18th International Conference on Space Operations, Montreal, Canada, 26 - 30 May 2025.

SpaceOps-2025, ID # 80

EUMESTAT Machine Learning Framework System and AI/ML Applications

Gianni Casonato[a]*, Ruth Britton[b], Luca Garegnani[c]

[a] *Technical and Scientific Support Department, EUMETSAT, Germany, gianni.casonato@eumetsat.int*
[b] *Infrastructure for IT and Data Division, EUMETSAT, ruth.britton@eumetsat.int*
[c] *Digital Solutions and Satellite Application Facilities Division, EUMETSAT, luca.garegnani@eumetsat.int*
* Corresponding Author

Abstract

The growing interest in AI/ML applications at EUMETSAT is driven by their significant potential to enhance task efficiency and effectiveness across a wide range of operational areas. Within that context, a concrete step in starting to work with AI/ML applications in EUMETSAT has been the realisation of the ML Framework. This centralised framework offers internal teams AI/ML tools to support the exploration and development of proof-of-concept applications across various operational areas. The ML Framework system delivers controlled-execution capabilities, providing automatized AI/ML model training, re-training, and serving (so-called ML-Ops capabilities) for small-to-large scale scenarios and a large variety of AI/ML models based on Deep Networks, Convolutional Networks and Large Language Models.

For the past four years, EUMETSAT has been leveraging the ML Framework to support approximately 10 different teams, facilitating the identification and validation of promising AI/ML applications that are potential candidates for future operational tools in routine activities .

As next steps the ML Framework transition into a permanent AI/ML platform in EUMETSAT will be assessed, for both keeping the proof-of-concept support role but also for running production AI/ML application for specific operational use cases.

Keywords: Artificial Intelligence, Machine Learning, Neural Networks, Large Language Models.

Acronyms/Abbreviations

AI	Artificial Intelligence
ANN	Autoencoder Neural Network
API	Application Programming Interface
CNN	Convolutional Neural Network
COCO	Common Objects in COntext
DNN	Dense Neural Network
IoU	Intersection over Union
IPR	Intellectual Property Rights
LLM	Large Language Model
LSTM	Long-Short Term Memory networks
mAP	mean Average Precision
ML	Machine Learning
MLF	Machine Learning Framework
MLOps	Machine Learning Operations
RAG	Retrieval Augmented Generation
RNN	Recurrent Neural Network
YOLO	You Only Look Once

1. Introduction

1.1 EUMETSAT Context

EUMETSAT is the European inter-governmental agency tasked with the exploitation of Earth-observation satellites to support the fields of operational meteorology, climatology and oceanography, and it is naturally interested in looking at new mature technologies and trends for improving its operational activities and processes. Among them, a special focus has been given in recent years to the AI/ML field, to identify potential application areas for EUMETSAT teams and to promote the benefits of integrating these technologies into operational tasks.

Within that context, a concrete step in starting looking at AI/ML applications in EUMETSAT has been the realisation of the ML Framework (MLF), for an organised and structured support to the exploration of AI/ML applications in various internal areas.

More specifically, MLF provides centralised and general-purpose AI-as-a-service capabilities to EUMETSAT internal teams covering their use cases proof-of-concept and supporting both AI/ML application result data analysts/consumers and AI/ML experts.

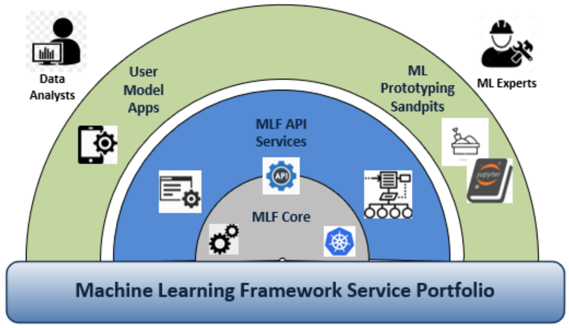

Fig. 1. MLF AI-as-a-Service Portfolio

The two classes of expected MLF users access the system in different ways, simply getting AI/ML results from relevant applications for Data Analysts/Consumer or actively interacting with MLF in development sandpits for ML Experts. Underlying service implementation and hosting on computing resources is completely transparent for both.

2. EUMETSAT Machine Learning Architecture and Services

2.1 High Level Architecture

The MLF system was designed and built with the main objectives of providing a flexible and scalable framework, able to support a wide range of different AI/ML applications (ideally "any model on any data"), and compliant with EUMETSAT corporate security and data protection constraints. In addition, the MLF system should also allow effective production execution administration, for minimising the required team effort.

Consequently, the following key design pillars were considered in the MLF architecture:

- AI/ML applications loose coupling, realised via containerization and making each AI/ML application independent from the others
- Wide range AI/ML applications support, obtained not binding to a proprietary specific AI/ML solution but building the MLF on top of state-of-art cloud service technologies
- Streamlined AI/ML application development and execution on MLF, realised by setting up a MLF API library wrapping common utility classes for EUMETSAT data sources access, ready to use models, datasets preprocessing algorithms, model scoring statistics, ML pipeline automation
- MLOps Level 1 production execution capabilities as initial target, with a goal to reach MLOps Level 2

A high-level view of the MLF system architecture is provided in the figure below. The different layers for hardware infrastructure hosting and Kubernetes cluster service, for the MLF API helper modules and finally for the AI/ML applications.

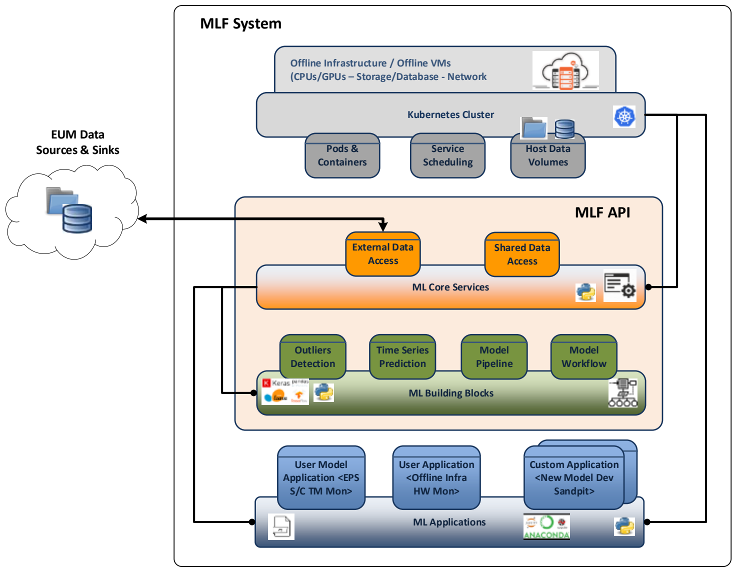

Fig. 2. MLF System Architecture

Current MLF base platform makes use of native automation Kubernetes capabilities for supporting MLOps Level 1 services. It will be upgraded by end of 2025 to a EUMETSAT version of the ESA AInabler platform [1], reaching MLOps Level 2, apart from other additional features for application pipelines handling, results presentation to users, and system monitoring.

The MLF AI/MLF applications are built following a standard execution pattern, based on a pipeline of steps consisting of:

- Data acquisition from the relevant EUMETSAT data source
- Dataset factory step, where feature engineering is performed for preparing the data in the format expected by the model, if needed. An engineered data caching is also performed on training datasets to keep them available for retraining
- Model data retrieval, picking up engineered datasets either from data sources of from cache
- Model pipeline execution for training of for prediction
- Results provided to EUMETSAT end users in the relevant data sinks

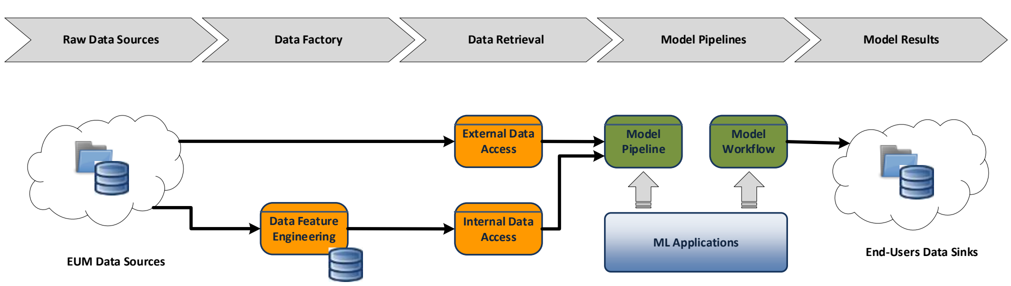

Fig. 3. MLF ML Pipeline Support

All those steps are automatised by the MLF API modules for minimising the code to write for a new application, having as a target a no-coding framework in future evolutions.

2.2 AI-as-a-Service and MLOps Approach

One key characteristic of the MLF is the capability of providing effective AI/ML application production execution as a service for EUMETSAT internal use cases. Specifically, the use of a centralised AI framework used as black-box for AI/ML applications brings important advantages for EUMETSAT as organisation in the area of:

- User effectiveness in getting AI/ML applications results, integrated in its operational processes. Application deployment in MLF is straightforward using Docker images, both for MLF “native” applications (i.e. based on MLF API) and 3-rd party ones.
- Cost efficiency in running production AI/ML applications. Maintenance and operational costs and team resources are minimised, also thanks to the use of MLF MLOps automation capabilities.
- Cost efficiency in framework scaling and evolutions to new AI technologies. The MLF layered architecture makes possible transparent hardware infrastructure upgrades as well as deployment of extra AI capabilities when available, without a disruptive effect on the framework and on the applications.
- Compliance with corporate guidelines on data protection and privacy, as well with AI regulatory in general. EUMETSAT data and documentation is subject to proprietary rights (IPRs), and it makes difficult if not impossible to share it on Internet-based cloud services. MLF supports a local storage of data and models, preventing the data leakage or IPR breach issues to happen.
- Protection vs. biased or unreliable data. MLF control over training data for any installed model, together to sanity controls on data quality if needed, avoid the risk of malicious data biasing by external entities.

MLF architecture is built targeting those objectives and trying to identify the best trade-off between AI service capabilities and performance and running maintenance and operational costs but always guaranteeing the maximum protection of EUMETSAT data.

3. EUMETSAT AI/ML Use Case Proof-of-Concept Applications

3.1 General Overview

The MLF capabilities described in the previous section are exploited for delivering automated production application control capabilities (target ML-Ops level 2) to EUMETSAT internal teams, for AI/ML use case proof-of-concept validation campaigns. The support covers small-to-medium scale scenarios and a large variety of AI/ML models, including Deep Neural Networks (DNNs) and Convolutional Neural Networks (CNNs), Object Detection networks, up to Large Language Models, Agents and Generative AI models.

Here below more details on the three main classes of applications currently supported in MLF, namely timeseries analysis models, large language models and object detection models. In addition, other type of models are under assessment for future extensions, specifically in the area of Reinforcement Learning and in support of Generative AI.

The MLF system is currently hosting several EUMETSAT use cases, in operations monitoring for spacecraft flight control [2], ground station, ground systems [3], in the area of documentation access for spacecraft and for data product knowledge bases, and in the area of satellite image classification.

3.2 Timeseries Analysis

Neural network models are the most common way to tackle AI/ML use cases where timeseries datasets should be analysed for looking at specific characteristics or data behaviours like trends and outliers. Those cases can be effectively tackled with a large variety of layouts, spanning from DNNs when just a large network size is enough, to CNNs if multi-dimensional datasets and large-scale features should be captured, to Recurring Neural Networks (RNNs) when the data points sequence plays a role, and finally to Autoencoder Neural Networks (ANNs) for cases where an input should be replicated with high accuracy. As further steps, ad-hoc combination of those basic building blocks can also be made for addressing specific problems, opening to a very large variety of Neural Network layout.

The application areas of timeseries forecast and outlier detection are supported by MLF through a series of predefined models based on the network layouts introduced above.

3.2.1 Outlier Detection Models

Neural Networks can be effectively applied for outlier detection use cases because of their flexibility and problem agnostic characteristics for setup and training, as well as for their capacity to scale with problem size and limited training data requirements, if compared with classical ML algorithms. They are also proven to be more accurate for non-linear problems, where the Neural Network input-output learning characteristics provides a step-up in the prediction accuracy.

Their application to outlier detection cases can be realised with different layouts, depending on the application domain dimensionality, but the most common ones are ANNs and RNNs, specifically Long-Short Term Memory (LSTM) networks. The way those outlier models work is similar and based on the calculation of the distance of a sample from the centroid of the training distribution using a certain metric and then comparing this distance against a threshold for determining if the sample is a “regular” or an outlier datapoint. The choice of the distance metric and the way the thresholds are defined can be done in different ways, and in MLF this is performed using a statistical metric (Mahlanobis distance), proven to be accurate for normal or normal-like distribution datasets. The thresholds are also derived from the training dataset by applying a regular-outlier datapoints split considering outlier the samples with a standard deviation above 3-sigma (i.e. 0.03% of the total in a normal distribution). The advantages of such approach are the parametric calculation of the thresholds and the capability to scale with problem dimensionality, making model setup very straightforward. The disadvantage is a statistical behaviour, with an intrinsic residual inaccuracy in the prediction, which can be anyway kept small (< 1%) by a proper model training. The model result is a dataset of normalised distance (mean absolute error usually) samples, which provides the timeseries distance trend and it can be compared with the detection threshold for identifying outlier datapoints.

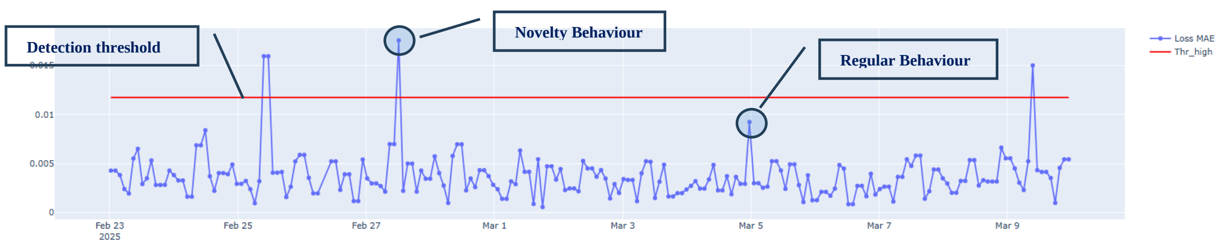

Fig. 4. MLF Outlier Detection Model Result Example

MLF API library implements both types of Neural Networks with different predefined layout depths and parametrised with the dataset dimensionality, so to provide a handy “no-code-development” support for AI/ML relevant applications setup. The hyperparameter considered for their setup and tuning are type, number and size of encoding and decoding layers, and size of latent representation layer. By playing with them different layout depths and input/output sizes can be configured, for adapting to the specific dataset dimensionality to address.

3.2.2 Timeseries Forecast Models

This case is focused on the forecast of the trend evolution of a certain parameter, based on its time history. A basic aspect to consider for time series prediction is that the forecast can work only on non-stochastic data, i.e. on data showing a quasi-regular and continuous trend. Noise effects on the parameter cannot be predicted. This can be seen as a limitation but is not in all the cases where the standard deviation of the parameter is limited (i.e. the “noise” level is limited too). Neural Network models generally used for forecast are RNNs based ones, or hybrid layouts mixing RNNs and CNNs. Both of them are supported in MLF API library.

An important concept of Timeseries Forecast models in general and applied also in MLF is the approach for using dataset samples for the forecast, and specifically the sliding window concept. This approach consists in using a moving subset of datapoints at each training or prediction step for actual datapoint forecast. Two different sliding windows are used: a “look-back” window identifying how many previous points are used for the forecast of a new sample, and the “forecast-window” defining how many datapoints in the future are predicted. Their setup of those two parameters is a crucial aspect, together in the identification of the proper training set and in tuning the model hyper-parameters, for achieving high prediction accuracy.

This type of model is generally quite large in terms of parameters (several thousands) due to the dense type of networks used, and meaning a wide-enough training dataset has to be used. The model output is a plain timeseries with length given by the forecast-window parameter, reconstructed by collecting the relevant predicted samples, and no further conversion is needed.

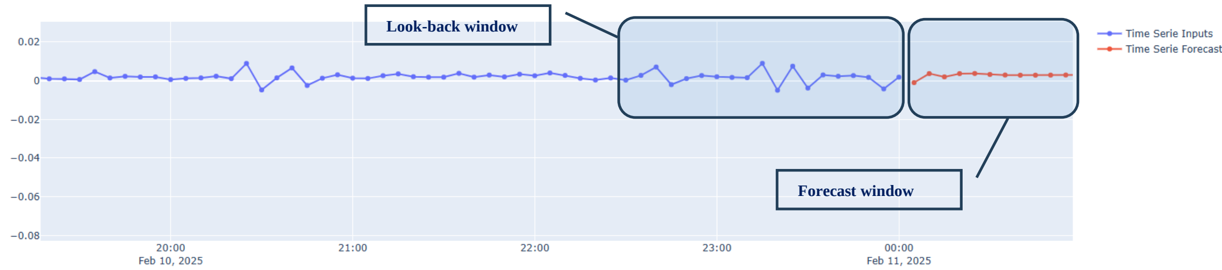

Fig. 5. MLF Timeseries Forecast Model Result Example

3.3 *Large Language Models*

Given the increasing global interest in Large Language Models (LLMs), EUMETSAT is exploring their application in operational contexts to enhance internal user support, facilitate efficient navigation of data services, and streamline the review of organizational documentation. Several internal teams are leveraging the ML Framework's LLM capabilities to develop AI/ML-powered chatbots, providing an intuitive and interactive tool for satellite product and documentation navigation.

One such use case, the EUMETSAT Data Store Access Assistant, utilizes a RAG-based LLM chatbot trained on product and user documentation for private knowledge base queries. The LLM is the core of this application, generating responses based on conversation history, user input, and contextual prompts. A user query is converted into a vector representation and matched with the vector database, with relevant information retrieved and augmented in context. Once trained on the necessary documentation, the chatbot can effectively answer user questions based on the provided text. The LLM's ability to comprehend and generate human-like responses makes it a versatile and valuable tool for navigating the EUMETSAT Data Store.

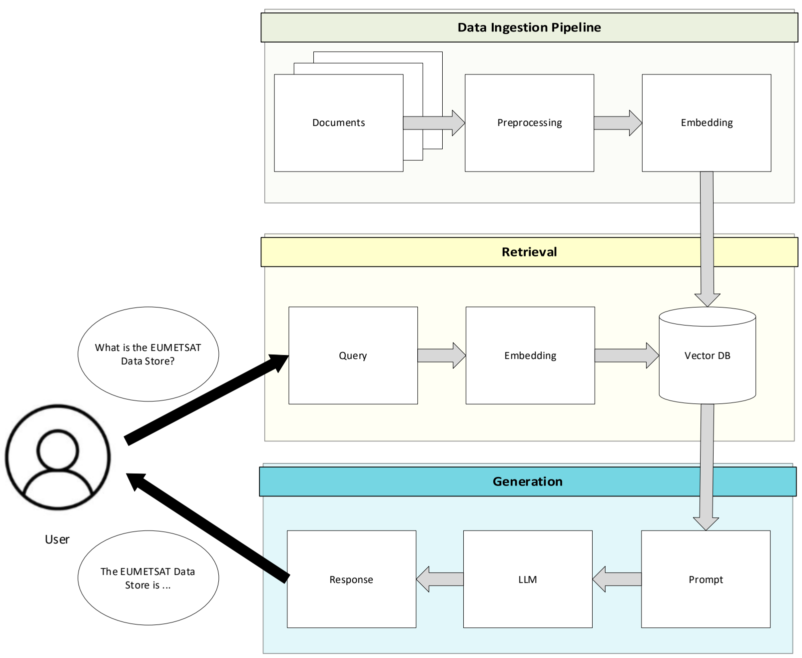

Fig. 6. RAG-based LLM Architecture for EUMETSAT Data Store Chatbot

To ensure data integrity and security, EUMETSAT utilizes Ollama, which enables models to run locally on the organization's secure infrastructure. This approach ensures compliance with security policies while facilitating seamless integration across multiple LLM-based applications hosted on the ML Framework, optimizing resource allocation and simplifying deployment.

Additional use cases leveraging LLM capabilities in the ML Framework include a Sentinel-3 Flight Operations chatbot, designed to assist engineers in navigating complex operational guides and procedures, and a Data Store "Text-to-Query" Access Assistant, which simplifies user interactions with the EUMETSAT Data Store by allowing natural language queries to be translated into structured search requests provided as hyperlinks for quick access to data and products.

To power these applications, the underlying ML techniques must be carefully designed to ensure accuracy and efficiency. The chatbot use cases primarily rely on a Retrieval-Augmented Generation (RAG) architecture, where a Large Language Model (LLM) is trained on product and user documentation to enable private knowledge base queries. To ensure high-quality responses, the text extracted from the documentation must be pre-processed before being used in the RAG application.

Raw sections extracted from the documentation vary significantly in structure and relevance. Directly inserting this text into the application introduces noise and unwanted context, which can degrade model performance. Since LLMs have a maximum context length, relevant material may not fit within the model's processing capacity. To address this, the text is first divided into smaller chunks that encapsulate distinct contexts, reducing noise and improving model efficiency.

Once the text is segmented into manageable chunks, an embedding model is used to transform each chunk into a vector representation. This step is crucial as it captures the semantic meaning of the information and encodes it into a numerical format suitable for querying. Various pre-trained embedding models are available, trained on extensive documentation datasets.

The embedded data is then indexed in a vector database for rapid retrieval. This allows the LLM to reference documentation directly when queried about its content. One of the most widely used open-source vector databases is Chroma DB, which enables efficient similarity search.

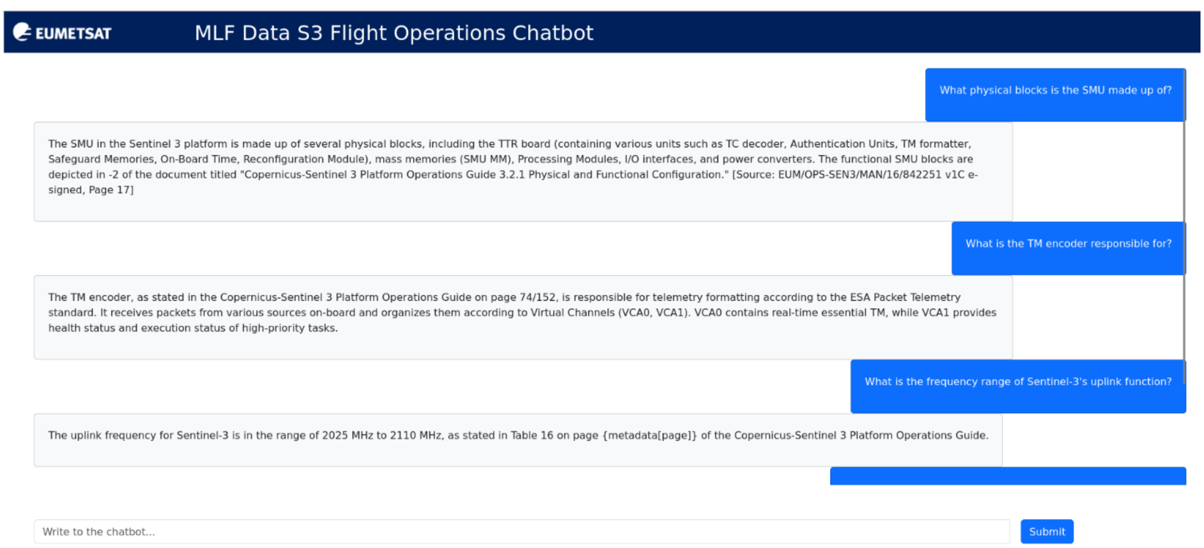

Fig. 7 WebUI for the S3 Flight Operations Chatbot

The primary advantage of RAG-based LLMs is their ability to combine generative AI with real-time, domain-specific knowledge retrieval, ensuring responses remain accurate and grounded in authoritative data sources. This mitigates the risk of hallucination—a common issue with LLMs where responses may be plausible but factually incorrect.

Despite these benefits, RAG-based LLMs also present certain challenges. Maintaining an up-to-date knowledge base requires continuous ingestion and processing of new documentation, which can be resource intensive. Additionally, the effectiveness of retrieval is highly dependent on the quality of the embedding model and the granularity of text chunking, as poorly structured data may lead to irrelevant or incomplete responses. Performance optimization also involves balancing retrieval precision with model response latency, ensuring that queries are processed efficiently without sacrificing accuracy. Finally, LLMs are prone to hallucination—producing seemingly confident but inaccurate responses—making external validation mechanisms essential. As a result, ensuring secure and responsible AI deployment within an organization requires ongoing monitoring, fine-tuning, and domain adaptation.

By leveraging LLMs within EUMETSAT's ML Framework, internal teams can benefit from scalable AI-driven solutions while addressing challenges through controlled deployment, integration with secure infrastructure, and continuous improvements to knowledge retrieval mechanisms.

Future evolutions of LLM-based use cases at EUMETSAT include Agentic AI, with chatbots transitioning from passive responders to proactive agents capable of autonomously performing complex tasks. Additionally, the use of Generative AI (GenAI) could enhance document summarization, automated report generation, and interactive data exploration, allowing users to engage with EUMETSAT's data services and documentation in a more intuitive fashion. These evolutions will require integrating reinforcement learning, fine-tuned models, and real-time contextual adaptation to ensure reliability and accuracy in mission-critical applications.

3.4 Object Detection

Object detection is a task in computer vision that involves identifying and localizing objects within an image, providing the class label and the precise location of objects via the usage of bounding boxes. Its application is being explored by EUMETSAT for assessing the possibility to use it for enriching satellite data with information for making data containing remarkable features of interest easier to discover.

In this context, the ML Framework's capabilities are being used to develop an application for detecting features of interest inside satellite images, with the first prototype focusing on the detection of tropical storms in images available on EUMETVIEW, but with the idea of further extending it to be able to detect other kinds of meteorological events in the future.

For performing the task, this application relies on YOLO (You Only Look Once), a state-of-the-art object detection model known for its speed and accuracy. To develop the storm detection model, a systematic approach involving dataset preparation, training, evaluation, and fine-tuning has been followed.

First, a dataset of satellite images annotated with labelled storm regions has been curated. These labels serve as ground truth for training the model. Instead of training YOLO from scratch, transfer learning has been leveraged, a technique where a pre-trained model (typically trained on a large dataset like COCO) is fine-tuned on a smaller, domain-specific dataset. This significantly reduced training time and improved model performance, as the pre-trained model has already learned general object features.

The model has been trained using the labelled satellite images, optimizing the detection accuracy through techniques such as data augmentation and hyperparameter tuning.

The model's performance has been assessed using mean Average Precision (mAP), a standard metric for object detection. mAP calculates the average precision across multiple intersection-over-union (IoU) thresholds, ensuring that both localization accuracy and classification performance are considered.

Based on the evaluation results, the model has been iteratively refined by adjusting hyperparameters, augmenting the dataset, and improving label quality to enhance detection performance.

Once the model is trained on the domain-specific dataset and fine-tuned, it is able to effectively detect tropical storms inside the satellite images that is fed with.

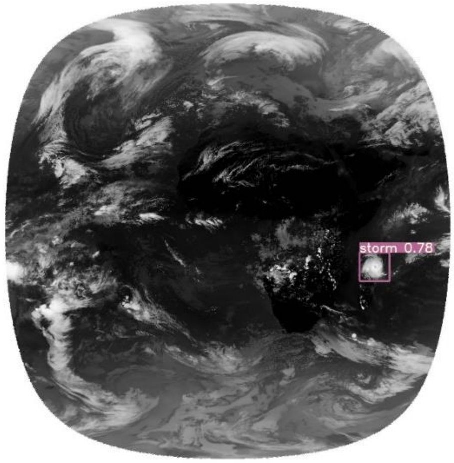

Fig. 8. MLF Object Detection Model Result Example

The application can run periodically in an automated way, directly retrieving images from suitable EUMETVIEW layers and feeding them to YOLO. The model then provides to the users the labelled images and, optionally, text files containing the related labels.

Users can benefit from the potential of an AI-driven object detection in meteorology, which provides a scalable and efficient solution for identifying remarkable meteorological events in satellite imagery.

4. Conclusions

EUMETSAT developed the MLF system for providing internal users with AI as service capabilities. This has been running for the past 3 years, supporting internal proof-of-concept use cases and allowing internal teams to identify, setup and validate AI/ML applications to assess their potential for enhancing operational activities.

The MLF architecture is scalable, flexible and open, enabling support for existing models as well as adapting to new technologies. Additionally, it provides a safe environment for AI/ML applications execution with respect to EUMETSAT data and intellectual rights protection regulatory. Production execution MLOps Level 1 automatisms are implemented with the plan to move to MLOps Level 2 with the MLF upgrade to ESA AInabler.

The supported AI/ML applications cover a large spectrum of models, spanning from timeseries analysis to large language models to object detection, with further extensions planned, and involving applications in the areas of satellite flight operations, ground stations monitoring, system operations monitoring, and user access services. Further extensions to other areas, e.g. science and system engineering, covering new AI technologies as Agents and Generative AI are planned in the short-term future.

The presented ML Framework is a unique resource in EUMETSAT for supporting large scale internal AI/ML uses cases proof-of-concept, with an increasing number of teams interested in making use of its services. Current phase will perform a platform upgrade reaching a high level of automatism for use case production-level execution. As next step, the ML Framework transition to a permanent AI-as-a-Service platform in EUMETSAT will be considered, for keeping the current proof-of-concept support role but also extending the provided services to operational use of AI/ML applications.

Acknowledgements

The authors would acknowledge all the EUMETSAT teams who have shown interest in running their use cases in MLF, specifically the Sentinel-3 flight control team, the MTG System team, the System Operation Mission Performance team, the Big Data services team and the Mission Control Applications and Tools team. They have interacted with use regularly and exchanged very useful ideas and findings, helping MLF team to make important progresses on the development of the platform as well as of the AM/ML applications running on it. Finally, a special thanks in all the EUMETSAT teams who have supported the MLF setup and extensions at infrastructure and data access level, namely Computer & Storage Management and Data Processing Systems & Applications teams.

References

[1] Space CODEV Platform, AInabler, https://ainabler.space-codev.org/

[2] E. Trollope et al, Enhancing Machine Learning Analysis of Satellite HKTM through the Application of Domain Specific Knowledge, SpaceOps 2025, ID #408

[3] V. Sierra Urena, N. Pilloni Leveraging Artificial Intelligence and Automatization to Improve Service and System Monitoring at EUMETSAT, SpaceOps 2025, ID #416