Universiteit van Amsterdam

Final Report

Digital Twin Project

15 oktober 2025

Studenten:
Artur Martins
16407636

Emad Mshalwat
15671933

Laxman
16507185

Shuo Lin
16474449

Tutor:
Victoria Degeler

Practicumgroep:
Group N

Cursus:
Digital Twin Engineering

Vakcode:
5284DITE6Y

1 Introduction & Purpose

Wildfires are a frequent concern in Portugal, exacerbated by land management policies and criminality. They do significant damage to wildlife ecosystems, properties, and human lives. Understanding when and where fires start is critical for prevention, preparation, and response. This project develops a **Digital Twin (DT)** focused on Portuguese wildfires. The DT acts as a virtual representation of wildfires by:

- **Preprocessing** a primary dataset requiring meticulous cleaning, organization, and structuring of information.
- **Visualizing** spatial and temporal patterns of fires to elucidate trends and identify critical areas. A real-time feature facilitates civil protection agencies in preparing for the wildfire season.
- **Supporting analysis** by establishing a framework that allows diverse users (researchers, policymakers, firefighters) to engage with and analyze data effectively.

The system aims to develop a dependable framework for converting wildfire data into understandable insights through the use of maps, charts, and exploration tools, while also generating predictions. This groundwork supports decision-making and prepares for future enhancements, such as incorporating more sophisticated AI models and integrating additional datasets (geographic, demographic, weather history) to enhance context and precision.

2 Users, Scenarios, and Requirements

The Digital Twin is designed as a practical tool that connects data-driven wildfire analysis with the real needs of those who must act upon it. Its users fall into three main groups:

- **Fire management agencies**, who require timely insights into historical fire patterns to plan preventive measures and allocate resources effectively.
- **Policymakers and environmental authorities**, who use aggregated trends to design long-term strategies for land management and climate adaptation.

- **Researchers**, who need structured data and visualizations to investigate the underlying dynamics of wildfires in relation to environmental factors such as weather, vegetation, and human activity.

These users interact with the system through concrete scenarios: identifying high-risk regions before the fire season, analyzing seasonal or annual variations in fire frequency and intensity, and validating or challenging existing scientific assumptions about wildfire behavior.

To support these scenarios, the Digital Twin must satisfy several requirements:

- **Data reliability:** preprocessing must ensure that irregular and incomplete wildfire records are cleaned, standardized, and consistent.
- **Spatio-temporal representation:** events must be modeled in a way that captures both geographic location and time evolution.
- **Visualization:** the system should offer interpretable outputs such as maps, heatmaps, and time-series plots to make patterns accessible to diverse stakeholders.
- **Extensibility:** the design should allow integration of additional data sources (e.g., meteorological or vegetation datasets) and advanced models (e.g., predictive analytics).

By meeting these requirements, the Digital Twin provides not only descriptive insights into past wildfire events but also establishes a flexible framework for future expansion toward real-time monitoring and predictive capabilities.

3 Data and Preprocessing

3.1 Dataset Overview

The dataset used in this project comes from the *Instituto da Conservação da Natureza e das Florestas* (ICNF), which maintains official records of wildfire incidents in Portugal. It spans the period from 3 January 2013 to 18 October 2022, covering nearly a decade of wildfire activity. The dataset contains a total of 143,403 recorded wildfire events, stored in a file of approximately 82 MB.

Each event is described by up to 71 features, which include:

- **Wildfire characteristics:** size of the fire, type of fire (e.g., agricultural, forest), flags for reignitions or small-scale incidents.
- **Spatial information:** geographic coordinates (latitude/longitude and national projection), as well as administrative divisions such as district and municipality.
- **Temporal information:** alert date and time, extinction date and time, intervention times, and calculated durations.
- **Environmental context:** meteorological conditions such as temperature, humidity, wind speed and direction, precipitation, and terrain data (altitude, slope, vegetation density).
- **Derived indices:** values from the Canadian Fire Weather Index (FWI) system, including fire danger codes (FFMC, DMC, DC, ISI, BUI, DSR).
- **Cause information:** recorded or suspected ignition causes, grouped into broad categories such as negligent, intentional, natural, and unknown, with subcodes for detailed circumstances.

3.2 Challenges in the Raw Data

Although the dataset is rich, it presents several issues:

- The dataset was entirely in Portuguese, including column names and categorical values, requiring careful translation for international use and clarity.
- High proportions of missing data in some features (e.g., simulation-derived burned area estimates and wind direction vectors).
- Redundancy, with overlapping representations of time (separate year/month/day vs. full timestamps) and space (district names vs. coordinates).
- Inconsistencies in temporal fields, with misaligned alert and extinction times or irregular formatting of hours and dates.

3.3 Preprocessing Pipeline

The preprocessing of the dataset involved several systematic steps:

- **Translation and standardization:** all Portuguese column names and categorical values were translated into English using dedicated dictionaries. For example, `DISTRITO` became `DISTRICT`, `TIPO` became `WILDFIRE_TYPE`, and values such as *Florestal* were translated to *Wildfire*. This step made the dataset readable and ensured reproducibility.
- **Attribute reduction:** columns with more than 80% missing values or a single unique value were dropped, and redundant fields were eliminated.
- **Event encoding:** categorical flags for reignitions and small fires were merged into the `WILDFIRE_TYPE` field, producing descriptive composite categories (e.g., *Wildfire_Small_Reignition*).
- **Temporal harmonization:** separate date and time columns were merged into unified Unix timestamps with minute-level precision. Extinction times were recalculated from alert times and reported durations where inconsistencies appeared.
- **Handling missing values:** for essential numerical features (temperature, humidity, wind speed, precipitation, vegetation indices), null entries were imputed with the column mean. Extreme placeholder values (e.g., -999) were treated as missing and removed.

After preprocessing, the dataset was exported as `ICNF_2013_2022_cleaned.csv`. The cleaned version retained the most relevant spatio-temporal and environmental variables, while discarding redundant or unreliable fields. This produced a dataset that is consistent, interpretable, and efficient to use in the Digital Twin framework, providing the basis for visualization and machine learning modelling.

4 System Representation (Model, State, Events, Transitions)

4.1 Model Overview

The Digital Twin is structured around a Digital Shadow Architecture that models wildfire events in Portugal by combining historical data replay, event-based simulation, and AI-driven forecasting. Instead of mirroring physical systems in real time, it constructs a digital reflection—a "shadow"—of wildfire dynamics, enabling analysis of past behavior and predictive insights for future risk.

The data replay process is handled by a dedicated Replayer component, which streams historical wildfire records into the **Kafka** messaging system under the Wildfire Topic. Kafka acts as the central event broker, decoupling data producers and consumers within the pipeline.

For the Kafka pipeline (Replayer and Consumer), we selected Go as the primary language due to its strong support for Kafka and InfluxDB libraries and its high performance, which is

essential for replaying our dataset at accelerated speeds. The Go programs run on Alpine Linux to maintain a lightweight environment.

Kafka was chosen for its proven reliability and widespread adoption in enterprise settings, while InfluxDB was selected for its specialization in storing and querying time-series data, making it ideal for our dataset.

The remaining two components were developed in Python to leverage its extensive data science and machine learning libraries. For visualization, we used Plotly Dash, which offers versatility and support for Leaflet, enabling interactive map visualizations. The Python programs run on Debian machines to accommodate the C++ dependencies required by many packages.

The architecture is composed of four main layers:

- **Time Series Storage Layer implemented in InfluxDB**: a time series database that stores both historical and predicted wildfire events for efficient querying and visualization.
- **Processing Layer**: simulates real-time message flow using Kafka, the most robust and tested event streaming framework, replaying historical data to emulate live wildfire activity. It is in blue in the diagram 1.
- **AI Layer**: applies machine learning algorithms, specifically Microsoft LightGBM, to predict future wildfire characteristics, including size, duration, and affected area. The AI model queries the InfluxDB for up to 1,250 records of recent events to generate new predictions and inserts the results back into the database.
- **Visualization Layer**: connects analytical outputs to an interactive dashboard, allowing users to explore patterns, compare historical and predicted data, and extract actionable insights. It can visualize Wildfires in real-time by connecting directly to Kafka or it can do more in-depth analysis by reading from InfluxDB. It is in light blue in diagram 1.

A Consumer Service subscribes to Kafka topics and inserts received wildfire events into InfluxDB, ensuring continuous data availability for visualization and analytics.

Through this layered design, the Digital Twin supports three complementary functions:

- **Historical Replay** visualizing past wildfire patterns to understand frequency, duration, and distribution.
- **Simulation** reproducing event flow and temporal evolution within the system.
- **Prediction** forecasting future wildfire risks to support preventive measures and strategic resource planning.

4.2 States and Events

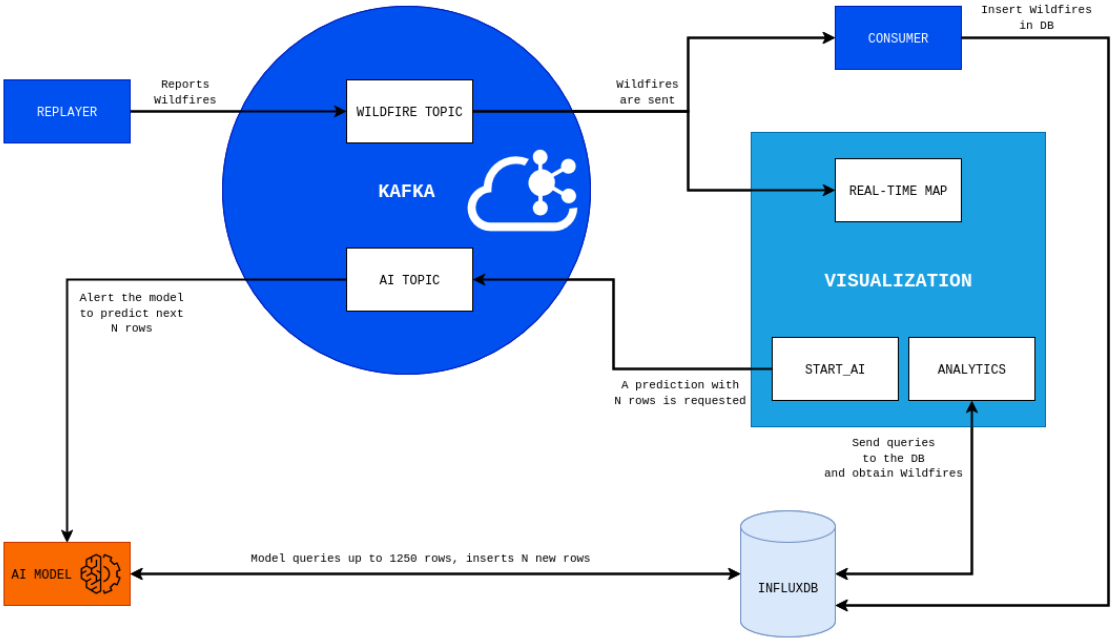
The central entity of the model is the **Wildfire Event**, representing a single recorded fire instance enriched with its spatial, temporal, and environmental context. Each event contains four groups of attributes:

- **Location:** latitude, longitude, district, and municipality.
- **Timeline:** alert, intervention, and extinction timestamps.
- **Impact:** total burned area, affected population density, and land category.
- **Context:** weather and environmental variables such as temperature, humidity, wind speed, and Fire Weather Index (FWI).

These attributes define the state of the system at a specific time. A state may represent a particular geographic region's wildfire condition or a moment in the national timeline.

Events act as **discrete triggers** that alter the system's state. The main types include:

- Alert Event – a new wildfire occurrence is recorded.



Figuur 1: Infrastructure of the DT

- Extinction Event – the fire is extinguished.
- Prediction Event – newly generated forecasted cases based on AI inference.

These events drive the system's temporal behavior, linking the data ingestion pipeline, simulation logic, and analytical updates.

4.3 Transitions and Dynamics

Each wildfire instance evolves through a series of discrete **state transitions** that describe its life cycle:

No Fire → Active Fire → Extinguished Fire.

When a user initiates a prediction via the **Start AI** button on the dashboard, a message is sent to the **AI Topic** in Kafka. The **AI Model** consumes this message, queries up to 1,250 historical rows from **InfluxDB**, generates new predicted records, and inserts them in the database. These predicted events appear in the visualization as future states extending the existing timeline.

This event-driven workflow creates a dynamic state machine in which data replay, AI prediction, and visualization remain continuously synchronized.

5 Visualization & Results

5.1 Visualization Objectives and Approach

The visualization component provides an intuitive interface for interpreting the complex spatio-temporal behavior of wildfires. Its primary objectives are to:

1. **Understand the Past** allow users to replay and observe historical wildfire patterns across Portugal.
2. **Simulate the Present** enable users to explore event progression and analyze correlations between environmental variables and fire spread.

3. **Predict the Future** visualize AI-generated forecasts to support strategic decision-making and preventive planning.

The approach prioritizes clarity, interactivity, and scalability. Through map-based layers and statistical dashboards, stakeholders—such as civil protection authorities, policymakers, and researchers—can analyze wildfire behavior, detect high-risk areas, and assess environmental conditions.

5.2 Visualization Implementation

Visualization is implemented through a custom interactive dashboard integrating geospatial and analytical components.

The dashboard includes a **Start AI** control that allows users to request new predictions directly. When triggered, the request is sent to Kafka's AI Topic, activating the AI Model pipeline and updating the map with forecasted events.

- **Interactive Map:** displays all wildfire events geographically. Users can zoom, pan, and click to access event-level details (district, size, duration, and weather context). Filtering options by year, region, or fire type enable targeted exploration.
- **Temporal Analysis:** a timeline chart presents fire frequency and burned area evolution from 2013 to 2022, highlighting seasonal variations and long-term trends.
- **Statistical KPIs:** key performance indicators summarize total events, cumulative burned area, and average intervention time.
- **AI Prediction Layer:** overlays predicted fire events on the map, allowing visual comparison between historical records and forecasted risk zones.

This multi-view interface transforms raw data into meaningful visual patterns, providing a clear understanding of wildfire dynamics. It serves as both an analytical tool and a communication platform, enabling evidence-based decision-making for emergency response and land management.

5.3 Results and Interpretation

The implementation of the Digital Twin produced several relevant analytical results. Historical data replay revealed **strong seasonal clustering** of fire incidents, with peaks typically occurring between July and September, correlating with low humidity and high temperatures. Spatial analysis indicated that **central and southern districts**—notably Coimbra, Leiria, and Faro—experience the highest fire frequency and burned areas.

The LightGBM predictive model achieved reliable accuracy, effectively forecasting high-risk periods and regions based on weather and historical features. It showed that temperature, FWI, and wind speed were among the most influential predictors. The visualization of these forecasts within the dashboard provided **actionable insights** for resource allocation, such as prepositioning firefighting teams in high-risk districts.

By integrating AI predictions with an interpretable visual interface, the Digital Twin evolved from a retrospective analytics tool into a **proactive decision-support system**. This combination of simulation, prediction, and visualization marks a significant step toward data-informed wildfire management in Portugal.

6 Insights, Limitations, and Future Work

6.1 Insights

Go proved to be an excellent choice for the processing layer of this project due to its speed and simplicity. Opting for Plotly Dash over Grafana provided us with greater flexibility, customization options, and additional features, enhancing our visualization capabilities. InfluxDB and Kafka fully met our expectations for data processing and real-time streaming.

From the outset, our goal was to containerize the project to streamline development, deployment, and future scalability—an objective we successfully achieved.

While we initially considered using more advanced AI models, we ultimately prioritized accessibility, ensuring that all group members could run and test the project on their personal computers.

6.2 Limitations

The contextualization of the predictions is poor, as the prediction model only uses wildfire data to predict the next wildfires. More parameters and a connection to other datasets are recommended. The rows used to train the model are limited to 1250, creating an artificial ceiling on accuracy.

The interactive map, built using Leaflet, struggles when more than 10.000 wildfires are shown. A purpose-built solution could attenuate this issue.

6.3 Future Work

- Tune Kafka batch size and sampling for smoother real-time visualization.
- Use a more refined AI model for predicting wildfires such as:
  - **ibm-granite/granite-timeseries-ttm-r2**: we tested this model running on the CPU and on a GPU, but, even with only 805k parameters, it was too slow. With more training data and more epochs, it is more accurate.
  - **thuml/sundial-base-128m and Datadog/Toto-Open-Base-1.0**: These are both some of the most advanced Time Series Forecasting models, both with over 100M parameters. This current version of the project was designed to run on a CPU, so that all team members could test the model.

  Naturally, leveraging more advanced models would require a GPU with over 12GB of VRAM, which would significantly enhance the accuracy of wildfire predictions. For this project, LightGBM was selected as it offered the best balance between speed and accuracy given our resource constraints.
- Integrating additional datasets (geographic, demographic, weather history, satellite imaging) to improve context and further improve the accuracy of the AI model. demographic, weather history) to enhance context and precision.

7 Closing Remarks

7.1 Challenges Faced

- **AI Model Selection:** Testing multiple AI models; including advanced LLMs, data mocking models, and gradient boosting models; was time-consuming, with each model requiring extensive evaluation.
- **Timestamp Format Mismatches:** Inconsistent timestamp formats between the AI model and database led to frequent errors due to format mismatches.
- **Container Boot Time:** The AI model container took several minutes to boot due to *PyTorch* and *NVIDIA* dependencies, significantly slowing down development.
- **InfluxDB Authentication:** Automating authentication for InfluxDB proved challenging, requiring the implementation of a complex script.
- **Kafka Message Handling:** Sending messages in Kafka required careful setup, and a clean Kafka container was needed after each restart to avoid remnants of old data.

- **Real-Time Map Synchronization:** Synchronizing the real-time map with Kafka messages was difficult and required fine-tuning. A **local clock** updates upon receiving new messages, with timing and accuracy being critical for declaring wildfires as extinct and removing them from the active list.
- **Data Processing Libraries:** While *Polars* was preferred for its speed, *Pandas* had to be used as well due to compatibility with InfluxDB's data format and most AI models.
- **InfluxDB Querying:** Counting unique wildfires was difficult because InfluxDB stores data in multiple rows. To improve performance and speed up filtering, a new column was added for faster counting.

7.2 Work Division

- Artur – selected the AI model for wildfire prediction, and managed project containerization and deployment using Docker, Docker Compose and multiple scripts.
- Emad – handled data cleaning and preparation, and established the Kafka pipeline to enable real-time data sharing and consumption between services.
- Laxman – developed database queries, designed RESTful APIs, and integrated backend services with both the frontend and the Kafka-based data flow.
- Shuo – implemented the frontend user interface, including the interactive map visualisation and the dashboard for data presentation and navigation.