

# Rapporto sulla progettazione degli algoritmi di Advanced Analytics

<b>Code</b>	O3.1
<b>Date</b>	22/1/2020
<b>Type</b>	Confidential
<b>Participants</b>	UNIMORE
<b>Authors</b>	Simone Calderara (UNIMORE), Francesco Del Buono, (UNIMORE), Francesco Guerra (UNIMORE), Fabio Lanzi (UNIMORE), Matteo Paganelli (UNIMORE), Maurizio Vincini (UNIMORE)
<b>Corresponding Authors</b>	Francesco Guerra

## Sommario

1. Abstract .....	3
2. Scenario industriale di analisi .....	3
3. Algoritmi di Advanced Analytics .....	4
3.1 Tecniche per il preprocessing .....	4
3.2 Tecniche per la diagnostica / prognostica .....	5
4. Risultati preliminari.....	6
4.1 Il Dataset .....	6
4.2 L'analisi preliminare .....	6
5. Algoritmi di Nowcasting.....	11
5.1 Acquisizione e Annotazione dei Dati.....	12
5.2 Progettazione e Allenamento del Feature-Extractor .....	13
5.3 Progettazione dei Modelli di Nowcasting .....	14
6. Bibliografia .....	16

## 1. Abstract

Questo documento vuole fornire una panoramica circa i progressi ottenuti nella prima fase di sviluppo del progetto SBDIOI4.0 relativamente alla realizzazione di algoritmi di advanced analytics sui dati messi a disposizione dalle aziende partecipanti al progetto stesso, relativi a parametri rilevati durante il funzionamento di macchine ed impianti produttivi.

In prima istanza si fornirà una descrizione dello scenario industriale di analisi (Sezione 2). In Sezione 3 verranno presentate le principali tecniche per l'elaborazione di dati industriali al fine di 1) ottenere informazioni di più alta qualità e 2) realizzare successive analisi di diagnostica e prognostica del comportamento dei macchinari analizzati. In Sezione 4 si presentano le attività finora condotte correlate con una serie di risultati preliminari. Infine, in Sezione 5 vengono presentate le principali tecniche di nowcasting.

## 2. Scenario industriale di analisi

Lo scenario considerato è quello relativo a macchinari industriali per i quali si vuole monitorare lo "stato di salute" nel tempo. Si tratta in particolare di analizzare il processo di degradazione di alcune componenti al fine di identificare e cercare di predire comportamenti anomali dei macchinari precedenti uno stato di guasto.

I macchinari possono operare con diverse modalità di funzionamento attivabili tramite il settaggio di opportune configurazioni di parametri di input (i.e. configurazioni). In virtù di ciò, l'analisi di comportamenti inattesi si traduce anche nel riconoscimento in maniera automatica delle varie modalità di funzionamento, il quale permette di escludere casistiche di errata interpretazione di condizioni anomale. In tale contesto individuare le grandezze fisiche più significative rispetto lo specifico settaggio del macchinario ha rappresentato fin da subito un task altrettanto prioritario.

In maniera più dettagliata, lo scenario analizzato richiede la risoluzione dei seguenti problemi:

- **Omogeneizzazione dati ed eliminazione di outliers:** i dati industriali e quindi anche quelli a disposizione tipicamente provengono da più sorgenti e si riferiscono ad intervalli temporali che possono anche non essere coincidenti. Si rende necessaria un'attività preliminare di integrazione e manipolazione dei dati al fine di creare un dataset unico ed omogeneo, che contenga tutte le informazioni (variabili) prelevate per il componente in un determinato periodo di tempo e con dati campionati alla stessa frequenza. Sono inoltre presenti sia picchi anomali (e.g., velocità nulla della macchina) che dati riferiti alla fase di avviamento della macchina che sarebbe opportuno rimuovere, o perlomeno tenere in considerazione per evitare errori nella fase di analisi delle prestazioni della macchina.
- **Feature extraction:** i segnali grezzi (i.e. provenienti dai sensori) spesso non sono in grado di fornire informazioni interessanti alla valutazione e classificazione del funzionamento reale sul campo e dei guasti. Si rende necessaria l'individuazione della rappresentazione più adeguata dei dati raccolti in funzione del task che si vuole realizzare. Questo vuol dire sia individuare le variabili di analisi più significative, sia aggregare i rilevamenti in gruppi omogenei.
- **Feature selection:** di norma, le variabili dei dataset sono molte e probabilmente ridondanti. È opportuno dunque fare una selezione di queste variabili, in modo da individuare delle variabili (o loro combinazioni/proiezioni) che siano in grado di:
  - Distinguere diversi settaggi (configurazioni) della macchina, si veda Figura 2 nella quale in corrispondenza dei diversi settaggi sono evidenziati diversi valori per la variabile analizzata;

- Mostrare il lento processo di usura verificatosi indipendentemente dalla ricetta implementata, come rappresentato in Figura 2.

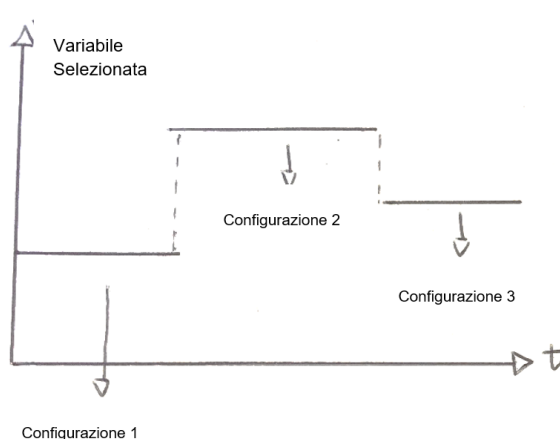


Figura 2: Individuazione settaggi

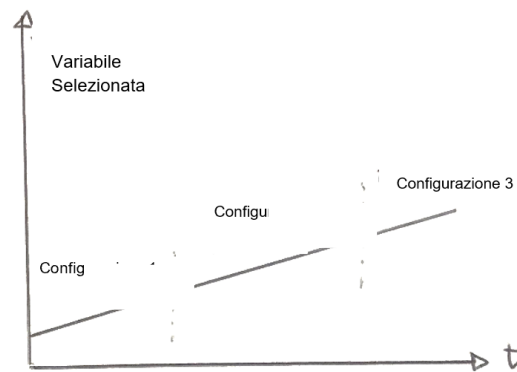


Figura 1: Usura del singolo componente

### 3. Algoritmi di Advanced Analytics

In questa sezione vengono introdotte le principali tecniche di elaborazione preliminare di dati industriali grezzi e di successiva analisi per monitorare lo “stato di salute” di macchinari (diagnostica), nonché la previsione di comportamenti futuri (prognostica).

#### 3.1 Tecniche per il preprocessing

La generazione di dati industriali di alta qualità rappresenta una premessa necessaria per poter estrarre conoscenze a supporto di un’industria intelligente. Sfortunatamente l’ambiente industriale, come quello analizzato, è intrinsecamente soggetto a fattori “disturbanti” che rendono tale processo un’attività sfidante e critica.

**Valori mancanti.** I dati possono contenere un’elevata percentuale di valori mancanti, i quali possono manifestarsi come singoli valori o come intervalli temporali più estesi. I sensori dei macchinari possono fallire nella registrazione di valori o possono esserci dei guasti nell’attività di trasmissione degli stessi verso lo storage di memorizzazione.

**Disallineamento temporale.** I dati derivano solitamente da molteplici sensori i quali possono raccogliere le informazioni con tempi di campionamento diversi. Ciò produce un disallineamento delle informazioni nel tempo e rende tali dati meno comparabili.

**Valori anomali.** Problemi inaspettati nella registrazione di dati da parte di un sensore possono produrre la memorizzazione di dati non coerenti. Analogamente, guasti inaspettati sui macchinari possono produrre significative variazioni di certe grandezze fisiche.

La risoluzione di queste tipologie di problematiche compete all'attività di pre-processing, la quale si articola in una serie di sotto-attività quali:

- Aggregazione di molteplici sorgenti dati in un'unica sorgente omogenea
- Analisi statistica preliminare dei dati (distribuzione dei valori, percentuale di valori mancanti, frequenza di campionamento, correlazione tra variabili, ecc...)
- Riempimento di valori mancanti (mediante media, moda, mediana o proporzionale alla distribuzione dei valori esistenti)
- Allineamento temporale (compattazione dei dati sull'asse temporale o interpolazione dei dati mancanti)
- Normalizzazione delle feature (min-max normalization, mean-std normalization o z-score...)
- Identificazione e rimozione di outlier (ovvero valori anomali)
- Feature extraction, ovvero identificazione della rappresentazione "migliore" dei dati in funzione del task che si vuole realizzare
  - Frequency domain (Trasformata di Fourier, FFT), time-frequency domain, wavelet domain, EMD
- Feature selection, ovvero identificazione delle feature più significative (PCA, SVD)

### 3.2 Tecniche per la diagnostica / prognostica

In questa sezione viene presentato lo stato dell'arte circa le attività di diagnostica e prognostica su macchinari industriali.

Con il termine **diagnostica**, si intende quell'insieme di tecniche volte all'identificazione dello "stato di salute" di un macchinario, mentre lo scopo della **prognostica** è quello di cercare di predire l'andamento futuro della macchina.

L'applicazione congiunta delle due attività è identificata in letteratura con il termine Prognostics and health management (PHM), il cui obiettivo è quello di massimizzare il tempo di funzionamento di una risorsa industriale riducendone i costi di manutenzione e migliorandone l'affidabilità e la sicurezza.

Il tempo stimato prima di un guasto del componente in esame è conosciuto con il termine Remaining Useful Life (RUL).

In letteratura esistono principalmente due categorie di approcci per la risoluzione di questi problemi:

- approcci basati su modelli multifisici
- approcci data-driven

I primi sfruttando un modello multifisico che riproduce il comportamento del macchinario considerato per realizzare delle predizioni sui suoi comportamenti futuri. Gli approcci appartenenti a questa categoria risultano essere estremamente affidabili, tuttavia il loro principale difetto risiede nella difficoltà (e talvolta incapacità) di modellare il comportamento del sistema reale mediante un'opportuna legge matematica.

Gli approcci data-driven invece prevedono un'attività di monitoraggio del funzionamento della risorsa in esame e di successiva analisi mediante tecniche di machine learning dei dati raccolti. Sebbene questa categoria di tecniche richieda la collezione di una sufficiente quantità di dati significativi, presenta il vantaggio di non

dipendere da una conoscenza a priori del dominio e di poter estendere le conoscenze acquisite ad altri domini applicativi.

Concentrandosi in particolare su quest'ultima categoria, molteplici sforzi sono stati compiuti negli ultimi anni con buoni risultati nell'attività di prognostica [Andrew K.S. Jardine 2005]. Tra i molteplici approcci proposti è possibile distinguere quelli che utilizzano tradizionali tecniche di machine learning, come modelli di support vector machine (SVM) [T. Benkedjough 2013], logiche Bayesiane [Jianbo Yu 2015] o Markoviane [Ming Dong 2007, Abdenour Soualhi 2013] e quelli che sfruttano le più recenti tecniche di deep learning tramite reti convolutive [Wu Chunzhi 2018, Li Xiang 2018] o ricorrenti [Shuyang Yu 2019].

## 4. Risultati preliminari

In questa sezione vengono discussi alcuni risultati preliminari relativi all'attività di pre-processing dei dati industriali analizzati. In particolare, verranno presentate le strategie adottate finora per 1) la pulizia dei dati, 2) l'individuazione dei settaggi unici dei macchinari e 3) il riconoscimento delle fasi di avviamento dei macchinari.

### 4.1 Il Dataset

Il dataset utilizzato si compone di 4 archivi descritti in Tabella 1. Il dataset4 contiene i setting della macchina, mentre i rimanenti memorizzano valori delle variabili del macchinario.

SORGENTE DATI	FREQUENZA DI CAMPIONAMENTO	DATA INIZIO	DATA FINE	NUMERO VARIABILI
dataset1	1 volta al giorno	13/02/2017 14:13:02	08/07/2019 00:00	187
dataset2	1 Hz	20/10/2017 10:03:11	17/09/2018 17:09:06	26
dataset3	10 Hz	2017-10-20 12:01:43.6	2018-10-02 18:03:27.2	8
dataset4	1 o più volte al giorno	10/10/2017 17:40:57	19/06/2019 10:13:52	36

*Tabella 1: Descrizione dataset*

### 4.2 L'analisi preliminare

#### 4.2.1 Identificazione delle configurazioni di funzionamento dei macchinari

Di seguito si fornisce una descrizione dell'attività svolta per identificare le configurazioni uniche dei macchinari per impostare diverse modalità di funzionamento.

L'attività di analisi considera come input il dataset 4 nel quale, come indicato in sezione 4.1, sono memorizzate le diverse configurazioni che sono state applicate ai macchinari nel corso del loro utilizzo. Ogni configurazione

viene identificata da un riferimento temporale circa l'istante di tempo nel quale tale configurazione è stata applicata al macchinario, ed una serie di valori associati a specifici parametri del macchinario.

L'obiettivo di questa analisi è quella di individuare all'interno di questo dataset le configurazioni uniche con le quali si possono attivare le diverse modalità di funzionamento. Si tratta in particolare di riconoscere all'interno della raccolta di settaggi quelli duplicati e successivamente di rimuoverli.

Sebbene il task da realizzare possa apparire semplice, i dataset industriali e quindi anche quello in esame presentano delle caratteristiche che rendono la sua analisi non agevole.

Il primo fattore "disturbante" riguarda il **formato dei dati**. Il dataset originale descrive un settaggio tramite una serie di righe ciascuna delle quali aventi il seguente formato: <istante di tempo>, <parametro macchinario> <valore parametro>. Naturalmente il campo relativo all'istante di tempo deve essere uguale per tutti i parametri associati alla configurazione in esame.

In secondo luogo, il dataset contiene un'**elevata percentuale di valori mancanti**. Circa il 75% dei valori sono inespressi.

Il dataset inoltre registra le informazioni dei settaggi in maniera irregolare: **il processo di memorizzazione dei settaggi non rispetta una costante cadenza temporale di registrazione dei dati**. Come conseguenza di questo aspetto, all'interno dell'arco temporale di copertura del dataset possono esistere giorni nei quali non è stato registrato alcun settaggio.

Infine, il dataset contiene delle **informazioni contraddittorie rispetto a quanto specificato dagli esperti del dominio** esaminato. In maniera più dettagliata, il dataset registra molteplici cambiamenti di parametri in corrispondenza dello stesso giorno, sebbene ci fosse stata confermata l'applicazione di un solo settaggio al giorno. Un altro fattore fonte di dubbi, risolto tramite esperti del dominio, riguardava la modalità di transizione di stato tra due settaggi applicati in giorni successivi. Non era chiaro se vi fosse una dipendenza tra il settaggio applicato un certo giorno rispetto al settaggio applicato il giorno successivo. A tal proposito ci è stato indicato che tra un settaggio e l'altro il macchinario veniva riavviato, permettendoci quindi di escludere alcuna forma di dipendenza tra settaggi successivi.

In virtù delle problematiche appena evidenziate, l'approccio per l'individuazione dei settaggi unici ha impiegato una strategia risolutiva ad-hoc, la quale si articola nelle seguenti fasi:

1. **Conversione del formato dei dati**. Il dataset originale è stato trasformato in modo tale da creare una matrice, in cui l'indice di riga rappresenta il timestamp dell'inserimento del settaggio, mentre le colonne rappresentano le variabili/parametri che vengono modificati. Il dataset trasformato coincide con una matrice 388x36.
2. **Rimozione dei dati non significativi**. Sono state eliminate le righe contenenti dei valori mancanti in corrispondenza di tutti i parametri registrati.
3. **Compressione dei dati giornalieri**. Sulla base della considerazione che vuole che in un dato giorno venga applicato una sola configurazione di funzionamento del macchinario, si è dedotto che la presenza di molteplici cambiamenti dei valori dei parametri in un dato giorno in realtà si riconducesse ad un unico cambiamento di stato corrispondente alla congiunta variazione dei diversi parametri considerati. Si è pertanto applicato un approccio di compressione di tali informazioni in un unico dato riassuntivo. La tecnica adottata si compone dei seguenti step:
  - a. Ordinamento sulla base temporale dei settaggi

- b. Creazione di un dato riassuntivo per ogni giorno di registrazione tramite la propagazione dei valori dei parametri a partire dal dato meno recente a quello più recente mediante tecnica di “fill forward”
- c. Sostituzione dei cambiamenti avvenuti giornalmente con il relativo dato di sintesi

A seguito dell’applicazione di questa pipeline di pulizia e compressione dati, è stato generato il dataset “*Settings*” contenente **14 settaggi unici**.

Al fine di validare l’approccio realizzato si è pensato di confrontare i risultati ottenuti (ovvero la conoscenza dell’esatto settaggio applicato in ogni istante temporale) con alcune misurazioni dell’evoluzione degli stessi parametri configurabili mediante l’applicazione di un settaggio. Tale informazione è in particolare disponibile all’interno del dataset “*dataset1*”.

I risultati di questa comparazione sono riassunti di seguito:

- 34 record/timestamp di *Settings* sono uguali in *dataset1*
- 9 record/timestamp sono in conflitto
- 21 record/timestamp di *Settings* non sono presenti in *dataset1* (*Giorni presenti in dataset4 ma non in dataset1*)
- 316 record/timestamp di *dataset1* non sono presenti in *dataset4* (*Giorni presenti in dataset1 ma non in dataset4*)

Se si considera il valore assunto da *dataset4* in un certo giorno confermato fino al successivo cambiamento tra i 316 record, solo 3 record sono in conflitto.

Questo risultato ci convince del fatto che i null values presenti nel file ricette corrispondono a valori non variati per la feature in esame.

#### 4.2.2 Identificazione della fase di avviamento (setup) di un macchinario

Di seguito si presenta il risultato preliminare circa l’analisi e l’individuazione della fase di avviamento di un macchinario. In particolare, la prima fase di questa attività è consistita nella verifica dell’eventuale presenza di una fase di assestamento dei valori delle grandezze fisiche misurate a seguito dell’applicazione sul macchinario di una nuova ricetta.

L’analisi realizzata ha considerato il dataset “*dataset3*” il quale si compone di una serie di file corrispondenti ciascuno alla misurazione di certi parametri nell’arco di una specifica giornata. Di seguito ci riferiremo a tali dati in uno specifico giorno con il termine di chunk.

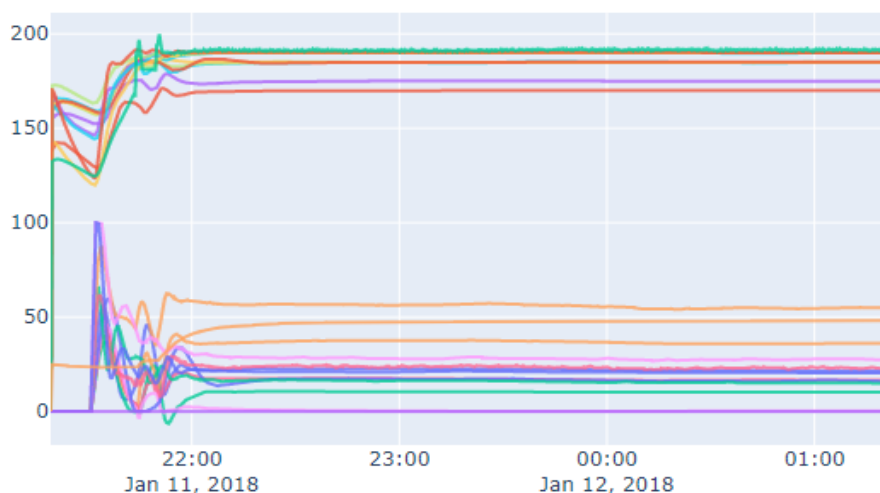
Al fine di realizzare questa prima analisi, è risultato necessario dapprima riuscire ad associare i dati di un chunk rispetto i settaggi applicati sui macchinari (file “*dataset4*”). Banalmente, osservando il tempo di applicazione di un settaggio e la componente temporale associata ad ogni misurazione, è possibile ottenere l’associazione desiderata tra la configurazione dei parametri e le misurazioni associate alla relativa modalità di funzionamento. Questa procedura preliminare ha prodotto dei risultati relativamente inattesi in quanto si è osservato come:

- possono passare anche parecchi giorni tra il settaggio di una configurazione di funzionamento del macchinario e la misurazione delle relative grandezze fisiche
- i chunk contengono misurazioni che si riferiscono solo a 3 su 14 settaggi totali
  - il settaggio 0 compare in 3 chunk



- il settaggio 3 compare in 2 chunk
- il settaggio 4 compare in 65 chunk

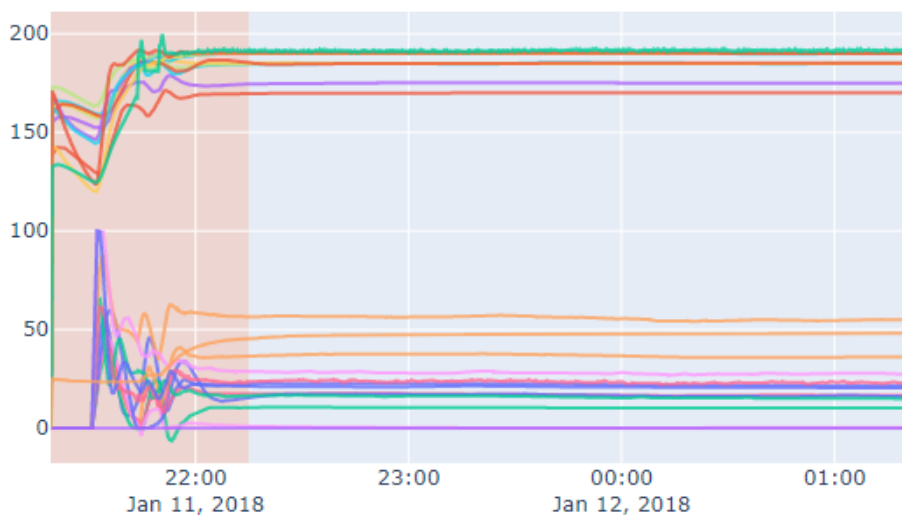
Si è inoltre rilevato, attraverso una analisi dei dati, che solo 9 dei 70 chunk contengono degli andamenti che possono ricondursi ad una fase di avviamento del macchinario. Nei restanti casi non sono presenti in fase iniziale di setting delle fluttuazioni dei valori che fanno pensare a una fase di avviamento e in altri casi non sono presenti rilevazioni sui sensori in corrispondenza di una variazione di setting. Questi 9 chunk si riferiscono unicamente al settaggio 4. In Figura 3 si fornisce una rappresentazione degli andamenti dei parametri che possono far pensare ad una fase di inizializzazione del macchinario.



*Figura 3: Fase di avviamento su diverse feature*

Come si può osservare, a seguito dell'applicazione di un settaggio, il macchinario presenta un comportamento irregolare ed instabile che con il passare del tempo tenderà a stabilizzarsi.

Sulla base di queste osservazioni, l'obiettivo imposto consiste nell'individuazione automatica di tali condizioni di funzionamento iniziali in modo da isolarle rispetto al comportamento "a regime" del macchinario. Una rappresentazione del risultato sperato è fornita in Figura 4.



*Figura 4: Identificazione della fase di avviamento*

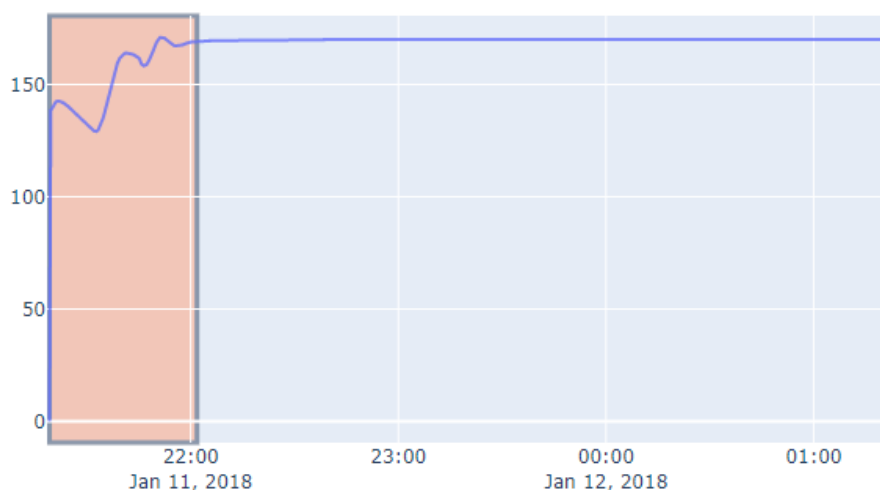
Sono stati sperimentati due approcci: una baseline basata su soglia (one threshold) ed una tecnica di clustering gerarchico.

**One Threshold:** tecnica base che consente di riconoscere variazioni nei dati specificando l'istante temporale dell'inserimento del setting e andando a individuare l'istante di tempo in cui si raggiunge la condizione di equilibrio. L'algoritmo prevede i seguenti step:

1. normalizzazione della serie temporale in ingresso
2. identificazione di una possibile condizione di equilibrio
3. mantenimento della condizione per un numero minimo di campioni consecutivi

L'algoritmo funziona correttamente solo nei casi in cui si deve avere il raggiungimento di una condizione di equilibrio, e sono presenti forti oscillazioni in fase iniziale. Di conseguenza, diventa impreciso in condizioni di avviamento che non subiscono forti variazioni, o in assenza di condizioni di equilibrio. Inoltre, deve essere indicato l'istante di tempo in cui è stato inserito il settaggio.

In Figura 5 si riporta un esempio di applicazione dell'algoritmo su una serie temporale che presenta una fase di avviamento iniziale evidenziata in arancione.



*Figura 5: Identificazione della fase di avviamento mediante One Threshold*

**Clustering.** Tecnica di estrazione di pattern dal comportamento normale del macchinario per l'individuazione di anomalie sulla base dello scostamento. L'algoritmo si divide in due fasi:

1. Training per identificare pattern dal normale funzionamento del macchinario, creando una lista di sequenze rappresentative
  - a. creazione di chunk di lunghezza fissa (con o senza overlap) dalle serie temporali in input
  - b. applicazione di un algoritmo di hierarchical clustering, utilizzando una misura di distanza specifica
  - c. identificazione di chunk rappresentativi del comportamento normale
2. Evaluation per analizzare lo scostamento di qualsiasi serie temporale e segnalando le eventuali anomalie, che nel nostro caso saranno le condizioni di setup
  - a. suddivisione della serie temporale in input in chunk di lunghezza uguale al training set

- b. misuriamo la distanza minima rispetto ai pattern normali per ciascun chunk
- c. ogni chunk in input se ha una distanza/scostamento superiore a un certo valore viene segnalato come anomalia

L'algoritmo impara pattern durante il normale funzionamento del componente, di conseguenza individua le condizioni di avviamento come anomalie (evidenziate in rosso nella Figura 6 Figura 6), senza dover indicare l'inserimento del nuovo settaggio. Inoltre, il modello può essere aggiornato online con nuove condizioni operative imparando nuovi pattern, e individuare outlier in qualsiasi momento.

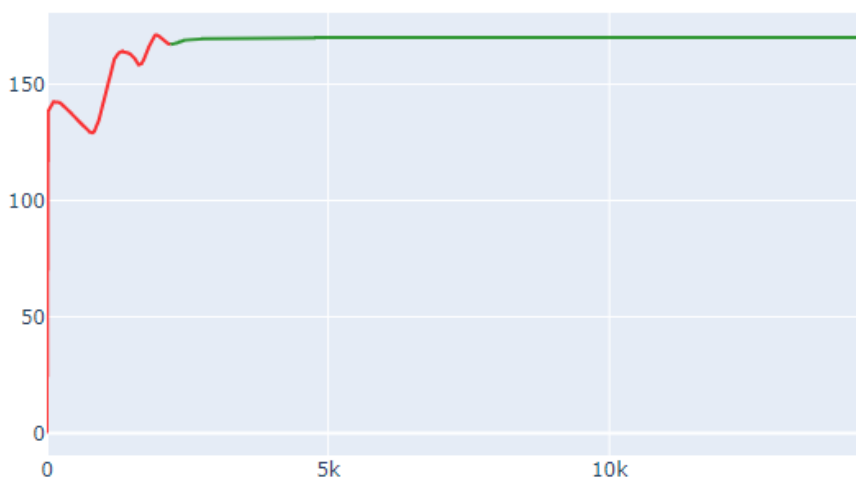


Figura 6: Identificazione della fase di avviamento mediante Clustering

## 5. Algoritmi di Nowcasting

Con il termine “nowcasting” facciamo riferimento all’insieme di tecniche finalizzate alla predizione delle condizioni meteorologiche all’istante di tempo attuale o comunque nell’immediato futuro (generalmente entro un massimo 5/10 minuti) circoscritte a una particolare zona di interesse. Questo concetto si affianca spesso a quello più noto di “forecasting”, che riguarda tuttavia previsioni di accuratezza inferiore, ma relative ad una finestra temporale più ampia, arrivando anche a stime di una settimana in avanti.

Sebbene ad un primo sguardo nowcasting e forecasting possano apparire molto simili tra loro, le finalità di questi strumenti presentano delle sostanziali differenze. Nel caso del nowcasting, infatti, più che fare previsioni su ciò che accadrà a livello meteorologico, il fine è quello di raccogliere precise statistiche relative a uno o più fenomeni di interesse in una zona circoscritta.

Il nowcasting è quindi un importante strumento statistico può fungere da supporto nelle analisi sul clima finalizzate a descrivere andamento, intensità e variabilità di fenomeni meteorologici, osservati a diversa scala temporale. Tuttavia, automatizzare il processo di nowcasting risulta essere un’operazione piuttosto complessa che, se affrontata tramite metodologia standard, richiede l’acquisto di strumentazione sofisticata e dai costi piuttosto elevati; per questo motivo, il nowcasting è spesso prerogativa delle stazioni meteo più attrezzate. L’idea è quindi quella di ridurre il più possibile il numero e la complessità dei sensori necessari al raccoglimento dei dati di nowcasting per rendere questa pratica più accessibile.

Pubblicazioni rilevanti per l’analisi che faremo sono [Qinxue Meng 2017, Kaiming He 2016, Joseph Redmon 2016, Joseph Redmon 2018].

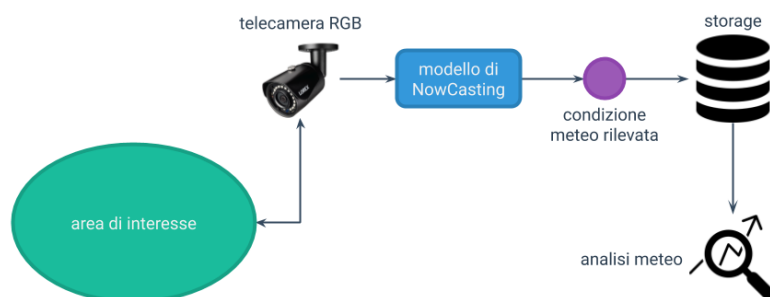


Figura 7: Schema semplificato del sistema di nowcasting proposto

Nel nostro caso specifico, si è scelto di affrontare il problema del nowcasting affidandosi esclusivamente a tecniche di Computer Vision e Deep Learning, limitando la richiesta di sensori a semplici telecamere RGB. Questo tipo di approccio, sebbene conveniente in termini di semplicità d'uso e di risorse impiegate, presenta svariate problematiche ed elementi innovativi che saranno attentamente studiati nell'ambito del presente progetto.

Il Deep Learning è una recente tecnica di apprendimento automatico (Machine Learning), molto utilizzata nel campo della moderna Computer Vision, che permette di realizzare modelli predittivi caratterizzati da prestazioni precedentemente inimmaginabili per i task più disparati. L'allenamento di modelli di questo tipo richiede necessariamente l'uso di un'ingente mole di dati annotati, in quanto l'apprendimento procede tramite l'analisi di una serie di esempi che portano al progressivo miglioramento del modello.

Le attività svolte possono essere suddivise in 3 fasi principali:

1. Acquisizione e annotazione dei dati
2. Progettazione e allenamento del feature extractor
3. Progettazione dei modelli di nowcasting

Di seguito andremo a illustrare più nel dettaglio ciascuna fase.

## 5.1 Acquisizione e Annotazione dei Dati

Per l'acquisizione dei dati, in questa prima fase del progetto, ci siamo rivolti all'Osservatorio di Modena, che ci ha fornito le registrazioni di un arco temporale di 5 anni; a partire dal 2013, infatti, l'osservatorio ha scattato, da telecamera fissa, un'immagine al minuto in modo quasi ininterrotto. Un esempio delle immagini fornite dall'Osservatorio è mostrato in Figura 8.

Si tratta tuttavia di dati grezzi in quanto privi di annotazione relativa alle condizioni meteo che si intendono rilevare. Nel nostro caso, definiamo "dato annotato" coppie del tipo  $\langle I, T \rangle$ , dove  $I$  è l'input del modello (ovvero un'immagine RGB come quelle in Figura 8) e  $T$  è l'output desiderato per quel particolare input (ovvero una label della condizione meteo da rilevare).



*Figura 8: Alcuni esempi di immagini raccolte dalla telecamera dell'Osservatorio di Modena*

Il processo di annotazione dei dati è un'operazione molto onerosa in termini di tempo, soprattutto considerando la grande mole di immagini a nostra disposizione; per questo motivo, in questa prima fase, abbiamo annotato solamente un piccolo sottoinsieme delle immagini fornite dall'Osservatorio, portando alla creazione di un dataset di dimensioni sufficienti all'allenamento di alcuni modelli predittivi preliminari che saranno descritti nelle sottosezioni successive.

Le condizioni meteorologiche attualmente considerate all'interno di questa versione preliminare del dataset sono le seguenti: classe di visibilità (definita sulla base di punti chiave disposti in scala logaritmica), nebbia, pioggia.

## 5.2 Progettazione e Allenamento del Feature-Extractor

Il processo di estrazione delle features consiste nel produrre una rappresentazione sintetica dell'immagine di input tale per cui risultino preservate le caratteristiche rilevanti per un particolare task di interesse (che nel nostro caso è la classificazione meteorologica). Per svolgere questa funzione, abbiamo pensato di utilizzare una rete neurale convolutiva; in particolare il feature extractor che intendiamo utilizzare coincide con l'encoder dell'architettura mostrata in Figura 9.

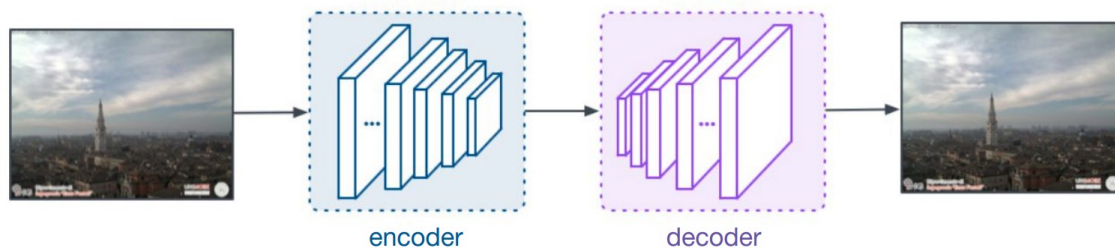


Figura 9: Schema semplificato dell'autoencoder utilizzato

La scelta di utilizzare una configurazione ad autoencoder è nata partendo dall'assunzione che le caratteristiche rilevanti all'interno dell'immagine per il task del nowcasting coincidano quasi esclusivamente con gli elementi variabili all'interno dei video del nostro dataset. Poiché stiamo trattando immagini a camera fissa, ciò che rimane costante sono gli edifici in primo piano e il resto del landscape, mentre a variare sono appunto le condizioni meteorologiche e le condizioni di illuminazione. Un autoencoder allenato a ricostruire le immagini di input nel modo più accurato possibile, tenderà quindi a conservare nel proprio bottleneck le sole informazioni rilevanti per la ricostruzione, andando ad eliminare gli elementi costanti e mantenendo quelli variabili (appunto quelli di interesse per il task di nowcasting).

L'input dell'autoencoder è un'immagine RGB appartenente al nostro dataset, l'output è la ricostruzione dell'immagine di input, mentre il target è l'immagine di input stessa; la loss utilizzata per il training è il mean squared error (MSE) tra input e target. Il bottleneck del modello, invece, rappresenta il set di features per i modelli predittivi di nowcasting che saranno descritti nella sottosezione seguente.

### 5.3 Progettazione dei Modelli di Nowcasting

Attualmente abbiamo progettato 3 modelli di nowcasting utilizzando il feature extractor descritto nella sottosezione precedente affiancato a una rete completamente connessa per la parte di classificazione della condizione meteo. Ciascuno dei modelli predittivi si compone quindi di una parte convolutiva che, presa in ingresso un'immagine RGB, estrae da essa features rilevanti per l'ambito meteorologico e una seconda parte completamente connessa che, date le features, restituisce la label di classificazione.

I modelli attualmente previsti sono i seguenti:

- modello di classificazione della visibilità: Figura 10
- modello di rilevazione di pioggia: Figura 11
- modello di rilevazione di nebbia: Figura 12

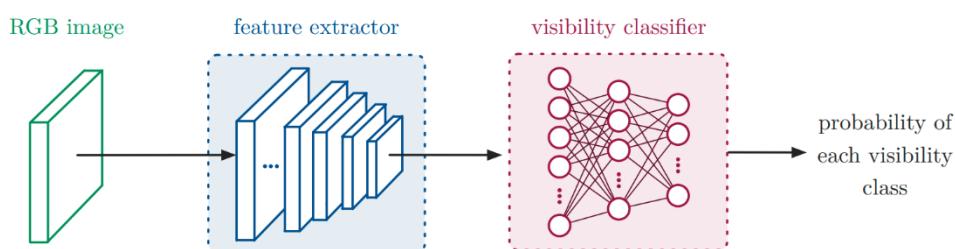


Figura 10: Schema semplificato del modello predittivo della classe di visibilità

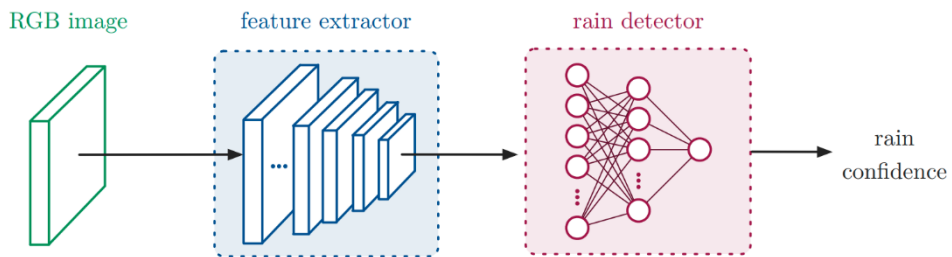


Figura 11: Schema semplificato del modello predittivo del fenomeno atmosferico "pioggia".

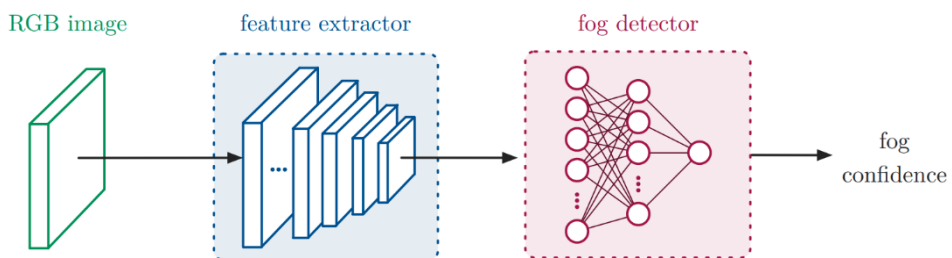


Figura 12: Schema semplificato del modello predittivo del fenomeno atmosferico "nebbia".

Ciascuno dei modelli di cui sopra sarà allenato e testato durante le prossime fasi del progetto, a seguito di un adeguato ampliamento dell'attuale dataset. Saranno inoltre valutati altri modelli predittivi.

## 6. Bibliografia

- [Andrew K.S. Jardine 2005] Jardine AKS, Lin D, Banjevic D. A review on machinery diagnostics and prognostics implementing condition-based maintenance. Mech Syst Signal Process 2006.
- [Jianbo Yu 2015] Yu J. Machine health prognostics using the Bayesian-inference-based probabilistic indication and high-order particle filtering framework. J Sound Vib 2015.
- [Ming Dong 2007] Dong M, He D. A segmental hidden semi-Markov model (HSMM)-based diagnostics and prognostics framework and methodology. Mech Syst Signal Process 2007.
- [T. Benkedjouh 2013] Tarak Benkedjouh, Kamal Medjaher, Nouredine Zerhouni, Saïd Rechak. Remaining useful life estimation based on nonlinear feature reduction and support vector regression. Eng. Appl. of AI 2013.
- [Abdenour Soualhi 2013] Abdenour Soualhi, Hubert Razik, Guy Clerc, Dinh Dong Doan. Prognosis of Bearing Failures Using Hidden Markov Models and the Adaptive Neuro-Fuzzy Inference System. IEEE Trans. Industrial Electronics 2014.
- [Wu Chunzhi 2018] Chunzhi Wu, Pengcheng Jiang, Chuang Ding, Fuzhou Feng, Tang Chen. Intelligent fault diagnosis of rotating machinery based on one-dimensional convolutional neural network. Computers in Industry 2019
- [Li Xiang 2018] Xiang Li, Wei Zhang, Qian Ding. Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction. Rel. Eng. & Sys. Safety 2019
- [Shuyang Yu 2019] Shuyang Yu ; Zhenyu Wu ; Xinning Zhu ; Michael Pecht. A Domain Adaptive Convolutional LSTM Model for Prognostic Remaining Useful Life Estimation under Variant Conditions. Prognostics and System Health Management Conference (PHM-Paris) 2019.
- [Qinxue Meng 2017] Qinxue Meng, Daniel R. Catchpoole, David B. Skillicorn, Paul J. Kennedy. Relational Autoencoder for Feature Extraction. IJCNN 2017
- [Kaiming He 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition. CVPR 2016.
- [Joseph Redmon 2016] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. CVPR 2016
- [Joseph Redmon 2018] Joseph Redmon, Ali Farhadi. YOLOv3: An Incremental Improvement. CoRR 2018