Artificial Intelligence for FAULT prediction

Complex System Design

In collaboration with:

LOCCIONI

A.Y. 2021/2022

Team

Damiano Cacchiarelli
damiano.cacchiarelli@studenti.unicam.it

Matteo Romagnoli
matteo02.romagnoli@studenti.unicam.it

Roberto Cesetti
roberto.cesetti@studenti.unicam.it

Supervisors

Prof. Andrea Polini
andrea.polini@unicam.it

Prof. Flavio Corradini
flavio.corradini@unicam.it

Luca Mazzuferi
l.mazzuferi@loccioni.com

Matteo Calisti
m.calisti@loccioni.com

Contents

Loccioni

Loccioni develops measurements and control systems to help manufacturers perform at their best, saving time and money while respecting human safety and the natural environment.

Application Domain

General Business Intelligent schema

Database

Storage of collected data

Rules & models

Use rules and templates to create data statistics

Objective

Improve the quality of the supplied product

IoT Sensors

Data production from IoT sensors, alarms and industrial machines

Analytical tools & ML

Data analysis and transformation using Machine Learning tools

Action

Data-driven actions performed by the company

Objective

A system that allows the company to integrate Machine Learning Algorithms, for each industrial machine used, in a simple way

Main task

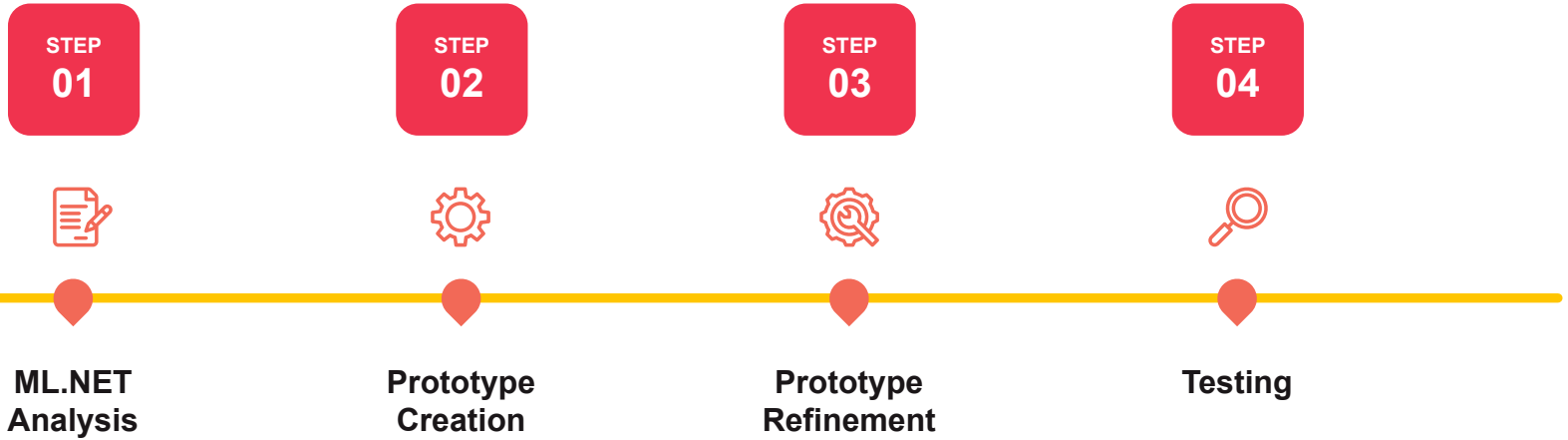
Pluggable ML algorithms

Language

Using .NET core and C# language

WORK PLANE

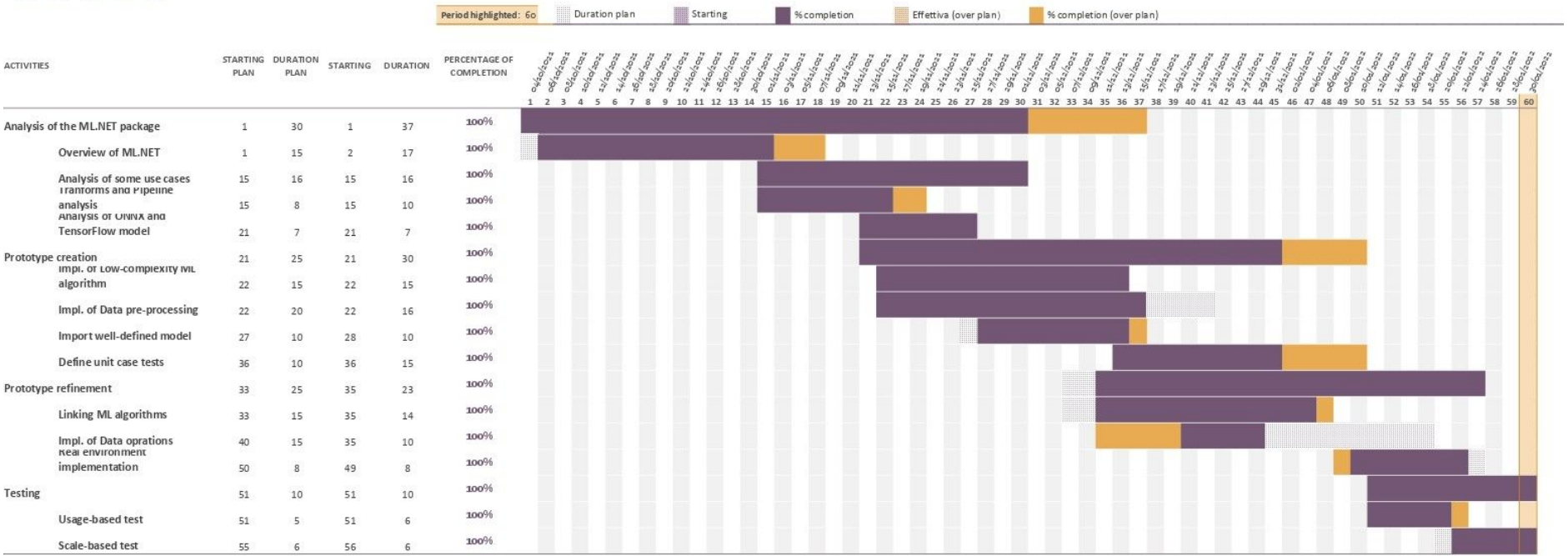
Development activities



Workflow

GANTT Diagram

Workflow



ACTIVITIES	STARTING PLAN	DURATION PLAN	STARTING	DURATION	PERCENTAGE OF COMPLETION
Analysis of the ML.NET package	1	30	1	37	100%
Overview of ML.NET	1	15	2	17	100%
Analysis of some use cases	15	16	15	16	100%
Transforms and Pipeline analysis	15	8	15	10	100%
Analysis of ONNX and TensorFlow model	21	7	21	7	100%
Prototype creation	21	25	21	30	100%
Impl. of Low-complexity ML algorithm	22	15	22	15	100%
Impl. of Data pre-processing	22	20	22	16	100%
Import well-defined model	27	10	28	10	100%
Define unit case tests	36	10	36	15	100%
Prototype refinement	33	25	35	23	100%
Linking ML algorithms	33	15	35	14	100%
Impl. of Data oprations	40	15	35	10	100%
Real environment implementation	50	8	49	8	100%
Testing	51	10	51	10	100%
Usage-based test	51	5	51	6	100%
Scale-based test	55	6	56	6	100%

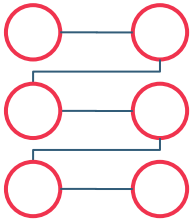
ARCHITECTURE

The architecture of the system has the following components.

In this way, the architecture provides some constraint about input and output at each stage of the pipeline.

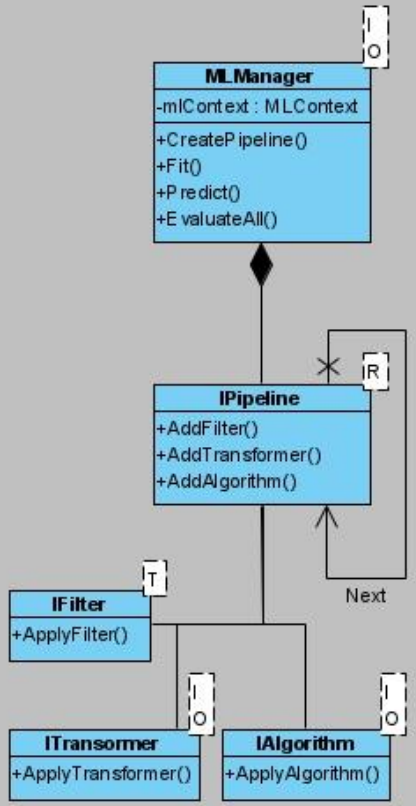
MLManager

- Create ML pipelines
- Train the model
- Evaluate the model
- Predict a value



Pipeline

- It represents a sequence of filters, transformations and ML algorithms
- Append filters, transformations and ML algorithms



ARCHITECTURE

The pipeline is the result of the concatenation of this 3 operations.

Pipeline

Filter

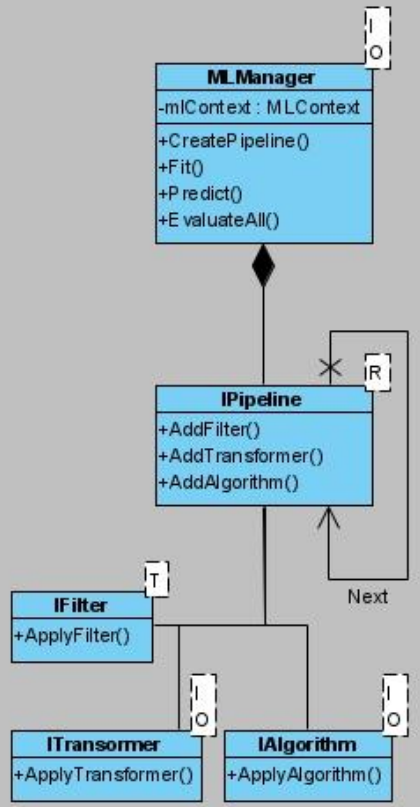
- Interface for data filters
- We are able to implement any kind of filter

Transformer

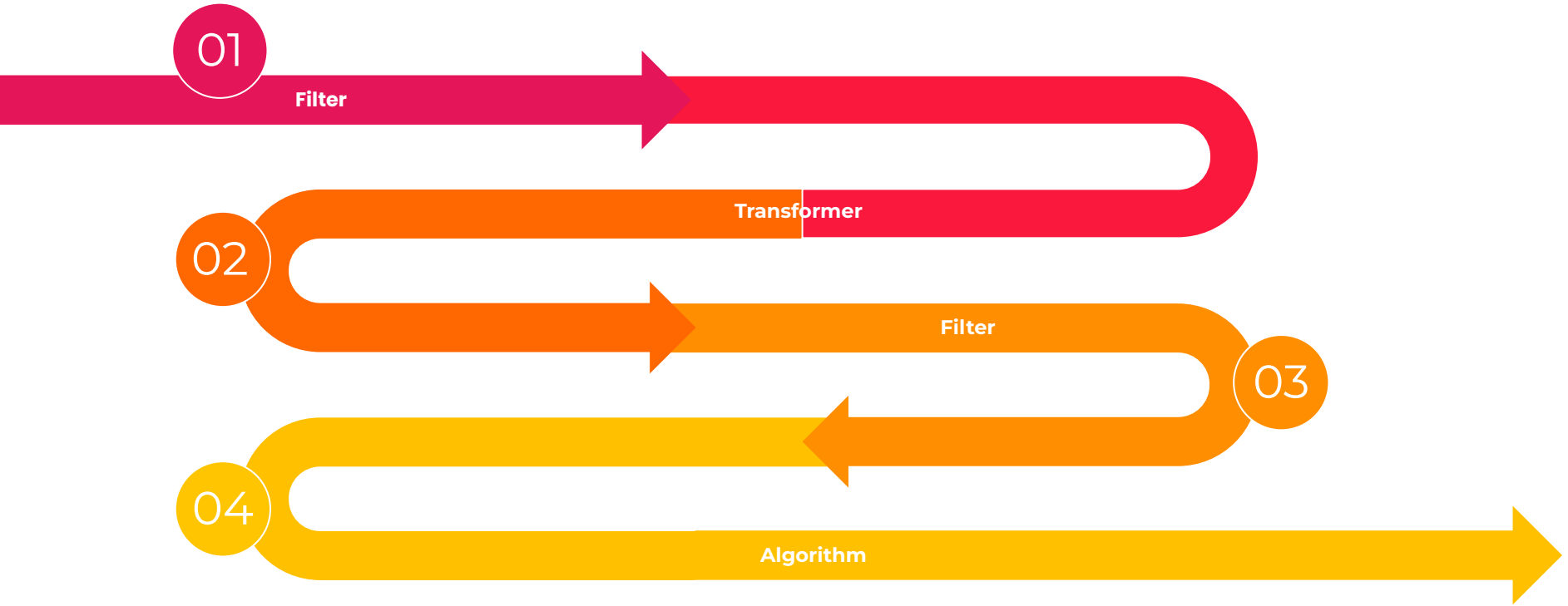
- Input Data transformation
- Implemented using ML.NET Trasformers

Algorithm

- ML or specific algorithm
- ONNX / Tensorflow pre-trained model

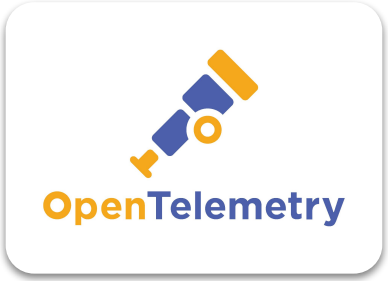


Pipeline constraints



Latest updates

REST API services



Open Telemetry

OpenTelemetry integration

01

Application

The system integrate the OpenTelemetry tracer and exporter

02

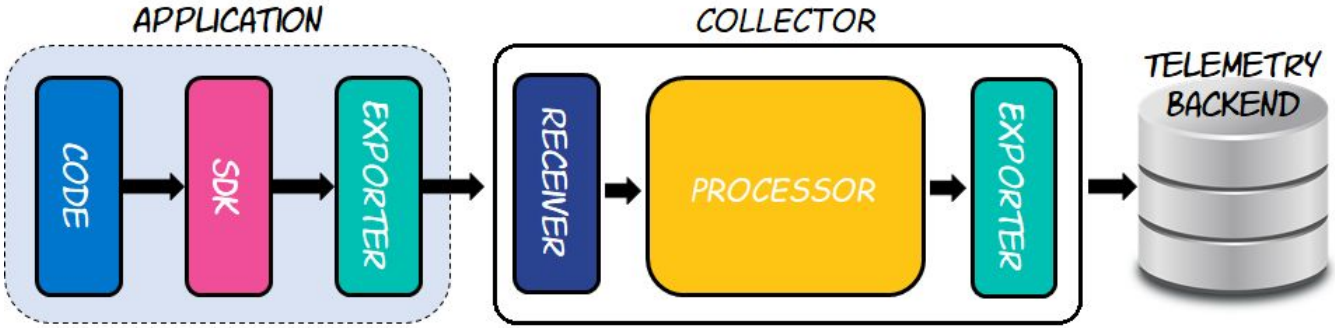
OT Collector

It is an external component that receive, process and export the received telemetries

03

Backend

It is a system that allow to store and analyze the telemetries



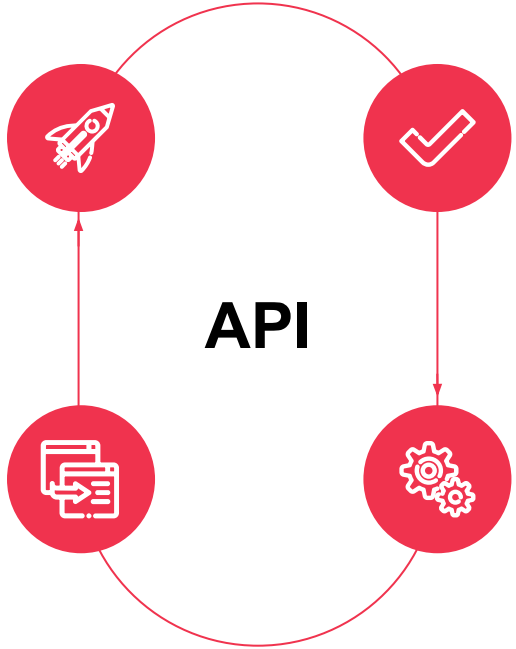
Flow of the program in REST API mode

The client send a request to the service, using a REST call.

CR
Client Request

SA
Server Accept

The server, if is available, accept the request and instantiate all the resources.



Once the server perform the computation, it return back the response.

SR
Server Response

SE
Server Execution

The server will process the request and perform some computation.

Example

Conclusion

Conclusion

The system meets all the requirements that were defined during the meetings with the development team

Future developments

However it could be improved by adding additional features such as:

- Automate the insertion and removal of algorithms
- Generic implementation for the input algorithms
- Perform asynchronous predictions

Thanks !