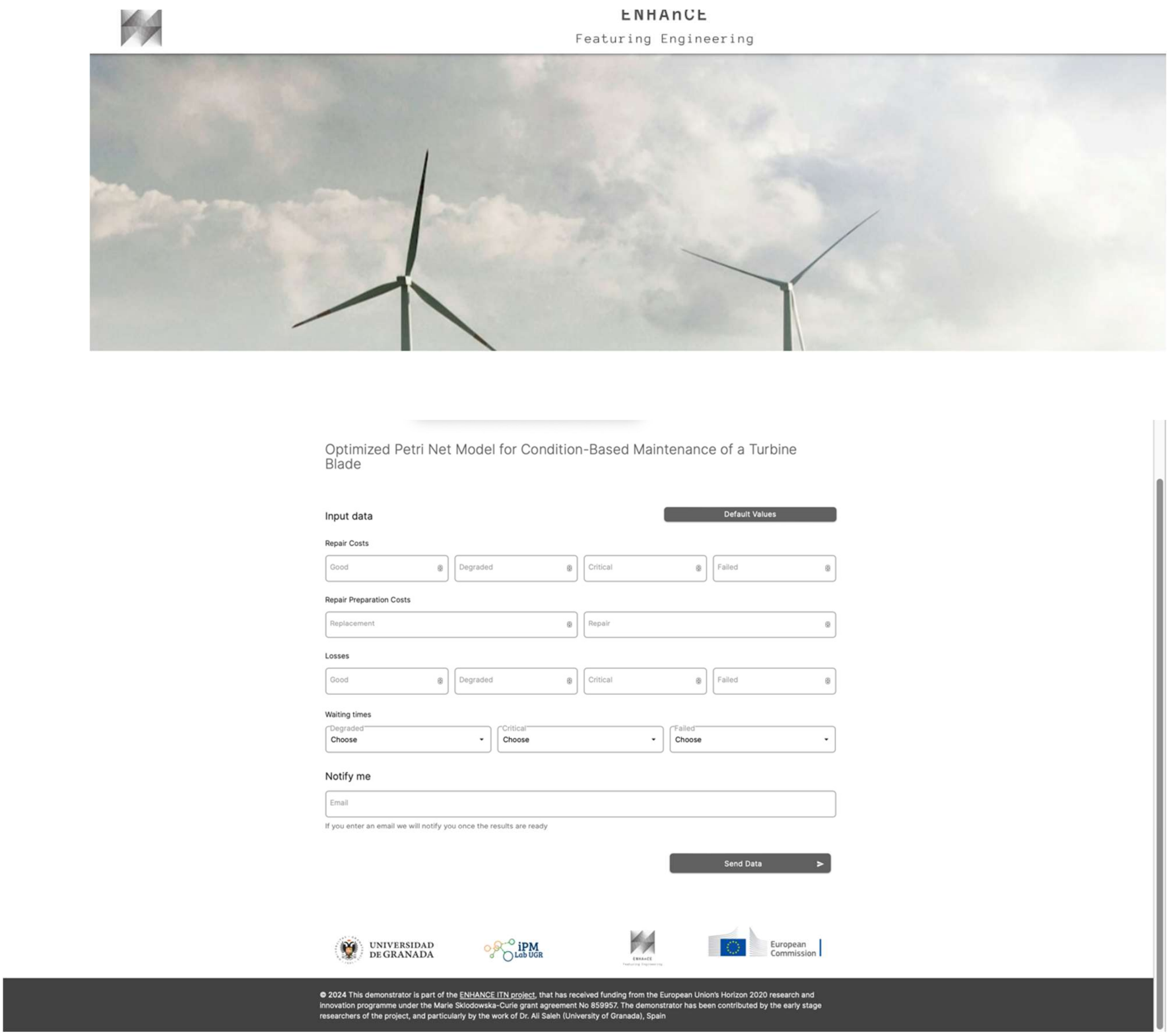

Figure 2. Front-end of the tool, screen 1: input data

When the form has been submitted, we will be redirected to a waiting page, telling us that the script is still running. This page will be reloaded from time to time until the results have been generated and displayed. The link is also provided in case we would like to close the page and need to access it later.

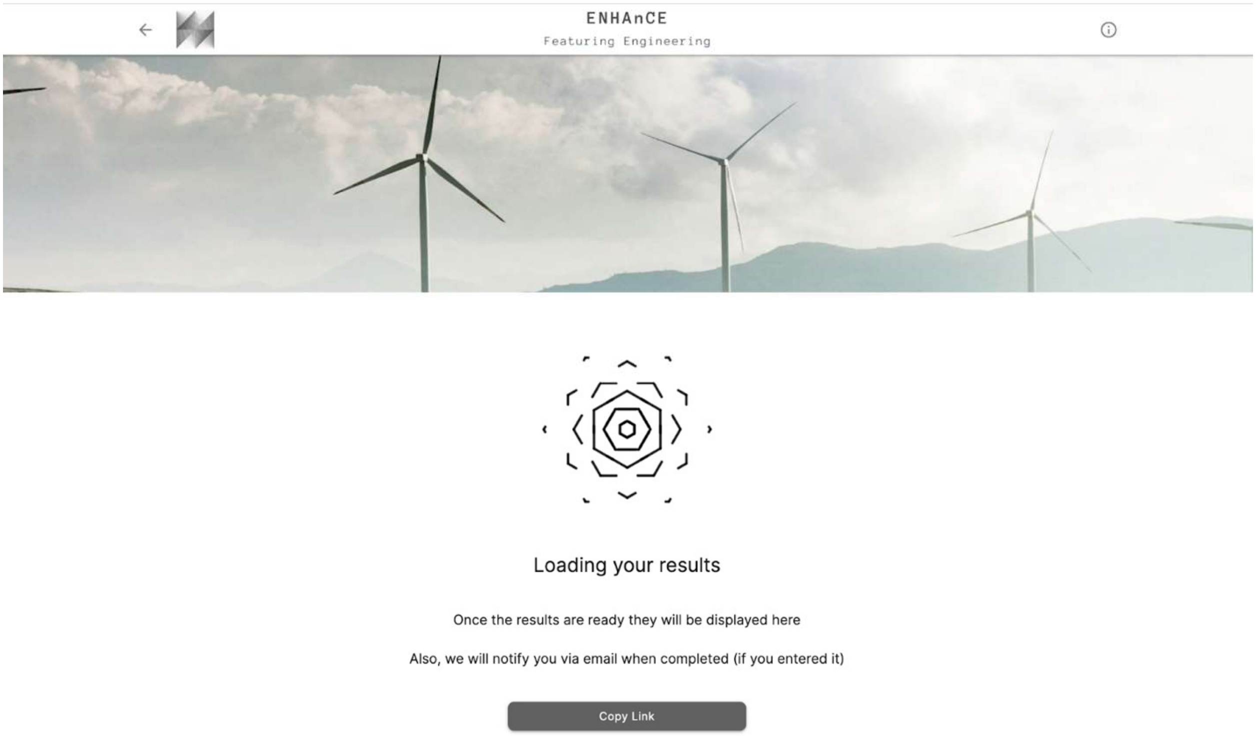

Figure 3. Front-end of the tool, screen 2: analysis

If the email has been entered, this would be the message that we would receive in the inbox. In it we can find the variables that have been entered in the form and also the link to the results page.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 859957.

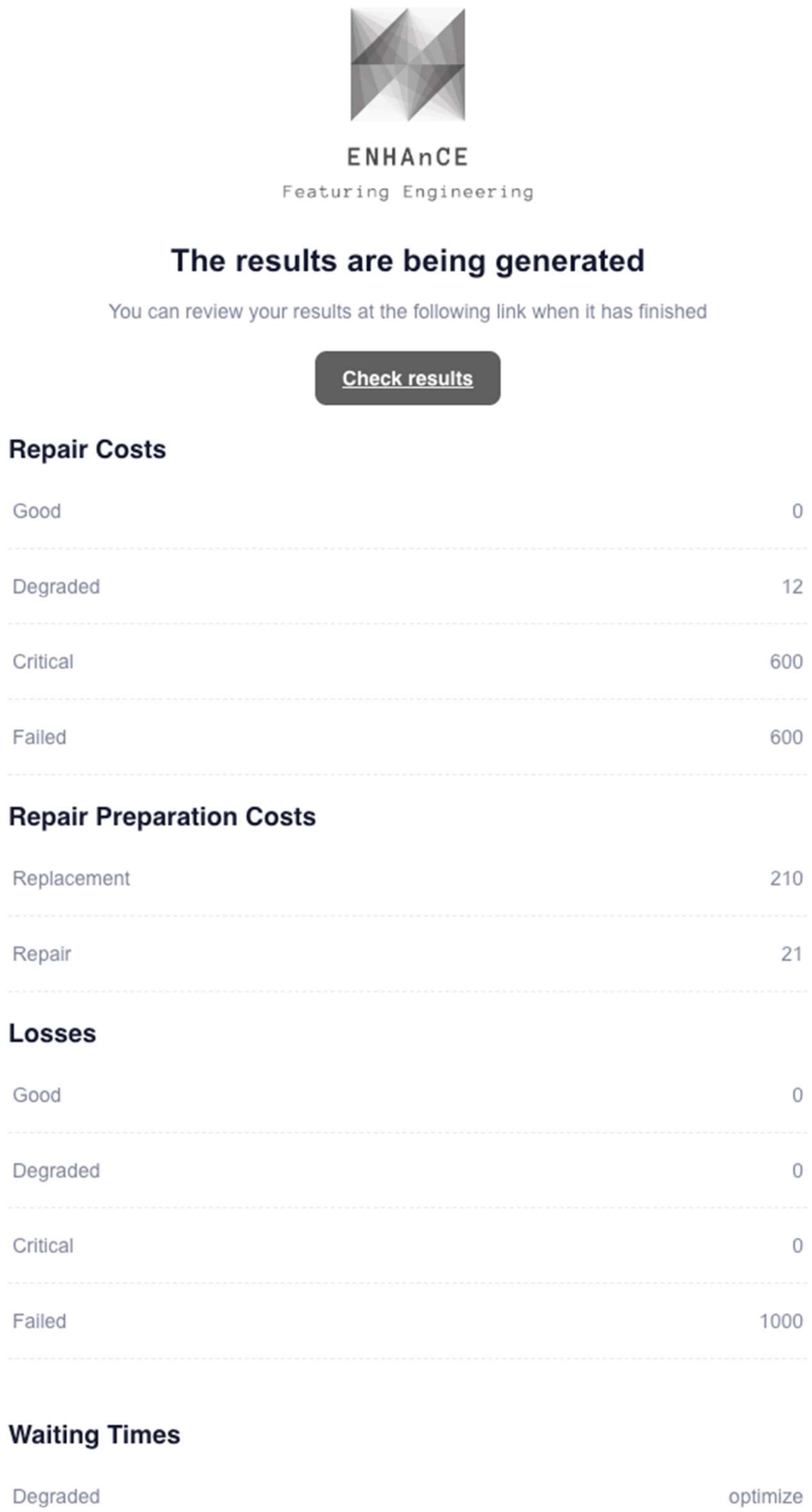

Figure 4. Front-end of the tool, screen 3: results

Finally, when the script finishes, all the generated graphs are displayed.

This tool allows for finding the optimal time to perform maintenance after different degraded conditions are reached. If the time is equal to 0, it means that repair should be done as soon as the condition is reached. On the other hand, if the time is too long, it is an indication that it is better not to repair at this condition and to wait until the next degraded condition.

The plots below show the evaluation of different strategies within different stages of the optimization process. Each point within these figures represents a specific strategy. Its coordinates represent the parameters of the strategy, and its color represents the cost of the strategy, which is evaluated using the Petri net model. This tool allows for finding the optimal time to perform maintenance after different degraded conditions are reached. If the time is equal to 0, it means that repair should be done as soon as the condition is reached. On the other hand, if the time is too big, it is an indication that it is better not to repair at this condition and to wait until the next degraded condition. The plots below show the evaluation of different strategies within different stages of the optimization process

Each point within these figures represents a specific strategy. Its coordinates represent the parameters of the optimal strategy, and its color represents the cost of the strategy, which is evaluated using the Petri net model.

For example, point (d2=10, d3=5) refers to the strategy where:

- After reaching critical condition, 10 weeks have to pass before performing maintenance.
- After reaching failed condition, 5 weeks have to pass before performing maintenance.

To further demonstrate how the tool can be used, a more general example of these results can be found in the following link: https://enhancedemonstrator.everyware.es/result/f31eb5d9-5059-47ab-8b50-a41daf571201

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 859957.

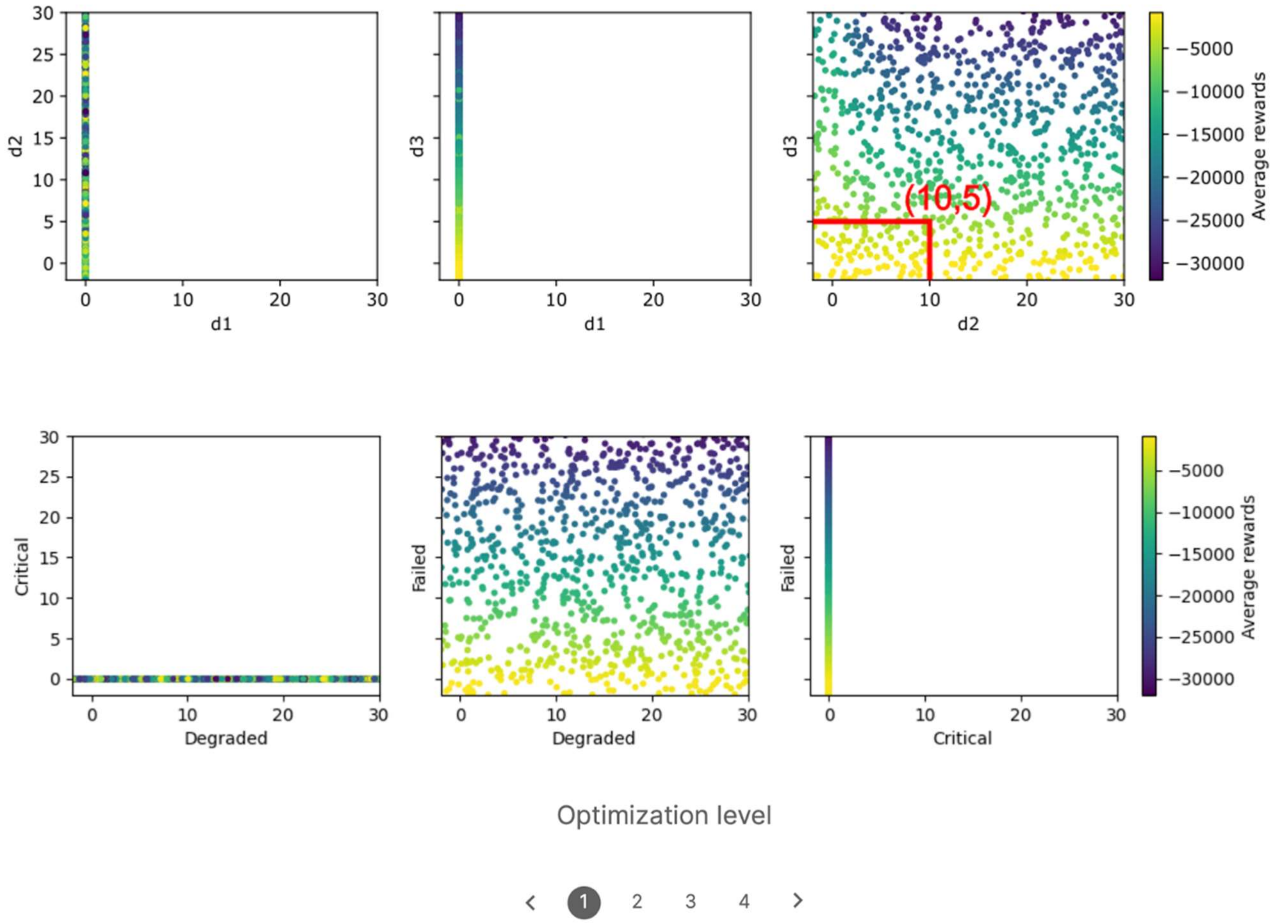

Figure 5. Plots evaluating different strategies

Once again, if the email has been entered, the message we would receive in our inbox would be the following, with the link to the results page and the variables with which it was executed: https://enhancedemonstrator.everyware.es/result/f31eb5d9-5059-47ab-8b50-a41daf571201

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 859957.

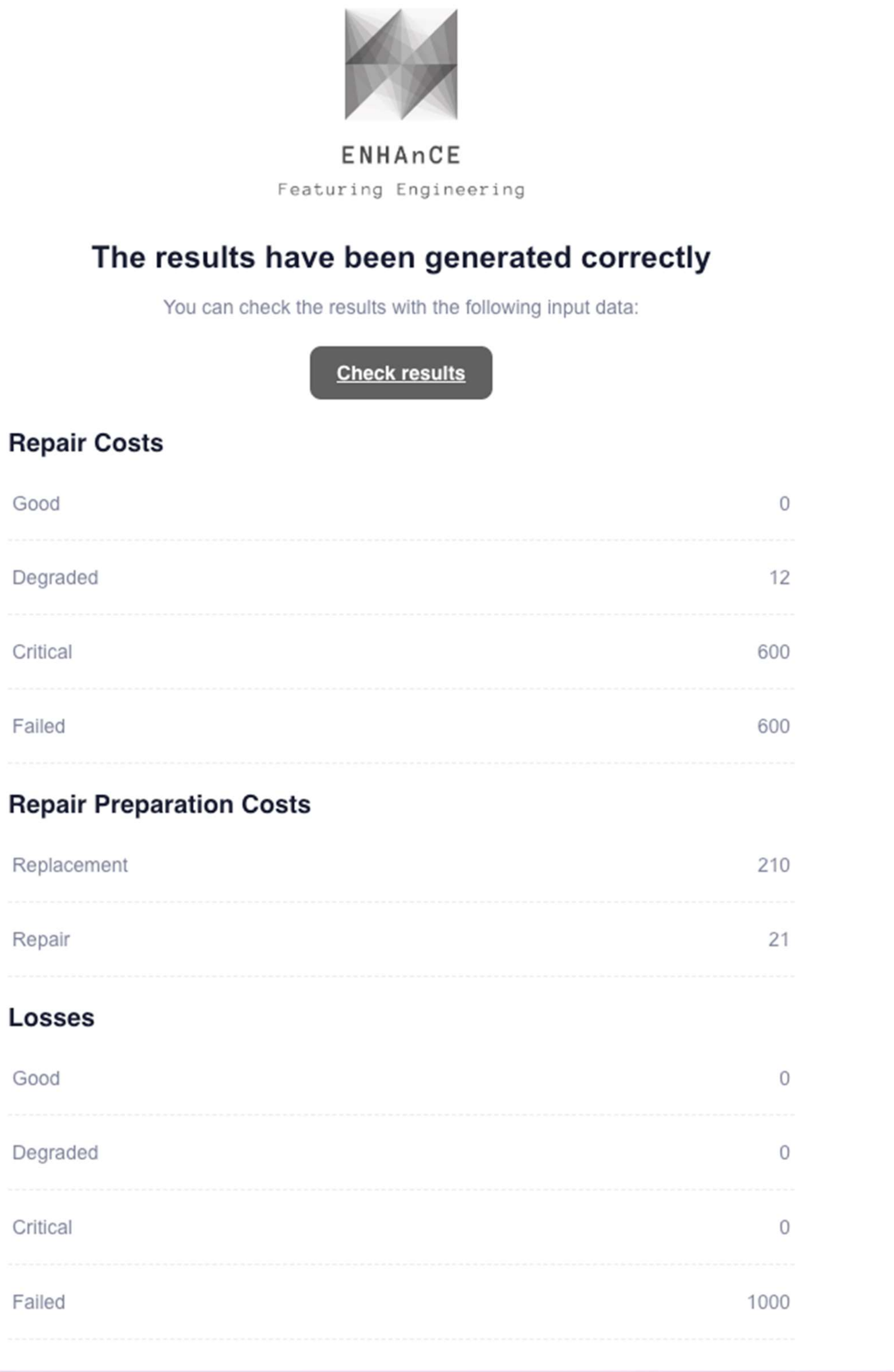

Figure 6. Front-end of the tool, screen 4: results checking

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 859957.

5. Access to the demonstrator

In this section, the link to access the Analytics O&M Platform (HAPMS) can be found here: https://enhancedemonstrator.everyware.es