



# **DIGITAL FUTURE**

## **OR4 – Soluzioni verticali**

**Analisi servizi di piattaforma per la  
Fabbrica Intelligente**

**TR4.3**

**AUTORE: EXPRIVIA S.P.A**

---



Digital Future – POR Regione Puglia  
Fondo Europeo di Sviluppo Regionale 2014-2020  
Titolo II – Capo 1 “Aiuti ai programmi di intervento delle Grandi Imprese”



UNIONE EUROPEA  
FONDO EUROPEO DI SVILUPPO REGIONALE



Ministero Sviluppo Economico



REGIONE PUGLIA  
AREA POLITICHE PER LO SVILUPPO ECONOMICO  
IL LAVORO E L'INNOVAZIONE

*Regolamento regionale della Puglia per gli aiuti in esenzione n. 17 del 30/09/2014*

*Titolo II – Capo 1 “Aiuti ai programmi di investimento delle Grandi Imprese”*

**POR PUGLIA FESR 2014 – 2020**

**CONTRATTO DI PROGRAMMA**

**“Digital Future”**

**CODICE PROGETTO: CP12PA6**

**Diritti di autore e riservatezza**

Questo documento è proprietà esclusiva della società Exprivia S.p.A. e non può essere riprodotto, anche in forma parziale, senza un'autorizzazione scritta della società stessa.

Exprivia

OR4 – Soluzioni verticali  
TR4.3 - Analisi servizi di piattaforma per la Fabbrica  
Intelligente

3



## Indice dei contenuti

<b>INDICE DEI CONTENUTI.....</b>	<b>4</b>
<b>1. INTRODUZIONE .....</b>	<b>8</b>
1.1. SCOPO DEL DOCUMENTO .....	8
<b>2. AMBITI APPLICATIVI DI RIFERIMENTO PER LA FABBRICA INTELLIGENTE.....</b>	<b>10</b>
2.1. PARADIGMA DI RIFERIMENTO PER LA DIAGNOSI DEI GUASTI NELLA SMART MANUFACTURING....	14
2.2. TIPOLOGIA DI DATI TRATTABILI E PROCESSO PER CREAZIONE MODELLI PREDITTIVI .....	20
2.2.1. Preparazione dei dati.....	24
2.2.2. Progettazione delle funzioni .....	27
2.2.3. Metodi di training e convalida.....	29
2.2.4. Metodi di test.....	31
2.3. TIPI DI MODELLI APPLICABILI NELLA MANUTENZIONE PREDITTIVA .....	33
2.4. FASI PER LA GESTIONE DELLA MANUTENZIONE PREDITTIVA.....	35
<b>3. APPROCCIO INTEGRAZIONE PIATTAFORMA DIGITAL FUTURE.....</b>	<b>43</b>
3.1. APPROCCIO INTEGRAZIONE CON SMART FACTORY.....	45
3.2. APPROCCIO INTEGRAZIONE CON SAP .....	48
<b>4. SERVIZI PIATTAFORMA PER LA FABBRICA INTELLIGENTE.....</b>	<b>50</b>
4.1. SERVIZI COMUNI .....	50
4.1.1. Device Managment.....	51
4.1.2. Data Ingestion.....	53
4.1.3. Data Preparation .....	56
4.2. SERVIZI PER MONITORAGGIO DELLO STATO DI SALUTE DI UN MACCHINARIO O COMPONENTE ...	57
4.2.1. Servizi per stimare la vita utile rimanente (RUL) .....	58
Data Acquisition da device.....	59
Data Preparation e feature engineering .....	62
Batch processing /model Regression .....	65
Real time Processing .....	69
Alerting.....	70



Start process mantanance .....	72
4.3. SERVIZI PER CLASSIFICARE LA PRESENZA DI UN GUASTO ENTRO UNA FINESTRA TEMPORALE .....	77
Data Preparation e feature engineering da storici .....	78
Training and creation of the neural network model of classfication.....	79
Validation and testing .....	82
4.4. SERVIZI PER LA RILEVAZIONE DI ANOMALIE (ANOMALY DETECTION) .....	84
Data Preparation e feature engineering .....	87
Anomaly Detection .....	89
Real time Processing .....	90
Alerting.....	92
Start process mantanance .....	93
Research Document .....	93
4.5. SERVIZI MONITORAGGIO DELLA LINEA PRODUTTIVA .....	94
4.5.1. Servizi per prevedere i valori dei sensori in instanti futuri.....	95
Data Preparation e feature engineering da dati storici .....	98
Creating ARIMA models .....	100
Validation model ARIMA.....	104
Real-time Processing.....	105
Alerting.....	106
4.5.2. Servizi per predire il numero di pezzi prodotti .....	106
Creating ARIMA models .....	107
Real time Processing .....	109
Alerting.....	110
<b>5. BIBLIOGRAFIA .....</b>	<b>112</b>



## Indice delle figure

Figura 1 - Modello di funzionamento di un sistema di manutenzione predittiva .....	16
Figura 2: Principali blocchi di un modello di Digital Twin .....	18
Figura 3: architettura di un sistema digital twin [5].....	19
Figura 4: Esempio di manutenzione predittiva .....	20
Figura 5: processo di <i>apprendimento</i> automatico .....	23
Figura 6: machine learning process .....	24
Figura 7: schemi di dati per monitoraggio dello stato di salute del macchinario e predizione di potenziali guasti .....	27
Figura 8: Funzione di aggregazioni in sequenza [8].....	28
Figura 9 Funzioni di aggregazione a cascata [1].....	29
Figura 10 Applicazione delle funzioni statiche sugli scenari della manutenzione predittiva .....	29
Figura 11: macro -fasi per l'approccio alla manutenzione predittiva.....	36
Figura 12 : Avvio Alerting.....	38
Figura 13 :Esempio di un sistema di CBM che combina diagnostica e prognostica [15].....	42
Figura 14 - Architettura Logica Fabbrica Intelligente. ....	46
Figura 15: - Architettura IOT basata su interazione tra Raspberry Pi e cloud.....	47
Figura 16 - Struttura Publish-Subscribe di una rete MQTT.....	48
Figura 17: Architettura Integrazione SAP-Piattaforma DF .....	49
Figura 18: configurazione device.....	52
Figura 19 Rappresentazione del collegamento tra Apache Kafka e il protocollo MQTT.....	54
Figura 20 - Kafka Architecture .....	55
Figura 21 - Architettura Kafka-Connect.....	55
Figura 22: stream processing .....	56
Figura 23 Dati acquisiti del valore di batteria una macchinina durante una corsa.....	62
Figura 24 Pipeline di processing.....	62
Figura 25:Andamento corse per macchina Nuke. ....	65
Figura 26: Training Model.....	67
Figura 27 :Training vs test corse Nuke .....	68
Figura 28:Training vs test tridimensionale .....	69
Figura 29:Tempo residuo di carica misurato vs stimato .....	70
Figura 30: pipeline real time Alerting.....	71
Figura 31 Workflow del processo di manutenzione generico.....	73
Figura 32 Blocco di attività della fase iniziale.....	73
Figura 33 Attività della fase di ispezione e della compilazione del relativo report .....	74
Figura 34 Attività legate alla decisione del referente e alle sue valutazioni finali.....	75
Figura 35 Attività della fase di attuazione dell'intervento e della compilazione del relativo report.....	76
Figura 36 time unit for an asset.....	79
Figura 37 funzione rete neurale .....	81
Figura 39: Pipeline di modelling for classification.....	84
Figura 40 Plot delle misurazioni macchinario ICAM.....	87
Figura 41 Differenze rilevazioni YActualPosition e YPositionFeedback1 Value .....	88
Figura 42 :Real-time Data Processing and Monitoring caso simulato.....	91
Figura 43 :Real-time Data Processing and Monitoring contest reale .....	92
Figura 44: Start process .....	93
Figura 45 - Modello di funzionamento scenario predizione numero di pezzi prodotti .....	94



Figura 46 - Modello di funzionamento scenario predizione velocità di rotazione.....	95
Figura 47: componenti di avvio ordine su sap .....	96
Figura 48 : Pipeline create modello ARIMA .....	99
Figura 49 - Spettro con criteri di definizione dei termin AR e MA .....	102
Figura 50 - Serie temporale relativa alla velocità di rotazione .....	102
Figura 51 - Rappresentazione del modello ARIMA per la predizione della velocità di rotazione.....	104
Figura 52 - Plot degli errori residui.....	104
Figura 53 - Metriche di valutazione del modello .....	105
Figura 54 - Forecast vs Actual modello ARIMA predizione velocità di rotazione .....	105
Figura 55 - Forecast vs Actual modello ARIMA predizione numero di pezzi .....	109
Figura 56 : pipeline real time .....	111



# 1. Introduzione

## 1.1. Scopo del documento

Obiettivo del presente TR è quello di descrivere nel dominio della Fabbrica Intelligente i fabbisogni, i contesti applicativi e i servizi che dovranno essere sviluppati e esposti dalla piattaforma Digital Future per essere sfruttati dai verticali facenti parte del dominio.

Nello specifico saranno definite per alcune delle problematiche trattate: la tipologia dei dati e i profili descrittivi dei dati trattati; le tipologie di tecniche di analisi e i modelli la cui qualità sarà validata mediante utilizzo di training .

Questo al fine di progettare attraverso l'uso del Piattaforma DF un processo additabile a diversi ambiti industriali capace di ideare una serie di regole e gestire una manutenzione predittiva e intelligente coadiuvata anche da supporti multimediali<sup>1,2</sup>.

Quindi si descrivono :

- **Capitolo 2.** descrizione :
- dei contesti applicativi analizzati e delle problematiche che sussistono in ogni ambito e i paradigmi di riferimento nell'ambito dei Big Data e Industria 4.0. il del flusso per la creazione di un modello e delle macro fasi per la generazione di manutenzione predittiva.
- **Capitolo 3** descrive quali sono i possibili approcci di integrazione con Piattaforma DF con l'applicativi destinati al dominio Fabbrica Intelligente come sistemi di Asset Management , di Monitoraggio e Diagnostica e sistemi ERP.

---

<sup>1</sup> Pega – Digital Prescriptive Maintenance – 2015

<sup>2</sup> Roland Berger- Predictive Maintenance - 2014



- **Capitolo 3:** descrive i servizi base e comune ad ogni problematica e in funzione degli obiettivi si descrive un specifico tipologia modello di data-driven .Nello scecifiso si decrive iservizi da progettare e implementare in Piattaforma DF che attraverso tecniche di machine learning e statistiche permettono di modellare delle pipeline rivolte principalmente a problemi di:
  - *Classificazione:* analisi de guasti e previsione ci sarà un guasto si o no e la classe di guasto
  - *Regressione:* attraverso inferenze e su base di serie storiche si può determinare
  - *Rilevamento delle anomalie:* individuazione di un fenomeno anomalo sulla base dell’apprendimento di gruppi di fenomeni etichettati come positivi e negativi il tempo utile rimanente.
  - *Analisi delle serie temporali di valori di telemetria.*
  - *Analisi e predizione dell’andamento produttivo.*





## 2. Ambiti applicativi di riferimento per la Fabbrica Intelligente

Fabbrica Intelligente è un termine coniato da molte agenzie tra cui il DoE (Department of Energy) e il NIST (National Institute of Standards and Technology) e prevede «l'utilizzo di applicazioni di Information and Communication Technology (ICT) e di tecniche di advanced analytics finalizzate a migliorare i processi industriali a tutti i livelli, dall'officina (shop-floor) al resto della catena di approvvigionamento (supply-chain)».

La sua introduzione permette di *creare* un ecosistema perfettamente integrato in cui macchine, prodotti, sistemi ICT e persone, sono connessi su Internet, creando una rete di produzione «*intelligente*» in cui i portatori di informazioni comunicano tra loro e scambiano dati in tempo reale o quasi reale generando «valore»

Trasforma il modello di produzione centralizzato in un modello decentralizzato più efficiente, flessibile e agile, in cui materiali e macchinari sono in grado di comunicare tra loro e prendere decisioni autonome, prevedendo e auto-adattandosi ai cambiamenti.

Mediante l'integrazione dei dati provenienti dal sistema fisico, operativo e risorse umane, guida attività quali la produzione, la manutenzione, Test di qualità, la tracciabilità delle scorte, il magazzino e la digitalizzazione delle operazioni attraverso il Digital Twin e come si riportata in tabella anche attraverso diverse tecnologie abilitanti in Industria 4.0



<b>Maintenance</b>	<p><i>Augmented reality</i>, per assistere il personale addetto alla manutenzione nella manutenzione e riparazione attrezzatura.</p> <p>Sensori sulle apparecchiature per guidare analisi di manutenzione predittiva e cognitiva.</p>
<b>Manufacturing operations</b>	<p><i>Additive manufacturing</i>, per produrre prototipi rapidi o pezzi di ricambio a basso volume</p> <p><i>Advanced planning and scheduling</i>, utilizzando dati di produzione e inventari in tempo reale per minimizzare perdite di tempo e sprechi</p> <p><i>Cognitive bots and autonomous robots</i>, per eseguire efficacemente processi di routine a costi minimi con un alto livello di accuratezza</p> <p><i>Digital twin</i>, per digitalizzare un'operazione andando oltre l'automazione integrando soluzioni quali l'analisi predittiva</p>
<b>Quality</b>	<p>Test di qualità, utilizzando optical- based analytics</p> <p>Real-time equipment monitoring, per predire potenziali problemi di qualità dei prodotti</p>
<b>Warehouse operations</b>	<p><i>Augmented reality</i>, per assistere il personale nelle attività di pick-and-place</p> <p><i>Autonomous robots</i>, per eseguire operazioni di magazzino</p>
<b>Inventory tracking</b>	<p>Sensori, per tracciare i movimenti e le posizioni in tempo reale delle materie prime, avanzamento del lavoro e prodotti finiti e utensili di alto valore</p> <p>Analytics, per ottimizzare l'inventario a portata di mano e segnalare automaticamente il rifornimento</p>
<b>Environmental, health, and safety</b>	<p><i>Sensori</i>, per georeferenziare apparecchiature pericolose</p> <p><i>Sensori</i>, applicati sul personale per monitorare le condizioni ambientali, la mancanza di movimento o altre potenziali minacce</p>

Tabella 1 Fabbrica Intelligente processi e tecnologie abilitanti



Pertanto, dopo un’analisi della definizione, le tecniche per preparazione dati (estrazione, manipolazione, correlazione) e i metodi per la preparazione delle funzioni e costruzione e addestramento di modelli descritti Nel TR 4.2 [1], il presente documento si focalizza ad esaminare quali sono le modalità e i possibili servizi che permettono di integrare dei dati provenienti dal sistema fisico con soluzioni che guidano ad analisi di manutenzione predittiva e cognitiva.

In particolare saranno presi in considerazione alcuni scenari . Per ciascuno scenario sarà descritto il problema che si vuole risolvere, l’obiettivo che si vuole raggiungere con la definizione del modello, i dati utilizzabili ed il modello predittivo applicabile e i servizi che la piattaforma Digital Future dovrà esporre per soddisfare i loro obiettivi.

Dalla descrizione delle problematiche dei scenari si potrà evincere come già evidenziato nel TR 4.2 [1] la necessità di integrare sistemi ERP e CMMS tradizionali con piattaforma IOT come Digital Future (Middleware) per passare da un gestione di interventi di manutenzione reattiva a una gestione proattiva e predittiva.

Come espresso nel TR 4.2 [1] l’obiettivo della soluzione proposta da Exprivia nell’ambito della Fabbrica Intelligente prevede la realizzazione di una soluzione che consenta di monitorare un impianto industriale e di prevederne l’andamento futuro al fine di prevenire possibili guasti o anomalie nel funzionamento e nel processo produttivo.

Per realizzare tale soluzione Exprivia intende avvalersi di una serie di tecnologie innovative in campo Big Data e Machine Learning resi disponibili attraverso i servizi della piattaforma DF che andranno a soddisfare diversi scenari applicativi.

Nello sviluppo dei servizi nell’ambito della manutenzione predittiva il modello generale sarà sperimentato su singole macchina/linee di produzione per i scenari applicativi :

- **Monitoraggio dello stato di salute di un asset** in cui si :
  - stimare la probabilità che un'apparecchiatura possa guastarsi entro X unità di tempo.
  - calcolare la vita utile rimanente di un'apparecchiatura definita come la quantità di tempo durante la quale l'apparecchiatura rimane operativa .



- individuare presenza di anomalia che possono causare un potenziale guasto di una determinata severità.
- **Monitoraggio della linea produttiva** in un dato periodo al fine di rilevare eventuali criticità nel processo produttivo.
- **Controllo della qualità della linea produttiva** per predire potenziali problemi di qualità dei prodotti. La trattazione di questo particolare problematica è descritta in maniera approfondita nell'allegato [2] . TR 4.3 Analisi Servizi di Piattaforma Product Quality Management.

Le fonti di raccolta per questi tipi di situazioni possono essere diverse e ottenuti da diverse fonti che vanno dai parametri di processo, dallo stato di lavoro, alle condizioni operative e ambientali e le tecniche di rilevamento intelligenti potranno essere utilizzate per misurare il stato di degrado di un sistema fisico secondo tre aspetti:

- a) **Modellazione geometrica:** parametri di base del sistema di fabbricazione ottenuti dalla fase di progettazione, come le proprietà dei materiali, le dimensioni delle attrezzature chiave e le relazioni di montaggio tra le parti. Questi dati possono essere supporta da una successiva analisi statica o dinamica attraverso simulazione meccanica, fluidodinamica o altre simulazioni multifisiche .
- b) **Stato di funzionamento:** Lo stato di salute di un sistema fisico si manifesta spesso con i suoi cambiamenti strutturali. Le informazioni possono essere raccolte da una varietà di sensori intelligenti per il monitoraggio e l'analisi delle condizioni di produzione, come quelli elettrici sensori di corrente per la misurazione della potenza, sensori accelerometrici per la misurazione delle vibrazioni.
- c) **Condizioni operative e ambientali:** Generalmente, la risposta dinamica di un sistema fisico non è influenzata solo dal suo stato di salute, ma anche dalle sue condizioni operative e ambientali. Lo stato di funzionamento meccanico è caratterizzato dalla complessità e dalla variabilità di diversi parametri, tra cui la profondità e la velocità di taglio, ecc. I dati ambientali che influenzano le



operazioni di un asset fisico includono la temperatura ambiente, la pressione barometrica, e il livello di umidità. Inoltre, le modifiche dei parametri operativi introdurranno inevitabilmente incertezza e eventuale condizione di degrado meccanico con il tempo.

Bisogna precisare che in un ambiente di Big Data i dati sono numerosi e spesso non strutturati. Capire quindi quali informazioni siano realmente importanti per lo sviluppo della soluzione non è semplice. Immaginare un modo per incrociare i dati provenienti da prodotti, clienti e ambiente è poi una sfida che solo i data scientist possono vincere. Una volta identificate come primo step le correlazioni ed effettuate le predizioni future adottando successivamente il paradigma Digital Twin for Product Design si può ottenere un valido strumento rispetto all'adozione sola delle tecniche classiche FMEA applicata in un CMMS citata nel TR 4.2 . [1]

### **2.1. Paradigma di riferimento per la diagnosi dei guasti nella Smart manufacturing**

Il modello Digital Twin è un nuovo paradigma per la diagnosi dei guasti applicabile in due fasi progettazione e manutenzione.

Nella fase di progettazione basandosi sull'addestramento di reti neurali (DNN) possono essere scoperte delle anomalie, imperfezioni, guasti su modello virtuale o prototipo del macchinario. [3] Mentre nella seconda fase il modello di diagnosi precedentemente addestrato può essere migrato dall'analisi di un sistema virtuale in uno spazio fisico reale con l'ausilio del deep learning permette la correlazione di più set di dati e il monitoraggio in tempo reale.

Pertanto nell'attuale iniziativa s'intende orientare la progettazione delle soluzioni applicative per la fabbrica intelligente basate su data-driven che differenzia da quella tradizionale, in quanto non dipende fortemente dall'esperienza del designer nell'identificare i dati rilevanti. Questo non vuol dire che l'attività umana viene sostituita da quella di un algoritmo, in quanto per identificare pattern consistenti, nei big data, serviranno comunque di figure come dei data scientist, ma piuttosto con questo approccio si vuole mostrare che si sta volendo progettare una soluzione misurata (tailor-



made) sui fabbisogni dei singoli scenari, contesti applicativi. In Tabella 2 viene illustrata il confronto.

tra l’approccio DFDD(Deep transfer learning (DFDD) e i metodi tradizionali di diagnosi dei guasti.

	Traditional fault diagnosis	DFDD
Data Source	Monitoring data of physical entity	Simulation data of virtual space and monitoring data of physical entity
Data Distribution	Same distribution of training and test data	Different distribution of source and target data
Model Training	Different situations need to be modeled separately	Models can migrate in different situations
Diagnosis Period	Operation and maintenance phase	From design phase to operation and maintenance phase

Tabella 2 Comparazione tra Dfdd and i metodi tradizionali di diagnostica dei fault

Pertanto nello sviluppo della soluzione proposta nell’ambito della manutenzione predittiva, ci siamo ispirati ad un modello di funzionamento generale basato su tre componenti principali e proposto in [4] Figura 1 :

- **sistema fisico**, nello spazio reale, vengono raccolte sempre più informazioni (dati geometrici sul prodotto, velocità, consumo energetico e condizioni ambientali quali umidità, temperatura, pressione barometrica) per determinare le condizioni reali del’ impianto e o prodotto mediante l’utilizzo di sensori intelligenti
- **Un modello digitale (Digital twin Model)**, in cui vengono correate le arricchite le informazione mediante l’aggiunta di numerose caratteristiche comportamentali dl macchinario i progettisti pertanto possono non solo visualizzare il prodotto, ma anche testarne le prestazioni per la diagnosi dei guasti fase di progettazione attraverso simulazione.
- **Sensori intelligenti** posti sul asset(macchinario) che rappresenta il modulo di connessione e raccolta dati , raccoglie lei misure di interesse dai sensori intelligenti posti sul macchinario e attraverso tecnologie basate sull’analisi dei parametri real time mette in relazione il modello digitale e il sistema fisico e permettono l’aggiornamento automatico del modello digitale in base alle condizioni operative e allo stato del sistema fisico

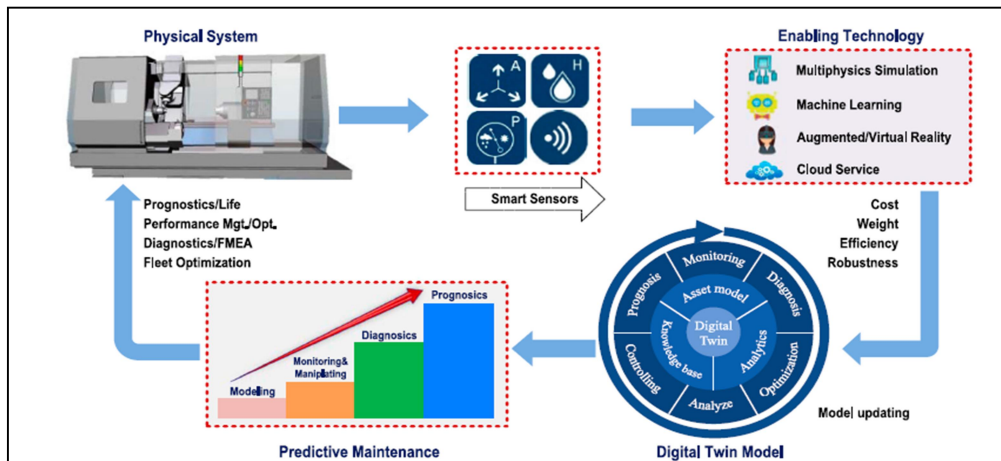


Figura 1 - Modello di funzionamento di un sistema di manutenzione predittiva

Il **Digital twin Model** è caratterizzato da dati che descrivono il modello fisico e da analisi dei dati per la diagnosi dei guasti. E' costituito essenzialmente da tre elementi principali: modello digitale (Digital Model), dati di analisi (analytics) e base di conoscenza, come mostrato Figura 1. In dettaglio:

- a) **Digital Model:** è la struttura dei sottosistemi e componenti che costituiscono l'asset. Ed per ogni sottosistemi specifico si indicano le misurazioni di rilevamento raccolte dalla produzione, dalle operazioni e dall'ambiente-Quindi rappresenta l'anagrafica tecnico funzionale del asset e permette di definire tutte le informazioni utili per creare in sistema virtuale capace di gestire processi simulati di comportamenti normali o anomali che aiutare nella fase di addestramento di un modello. Quindi con l'ausilio del sistema Asset Management componente della soluzione Fabbrica Intelligente descritto nel [1]si descriveranno i modelli digitali dei scenari trattati.
- b) **Analytics** rappresentano i dati utile per descrivere, diagnosticare, prevedere e prescrivere il comportamento del sistema fisico per la diagnosi dei guasti. In particolare dopo la raccolta dei dati in tempo reale sui dati inerenti il processo produttivo e/o le condizione di utilizzo, l'utente esperto del dominio accederà alla piattaforma DF e con l'ausilio della modulo **Algorithm Recommender** recupera una classifica ordinata di algoritmi di Machine Learning i primi algoritmi in classifica sono quelli con più probabilità di successo per definire uno



o più modelli da addestrare adatti alla tipologia di dati trattati. Successivamente per avviare l’esperimento e quindi l’analisi del problema attraverso il modulo Data Governance potrà eseguire una prima fase analisi statistica esplorativa sulle serie temporali e con il modulo di **Correlation Analyser** potrà dedurre delle una correlazione tra i dati raccolti e presi in esame. A seguire mediante algoritmi addestramento di machine learning e/o potrà definire un classe di potenziali guasti e/o breakdown della macchine. Tale classi serviano ad arricchire il modello Digital Twin.

- c) **Knowledge base:** La conoscenza acquisita dall'analisi dei rapporti di risoluzione dei problemi e di riparazione, così come dalla interazione con gli esperti di diagnostica e gestita invece dal sistema CMMS (Asset Mangament e Workflow Managment). Il quale storicizza tutte le informazioni utili come le modalità di guasto, gli indicatori di salute, le regole diagnostiche, la definizione di soglie e i rischi operativi, le classificazione delle anomalie riscontrate e alert inviati, nonché lo storico dei dati inviati dai sensori nel corso del tempo. Questa capacità crescerà continuamente attraverso l'apprendimento dalle interazioni degli utenti. Consentendo di conseguenza ad un'impresa manifatturiera di mettere in atto azioni correttive finalizzate ad evitare guasti o deterioramenti e/o migliorare il processo produttivo determinando l'ideale programma di manutenzione, e ottimizzare le prestazioni in base alle specificità del singolo asset

Il Modello di Digital Twin in linea generale avrà il compito di:

- **Supportare Estrazione dei dati** da apparecchiature, sensori e dispositivi in quanto avrà l’anagrafica dei sensori e le misure da rilevare.
- Supportare l’**analisi e il controllo real-time**
- Supportare l’**Edge Analytics** per analizzare grandi volumi di dati;
- **Archiviare grandi volumi di dati**, che potranno crescere in modo esponenziale, con un costo minimo;



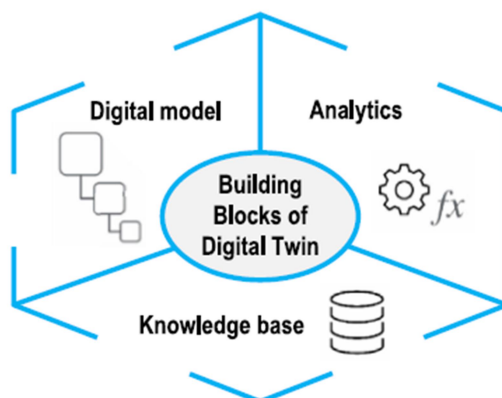


Figura 2: Principali blocchi di un modello di Digital Twin

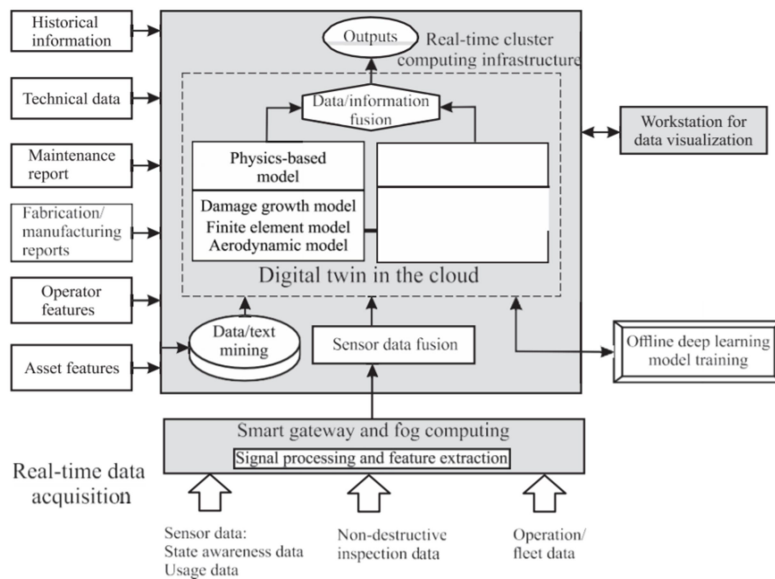
La realizzazione del prototipo Fabbrica Intelligente quindi, come definito, si baserà su un approccio totalmente open source.

Sarà configurato e adattato il sistema di base CMMS OpenMaint per realizzare la componente Asset Management che rappresenterà Knowledge base a supporto DF IOT Platform e permetterà la definizione del Digital Model.

Mentre la parte analytics sarà gestita attraverso Digital Future IOT Platform con lo sviluppo di servizi aventi come obiettivo automatizzare il processo di gestione de dati : raccolti dati; l'estrazione delle informazione; definizioni di modelli sulla analisi dati, correlazione dati esistenti con altre fonti esterne e/o con diversi dati di monitoraggio al fine di valutare le relazioni e/o dipendenze; addestrare uno o più modelli predittivi per la valutazione della probabilità di accadimento e delle finestre temporali di interesse;

In generale l'architettura proposta si basa sul paradigma Digital Twin che come definito dalla piattaforma GE Digital Twin di GE Power Digital Solution [5] e da Zheng Liu<sup>1</sup>, Norbert Meyendorf and Nezhir Mrad [6] è utilizzata per identificare in modo proattivo potenziali problemi e permette di prevedere la vita utile residua (RUL) ossia lo stato di salute attraverso:

- la raccolta di dati operativi provenienti da sensori ;
- l'analisi, la correlazione e fusione di informazioni specifici del asset derivati dalla fase di progettazione, produzione, ispezione, riparazione;
- l'addestramento di modelli.



**Figura 3: architettura di un sistema digital twin [6]**

Pertanto con l'adozione di questo approccio basati su dati sarà possibile individuare di ogni asset:

- come invecchierà rispetto al suo funzionamento e quindi lo stato di salute;
- le anomalie che possono portare a potenziali guasti;
- le cause come la fatica, lo stress, l'ossidazione e altri fenomeni.

in modo da ottimizzare la manutenzione aumentare la precisione delle curve di vita delle macchinari e riducendo i tempi di inattività non pianificati.

In particolare in molti processi produttivi [7] evidenzia che il controllo delle vibrazioni dei motori delle macchine può mettere in luce un possibile guasto in arrivo.

Piccole vibrazioni iniziali anomale, causate da minuscoli disassamenti, si amplificano sempre di più fino a mettere sotto sforzo il motore e guastarlo. Esse non sono percepite in mezzo al rumore dell'intero impianto, almeno nei primi tempi, ma possono essere rilevate da sensori.

Quindi fornire la capacità di conoscere questo tipo di anomalie permette di prevedere con largo anticipo la rottura in si evidenzia un possibile esempio in Figura 4 di gestione di manutenzione predittiva su un motore.

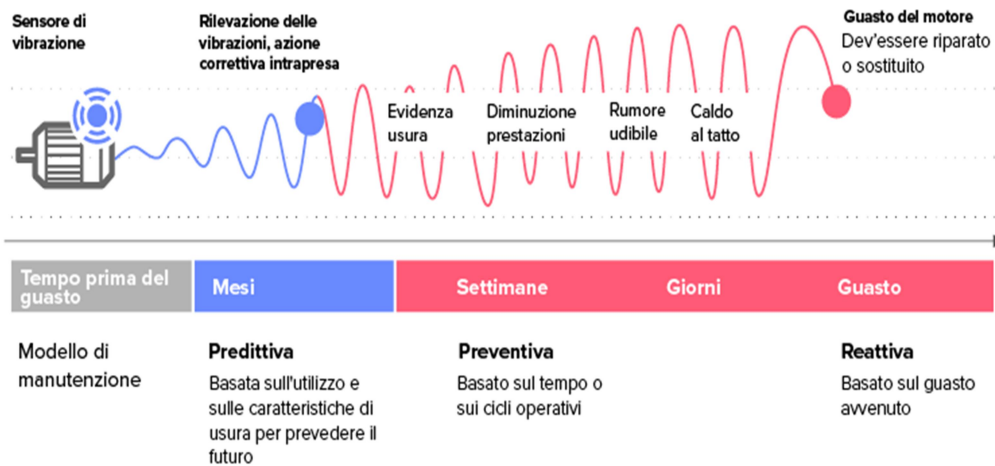


Figura 4: Esempio di manutenzione predittiva

## 2.2. Tipologia di dati trattabili e processo per creazione modelli predittivi

In generale in contesti reali prevedere e prescrivere e diagnosticare potenziali guasti è necessario una piattaforma IOT in grado di raccogliere grande quantità di dati che in real-time da sensori e attraverso un serie di passi di analisi e correlazioni tra essi deve fornire delle valide segnalazione di alerting. Deve pertanto possedere componenti deputate al **Data Ingestion** e allo batch e **Stream Processing** e deve essere capace di gestire:

- **Serie temporali**, i sensori misurano i cambiamenti nel tempo. Per ottenere informazioni utili, pertanto i sistemi hanno necessità di valutare e analizzare i dati in base a intervalli temporali e pertanto è necessario che le piattaforme IOT devono essere adattarsi a funzionare con serie temporali





- **Streaming di dati:** che sono generati da una miriade di sensori che emettono un flusso continuo di dati pertanto le piattaforma IOT devono essere in grado di gestire lo streaming di dati che arriva dai sensori.

La soluzione proposta per la Fabbrica Intelligente, Exprivia intende avvalersi come Indicato nel TR 4.2 [1] della Componente “Data Governance” della Piattaforma DF, nello specifico per la manutenzione predittiva s’intende adottare un sistemi di intelligenza artificiale e basato sull’*apprendimento automatico* al fine di costruire di modelli, i quali usando dati storici e addestrandoli su casi noti, saranno in grado d’identificare e classificare situazioni di potenziali guasti e predire a regime situazioni di guasti nel corso del funzionamento dell’impianto.

Bisogna precisare che i modelli prima di applicarli a regime dovranno essere validati usando dati reali di test. La fase di validazione fornisce indicazioni (matrice di confusione) sull’attendibilità dei modelli individuando i veri positivi, i veri negativi, falsi positivi e i falsi negativi.

La pipeline del processo di apprendimento sarà praticamente la stessa per ogni caso preso in esame e comprenderanno in generale i seguenti macro step [8] :

1. set di dati vengono inviati al sistema di intelligenza artificiale (DF)
2. I modelli di dati vengo realizzati in funzione dei set di dati.
3. Si individua una potenziale anomalia ogni volta che una transazione si discosta dal modello.
4. Si verifica con il stehoder che la deviazione indentificata come anomalia sia riscontrabile nel contesto reale.
5. Il sistema apprende dall'azione e si basa sul modello di dati per previsioni future.
6. Il sistema continua ad accumulare schemi in base alle condizioni preimposta.

Gli algoritmi per rilevare anomalie si basano su tecniche di apprendimento automatico supervisionate o non supervisionate.

Il metodo supervisionato richiede un set di addestramento etichettato con campioni normali e anomali per la costruzione di un modello predittivo. I metodi supervisionati



più comuni includono neural networks, support vector machine, k-nearest neighbors, Bayesian networks and decision trees.

Probabilmente, la tecnica non parametrica più popolare è il vicino K più vicino (k-NN) che calcola le distanze approssimative tra i diversi punti sui vettori di input e assegna il punto senza etichetta alla classe dei suoi vicini K-mean. Un altro modello efficace è la rete bayesiana che codifica le relazioni probabilistiche tra le variabili di interesse.

Si ritiene che i modelli supervisionati forniscano un tasso di rilevazione migliore rispetto ai metodi non controllati a causa della loro capacità di codificare le interdipendenze tra le variabili, insieme alla loro capacità di incorporare sia la conoscenza precedente che i dati e di restituire un punteggio di confidenza con l'output del modello.

Le *tecniche non supervisionate* non richiedono dati di addestramento etichettati manualmente ed è possibile applicare l'apprendimento senza supervisione per poter prevedere i prossimi punti dati della serie. Per implementare questo, viene definito un intervallo di confidenza o un errore di previsione. Pertanto, per rilevare punti di dati anomali, viene implementato un test per verificare quali punti di dati sono presenti all'interno o all'esterno dell'intervallo di confidenza. Gli algoritmi non supervisionati più popolari includono K-mean, PCA e analisi basate su test di ipotesi.

L'approccio proposto per l'analisi predittiva è un approccio ibrido che abbraccia varie tecnologie: da una parte tecnologie Big Data per l'elaborazione in tempo reale di grosse mole di dati provenienti dai sensori (quali appunto Spark) e dall'altra l'utilizzo combinato e quindi ibrido di tecnologie statistiche, tecniche tradizionali di machine Learning e di deep Learning. La rete neurale ricorrente è uno degli algoritmi di deep learning per rilevare punti di dati anomali all'interno delle serie temporale

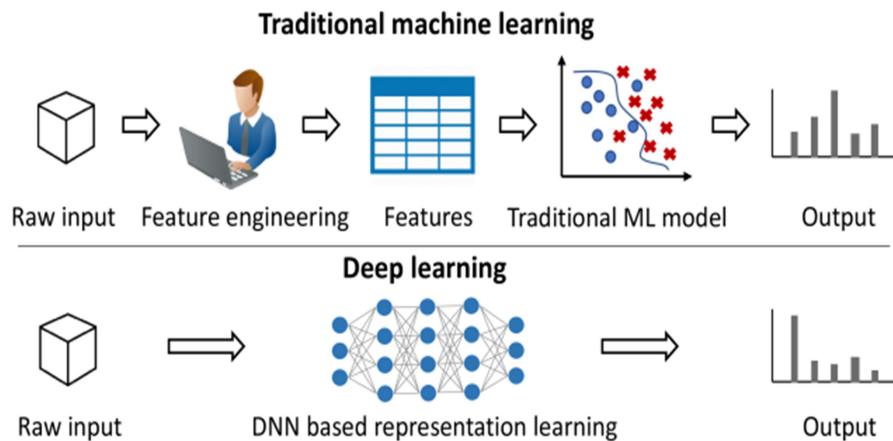


Figura 5: processo di *apprendimento* automatico

L’innovazione della soluzione dal punto di vista tecnologico consiste infatti nell’implementazione di soluzioni allo stato attuale non ancora presenti sul mercato che abbracciano mediante un approccio integrato più tecnologie differenti.

Sarà dunque deputato alle funzionalità di stream processing per il reperimento dei dati in tempo reale e successivamente alle elaborazioni dei dati per l’analisi e la manutenzione predittiva che saranno, come vedremo più avanti saranno inviati alla piattaforma DF per essere infine visualizzati successivamente attraverso apposite dashboard a sistema di Business intelligente di riferimento per lo specifico Verticale.

La costruzione di un modello applicabile agli scenari di manutenzione predittiva prevede l’esecuzione in sequenza di una serie di step che inizia con la preparazione dei dati a disposizione e termina con la valutazione del modello costruito.

Ovviamente prima della costruzione di un modello applicabile agli scenari di manutenzione predittiva si prevede l’esecuzione in sequenza step:

- **Acquisizione e raccolta dei dati** da fonti eterogenee sensori o file csv ,o database esterni erp, sui quali le attività di analisi avanzata dovranno essere condotte e nella formulazione delle ipotesi che devono essere verificate;
- **Preparazione dei dati:** consiste nella individuazione operativa dei dati da elaborare, nel recupero dei dati e nella preparazione dei dati in funzione degli strumenti di modellazione ed elaborazione selezionati;



- **Modellazione:** applicazione delle tecniche di modellazione ed elaborazione prescelti;
- **Valutazione:** dedicata all'analisi dei risultati delle elaborazioni eseguite. Attraverso tale fase si valuta se gli obiettivi del progetto siano stati raggiunti o se è necessario reiterare il processo;

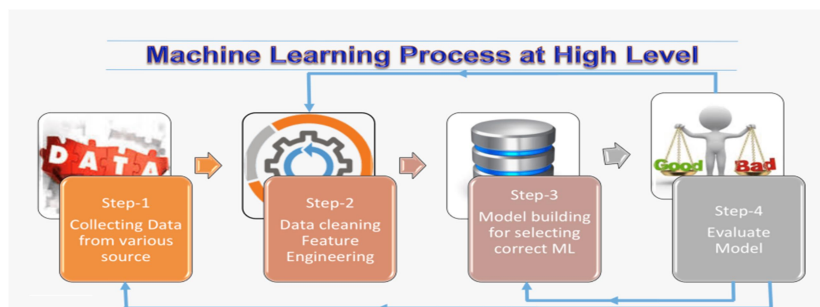


Figura 6: machine learning process

Nel caso in cui la valutazione si sia conclusa positivamente step successivo con il supporto della Piattaforma DF è possibile portare in erogazione il modello con l’acquisizione dei dati real-time e avviare un monitoraggio continuo in modo da acquisire i dati dal campo ed eseguire il modello addestrato.

### 2.2.1. Preparazione dei dati

La preparazione dei dati è il primo passo per la definizione del modello predittivo.

Nel caso degli scenari di analisi predittiva per prevedere potenziali le tipologie di dati più comuni da prendere in considerazione sono:

- **Cronologia dei guasti:** in un macchinario o in un componente del macchinario.
- **Cronologia di manutenzione:** ossia la cronologia delle riparazioni di una macchina. Esempi sono una lista di codici di errore, le attività di manutenzione precedenti o la sostituzione di componenti.
- **Condizioni e utilizzo della macchina:** ossia le normali condizioni operative di un macchinario; tali dati sono in genere quelli raccolti dai sensori.
- **Caratteristiche del macchinario:** ad esempio nel caso delle auto: cilindrata, marca e modello, etc.



Si precisa che la definizione delle fonti dati deve essere fatta consultando gli esperti del settore al momento della definizione del modello predittivo.

Non sempre le fonti dato sono nettamente distinguibili l’una dall’altra spesso la cronologia dei guasti è inclusa nella cronologia di manutenzione sotto forma di codici di errore speciali o date degli ordini delle parti di ricambio. Altre volte la cronologia dei guasti può essere dedotta dal verificarsi di un altro evento. Per maggior chiarezza di seguito vengono elencati degli esempi delle tipologie di dati che potrebbero essere presi in esame [9]:

Cronologia dei guasti:

- date e tipi di guasti dei componenti
- errori delle transazioni di prelievo di deposito di sistema di magazzini automatico
- date degli ordini di sostituzione dei catene
- dati dei guasti delle turbine

Cronologia di manutenzione:

- registri degli errori delle transazioni degli sportelli prelievo e deposito
- record di manutenzione dei catene e tipo di manutenzione
- record di manutenzione degli interruttori

Condizioni e utilizzo delle macchine:

- letture dei sensori dalle transazioni
- letture di sensori di turbine
- Caratteristiche della macchina:
- caratteristiche del veicolo come marca, modello, cilindrata, impianto di produzione

Osservando con attenzione i tipi di dati utilizzati, è possibile identificare due tipi di dati principali: dati temporali e dati statici; se i dati statici cambiano nel tempo (ad es. cambia l’operatore) essi diventano dinamici.

La cronologia dei guasti, della manutenzione e le condizioni di utilizzo del macchinario sono dati temporali e sono quasi sempre accompagnati da timestamp che indicano l'ora





della raccolta di ogni singolo dato o comunque da un'altra unità di misura assimilabile ad un ordine cronologico come il numero di cicli, i chilometri, le transazioni, etc.

Le caratteristiche della macchina e dell'operatore sono, invece, dati statici poiché descrivono le specifiche tecniche dei macchinari o le caratteristiche dell'operatore che vi opera.

Una delle prime attività da compiere per la preparazione dei dati è l'unione dei dati, ossia la generazione di un record per ogni unità di tempo di ogni macchinario (asset) con le relative funzioni ed etichette da inserire nell'algoritmo di machine learning che sarà alla base per la creazione di un modello predittivo.

È importante fare attenzione che l'unità di tempo non sia la stessa della frequenza di raccolta di dati, perché in tal caso i dati potrebbero non mostrare alcuna differenza da un'unità all'altra. In altre parole, se, ad esempio, i valori di pressione vengono raccolti ogni 60 secondi, la selezione di un'unità di tempo di 60 secondi per l'intera analisi aumenta il numero di esempi, senza fornire però informazioni aggiuntive utili; in tal caso sarebbe più opportuno, ad es., considerare la media in un'ora.

In particolare gli schemi di dati che sarà possibile predisporre in base alle tipologie di fonti dati generalmente utilizzate per un generale caso di *monitoraggio dello stato di salute del macchinario e predizione di potenziali guasti* in un dato lasso di tempo di un certa severità sono schematizzate nella Figura 7.



## Schemi di dati

### Record di manutenzione

ID asset	Timestamp	Azione di manutenzione
----------	-----------	------------------------

### Record dei guasti

ID asset	Timestamp	Guasto o motivo guasto
----------	-----------	------------------------

### Condizioni delle macchine → dati di monitoraggio in tempo reale

ID asset	Timestamp	Valore della condizione
----------	-----------	-------------------------

### Dati delle macchine + operatore → questi dati possono essere uniti in uno schema per identificare gli asset gestiti da un dato operatore insieme alle proprietà degli asset e dell'operatore

ID asset	Caratteristiche asset	ID operatore	Caratteristiche operatore
----------	-----------------------	--------------	---------------------------

Se un veicolo assegnato ad un operatore cambia nel tempo, dovrà essere inclusa anche una colonna per data e ora

Figura 7: schemi di dati per monitoraggio dello stato di salute del macchinario e predizione di potenziali guasti

### 2.2.2. *Progettazione delle funzioni*

L'obiettivo della progettazione delle funzioni è quello di descrivere e astrarre la condizione di integrità di un macchinario in un dato momento usando i dati cronologici raccolti fino a quel momento [9].

Esistono diverse tecniche di progettazione delle funzioni applicabili per la manutenzione predittiva ed ogni tecnica prevede l'assegnazione di etichette ai dati.

La scelta di una tecnica anziché un'altra dipende dai dati a disposizione e dalle problematiche da risolvere. Durante la generazione delle funzioni sarà necessario gestire i valori mancanti e la normalizzazione. Le funzioni applicabili sono quelle:

- di ritardo se create da dati temporali (sono di natura numerica)
- statiche se create da dati statici (variabili categoriche)

#### **Funzioni di ritardo**

Nei scenari analizzati e tratti per la predizioni i dati includono il *timestamp* che indica l'ora della raccolta di ogni singolo dato. Esistono diversi modi per creare delle funzioni dai dati temporali; due esempi di questi modi usati per le analisi predittive sono *l'aggregazione in sequenza* e *l'aggregazione in cascata*.

#### Aggregazioni in sequenza



Avviene attraverso le seguenti modalità [9]:

1. Per ogni record di un asset viene selezionata una finestra di dimensione  $W$  che corrisponde al n° di unità di tempo per cui si vuole calcolare le aggregazioni cronologiche. Es. di aggregazioni in sequenza sono:
  - a. Conteggi incrementali
  - b. Medie
  - c. Deviazioni standard
  - d. Misure della somma cumulativa (CUSUM)
  - e. Valori min e max della finestra
2. Lo step 2 può essere condotto alternativamente attraverso:
  - a. Il calcolo delle funzioni di aggregazione di sequenza usando i periodi  $W$  prima della data di quel record
  - b. L’acquisizione delle variazioni di tendenza, picchi e livelli usando algoritmi che rilevano anomalie nei dati.

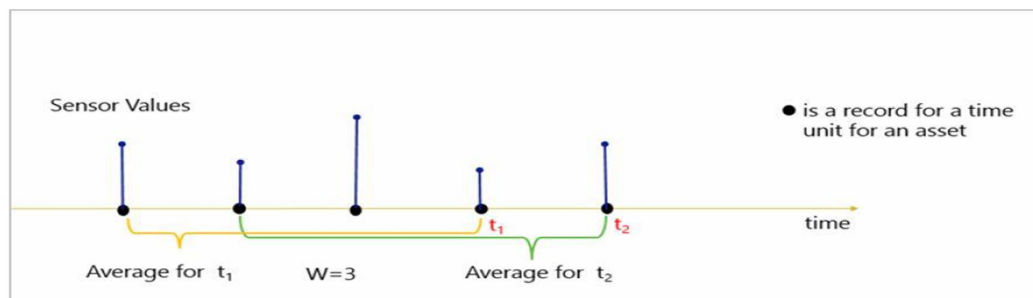


Figura 8: Funzione di aggregazioni in sequenza [9]

### Aggregazioni a cascata

La creazione delle funzioni temporali tramite aggregazione a cascata avviene attraverso la seguente modalità [1]:

1. Per ogni record di un asset si seleziona una finestra di dimensione  $W_k$  dove  $k$  rappresenta il valore o la finestra di dimensione  $W$  per cui si desidera creare la funzione di ritardo. Si usa:
  - a. k elevato per acquisire schemi di degradazione a lungo termine



- b. K piccolo per acquisire gli effetti a breve termine
2. Si usano le funzioni a cascata  $k$  ( $W_k, W_{(k-1)}, \dots, W_2, W_1$ ) per creare funzioni di aggregazione per i periodi precedenti la data e l'ora del record.

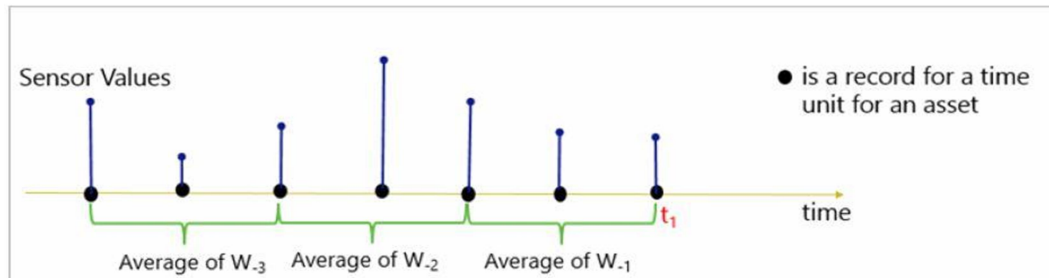


Figura 9 Funzioni di aggregazione a cascata [1]

#### Funzioni statiche

Per i macchinari sono disponibili specifiche tecniche come la data di fabbricazione, il numero di modello, la posizione, etc. Mentre le funzioni di ritardo sono di natura numerica, le funzioni statiche sono variabili categoriche nei modelli.

Nella figura che segue sono indicate le proprietà che possono essere utilizzate per le funzioni statiche definite per gli scenari di manutenzione predittiva elencati in precedenza.

Caso d'uso	Cosa può essere usato per creare la funzione
Errori degli interruttori	Proprietà dell'interruttore come le specifiche per la tensione, corrente e potenza Tipi di trasformatore Fonti di alimentazione

Figura 10 Applicazione delle funzioni statiche sugli scenari della manutenzione predittiva

### 2.2.3. Metodi di training e convalida

Lo step successivo alla definizione dei modelli previsionali è quello che prevede il training, la convalida ed il test del modello. La fase di training e test deve considerare gli aspetti variabili nel tempo per una migliore generalizzazione dei dati futuri.



Gli algoritmi di Machine Learning dipendono da iperparametri<sup>3</sup> in grado di modificare le prestazioni del modello. I valori degli iperparametri sono specificati dai data scientists all’atto della definizione del modello.

Un metodo per trovare i valori ottimali degli iperparametri è quello della **convalida incrociata** che prevede i seguenti passi [9]:

1. Si suddividono gli esempi in modo casuale in k sezioni
2.  $\forall$  set di valori degli iperparametri, l’algoritmo di apprendimento viene eseguito k volte
3.  $\forall$  iterazione
  - Gli esempi nella sezione corrente vengono usati come set di convalida
  - Il resto come set di training
  - L’algoritmo esegue il training sugli esempi di training
  - Le metriche delle prestazioni sono calcolate sugli esempi di convalida
4. Alla fine del ciclo,  $\forall$  set di valori degli iperparametri:
  - Si calcolano i valori delle metriche delle prestazioni k
  - Si scelgono i valori degli iperparametri con le prestazioni medie migliori

Nella definizione dei dataset di training e convalida bisogna tenere conto che i dati nei problemi di manutenzione predittiva:

1. sono registrati in serie temporali di eventi
2. possono essere ordinati cronologicamente in base all’ora di assegnazione di un’etichetta

---

<sup>3</sup> In statistica bayesiana, un iperparametro è un parametro di una distribuzione di probabilità a priori; il termine è impiegato per distinguerlo dai parametri del sottostante sistema in esame. Se si sta usando una distribuzione Beta per modellare la distribuzione di un parametro p di una distribuzione di Bernoulli, allora p è un parametro del sistema sottostante (la distribuzione di Bernoulli) mentre  $\alpha$  e  $\beta$  sono parametri della distribuzione a priori (la distribuzione Beta), quindi iperparametri [12].



Queste caratteristiche dei dati possono determinare il verificarsi di un problema nella definizione dei dataset di training, convalida e test; infatti, se si suddivide in modo casuale il set di dati tra training e convalida, il rischio è che alcuni esempi di training siano successivi agli esempi di convalida e quindi si avrebbe la stima delle prestazioni future dei valori degli iperparametri su dati arrivati prima dell'esecuzione del training del modello. Alla luce di questa considerazione, il metodo più adeguato a trovare valori ottimali degli iperparametri consiste nel:

- suddividere gli esempi di training e quelli di convalida in modo dipendente dal tempo in modo che gli esempi di convalida siano tutti successivi agli esempi di training
- per ogni set di valori degli iperparametri si esegue il training dell'algoritmo sul set di training
- si misurano le prestazioni del modello sullo stesso set di convalida
- si scelgono i valori degli iperparametri che mostrano le prestazioni migliori

Il modello finale è generato eseguendo il training di un algoritmo di apprendimento su tutti i dati, usando i valori degli iperparametri migliori trovati usando la suddivisione training/convalida o la convalida incrociata.

#### 2.2.4. **Metodi di test**

Dopo la compilazione, il test del modello è necessario per valutare le prestazioni del modello su nuovi dati.

Un modo per effettuare il test del modello consiste nella seguente procedura:

1. suddividere i dati in modo casuale in set di training, convalida e test
2. i set di training e convalida vengono usati per selezionare i valori degli iperparametri ed eseguire il training del modello con questi valori
3. le prestazioni del modello vengono misurate sul set di test.

Nel caso della manutenzione predittiva, un metodo efficace per il test del modello prevede i seguenti passi:



1. suddividere gli esempi in set di training, convalida e test in modo dipendente dal tempo
2. generazione del modello
3. misurazione delle prestazioni secondo questo schema:
  - a. suddividere ciascun set di dati in modo casuale
  - b. i set di training e convalida vengono usati per selezionare i valori degli iperparametri
  - c. il training del modello viene eseguito su tali valori
  - d. le prestazioni del modello vengono misurate sul set di test.

Quando le serie temporali sono fisse e facili da stimare, entrambi gli approcci generano valutazioni simili delle prestazioni future. Quando però le serie temporali non sono fisse oppure sono difficili da stimare, il secondo approccio permette di generare stime più realistiche delle prestazioni future rispetto al primo [9].

Le metriche di valutazione da considerare per i modelli predittivi sono:

#### **Accuratezza**

È la metrica più comune per descrivere le prestazioni di un classificatore. Rappresenta il grado di corrispondenza del dato teorico, desumibile da una serie di valori misurati (campione di dati), con il dato reale o di riferimento; in sostanza, indica la vicinanza del valore trovato a quello reale [10]

Come metrica di valutazione ha lo svantaggio di non riflettere le prestazioni reali perché molto sensibile alle distribuzioni dei dati [9].

#### **Frequenza di richiamo**

Nella pratica denota quanti guasti sono stati identificati correttamente nel set di test da parte del modello; quindi, tassi di richiamo elevati indicano che il modello individua correttamente i veri guasti.

#### **Precisione**

La precisione è il grado di "convergenza" di dati rilevati individualmente rispetto al valore medio della serie cui appartengono. La precisione è correlata alla frequenza di



falsi allarmi [11] Tassi di precisione più bassi corrispondono a maggiori quantità di falsi allarmi.

### F1 score

Nell'analisi statistica della classificazione binaria, l'F1 score è una misura dell'accuratezza di un test. La misura tiene in considerazione precisione e recupero del test, dove la precisione è il numero di veri positivi diviso il numero di tutti i risultati positivi, mentre il recupero è il numero di veri positivi diviso il numero di tutti i test che sarebbero dovuti risultare positivi (ovvero veri positivi più falsi negativi) [12]

### 2.3. Tipi di Modelli applicabili nella manutenzione predittiva

Una volta terminata la progettazione delle funzioni è possibile passare alla fase di modellazione con l'uso di serie temporali in quanto come definito i dati trattati in questo progetto sono **dati IOT** che di fatto sono :

- ✓ **serie temporali**, i sensori misurano i cambiamenti nel tempo in un determinato ciclo o stazione.
- ✓ **streaming di dati** che sono generati da una miriade di sensori che emettono un flusso continuo di dati.
- ✓ **dati in tempo reale**: i sensori acquisiscono ed emettono dati in tempo reale.

Tra i tipi di modelli applicabili in grado di prevedere il comportamento futuro di un componente si utilizzano modelli di :

- Classificazione
- Regressione
- Clustering
- Anomaly detection con Time Series Forecasting

Classificazione e regressione sono due tipi di modelli supervisionati: ovvero modelli per basati su dati raccolti storici e Clustering sono invece modelli non supervisionati su cui e storicamente non abbiamo una risposta al quesito a cui vogliamo rispondere. [13]

I Modelli supervisionati e non supervisionati differiscono rispetto al modo in cui il problema di Machine Learning viene strutturato. Nello specifico la Classificazione,





classification, è una tecnica di Analisi Predittiva utilizzata quando vogliamo indentificare la prevedere un potenziale guasto o anomalie e quindi si suddividere la nostra base dati in classi:

- che il guasto oppure no) (faliure vs non faliure)
- il prodotto è difettoso oppure no? (difetto vs. non difetto)
- a quale tipologia di difetto appartiene un prodotto difettoso? (difetto A vs. difetto B vs. difetto C)

La Regressione (Regression), invece è una tecnica di analisi predittiva utilizzata quando vogliamo stimare il valore atteso in futuro:

- quanta sarà la domanda o pezzi da produrre nei prossimi 7 giorni?
- quale sarà il tempo di vita dell’asset nei prossimi 30 giorni?

La differenza tra le tipologie di output scaturite da queste due diverse tecniche di analisi dei dati richiederanno l’utilizzo di differenti metriche di performance per testare la bontà del modello. E’ importante comprendere che, partendo da una base dati, si possono porre sia quesiti di Classificazione che quesiti di Regressione.

Clustering invece tra le tecniche di analisi predittiva di tipo non supervisionato è possibile ricorrere al clustering o analisi dei gruppi. Il clustering è un set di tecniche di analisi multivariata dei dati che consente di selezionare e raggruppare gli elementi omogenei in un insieme di dati (gruppi di clientela aziendale che hanno preferenze e comportamenti di acquisto simili).

In fine l’anomaly detection a seconda se i campioni di dati di un dataset vengono forniti con delle etichette assegnate da esperti in materia, le tecniche usate anche in questo caso vengono classificate in modelli con apprendimento supervisionato, semi supervisionato e non supervisionato. La principale differenza tra le tre tipologie di apprendimento è la natura del dataset utilizzato in fase di addestramento del modello. Quando ad ogni singola istanza del dataset è associata un’etichetta predefinita, parliamo di modelli con apprendimento supervisionato, viceversa quando il dataset non contiene alcuna etichetta ci troviamo nel caso di modelli con apprendimento non supervisionato. Si parla invece, di modelli semi supervisionati, quando una piccola porzione di istanze del dataset viene fornita etichettata mentre il resto dei dati senza etichette. Il processo di etichettatura di un dataset viene spesso eseguito manualmente da esperti del settore



e richiede, solitamente, un notevole sforzo sia in termini di analisi che in termini di tempo.

In generale, a causa del processo oneroso di etichettatura dei campioni di dati, in scenari reali, non sempre si hanno a disposizione dataset etichettati oppure il supporto di esperti del settore oggetto di valutazione, in tal caso può adottare tecniche di Deep learning [14] per rilevare punti di dati anomali all'interno delle serie temporali. L'esecuzione dell'analisi in tal caso è noto come analisi delle serie temporali.

Pertanto, partendo dall'assunzione di base che qualsiasi apparecchiatura nell'uso subisce un degradazione progressivo che si riflette nelle misurazioni dei sensori dell'apparecchiatura.

Si può quindi assumere che monitorando ed esaminando i valori dell'apparecchiature in questione nel corso del tempo attraverso diverse modelli statistici si può apprendere la relazione tra i valori dei sensori e la variazione dei valori che possono generare potenziali guasti.

La componente della piattaforma Digital Future che sarà deputato all'addestramento dei modelli statistici e di machine learning e quindi rappresenta la componente base di integrazione con la soluzione Fabbrica Intelligente sarà il **Data Governance**

#### 2.4. Fasi per la gestione della manutenzione predittiva

La Manutenzione Predittiva poggia le sue fondamenta come marcato in precedenza sulla possibilità di comprendere in tempo reale cosa stia succedendo al macchinario sotto esame.

Al fine di raggiungere tale obiettivo la soluzione **Fabbrica Intelligente** prima della fase di pura erogazione ed esecuzione del processo di manutenzione predittiva deve eseguire come si evince in Figura 11 una fase *preliminare*.

Questa fase è necessaria per acquisire dal macchinario dei dati oggettivi che permettano di definire lo stato di funzionamento dove sono presenti delle anomalie che hanno comportato dei segnali di guasto.



L’acquisizione dei dati è un processo molto importante perché i dati di scarsa qualità o poco significativi non possono risultare utili.

Inoltre sono necessari strumenti di misurazione adeguati, strutture d’interconnessione e programmi dedicati per poter riuscire a far fluire i dati dalla macchina a sistemi che possano analizzarli. Per cui la fase richiede del tempo perché per poter costruire un modello adeguato è in genere necessario avere una grande quantità di informazioni sulle condizioni in cui si verificano i guasti. I tempi di raccolta dei dati possono durare da qualche settimana a qualche mese.

Al processo di acquisizione dei dati segue poi la loro ri-elaborazione per renderli pronti ad essere interpretati in un modello. Infine viene costruito di modello predittivo.

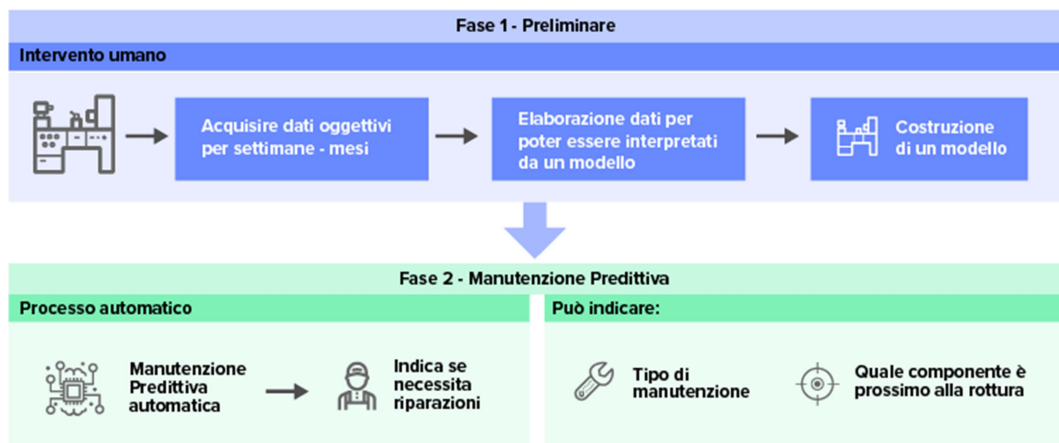


Figura 11: macro -fasi per l'approccio alla manutenzione predittiva

Ovviamente la manipolazione dei dati, anche in questa fase, non può essere lasciata al caso e l’interpretazione dei risultati in base al contesto richiede sempre l’intervento di data scientist esperto di dominio .

Terminate queste fasi si avrà un processo automatico nel quale i dati provenienti dalla macchina saranno raccolti, elaborati e interpretati da un modello che fornirà alert su possibile anomalie e attivano un processo automatico di intervento indicando sia il caso di anomalie ma anche il tipo di manutenzione che è necessario svolgere, e la componente si sta comportando in modo anomalo ed la prossima alla rottura.

Inoltre sono numerosi fattori possono contribuire anche sulla qualità di un prodotto realizzato con il supporto di specifiche attrezzature, uno delle fonti più comuni di



problemi di qualità può essere strettamente legata proprio all'attrezzatura difettosa che non è stata mantenuta correttamente. Quindi, il monitoraggio delle condizioni di macchine e componenti come ventole di raffreddamento, cuscinetti, turbine, ingranaggi, cinghie e il mantenimento di uno stato di funzionamento desiderabile diventa molto cruciale.

Quando una macchina o un componente si guasta, viene eseguita una manutenzione correttiva per identificare la causa del guasto e decidere le procedure di riparazione necessarie per mantenere e riavviare la macchina alle sue normali condizioni di lavoro.

Tuttavia, poiché la macchina si è già guastata senza alcun preavviso, è necessario del tempo per procurarsi e riparare il componente guasto. Pertanto, è necessario prendere in considerazione una strategia di manutenzione per ridurre al minimo i tempi di fermo del servizio. Ma le macchine e i loro componenti si degradano nel tempo e il tempo di guasto non è noto in anticipo. Pertanto, le strategie di manutenzione basate sul tempo vengono utilizzate principalmente per monitorare le condizioni di macchine e attrezzature. Qualsiasi macchina, che sia una macchina rotante (pompa, compressore, turbina a gas o a vapore, ecc.) o una macchina non rotante (scambiatore di calore, colonna di distillazione, valvola, ecc.) raggiungerà prima o poi un punto di cattiva salute. Questo punto potrebbe non essere quello di un guasto o di un arresto effettivo, ma quello in cui l'apparecchiatura non agisce più nel suo stato ottimale.

Questo segnala che potrebbe essere necessario attività di manutenzione per ripristinare il pieno potenziale operativo. In termini semplici, identificare lo "stato di salute" delle apparecchiature può avvenire con un monitoraggio delle condizioni [15].

Il modo più comune per eseguire il monitoraggio delle condizioni è esaminare ogni misura di sensore dalla macchina e imporre un limite di valore minimo e massimo su di essa. Se il valore corrente rientra nei limiti, la macchina è integra. Se il valore corrente non rientra nei limiti, la macchina non è salutare è necessario inviare un allarme dal sistema di monitoraggio e un alert al referente per avviare un workorder di ispezione come rappresentato in Figura 12

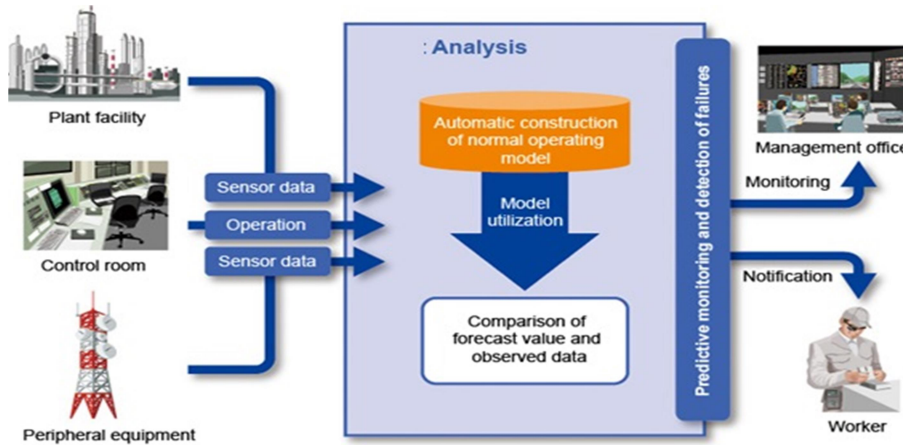


Figura 12 : Avvio Alerting

È noto che questa procedura di imposizione di limiti di allarme codificati in modo rigido invia un gran numero di falsi allarmi, ovvero allarmi per situazioni che sono effettivamente stati sani per la macchina. Ci sono anche allarmi mancanti, ovvero situazioni problematiche ma non allarmate. Il primo problema non solo fa perdere tempo e fatica, ma anche la disponibilità dell'attrezzatura. Il secondo problema è più cruciale in quanto porta a danni reali con i costi di riparazione associati e la perdita di produzione.

Entrambi i problemi derivano dalla stessa causa: la salute di un'apparecchiatura complessa non può essere valutata in modo affidabile sulla base dell'analisi di ciascuna misurazione da sola. Dobbiamo piuttosto considerare una combinazione delle varie misurazioni per ottenere una vera indicazione della situazione.

Con il presente progetto sono state analizzate e sviluppate varie tecniche che possono essere utilizzate per rilevare l'insorgenza di guasti che si verificano nelle macchine.

Le principali tecniche che s'intendono attuare e sperimentare per rispondere a predire situazioni di probabili guasti nel presente progetto di Ricerca Digital Future è l'applicazione di tecniche di machine learning (apprendimento automatico), analisi dei dati. Tale approccio sarà basato sull'adozione di un monitoraggio continuo dei dati e su



condizione attraverso il Framework *Condition Based Maintenance* (CBM), manutenzione basata su condizione.

*Manutenzione su condizione eseguita a fronte di una previsione derivante dall’analisi e dalla successiva valutazione dei parametri significativi afferenti al degrado dell’entità d’analisi (norma UNI EN 13306:2003)»*

La CBM è la più moderna politica di manutenzione, è una specializzazione della manutenzione preventiva che include una combinazione di attività automatiche o manuali di monitoraggio delle condizioni e la loro analisi. Ciò significa quindi che prima di agire effettivamente sui componenti si effettua una rilevazione del loro stato di salute e si interviene solamente nel caso in cui venga evidenziata una certa probabilità di rischio. La rilevazione di una condizione di anomalia si ha quando un qualche parametro fisico della macchina risulta essere non conforme con il normale funzionamento. Esempi tipici sono l’aumento del rumore, delle vibrazioni o della temperatura. Essa può essere messa in atto sia dall’uomo, ad esempio da manutentori con apposite ispezioni o da operatori esperti che si accorgono delle variazioni durante l’uso, oppure tramite appositi sensori, che servono al monitoraggio continuo dei parametri ritenuti significativi.

L’idea alla base della CBM è pertanto proprio quella di analizzare i parametri operativi di funzionamento di un asset, ricavando, secondo modelli appropriati, il tempo residuo prima del guasto e riconoscendo la presenza di un’anomalia che si sta sviluppando, attraverso la scoperta e l’interpretazione di deboli segnali premonitori del guasto finale.

Se si analizzano i componenti e le funzionalità necessarie per implementare un sistema di Condition-Based Maintenance (CBM), si ottengono i seguenti elementi:

- Sensori, installati sulle macchine per effettuare misurazioni riguardo i parametri fisici di interesse;
- Comunicazione, per poter trasmettere in qualche modo i dati raccolti a un centro di aggregazione;
- Archiviazione, per mantenere lo storico dei valori dei sensori, ed eventualmente integrarlo con una lista di eventi e attività;



- Analisi, per estrapolare correlazioni tra variabili e stato della macchine, e di conseguenza riconoscere e predire malfunzionamenti.

E come si evince pertanto una forte corrispondenza fra essi e i servizi e le tecnologie abilitanti dell’Industry 4.0 e della Smart Factory. I vantaggi provenienti dall’applicazione di questa metodologia sono:

- la riduzione dei guasti, e in particolare delle avarie distruttive;
- la possibilità di pianificare le manutenzioni sulla base della reale situazione d’efficienza e della previsione di vita residua (RUL) degli apparati;
- l’ottimizzazione del piano delle manutenzioni sulla base del comportamento rilevato nel tempo;
- esecuzione delle operazioni di manutenzione in base alle condizioni correnti delle risorse. verificate attraverso ispezioni visive, test e analisi dei dati sulle prestazioni (raccolti da sensori e/o strumenti diversi).

L’ulteriore vantaggio della CBM è dato inoltre dal fatto che si tratta di un approccio che integra la diagnostica basata sulle condizioni (monitoraggio preventivo) con metodologie predittive complesse per prevedere la probabilità di accadimento di una anomalia e la finestra di tempo entro la quale una potenziale rottura può verificarsi. Infatti, in conformità alla norma *UNI 10147*, la CBM condizione utilizza principalmente test non distruttivi, ispezioni visive e analisi dei dati per conoscere lo stato di salute delle macchine, permettendo così di predire il verificarsi dei guasti e aumentare la vita dei componenti, con i conseguenti vantaggi sui costi di manutenzione. A differenza dei metodi preventivi, la CBM fornisce un framework per il monitoraggio dello stato di salute di uno o più dispositivi valutando anche l’interazione tra le differenti componenti (ad es. correlazioni tra feature) individuando un valore di soglia, oltre il quale il sistema può presentare una significativa probabilità di guasto. Il processo di Condition-based maintenance prevede i seguenti fasi:

1. Acquisizione dati ossia il processo di raccolta di tutte quelle informazioni che si ritengono rilevanti per poter dedurre lo stato della macchina o dei suoi componenti;



2. Elaborazione Dati: ossia la gestione e l'analisi dei dati raccolti per poterne fornire un'interpretazione e la loro trasformazione in conoscenza sulla macchina;
3. Decisioni di Manutenzione: ossia la definizione di una politica di decisioni riguardo alle azioni di manutenzione da eseguire che dipende anche dalle informazioni aggiuntive ottenute tramite lo step di elaborazione. Quest'ultima fase rappresenta la fase finale del processo

La distinzione principale all'interno delle applicazioni di CBM è quella fra diagnostica e prognostica. Lo scopo di un sistema di diagnostica è quello di rilevare ed identificare un guasto quando quest'ultimo accade. Nel caso ideale, ciò significa quindi monitorare un sistema, segnalare quando qualcosa non sta funzionando nel modo atteso, indicare quale componente è colpito dall'anomalia e specificare la tipologia di anomalia. La prognostica invece ha come obiettivi quello di determinare se un guasto è prossimo al verificarsi o quello di dedurre la probabilità di accadimento. Ovviamente, essendo la prognostica un'analisi a priori, può fornire un contributo maggiore per quanto riguarda la riduzione dei costi degli interventi, ma è un obiettivo più complesso da raggiungere.

Quindi in questo progetto si propone come si vede Figura 13 in di utilizzare contemporaneamente soluzioni di diagnostica e prognostica. La loro combinazione fornisce infatti due preziosi vantaggi :

- La diagnostica permette di intervenire al supporto delle decisioni nei casi in cui la prognostica fallisce; questo scenario è infatti inevitabile, in quanto ci sono guasti che non seguono un modello tale da essere previsti, e anche i guasti che sono prevedibili con buona precisione non riescono a essere individuati nella totalità dei loro accadimenti;
- Le informazioni ottenute tramite applicazioni di diagnostica possono essere utilizzate come input aggiuntivo ai sistemi di previsione, permettendo quindi la realizzazione di modelli più sofisticati e precisi.

Gli obiettivi principali dell'approccio sono quindi:





- Predizione del Tempo Residuo di Funzionamento (Time to Failure);
- Previsione di eventuali anomalie entro un determinato periodo di tempo o differenti finestre temporali;
- Individuazione di pattern o caratteristiche specifiche che possono interagire e influenzare il corretto funzionamento della macchina.

Il cuore dell’approccio CBM sarà il layer che fornisce stime di predizione, sulla base di tecniche di machine learning e tecnologie big data applicate ai dati di anomalia generati dal sistema di monitoraggio e dall’analisi dei trend. L'applicazione di approcci di machine learning per la predizione di probabili guasti si basa sulla costruzione di modelli che “imparando” da dati storici e casi noti, sono in grado di identificare o classificare situazioni di potenziali rotture. Per eseguire tale monitoraggio, è necessario definire indicatori di performance articolati per tipologia di asset che si possono basare su: performance fisiche delle risorse, frequenza dei fallimenti e attuali manutenzioni preventive messe in atto nei sistemi. Le misurazioni possono essere inoltre raccolte in modo non distruttivo su base continua o ad intervalli regolari, mediante sensori e con strumenti di monitoraggio delle condizioni (strumenti portatili) remoti.

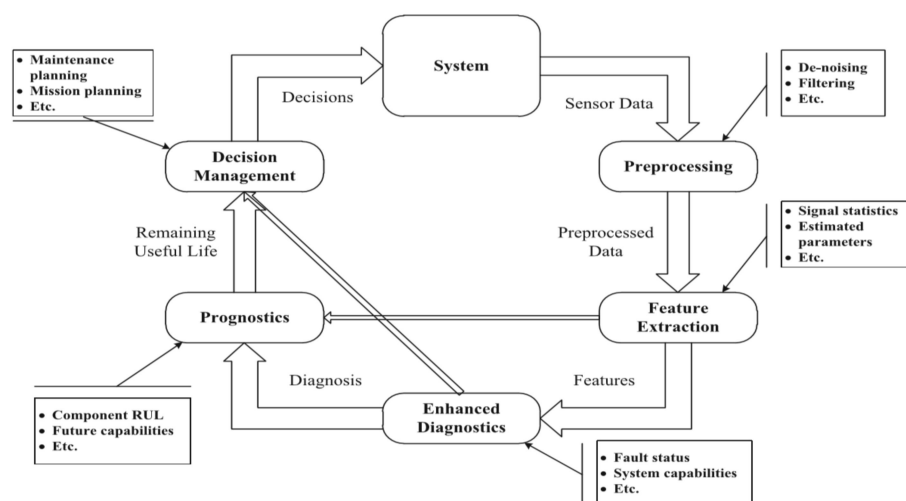


Figura 13 :Esempio di un sistema di CBM che combina diagnostica e prognostica [16]



### 3. Approccio integrazione Piattaforma Digital Future

In funzione di quanto su menzionato per l'adozione del framework *Condition Based Maintenance* (CBM) e quindi per:

- L'acquisizione e analisi preliminare dei dati.
- Il Monitoraggio continuo dello salute dell'asset e l'individuare anomalia che possono causare un potenziale guasto di un determinata severità.
- l'elaborazione e trattamento dei dati attraverso algoritmi di data mining e di machine learning; e/o deep learning;

Sono state eseguite diverse valutazioni al fine di individuare una piattaforma abilitante alla coniugazione tra sensori, e sistemi Asset Management remoti e la capacità di elaborazione dei dati ricavati tramite soluzioni di Business Intelligence di Realtime Monitoring sistemi base della soluzione base della Fabbrica Intelligente.

Nella fattispecie è stata valutata l'adozione tra una piattaforma open source IOT o la piattaforma **SAP Cloud Platform** in quanto Exprivia segue storicamente il mercato Manifatturiero, attraverso la suite di prodotti SAP, ed in particolare nella fornitura di soluzioni per le aree: magazzino, gestione impianti, supply chain, ecc...

la piattaforma Paas (Platform as a Service) **SAP Cloud Platform** ha il vantaggio di avere a disposizione in un'unica piattaforma molti dei servizi utili alla realizzazione di un'applicazione IOT traendo vantaggio dalle funzionalità già implementate con un conseguente TTM (Time to Market) più veloce;

Ma dall'analisi e sperimentazione è emerso che non avrebbe un elevato livello di flessibilità e le funzionalità attive e standard messe a disposizione dalle componenti IOT della piattaforma SAP Cloud Platform a disposizione, avrebbero lunghi tempi di progettazione per adattamento a particolare scenari di predizione. Sorgono delle difficoltà anche per la manutenzione dell'ERP perché si viene a creare un vincolo tra cliente e vendor in quanto non vi è la possibilità di modificare il codice sorgente dei



componenti acquistati. inoltre tali soluzioni, per l’elevato costo delle licenze SAP, sono generalmente al di fuori delle capacità di spesa delle PMI.

Pertanto grazie alla opportunità fornite dal progetto, Exprivia si intende costruire un’offerta innovativa sul segmento di mercato relativo ai processi di produzione e post-vendita, maggiormente indirizzato verso le esigenze delle PMI, colmando le sue attuali carenze in ambito. con l’adozione di una soluzione open source come la piattaforma DF ,non solo è senza costi di licenza ma soprattutto risulta adattabile in ecosistema basati su microservizi come Fabbrica Intelligente e su cui si può sviluppare microservizi applicativi a diversi scenari e integrabile anche alla soluzione SAP.

Pertanto l’approccio architetturale per un sistema *Condition Based Maintenance* e che consentirà l’**interoperabilità** tra le applicazioni di Fabbrica Intelligente e la piattaforma Digital Future, definita come ‘Digital Future IOT Platform Integration’ come indicato nel TR 4.2 [1] è un approccio serverless o FaaS (Function as a Service), che ben si adatta al tipo di interazione fra due soluzioni.

Infatti, questo tipo di approccio, ormai tipico del mondo dei servizi cloud, si adatta perfettamente nei contesti in cui le applicazioni interagiscono sulla base di eventi asincroni e che operano su sorgenti di dati distribuite;

Non a caso è molto utilizzato nell’elaborazione di dati e delle segnalazioni provenienti da sistemi IoT e soprattutto per fare interagire sistemi eterogeni basati su modelli diversi come sistemi CMMS ed ERP (SAP) ma complementari nella gestione di un processo di un asset e del suo ciclo di vita. Pertanto per garantire:

- ***l’interoperabilità tra le diverse applicazioni integrate;***
- ***la standardizzazione dei dati scambiati;***

sarà necessario pertanto disporre una soluzione globale come la Piattaforma DF che attraverso il middleware permette di interagire diversi i sistemi distribuiti e diversificati dal punto di vista formativo renderli interoperabili non solo al livello sintattico ma anche semantico in modo da poter fornire in maniera appropriata le giuste informazione al momento giusto.



### 3.1. Approccio integrazione con Smart Factory

Nello specifico si descrive l'architettura integrata tra la piattaforma DF e la Fabbrica Intelligente è suddivisa logicamente e principalmente dai moduli come si evidenzia in Figura 14:

- **IOT Gateway**, ha la funzione di stabilire connessioni senza interruzioni ai sensori e alle apparecchiature. E' deputato principalmente alle comunicazioni, ai protocolli, alla sicurezza, e alla topologia di rete e di infrastruttura. I dispositivi sono abilitati mediante IP e identificati in modo univoco nella rete.
- **Data Ingestion**, I dati vengono reperiti da fonti eterogenee (serie temporali, flussi di eventi, log stream, fonti strutturate, fonti semi-strutturate e non strutturate) in tempo reale o in modalità batch attraverso il Data Ingestion Layer, e infine normalizzati per essere compliant con i formati standard aziendali.
- **Data Lake**, modulo deputato alla fase di Data Analysis ed componente core del middleware della Piattaforma DF. Utile nel contesto della Fabbrica Intelligente per la generazione dei modelli statistici e di machine learning, ed reperimento e elaborazione in tempo reale dei dati storici per la generazione di dati previsionali e per il rilevamento in tempo reale di anomalie del macchinario e avvio di manutenzione predittiva o problematiche nel processo produttivo come numero di pezzi prodotti non in linea con quelli richiesti/previsti nell'ordine di produzione
- **Data, Monitoring and Alerting** modulo costituito da strumenti di data visualization della Fabbrica Intelligente in cui si recuperano i dati real-time aggregati e attraverso dashboard si offre una vista riepilogativa per i diversi componenti delle metriche ossia kpi operativi analizzate e offre : la rilevazione di Alert specifici per il caso d'uso rappresentato; le regole di attivazione degli alert devono essere definite a priori in base al caso d'uso rappresentato.
- **Remote Asset Mangament** modulo di Knowledge base e dove viene modellato anche il modello digitale dell' asset.
- **Workflow Management** modulo per la gestione dei processi di intervento.



Ed la dove è possibile Edge Compute, costituito da server gateway o servizi router ed esegue il calcolo in tempo reale al fine di gestire localmente all'apparecchiatura e velocizzare le analisi sui flussi di dati per i controlli a bassa latenza.

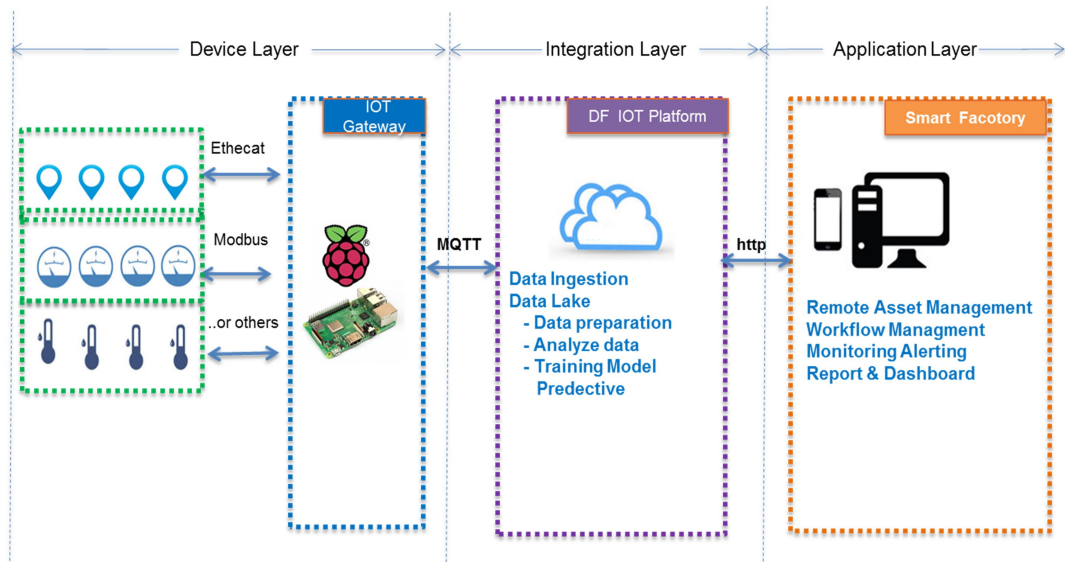


Figura 14 - Architettura Logica Fabbrica Intelligente.

In particolare in un'architettura IOT come quella proposta in questo progetto e nei scenari applicativi analizzati, il Raspberry PI fungerà da vero e proprio dispositivo Internet Gateway ma per lo sviluppo di componenti dedicate al controllo dei sensori e dei device è necessario installare sul Raspberry un sistema operativo Linux e delle

librerie Python. E' possibile utilizzare servizi quali Google Cloud Messaging (GCM) da utilizzare come web server per effettuare l'upload dei dati provenienti dai sensori che possono in seguito essere inviati ad una qualunque altra piattaforma.

Ma nella Figura 15 si offre una rappresentazione di come una scheda Raspberry PI può interagire con una piattaforma IOT mediante invio dei dati ad un broker MQTT.

Pertanto nel presente progetto Il comportamento reale di scambio tra i sensori e la Piattaforma Cloud avvierà potenzialmente attraverso il Raspberry Pi.

Quest'ultimo comunica con La piattaforma tramite protocollo MQTTs, ed dal momento che nell'implementare tale meccanismo non sono sempre sarà possibile avere a



disposizione dei sensori reali da cui prendere dati in real-time, sarà necessario per i diversi scenari applicativi implementare dei clients che attraverso il protocollo MQTT simulassero l’invio di dati di tali sensori e/o eseguiranno le prime fusioni o aggregazioni di dati.

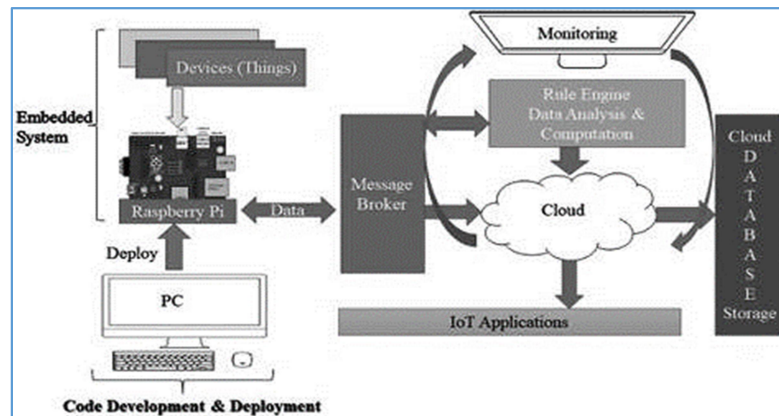


Figura 15: - Architettura IOT basata su interazione tra Raspberry Pi e cloud

MQTT (Message Queue Telemetry Transport) è un protocollo di comunicazione open-sourcedi tipo M2M (Machine to Machine) adatto al moderno scenario IoT (Internet of Things) . Il protocollo MQTT è utilizzato per dispositivi vincolati e con larghezza di banda limitata, e per reti ad alta latenza. E’ un protocollo multi-a-molti molto leggero asato sul meccanismo di Messaging Transport Publish/Subscribe, e pertanto si adatta bene per collegare piccoli dispositivi a reti vincolate. Si basa su un meccanismo di scambio di messaggi tra molteplici client attraverso un broker centrale (come mostra La Figura 16): la rete MQTT contiene un broker MQTT, che fa da mediatore tra i client MQTT.

Un client MQTT può essere un Publisher o un Subscriber: i publisher hanno il compito di inviare (Publish) messaggi su un Topic (argometo), che rappresenta un canale tematico di comunicazione, mentre i Subscriber hanno il compito di “isciversi” ad un topic al fine di ricevere le informazioni inviate dai Publisher.

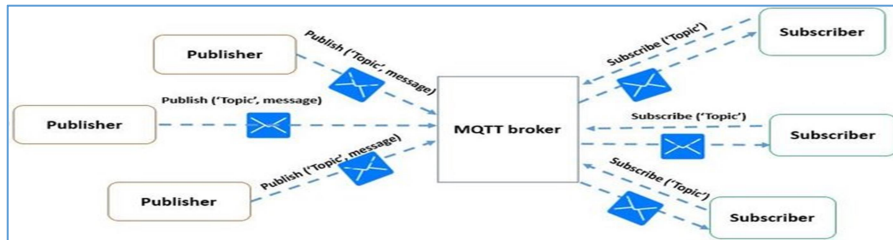


Figura 16 - Struttura Publish-Subscribe di una rete MQTT

### 3.2. Approccio integrazione con SAP

Al fine di rendere analizzabile la possibilità di sviluppare microservizi applicativi a diversi scenari e integrabile anche soluzione SAP da cui sono possibile recuperare informazioni anche dalle linee produttive a cui sono sottoposti gli impianti oggetto di manutenzione. Pertanto in questo capitolo analizziamo quale potrebbe essere l’approccio di integrazione della Piattaforma DF con le componenti della piattaforma SAP Leonardo adotta attualmente da Exprivia i cui contesti di analisi saranno descritti nei paragrafi 4.5 servizi per il monitoraggio delle linee produttive. In particolare quest’ultima è costituita da :

- *SAP Cloud Platform IoT Service Cockpit*: interfaccia grafica del servizio SAP Internet of Things che consente di effettuare la definizione e la gestione dei device, assegnare ad essi sensori e specificare le loro capacità, monitorare il flusso dei dati in tempo reale e gestire l’accesso utenti in ambiti diversi; nel presente progetto è stata usata per l’implementazione del Data Model del singolo device fisico ovvero per costruirne la rappresentazione digitale in termini di capabilities associate alle grandezze misurate dai sensori ad essa collegati;
- *SAP IoT Application Enablement*: offre servizi HTTP di tipo REST e OData usati per creare un ambiente per la gestione e il monitoraggio di dati forniti da sensori generati da dispositivi che fanno parte del mondo Internet of Things. L’applicazione **Thing Modeler Uis** è deputata alla modellazione del device in termini di oggetto virtuale (Thing) e per l’associazione del device fisico descritto dal Data Model al suo gemello virtuale (**Digital Twin**), attività nota come Thing onboarding che consente il



recupero dati forniti dai sensori e la messa a disposizione per le applicazioni sviluppate in ambiente SAP Web IDE per lo sviluppo di dashboard.

- *SAP Web IDE Full-Stack*: ambiente di sviluppo integrato (IDE) basato su Web per lo sviluppo di applicazioni SAP end-to-end full-stack; risulta costituito da moduli integrati che interagiscono tra loro e con qualsiasi sistema SAP.

Quest’ultimo consente di gestire il ciclo di vita completo dell’applicazione (sviluppo semplificato, test, build, rilascio) e di ampliare applicazioni esistenti sia per utenza di tipo consumer che business. Tramite l’utilizzo di SAP Web IDE è possibile sviluppare applicazioni in tempi ridotti e valutare subito l’esperienza utente (User Experience). Nell’ambito del presente progetto pertanto nei casi di monitoraggio della linea produttiva sarà analizzato il modulo SAP Web IDE Full-Stack utilizzato per lo sviluppo delle applicazioni SAP Fiori e sarà deputato alla gestione dei rilasci degli ordinativi di produzione (Release Orders) e alla visualizzazione della media dei dati di telemetria interessati e del risultato predizione dello stato di salute del macchinario (PredictiveModelLSTM) . Mentre SAP Cloud for Customer (C4C), piattaforma di servizi software (SaaS) per vendite e servizi che nel presente progetto potrà essere adottato per la creazione e gestione dei Service Ticket per la notifica delle anomalie riscontrate sul macchinario. Nei prossimi capitoli si analizzano e si descrivono possibili scenari applicativi.

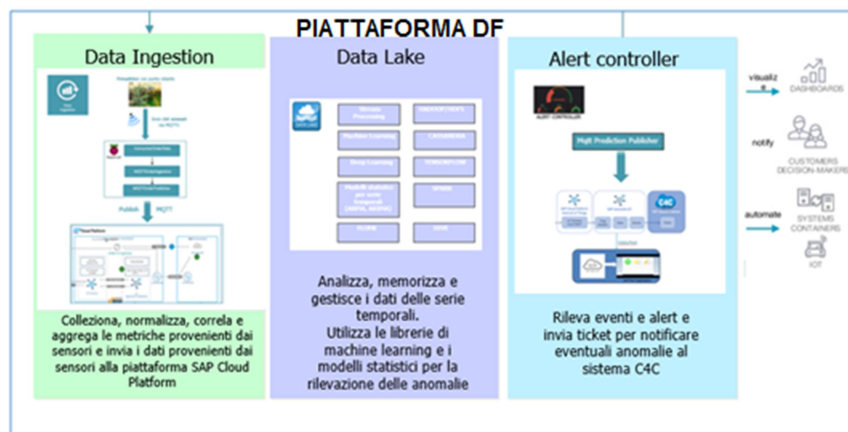


Figura 17: Architettura Integrazione SAP-Piattaforma DF





## 4. Servizi Piattaforma per la Fabbrica Intelligente

Al fine di analizzare e individuare con il supporto della piattaforma DF le modalità, l'approccio più opportuno e le macro fasi generali da attuare per individuare algoritmi e creare modelli produttivi in funzione delle problematiche in questa sezione si descrivono per mezzo di alcuni esempi applicativi simulati all'interno del laboratorio (ILAB), i servizi fondamentali individuati per la gestione della manutenzione predittiva che verranno esposti dalla piattaforma Digital Future per essere utilizzati dalle diverse soluzioni verticali facenti riferimento al dominio Fabbrica Intelligente e per gestire i diversi scenari esposti. Nello specifico sono stati presi in esame casi pilota sono stati analizzati i parametri di telemetria e sono stati individuati dei microservizi (Servizi Function as a Service), tramite i quali sarà possibile innescare eventi, automatismi o eseguire delle operazioni applicabile su Digital Future IOT Platform Integration e che gestiranno dati provenienti da sensori o da sistemi esterni ERP, CRM, CMMS e per qualsiasi scenario della Fabbrica Intelligente. L'obiettivo che ci si pone è quello di riuscire a dedurre lo stato di salute delle macchine e dei loro componenti critici, ed usare queste informazioni per pianificare quando intervenire con sostituzioni o riparazioni.. Lo schema generale di risoluzione prevedrà quindi di utilizzare dati telemetria posizionati dalle macchine, comunicare in tempo reale queste informazioni tramite simulatori alla Piattaforma DF la quale permette di aggregazione analizzare, ed applicare modelli di machine learning per estrapolare informazioni utili riguardo lo stato di salute delle macchine, che verranno utilizzati a supporto delle decisioni e delle pianificazioni degli interventi di manutenzione

### 4.1. Servizi Comuni

I servizi base quindi identificati sono: Device Management Data Ingestion , Data Preparation.

Essi saranno allestiti sulla piattaforma e potranno essere personalizzati in funzione del contesto applicativi e funzione della problematica e ci saranno servizi custom dipendenti



strettamente al contesto e in funzione della tipologia di creazione e addestramento di modelli predittivi che si intende attuare.

fra le tre tipologie di approccio al problema riportate ci si concentra nella descrizione di lavori che utilizzano strategia data-driven in quanto, anche se molti di essi sono pensati e testati in scenari specifici, l'idea e il principio di funzionamento sono facilmente generalizzabili ed utilizzabili anche in contesti differenti. Si riportano quindi casi con l'adozione di modelli presenti in letteratura ritenuti più significativi, evidenziando il tipo di problema di CBM che risolvono e la tipologia di dati che utilizzano.

#### 4.1.1. *Device Managment*

Per la realizzazione dei casi pilota in oggetto il primo passo è stato quello di configurare correttamente l'interazione con la piattaforma Digital Future, mediante la registrazione del device e la creazione di uno specifico topic sul broker EMQ per le operazioni di scambio dati mediante il paradigma publish/subscribe.

La fase di censimento ha scopo di eseguire la registrazione del sensore e autorizzare il sensore ad inviare dati in modo da memorizzarli, renderli persistenti ed eventualmente elaborarli .

L'operazione di registrazione del sensore sulla piattaforma e configurazione si articola come si evince nei seguenti passi:

1. Censimento di Anagrafica Asset e sensori IoT e misure da rilevare nella componente Asset Managment da parte del supervisore;
2. Registrazione utenza associata al tipo di dispositivo (Device type).
3. Connessione e configurazione plug & play del tipo di dispositivo che invierà dati sul topic MQTT in Piattaforma DF.

In particolare la configurazione del device comporta l'invocazione di un servizio “Registrazione device Type”, il quale crea 2 topic sul broker MQTT EMQ rispettivamente di ingresso e uscita e 2 topic su Kafka rispettivamente di ingresso e uscita, 1 connettore MQTT, 1 connettore al database Cassandra e uno Schema Registry.



Successivamente come mostrato in Figura 18 è necessario configurare un client mqtt per l'invio del messaggio sul topic plug play/username\_dispositivo e dal termine di questa procedura descritta si è autorizzati all'invio dei propri dati di telemetria al relativo topic. Il modello di dati inviato sarà informato json costituito da:

- Id : Identificativo unico device;
- Type: Tipo di entità. Deve essere uguale a Device.
- Versione del device
- Timestamp dell'invio dell'evento
- nome della rilevazione
- valore della rilevazione quindi la misura rilevata
- ore rilevazione
- timestamp della rilevazione
- nome asset sui è stata raccolta la rilevazione
- frequenza di rilevazioni

La pubblicazione dei dati sul topic EMQ in Piattaforma DF comporterà automaticamente la pubblicazione dei dati sul relativo topic Kafka generato con la procedura di registrazione.

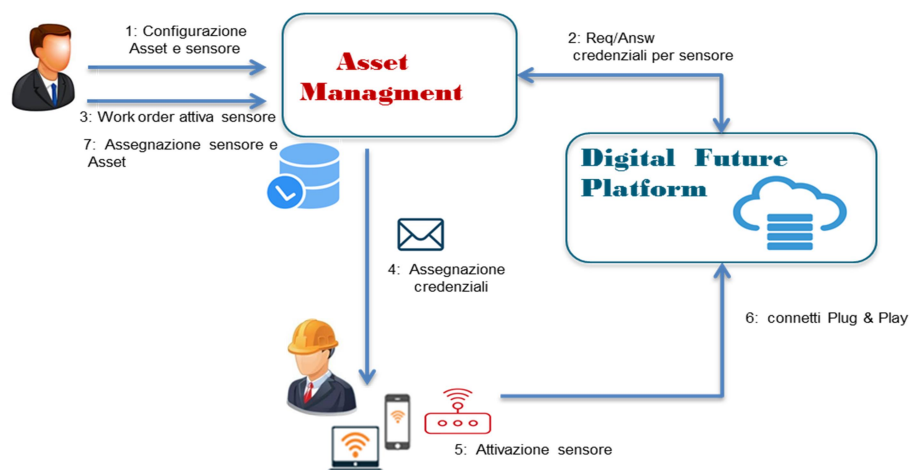


Figura 18: configurazione device



#### 4.1.2. *Data Ingestion*

Come su menzionato nel capitolo 3 il reale scambio tra i sensori e la Piattaforma Cloud e quindi acquisizione de dati avverrà potenzialmente attraverso il Raspberry Pi (IOT Gateway).

Quest’ultimo comunica con la piattaforma tramite protocollo MQTTs, ed dal momento che nell’implementare tale meccanismo non sempre sarà possibile avere a disposizione dei sensori reali da cui prendere dati in real-time, sarà necessario per diversi scenari applicativi implementare dei clients che attraverso il protocollo MQTT simulassero l’invio di dati di tali sensori e/o eseguiranno le prime fusion o aggregazioni di dati.

A tal fine per simulare il comportamento del macchinario, per i diversi scenari si utilizzerà un simulatore . Il simulatore potrà inviare periodicamente un set di dati, tramite il protocollo MQTT attraverso il servizio il quale reperimento dei dati provenienti dai sensori in tempo reale e la normalizzazione dei dati nel formato JSON conforme al datamodel definito sulla piattaforma Digital Future;

MQTT (MQ Telemetry Transport) è un protocollo di comunicazione di tipo M2M (Machine to Machine) adatto al moderno scenario IoT (Internet of Things). Successivamente Il client MQTT, inviando messaggi al Broker (**MQTT**) **EMQ**.

**EMQ** broker di messagistica MQTT distribuito e completamente open-source, altamente scalabile e indicato per applicazione IoT, in grado di gestire decine di milioni di client contemporaneamente.

Nella versione più aggiornata, EMQ supporta a pieno le specifiche del protocollo MQTT V5.0 e può essere dimensionato ad oltre 10 milioni di connessioni MQTT simultanee su un cluster.

EMQ è installato sulla piattaforma Digital Future è permetterà di agganciare il topic MQTT al topic Apache Kafka e in questo modo consentirà il trasferimento dei dati su Kafka dove potranno essere elaborati a sua volta Apache Spark.

Il collegamento tra il topic MQTT e il topic Kafka verrà implementato da un Kafka Connector come mostrato in Figura 19.

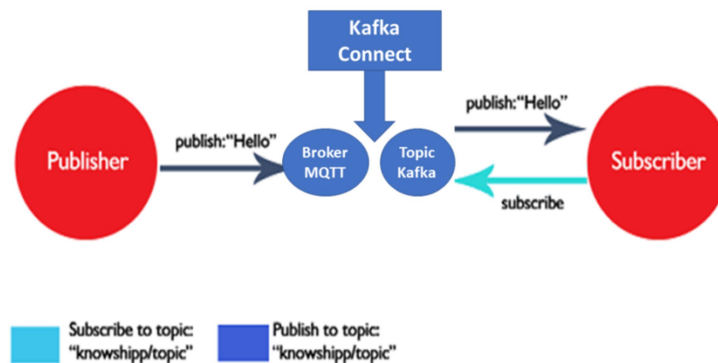


Figura 19 Rappresentazione del collegamento tra Apache Kafka e il protocollo MQTT.

**Apache Kafka**<sup>4</sup>, è piattaforma open-source distribuita per lo stream processing dei dati in tempo reale, ossia per l’elaborazione in tempo reale di dati generati in maniera continua da migliaia di fonti contemporaneamente.

Combina funzionalità di messaggistica, archiviazione ed elaborazione di flussi di dati per la memorizzazione e l’analisi dei dati storici in tempo reale.

Per le sue caratteristiche è quindi spesso adottata nelle applicazioni IoT, in quanto in grado di gestire l’enorme quantità di dati in tempo reale provenienti con flusso continuo da più sensori. In questo progetto Apache Kafka, installato sulla piattaforma Digital Future, sarà utilizzato come sistema di messaggistica.

Osservando la Figura 20, i processi che pubblicano messaggi su un topic Kafka vengono denominati producers, mentre i processi che effettuano la sottoscrizione al topic ed effettuano l’elaborazione dei feed relativi ai messaggi pubblicati vengono denominati consumers. Kafka viene eseguita su un cluster che consiste di uno o più servers ognuno dei quali viene denominato broker. Ad alto livello possiamo dire che i producers mandano messaggi in rete al cluster Kafka che li mette a disposizione dei consumers. La

<sup>4</sup> “Apache Kafka”, [“https://kafka.apache.org”](https://kafka.apache.org)



comunicazione tra i client e i server avviene con un semplice ma altamente performante protocollo TCP.

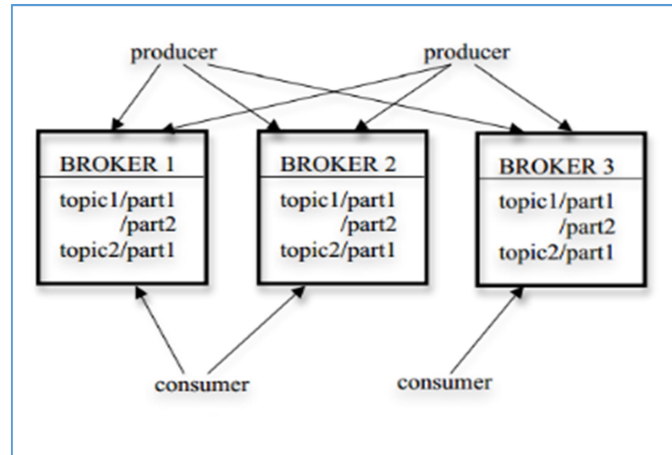


Figura 20 - Kafka Architecture

Mentre **Apache Kafka Connect**, consente di integrare kafka con altri sistemi (Figura 21). La finalità è quella di semplificare l’aggiunta di nuovi sistemi a pipeline pre-esistenti di data streaming scalabili e sicure sfruttando al massimo le potenzialità di Kafka quali: l’elevato throughput, la scalabilità e l’affidabilità. L’MQTT Connector presente in Kafka-Connect è un plug-in per l’invio e la ricezione dei dati da un broker MQTT.

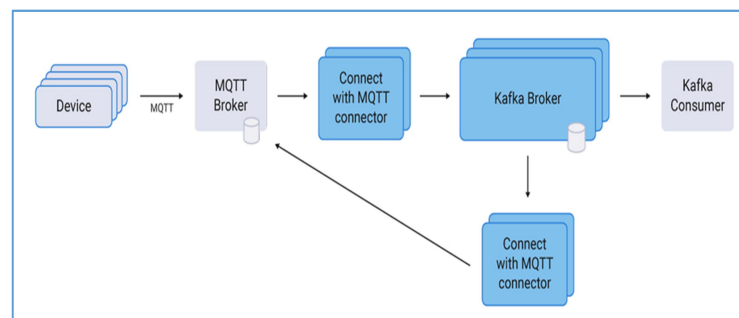


Figura 21 - Architettura Kafka-Connect

Perciò per acquisire, archiviare i messaggi in tempo reale e una successiva elaborazione in streaming dopo aver registrato il sensore ed eseguito la configurazione plug & play sarà necessario realizzare in un servizio “**Publisher**” preposto proprio alla pubblicazione su il topic del broker MQTT (EMQ) interno alla piattaforma Digital Future su connessione sicura TLS/SSL (883) come mostro in **Errore. L'origine riferimento non è stata trovata..**



#### 4.1.3. *Data Preparation*

In generale l’elaborazione dei flussi (stream processing) utilizza paradigmi seguenti: Batch, micro-batch, Elaborazione stream. Nel primo caso di elaborazione (in batch) lo stream di dati è collezionato in un storage e l’analisi del dato raccolto è effettuata mediante elaborazioni batch schedulate a tempo; L’approccio basato su micro-batch comporta l’utilizzo di tecnologie che si occupano di raggruppare lo stream di dati in blocchi, determinati dal raggiungimento di una soglia che può essere di varie tipologie: Intervalli di tempo schedulati (ogni x minuti) Al raggiungimento di X elementi Al raggiungimento di una soglia di X MB di dati Il blocco così determinato è poi elaborato in batch.

Ed il framework che utilizza questo approccio è Spark Streaming, framework reso disponibile dalla Piattaforma DF. Si precisa che nel caso di elaborazione a flusso il dato è analizzato nel momento stesso in cui viene raccolto; la conseguente mancanza di ritardo nell’elaborazione rende questa modalità più efficace nell’analisi real time.

Quindi in generale il Batch processing è appropriato quando non è cruciale avere il dato più aggiornato possibile e non è critico avere risposte in tempi più lunghi, mentre l’elaborazione a micro-batch e a flusso sono appropriate quando è necessaria la massima reattività tra l’arrivo del dato e la disponibilità dell’informazione.

In particolare la piattaforma DF per quanto concerne la modalità Batch la sorgente dati per le varie elaborazioni sarà il database Cassandra mentre per l’analisi in real-time dei dati provenienti dal simulatore, verrà utilizzata la modalità Streaming, per cui i dati per l’elaborazione saranno presi direttamente dallo specifico topic Kafka e con Spark Streaming si permetterà di analizzare flussi di dati in tempo reale, ad esempio log di errori come mostrato in Figura 22. **Errore. L'origine riferimento non è stata trovata..**

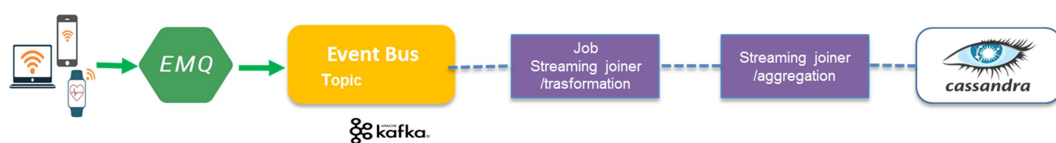


Figura 22: stream processing



In particolare, Spark streaming, [17] crea processi con esecuzione prolungata durante i quali è possibile applicare trasformazioni ai dati e quindi eseguire il push dei risultati in filesystem, database, dashboard e nella console.

Spark Streaming elabora micro-batch di dati, raccogliendo prima un batch di eventi per un intervallo di tempo definito. Tale batch viene poi fatto proseguire per l'elaborazione e l'output. Gli intervalli di tempo dei batch sono in genere definiti in frazioni di secondo.

I flussi di stream possono arrivare da fonti come Kafka, o HDFS, e vengono ingeriti in piccoli lotti per essere analizzati. Si possono svolgere tutte le operazioni fattibili sui dati statici, ma in più sono presenti funzionalità specifiche relative al tempo come le sliding windows. Il Job quindi verrà eseguito per una durata configurabile e con una finestra temporale di aggregazione dei dati anch'essa configurabile.

Pertanto per eseguire pre-analisi, preparare dati, progettare e definire una funzione al fine creare e addestrare modelli predittivi su dati real time e o su dati storici, sarà necessario definire servizi che recuperano i dati con lo schema concordato dal simulatore in un intervallo di tempo definito a priori, come si vedremo nel caso diversi contesti applicativi di seguito analizzati.

#### 4.2. **Servizi per Monitoraggio dello stato di salute di un macchinario o componente**

Come su menzionato nella soluzione proposta da Exprivia nell'ambito della Fabbrica Intelligente si prevede la realizzazione di una soluzioni prognostica e diagnostica che consentano di monitorare un impianto industriale e di prevederne l'andamento futuro al fine di prevenire possibili guasti nel funzionamento e nel processo produttivo sono stati analizzati due problematiche rivolata in particolare a:

- **Caso 1** : predire la vita utile rimanente (Remaining Useful Life: RUL) di una batteria.
- **Caso 2**: stimare la probabilità che una turbina idroelettrica possa guastarsi entro X unità di tempo.
- **Caso 3** : monitorare lo stato di salute dell'asset effettuando anomaly detection.





Nel caso 3 per allenare il modello, che viene poi utilizzato per fornire previsioni dei prossimi punti. Si confrontano poi i valori effettivamente forniti dal sensore con quelli stimati, e se la differenza supera una soglia prefissata si segnala una anomalia.

#### 4.2.1. **Servizi per stimare la vita utile rimanente (RUL)**

Al fine di definire e analizzare quali sarebbero stati i potenziali servizi che la Piattaforma potrebbe mettere a disposizione per modellare questo problema di prognostica in questa attività è stato simulato un caso pilota. In particolare attraverso l'uso Anki-overdrive all'interno del laboratorio (ILAB) sono stati analizzati i parametri rilevati da sensori di position, speed, battery charge level di micro-automobili al fine di monitorare lo stato di carica della batteria e predire la stima del tempo utile residuo; In particolare il kit Anki-overdrive è costituito da macchinine controllate da remoto e fatte girare su diversi tipi di piste dalla forma e lunghezza variabile e da una pista. La pista è costituita da pezzi componibili dalla diversa forma (curve, rettilinei) il cui assemblaggio permette una completa personalizzazione della pista realizzando diversi tipi di tracciato e permettendone quindi una buona diversificazione dello situazione. Le macchinine che gireranno in pista sono state tre diversi modelli, le cui caratteristiche e performance sono simili ma non identiche. Inoltre velocità, verso di percorrenza e accensione o spegnimento potranno cambiare a run-time aggiungendo maggiore complessità allo simulazione. Su ciascuna di queste è installato un sensore che permetterà di conoscere una serie di informazioni legate alla corsa delle macchinine come velocità, posizionamento sulla pista e livello di carica batteria. Il protocollo di comunicazione tra i sensori delle macchine e il Raspberry PI è stato Bluetooth Low Energy . Per quanto riguarda la scarica della batteria da una pre-analisi si verificata che essa dipenderà dalla velocità, dal tempo in cui i led sono stati attivati e dal modello della macchinina, in quanto ogni modello avrà una batteria con caratteristiche diverse e quindi anche capacità e livello di usura diverso. I sensori installati sulle macchinine permetteranno di conoscere il livello di carica in qualsiasi momento. Questo tipo di richiesta verrà implementata con l'ausilio di specifiche interfacce lato cruscotto che interrogheranno le macchinine per conoscere il livello di Amperaggio.



Ipotizzando però in cui tale funzionalità non fosse sempre disponibile potrebbe nascere la necessità di prevedere il livello di carica in un determinato istante non potendo richiedere tale livello direttamente al sensore si analizzano e preparati i dati applicandoli in tutti i casi possibili derivanti da questa casistica:

- avere a disposizione un'unica lettura del livello di batteria registrata ad inizio corsa.
- avere a disposizione diverse letture del livello di batteria a intervalli prefissati.
- avere a disposizione diverse letture del livello di batteria a intervalli irregolari.
- nessun valore a disposizione.

Ed a prescindere delle situazione sono stati analizzati le modalità attuazione dei servizi di:

- *Data Acquisition* acquisizione dati da sensori
- *Stream processing*
- Batch processing/ model building
- Real time Processing
- *Alerting*
- *Start processing*

#### *Data Acquisition da device*

L'acquisizione dei dati da ciascun sensore installato sulle macchinine sono mostrati nella tabella 3.

device_id	Identificativa univoca macchinina
timestamp_	Timestamp della rilevazione
car_name	Nome macchinina
message_type	Tipo di messaggio restituito dalla macchina. I possibili valori possono essere: <ul style="list-style-type: none"><li>• ANKI_VEHICLE_MSG_V2C_LOCALIZATION_TRANSITION_UPDATE</li><li>• ANKI_VEHICLE_MSG_V2C_BATTERY_LEVEL_RESPONSE</li></ul>



	<ul style="list-style-type: none"> <li>• ANKI_VEHICLE_MSG_V2C_LOCALIZATION_POSITION_UPDATE</li> </ul>
track_location	Identificativo del pezzo di circuito che la macchinina sta percorrendo in quel determinato momento
road_piece	Identificativo di una sottoparte del pezzo di circuito su cui la macchinina si trova
offset	Indica la distanza tra la posizione corrente della macchinina e l’inizio del pezzo del circuito
speed	Velocità registrata dalla macchinina al momento della rilevazione
clockwise	Indicatore del verso di percorrenza della macchinina sulla pista
intersection_code	Indica se il pezzo di circuito che la macchinina sta percorrendo è un pezzo di tipo incrocio
is_exiting	Indicatore utilizzato per segnalare l’ingresso della macchinina in una zona riservata della pista(zona box)
road_piece_prev	Identificativo di una sottoparte del pezzo di circuito su cui la macchinina si trovava alla rilevazione precedente
trackstyle	Indica se la macchinina sta girando a destra, a sinistra o sta procedendo dritta al momento della rilevazione
battey_level	Livello di batteria espresso in percentuale
delocalized	Indica se la macchinina è andata fuori pista, non riuscendo più a identificare un pezzo di circuito da percorrere
username	Alias utilizzato per identificare la macchinina



Prevedere l’attuazione del servizio “Publisher” preposto alla pubblicazione sul topic del broker MQTT (EMQ) per recuperare i dati Raspberry PI con il supporto del servizio infrastrutturale Event layer della piattaforma Digital Future.

Dato che la rilevazione sullo stato di carica della batteria avviene per via indiretta tramite l’amperaggio di corrente rilevato dal sensore di batteria, le rilevazioni non sono precise. Ai fini della preparazione del dataset si prevede stabilire di interrogare la macchinina a intervalli regolari di 30 o 60 secondi, non mediante una singola chiamata ma effettuando una serie di 15 chiamate consecutive. I dati vengono rilevati in momenti differenti in base al tipo di messaggio che le macchinine mandano. In particolare:

- La macchinina manda un messaggio di tipo  
NKI\_VEHICLE\_MSG\_V2C\_LOCALIZATION\_TRANSITION\_UPDATE  
ogni volta che il sensore posto sulla macchinina rileva il superamento di un pezzo di circuito. I dati significativi per questo tipo di messaggio sono:
  - *track\_location*;
  - *trackstyle*.
- La macchinina manda un messaggio di tipo  
NKI\_VEHICLE\_MSG\_V2C\_LOCALIZATION\_POSITION\_UPDATE  
ogni volta che il sensore posto sulla macchinina rileva il superamento di un sotto-componente del pezzo di circuito. I dati significativi per questo tipo di messaggio sono:
  - *road\_piece*;
  - *road\_piece\_prev*;
  - *speed*.
- La macchinina manda un messaggio di tipo  
ANKI\_VEHICLE\_MSG\_V2C\_BATTERY\_LEVEL\_RESPONSE  
ogni volta che alla macchinina viene richiesto il livello di carica della batteria. Il dato significativo per questo tipo di messaggio è *battey\_level*.

Riassumendo, i dati arrivano dalle macchinine ogni volta che succede un evento nei casi di messaggi di tipo ANKI\_VEHICLE\_MSG\_V2C\_LOCALIZATION\_TRANSITION\_UPDATE e



ANKI\_VEHICLE\_MSG\_V2C\_LOCALIZATION\_POSITION\_UPDATE, mentre il livello di carica della batteria con i messaggi di tipo

ANKI\_VEHICLE\_MSG\_V2C\_BATTERY\_LEVEL\_RESPONSE avviene su richiesta tramite interrogazione della macchinina. In quest’ultimo caso, per la fase di addestramento del modello, le interrogazioni avverranno ogni 30 o 60 secondi.

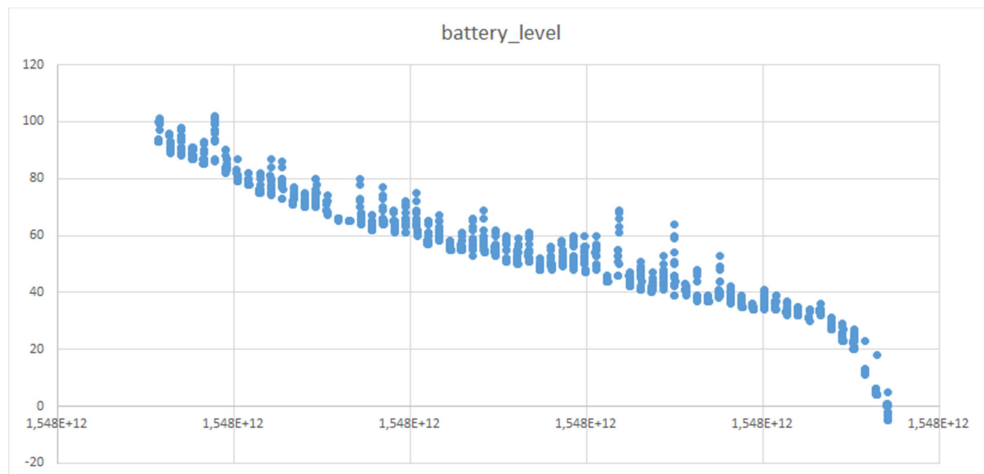


Figura 23 Dati acquisiti del valore di batteria una macchinina durante una corsa

#### Data Preparation e feature engineering

Al fine di predisporre i dati per l’addestramento del modello, i dati sono stati sottoposti a variazioni di filtraggio, elaborazione e raggruppamento pertanto sono stati realizzati due job come si evince in Figura 24.

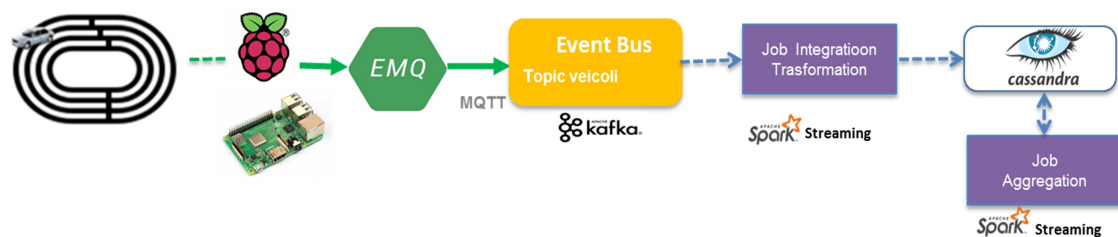


Figura 24 Pipeline di processing

Il primo carica i dati essenziali sul datastore della Piattaforma DF Cassandra. Il dataset iniziale è composto da una moltitudine di record aventi come struttura quella esposta nella tabella precedentemente descritta. I passi di trasformazione che vengono eseguiti sono i seguenti:



1. Divisione del dataset per tipo di messaggio. Si avranno quindi i seguenti sotto-dataset:
  - a. Dataset con *message\_type* riguardante lo stato di carica della batteria;
  - b. Dataset con *message\_type* riguardante l’aggiornamento di posizione o di transizione;
  - c. Dataset con *message\_type* sconosciuto, che viene scartato in quanto considerato outlier.
2. Nel dataset riguardante lo stato di carica della batteria:
  - a. Recupero del timestamp precedente per ogni record;
  - b. Identificazione del numero della richiesta del livello di batteria (differenza di timestamp > 15 sec) e identificazione del numero di corsa (*n\_race*) (differenza di timestamp > 1000 sec, nome campo: *n\_race*);
  - c. Raggruppamento dei record per numero della richiesta del livello di batteria, prendendo come valore di batteria il valore medio per ogni serie;
  - d. Unione del dataset riguardante la batteria con il dataset riguardante la posizione e transizione.

A questo punto il dataset viene privato delle colonne ritenute inutili e si riportata nel datastore le seguenti feature:

- *car\_name*;
- *timestamp\_*;
- *battery\_level*;
- *speed*;
- *message\_type*.

Successivamente le altre fasi di elaborazione hanno riguardano un altro job di sezione:

1. Ordinamento per *car\_name* e *timestamp\_*;
2. Recupero del livello di batteria precedente per ogni record;
3. Calcolo del tempo trascorso tra ogni dato inviato;
4. Assegnazione di un numero a ogni record, che parte da 0 e aumenta di 1 ogni volta che si incontra un messaggio riguardante la batteria (nome campo *n\_blr*);



5. Raggruppamento per car\_name, n\_race, n\_blrt, message\_type, calcolando la media della velocità (nome campo: avg\_speed).

La parte finale di preparazione dei dati per l’addestramento del modello riguarda il calcolo del tempo residuo di carica per ogni macchinina, per ogni corsa e ad ogni livello di batteria. Per fare questo i passi eseguiti sono i seguenti:

1. Calcolo del minimo timestamp per ogni macchina e per ogni corsa;
2. Differenza per ogni riga tra il timestamp reale e il minimo timestamp calcolato precedentemente.

Un estratto del dataset così ottenuto, filtrato per ogni campo strettamente necessario e pronto per essere dato in input al modello di machine learning, è riportato nella tabella

car_name	battery_level	avg_speed	sec_to_end
NUKE	85	301,7143	2034
NUKE	79	302,2857	1970
NUKE	77	298,7857	1907
NUKE	75	303,9286	1843

Tabella 3 dati rielaborati nuke

L’estratto riportato è riferito ad una specifica corsa della macchinina NUKE, e riporta a diversi intervalli di tempo il proprio livello di batteria, la velocità media a cui la macchinina stava percorrendo il tracciato e il tempo rimanente alla fine della batteria espresso in secondi.

Di seguito viene riportato il grafico del dataset ottenuto secondo l’elaborazione espressa precedentemente, sovrapponendo i risultati ottenuti per quattro corse complete della macchinina NUKE.

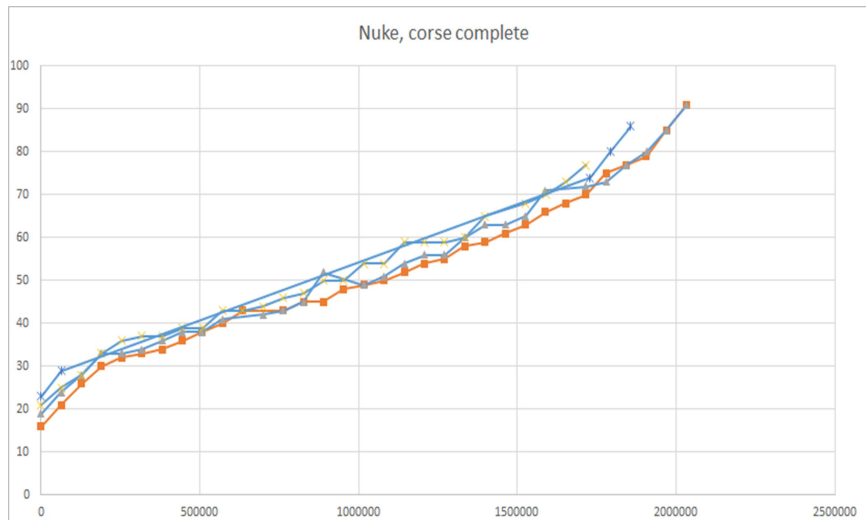


Figura 25: Andamento corse per macchina Nuke.

Sull'asse delle ordinate viene espresso il valore percentuale di batteria residua in un determinato momento, mentre sull'asse delle ascisse viene riportato il tempo rimanente alla fine della carica di batteria per una determinata corsa effettuata espresso in millisecondi. Il grafico è quindi da intendersi a specchio orizzontalmente in quanto il tempo residuo naturalmente va a decrescere.

#### *Batch processing /model Regression*

Per la costruzione del modello si provvederà di utilizzare i servizi infrastrutturale messa dalla diposizione Piattaforma IOT DF. L'algoritmo di Machine Learning che verrà adottato per questo caso specifico è la Regressione Lineare. La regressione può essere utilizzata per modellare problemi di prognostica. Il che significa permettere di stimare la vita utile rimanente di un componente in termini di un numero continuo (fornito appunto dal modello di regressione) di unità di tempo prefissata. In questo specifico caso il dataset di training deve contenere solamente dati relativi a componenti che sono stati soggetti a guasti, per poter permettere l'etichettatura degli input a ritroso a partire dall'istante di guasto .





Pertanto partendo dal dataset che si è ottenuto dalle operazioni citate nel paragrafo precedente, si andranno a individuare le feature e il label del modello. Le feature del modello sono le seguenti:

- *BatteryLevel*: livello di batteria, il cui valore verrà associato al valore iniziale comunicato dal sensore installato sulla macchinina (questo valore verrà ricevuto o all’inizio di ogni corsa o ad intervalli regolari) per poi essere scalato a seguito appunto della previsione fatta all’istante precedente
- *Speed*: velocità media della macchinina nella finestra temporale presa in considerazione dal modello.
- *Sigmoide*: fattore ottenuto a partire dal valore di batteria e applicandolo alla formula

$$\text{Sigmoide} = 1/(1 + 2.71^{(-1/10)*(x-50)})$$

Questo fattore sarà utile per smorzare la retta ottenuta dalle prime due feature e adattarla all’andamento dei dati reali e ottenere un drastica riduzione dell’errore di cui sarà affetto il modello. Per ciascuna macchinina si costruirà uno specifico modello che in fase di streaming verrà utilizzato permettendo di ottenere le predizioni sul tempo di vita della stessa.

Per la fase di addestramento e testing si utilizzeranno gli strumenti messi a disposizione della Piattaforma DF. I modelli creati per saranno salvati su datastore HDFS presente nella Piattaforma DF come si evince in Figura 26.

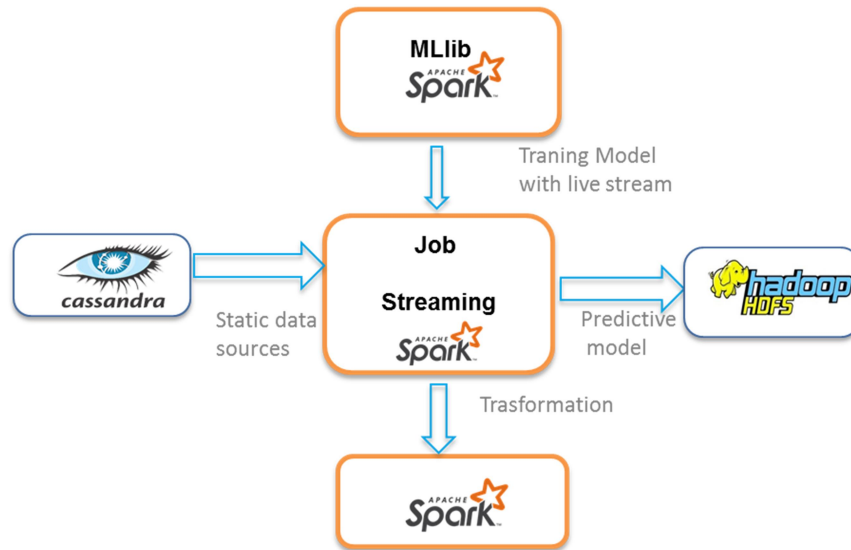


Figura 26: Traning Model

L’idea alla base del test del modello è quella di selezionare in maniera casuale una parte del dataset da usare in input al modello predittivo, e la restante parte da usare come test, eventualmente calcolando degli indici di bontà della predizione come può essere la distanza media tra il valore vero e il valore predetto.

Nel grafico in Figura 27 si veda sull’asse delle ordinate troviamo i minuti residui alla scarica completa della macchinina, e in cui sull’asse delle ascisse troviamo la percentuale di batteria, colorando in arancione i punti estrapolati dalle corse reali e in blu i punti predetti.

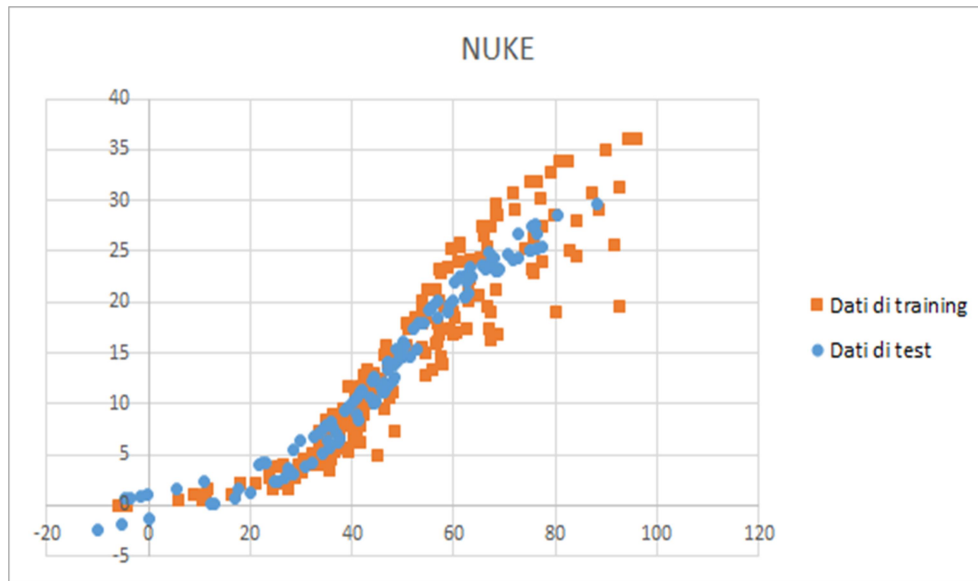


Figura 27 :Training vs test corse Nuke

Dal grafico Figura 27 è possibile notare come le previsioni tendano a ricalcare abbastanza fedelmente i comportamenti reali delle macchine, soprattutto quando il livello di batteria inizia a decrescere sotto una certa soglia.

È importante notare che la differenza dei valori di tempo residuo di carica quando la batteria è pressoché carica dipende dalla differenza di velocità alla quale la macchinina ha percorso il tracciato.

Nel grafico Figura 28 è riportata anche la velocità, espressa in cm/s, il quale asse unito agli altri due rende il grafico tridimensionale. Esso è di seguito riportato.

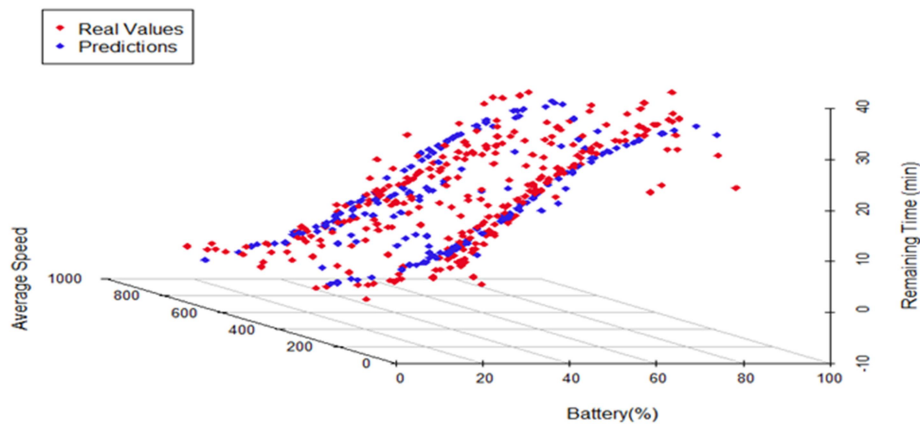


Figura 28: Training vs test tridimensionale

### *Real time Processing*

I dati che arrivano in streaming dalle macchinine durante le loro corse richiedono una elaborazione in tempo reale per monitorare il livello di batteria residuo e la velocità alla quale ogni macchinina sta viaggiando, al fine di utilizzare il modello predittivo e stimare così il tempo residuo di carica della batteria.

Dato che l'elaborazione in streaming avviene a finestre temporali di grandezza configurabile e definita inizialmente di 5 secondi, sarà necessario effettuare e implementare un servizio per trasformare i dati grezzi che arrivano di volta in volta in dati aggregati e semilavorati. I passi di elaborazione da eseguire sono aggregare i record per tipo di macchina e tipo di messaggio inviato, filtrando i messaggi di tipo sconosciuto, calcolando la media della velocità e il valore medio di batteria registrato;

Successivamente a queste operazioni preliminari di elaborazione del dataset preso in ingresso per ogni finestra temporale considerata, il job recupererà il modello predittivo salvato nel file system distribuito Hadoop Distributed File System (HDFS), in modo da utilizzarlo per trasformare i nuovi dati in ingresso modello di macchina, la velocità attuale e l'attuale livello di carica della batteria, ed effettuare la nuova stima relativa al tempo residuo di carica della batteria veda Figura 29 .



### Nuke - Remaining Time - Predicted vs Actual

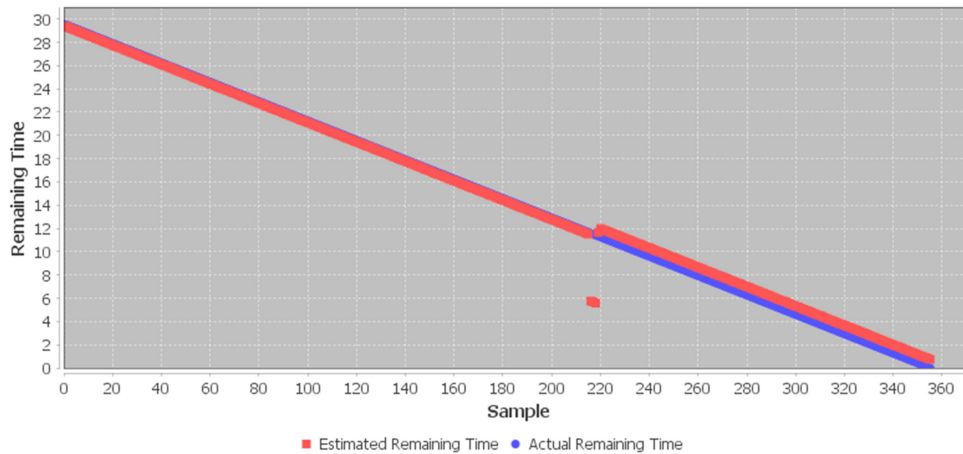


Figura 29: Tempo residuo di carica misurato vs stimato

#### Alerting

Una volta creato il modello addestrato, testato, salvato sulla Piattaforma, in fase di erogazione arrivano i dati in Piattaforma un job esegue l'elaborazione su menzionata e successivamente si avvia un servizio composto che avrà in input Identificativo\_asset, BatteryLevel\_Actual, Speed attuale ed eseguirà i seguenti passi:

- Invocazione servizio che recupera il valore predittivo “*RUL\_BatteryLevelPredict*” ossia il potenziale livello di batteria in funzione della velocità attuale dalla funzione modello creato e addestrato in precedenza.
- Invocazione del servizio con il quale recupera il kpi “Estimated Time” dal sistema Asset Management che rappresenta il valore Target di tempo massimo al di sotto del quale è necessario avviare un processo di interventi. Pertanto si confronta il valore “*RUL\_BatteryLevelPredict*” con Estimated Time al superamento si fornisce **in output**: *Identificativo\_asset, message : start intervento predittivo*

Una volta creato il modello addestrato e testato e salvato sulla Piattaforma in fase di erogazione arrivano i dati Piattaforma e come mostrato in Figura 30 un job esegue elaborazione su menzionata e successivamente sia un avvia un job il quale recupera in real-time tutte le metriche relative lo stato degli asset in un determinato, e li confronto



con i KPI prefissati in fase iniziale e recuperati tramite api dall’ asset Management in caso del superamento di almeno una soglia del KPI invia un messaggio di alerting indirizzato all’ applicazione Fabbrica Intelligente in particolare il modulo di Field Service Management il quale provvederà ad avviare il processo di manutenzione.

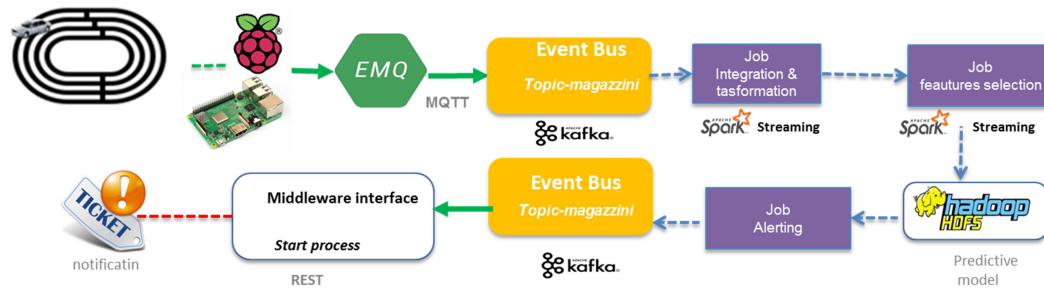


Figura 30: pipeline real time Alerting

Il confronto tra il valore di soglia di ciascun KPI attivo e i valori registrati o predetti di ogni macchinina, avverrà periodicamente in una finestra temporale di 5 secondi. Le tipologie di KPI sono le seguenti:

- *Estimated Time* : rappresenta il tempo di vita rimanente della batteria di una specifica macchinina. Fissando questo valore e comunicandolo al Job in esecuzione, esso controllerà, dopo ogni predizione effettuata su tutte le macchinine in corsa, quali tra queste è associata ad un KPI di questo tipo e il valore del tempo di vita predetto della macchinina. Al di sotto del valore di KPI la notifica verrà inviata avviando così un processo di manutenzione di tipo predittivo comunicando tutte le informazioni legate all’ Asset coinvolto.
- *Elapsed Time*: rappresenta il tempo trascorso dall’inizio della corsa. La manutenzione di questo tipo non può essere definita predittiva ma ordinaria, in quanto fissando questo valore, senza l’ausilio del modello predittivo, la manutenzione potrà iniziare tenendo traccia solo del tempo da inizio corsa e non a valori ottenuti come output dal modello predittivo.

Al superamento del valore associato a questo KPI del tempo trascorso della macchina, una notifica verrà inviata permettendo così l’avvio di un processo di manutenzione ordinaria.



- *Laps Number*: rappresenta il numero di giri che la macchinina dovrà compiere prima del lancio del processo di manutenzione ordinaria. Il controllo verrà effettuato in ogni finestra temporale.
- *Battery Level*: questo tipo di KPI rappresenta il valore di batteria della macchinina al di sotto del quale il processo di manutenzione predittiva dovrà essere eseguito. Come il KPI “Estimated Time” il controllo dovrà essere eseguito a seguito dell’utilizzo del modello.

I valori di KPI verranno richiesti dalla piattaforma invocando un endpoint specifico e memorizzati periodicamente all’interno del Job eseguito in modalità in streaming.

L’aggiornamento dei valori di KPI dovrà essere eseguito ad ogni nuova corsa riconosciuta dal Job in modo da poter utilizzare KPI diverse per ogni nuova corsa.

Il singolo KPI sarà caratterizzato da una tipologia, un valore di soglia fissato, la macchinina a cui è associato e un flag che ci informerà se esso è attivo o meno.

Al superamento della soglia fissata del valore specificato dal KPI, da parte della macchinina associata al KPI, il sistema notificherà tale evento, permettendo così l’avvio di un processo di manutenzione nel quale si potrà procedere con azioni correttive.

#### *Start process mantanance*

L’avvio del processo, è demandato al Middleware Interface componente della soluzione Smart Factory il cui compito, in questa fase, sarà di avviare il processo con le informazioni legate all’evento che ha generato l’Alert pertanto attraverso il subscribe recupera il messaggio dal connettore kafka. Con l’utilizzo del modulo di Workflow Management sarà infine possibile far avanzare il processo e completarlo. Un esempio processo di manutenzione utilizzabile nel modulo di Workflow Management è riportato nella Figura 31.

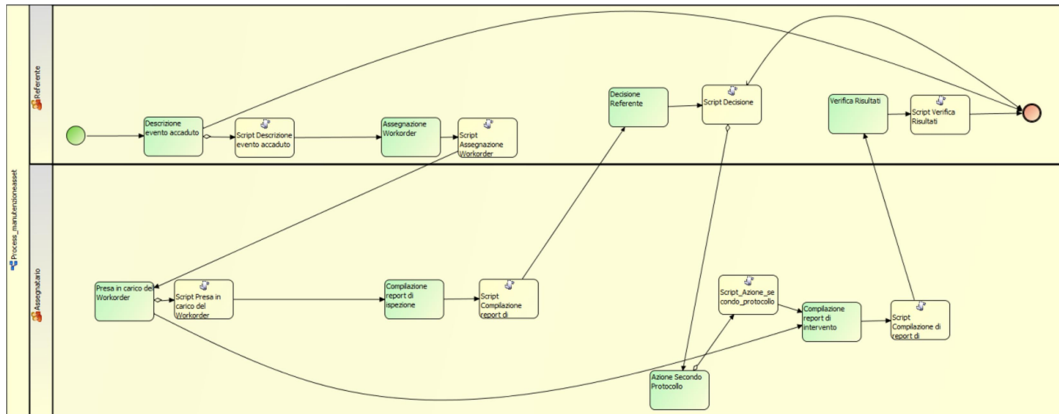


Figura 31 Workflow del processo di manutenzione generico

Il processo di manutenzione avrà due unici attori, il Referente e l’Assegnatario.

Queste due tipologie di utenti permetteranno l’avanzamento del processo eseguendo determinate azioni che di seguito verranno descritte. Le attività coinvolte nel processo saranno quelle mostrate in Figura 32.

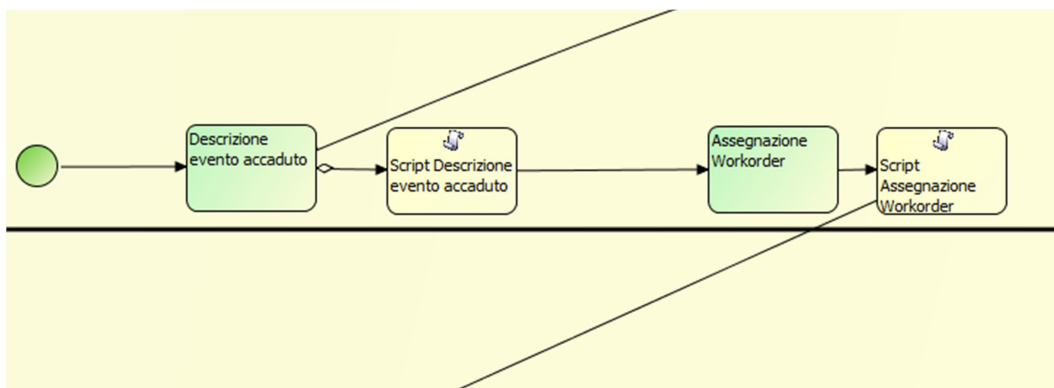


Figura 32 Blocco di attività della fase iniziale

Descrizione evento accaduto: Attività iniziale che prevede la visualizzazione da parte del Referente di tutte le informazioni che hanno generato il processo di manutenzione.

In questa attività sarà presente, una descrizione dell’evento accaduto al cui interno saranno anche presenti i valori al di sopra del KPI configurato, l’impianto dove si è presentato l’evento che ha generato il Workflow, il macchinario e il componente coinvolti, data dell’evento e infine se l’evento dovrà essere assegnato o meno. In questa





fase il Referente potrà modificare le informazioni o passare direttamente all’attività successiva per gestire l’evento.

Assegnazione del Workorder: In questa fase il Referente, dopo aver analizzato tutte le informazioni provenienti dalla piattaforma, dovrà scegliere se effettuare un Workorder di tipo ispezione o di tipo intervento. In entrambe le scelte, il Referente dovrà specificare in quale ottica l’azione dovrà essere svolta tra Riparazione, Sostituzione e Rottamazione). Infine, potrà inserire una nota relativa alle azioni da eseguire o le considerazioni sul tipo di azione da eseguire che l’Assegnatario potrà visualizzare nelle fasi successive.

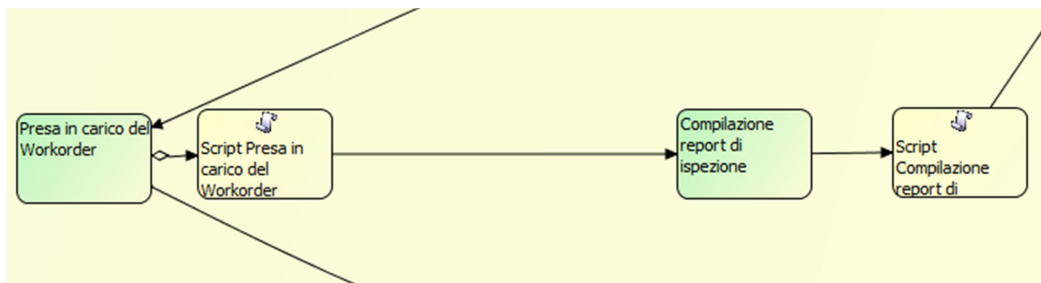


Figura 33 Attività della fase di ispezione e della compilazione del relativo report

Presenza in carico del Workorder: Con la scelta del referente di attivare un processo di ispezione, l’avanzamento del processo potrà avvenire solo da parte dell’Assegnatario. In questa fase, si potranno visualizzare tutte le informazioni legate all’evento che ha avviato il processo di manutenzione e inoltre verranno visualizzate le informazioni inserite dal Referente nella fase precedente. Infine, l’Assegnatario potrà accettare o non accettare il compito richiesto, solo l’accettazione della richiesta comporterà l’avanzamento dello stato del processo.

Compilazione report di ispezione: Nel caso si scegliesse un’azione di tipo ispezione, e successivamente tale incarico venisse accettato dall’Assegnatario, l’attività coinvolta nel processo sarà “Compilazione report di ispezione”. In questa fase l’Assegnatario compilerà un breve rapporto sulle operazioni svolte. I campi che potranno essere valorizzati saranno i seguenti:

- Data Inizio Ispezione
- Data Fine Ispezione



- Attività eseguite in fase di ispezione
- Attività non eseguite in fase di ispezione
- Condizione asset: campo obbligatorio che permetterà di definire lo stato dell’asset riscontrato in fase di ispezione in un range di 5 valori (1 - Scarsa, 2 - Discreta, 3 - Sufficiente, 4 - Buona, 5 - Eccellente)
- Criticità riscontrate in fase di ispezione
- Altre note dopo ispezione.

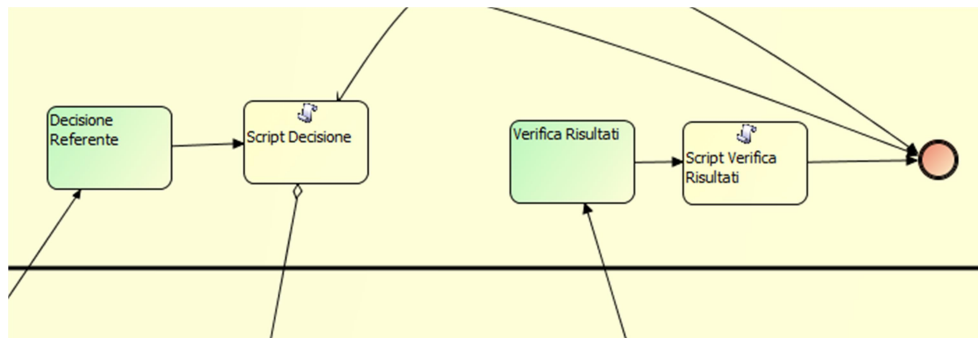


Figura 34 Attività legate alla decisione del referente e alle sue valutazioni finali

Decisione Referente: In questa fase, il Referente valuterà il report stilato dall’Assegnatario nella fase di ispezione e valuterà l’attuazione di un intervento. Si potrà scegliere se eseguire un intervento di riparazione, sostituzione, rottamazione o nessuna delle precedenti lasciando lo stato del componente sotto esame inalterato. A questa scelta potrà anche essere associata una priorità dal livello 1 (priorità molto bassa) fino a 5 (priorità molto alta) nel caso si scegliesse di eseguire un intervento o “Nessuna Priorità” nel caso si decidesse di non eseguire nessuna azione. Con quest’ultima scelta il processo terminerebbe, venendo etichettato come completato.

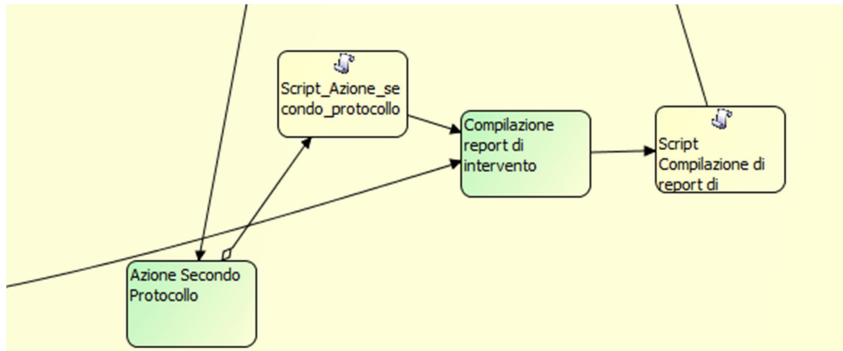


Figura 35 Attività della fase di attuazione dell'intervento e della compilazione del relativo report

Azione Secondo Protocollo: Nel caso il Referente scegliesse di eseguire un qualsiasi intervento tra Riparazione, Sostituzione o Rottamazione, il controllo passerà nuovamente all' Assegnatario, che, come nella fase di accettazione del Workorder, potrà accettare l'incarico o rifiutarlo. La sola accettazione potrà permettere l'avanzamento del processo e il passaggio all'attività di "Compilazione del report di intervento".

Compilazione report di intervento: Dopo l'attuazione dell'intervento deciso dal Referente, l'Assegnatario dovrà compilare un report relativo all'intervento appena svolto. I campi da dover valorizzare saranno i seguenti:

- Data inizio intervento
- Data fine intervento
- Attività eseguite durante l'intervento
- Attività non eseguite durante l'intervento
- Criticità riscontrate dopo l'intervento
- Altre note dopo l'intervento

Verifica Risultati: Al termine delle attività eseguite durante tutto il processo, il Referente avrà a disposizione tutte le informazioni delle attività precedenti e potrà emettere un giudizio sulle azioni svolte. Questo giudizio verrà espletato con un voto da 1(Scarso) a 5(Eccellente) e una nota in cui poter inserire valutazioni più specifiche e dettagliate.



#### 4.3. Servizi per classificare la presenza di un guasto entro una finestra temporale

Obiettivo nell’analizzare questa problematica è stato quello di individuare un servizio in grado di fornire diagnostica e prognostica. In caso di problema di diagnostica, questo significa decidere se la macchina sta funzionando correttamente o non correttamente, facendo rientrare tutti i possibili stati in due classi. Ma l’applicazione della la classificazione binaria può rientrare anche per decidere se la macchina può guastarsi entro un intervallo di tempo fissato. La differenza fra le due accezioni è data semplicemente dalla diversa interpretazione delle etichette. Ciò significa che uno stesso modello può risolvere entrambi i problemi. Quello che sarà differenziato è il labeling del dataset utilizzato per svolgere la fase di training del modello. In particolare si voleva riuscire a distinguere fra uno stato di normale funzionamento della macchina e stati con presenza di anomalia /guasto entro un determinato tempo. Grazie ai test descritti in questa sezione è dimostrare che modelli di machine learning in particolare modelli di classificazione sono in grado di distinguere differenti stati di funzionamento della macchina tramite i valori raccolti dai sensori.

Il dataset utilizzato per alimentare il trattamento questo problema e analizzare quello sarà il migliore approccio di integrazione tra la Piattaforma e la soluzione Fabbrica intelligente è stato reperito su I dati originali, contenuti in un file CSV, contenevano inizialmente i dati in tempo reale di telemetria simulata (e semplificata) e il registro macchina di una turbina idroelettrica. In particolare i dati rappresentavano misure (features) quali tensione, rotazione, pressione e vibrazione raccolte in tempo reale da 100 differenti macchine mediamente su base oraria (ogni ora) durante l’anno 2015. Di seguito forniamo la rappresentazione originale per i primi 10 record dei dati di telemetria della macchina 1.



datetime	machineID	volt	rotate	pressure	vibration
2015-01-01 06:00:00	1	176.2179	418.5041	113.07794	45.08769
2015-01-01 07:00:00	1	162.8792	402.7475	95.46053	43.41397
2015-01-01 08:00:00	1	170.9899	527.3498	75.23790	34.17885
2015-01-01 09:00:00	1	162.4628	346.1493	109.24856	41.12214
2015-01-01 10:00:00	1	157.6100	435.3769	111.88665	25.99051
2015-01-01 11:00:00	1	172.5048	430.3234	95.92704	35.65502
2015-01-01 12:00:00	1	156.5560	499.0716	111.75568	42.75392
2015-01-01 13:00:00	1	172.5228	409.6247	101.00108	35.48201
2015-01-01 14:00:00	1	175.3245	398.6488	110.62436	45.48229
2015-01-01 15:00:00	1	169.2184	460.8507	104.84823	39.90174

Tabella 4 dataset origine

Oltre ai dati di telemetria, per addestrare il modello sono stati utilizzati:

- ✓ Un dataset in CSV contenente i dati relativi ai log degli **errori irreversibili** generati mentre il macchinario è ancora operativo e non rappresentano guasti;
- ✓ Un dataset CSV contenente le **informazioni sul macchinario** ossia il tipo di modello e l'età indicante da quanti anni è operativo;
- ✓ Un dataset CSV contenente i **fallimenti**, ossia i dati associati ai guasti del macchinario. contenenti data in cui si è verificato il guasto e id della macchina;

#### *Data Preparation e feature engineering da storici*

Per questo specifico caso d'uso, il dataset originale sarà stato modificato in modo da descrivere concettualmente e astrarre le condizioni di salute del macchinario utilizzando i dati storici raccolti fino ad un dato momento. Il risultato di questa fase è la suddivisione del dataset in training/validation ottenuta mediante una combinazione dei dati storici associati al macchinario e nello specifico oltre ai dati di telemetria anche altre tipologie di dati quali errors, failures e machine properties già citate nel precedente paragrafo.

E' stata pertanto definita una finestra di ritardo, da utilizzare per generare alcune misure di aggregazione quali la rotazione media, la deviazione standard, etc.. al fine di rappresentare la history dei dati di telemetria in un determinato lasso temporale. L'ampiezza della finestra, come mostra la figura che segue, è stato inizialmente selezionata pari a 3 ore come mostra Figura 36 che segue.

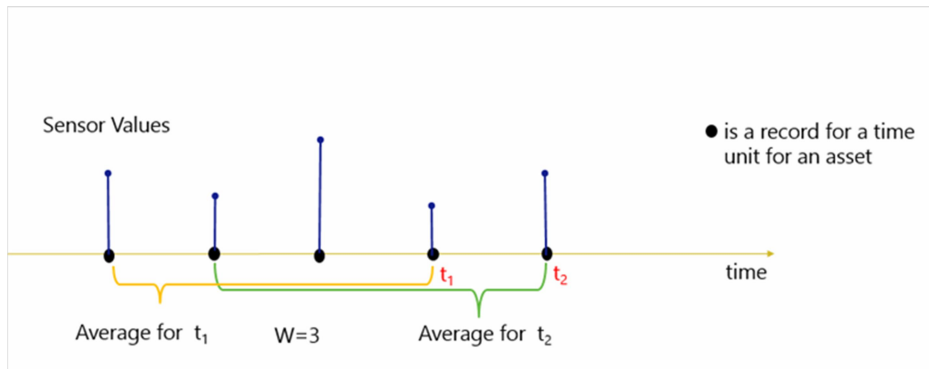


Figura 36 time unit for an asset

Il risultato di questa prima fase di preparazione dei dati è rappresentato nella figura seguente:

	datetime	machinelD	voltmean	rotatemean	pressuremean	vibrationmean	voltsd	rotatesd	pressured	vibrationsd	error1count	error2count	err
0	2015-01-02T05:00:00Z	1	169.733809	445.179865	96.797113	40.385160	11.233120	48.717395	10.079880	5.853209	0.0	0.0	
1	2015-01-02T08:00:00Z	1	170.525721	443.906847	97.667249	39.786670	12.591948	46.930282	9.406795	6.098173	0.0	0.0	
2	2015-01-02T11:00:00Z	1	170.049722	446.461279	96.906162	40.016513	13.277336	42.836780	9.071472	5.481724	0.0	0.0	
3	2015-01-02T14:00:00Z	1	170.341974	447.355315	96.229522	39.921963	13.817158	42.808633	8.256794	5.862312	0.0	0.0	
4	2015-01-02T17:00:00Z	1	170.060643	452.163407	96.357441	39.990470	14.792869	42.525293	8.669605	5.907157	0.0	0.0	
5	2015-01-02T20:00:00Z	1	169.369283	453.336163	98.042007	39.531667	15.674787	41.689624	10.607947	6.205887	0.0	0.0	
6	2015-01-02T23:00:00Z	1	169.795758	446.832666	98.454608	39.271645	15.742155	38.800266	11.679314	5.579524	0.0	0.0	

Tabella 5 : pre-processing dei dati con impostazione di finestra di ritardo

Al fine di adattare il dataset così ottenuto al caso d’uso in oggetto, i dati inizialmente aggregati a 3 ore hanno subito una ulteriore fase di pre-processing modificando gli intervalli temporali tra i dati delle serie temporali a 10s. Lo scenario qui presentato prevede infatti un invio dati continuo ed in tempo reale, con un livello di granularità, quindi, molto superiore rispetto a quello originale.

*Training and creation of the neural network model of classification*

A partire dal dataset ottenuto durante l’intera fase di pre-processing, per l’addestramento del modello è stata effettuata una suddivisione in **training**, **test** e **validation** set:

1. Evitando di sovrastimare le performance del modello;



2. Basandosi sul tempo e sull’osservazione che in scenari di manutenzione predittiva come quello qui descritto, esempi consecutivi che rientrano nella stessa finestra temporale hanno comportamenti simili;
3. Tenendo presente che per la natura previsionale del task, la fase di addestramento deve essere effettuata sui dati storici mentre la fase di test sui dati futuri;
4. Evitando, quindi, di utilizzare suddivisioni casuali o basate su tecniche classiche adottate nel machine learning quali k-fold cross validation, che non si adattano a questo tipo di scenario. La tecnica del k-fold cross validation si basa infatti sulla suddivisione del dataset totale in k parti di uguale numerosità e, ad ogni passo, la k-esima parte del dataset viene ad essere il validation dataset, mentre la restante parte costituisce il training dataset; un approccio del genere non permetterebbe di utilizzare l’assunto definito al punto 2;

In base a quanto espresso è stato selezionato un istante di tempo  $t$  presente nel dataset: il modello viene addestrato utilizzando i dati associati ad istanti di tempo precedenti all’istante  $t$  e validato sui dati associati ad istanti di tempo successivi a  $t$  assumendo che i dati futuri successivi al punto di suddivisione utilizzati per il test non siano noti.

Nello specifico i passi eseguiti per l’addestramento del modello sono stati i seguenti:

- partendo dal dataset dei dati di telemetria ottenuto durante la fase di pre-processing, per l’addestramento del modello sono state selezionate circa **15 features** ritenute significative e già rappresentate nella Tabella 5.
- per la fase di **addestramento** del modello è stato utilizzato come dataset di **training**, lo storico precedente ad un certo timestamp;
- per la fase di **testing**, sono stati selezionati tutti i dati successivi al timestamp precedentemente definito: i dati di test corrispondono a circa 1/12 dell’intero dataset;
- per la fase di **validazione** è stata utilizzata una percentuale del dataset di training pari al 5%: il **validation set** è la quantità di dati utilizzata per stabilire se la rete neurale sta convergendo;



Come descritto in dettaglio nel TR 4.2 [1], una rete neurale è composta da neuroni organizzati in layers. Ogni neurone della rete (tranne quelli contenuti nell’input layer), è la somma di tutti i suoi input che di fatto sono gli output del layer precedente moltiplicati per alcuni pesi. Un termine addizionale, il bias, viene aggiunto alla somma, ed infine una funzione di attivazione è applicata al risultato. Una rappresentazione di quanto appena espresso è rappresentata nella figura che segue Figura 37.

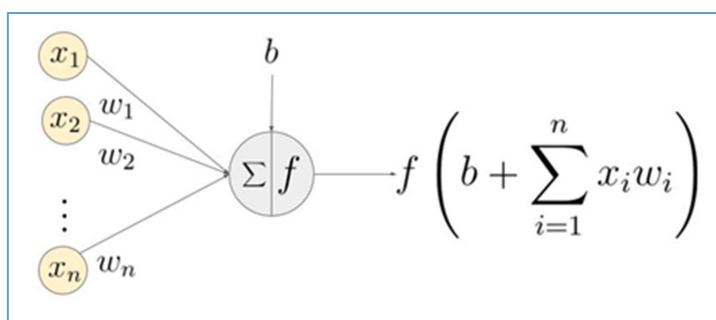


Figura 37 funzione rete neurale

I parametri rappresentati (weight, bias) sono esattamente i valori numerici che verranno di volta in volta calcolati addestrando la rete mediante un insieme esempi (dataset di training) già etichettati come avviene in un qualsiasi task di machine learning supervisionato. Il risultato finale darà un modello costruito su questi dati, che sarà in grado di prevedere i valori futuri non noti.

Nel caso specifico, la rete neurale, è stata addestrata utilizzando di default un algoritmo di ottimizzazione: l’algoritmo della Discesa del Gradiente (**Gradient Descent Algorithm**<sup>5</sup>) il cui obiettivo è quello di determinare l’insieme dei parametri interni (pesi, bias) del modello della rete neurale in modo da minimizzare una **loss function**, come la funzione di perdita logaritmica o lo scarto quadratico medio. Il “gradiente” si riferisce al calcolo di un gradiente di errore o pendenza di errore mentre per discesa si intende lo spostamento verso il basso lungo tale pendenza al fine di minimizzare l’errore. L’algoritmo è iterativo, in quanto l’obiettivo è quello di migliorare ad ogni passo i

<sup>5</sup> Gradient descent, [https://en.wikipedia.org/wiki/Gradient\\_descent](https://en.wikipedia.org/wiki/Gradient_descent)





parametri del modello. Ogni passaggio prevede l'utilizzo del modello con l'insieme corrente di parametri interni per fare previsioni su alcuni esempi, confrontando di volta in volta le previsioni con i risultati reali previsti, calcolando l'errore e usando l'errore per aggiornare i parametri interni del modello. La procedura di aggiornamento utilizzata nel caso delle reti neurali è il **backpropagation update algorithm**.

Oltre all’algoritmo di default di discesa del gradiente, per l’addestramento della rete è necessario impostare degli iperparametri: la dimensione del **batch**, il numero di tuple, le **epoch** ed infine una **funzione di validazione** da utilizzare per valutare il modello. Nella Tabella 6 sono rappresentati gli iperparametri utilizzati.

<b>Dimensione del batch</b> (numero totale di esempi di training presenti in ogni blocco in cui il dataset di training viene suddiviso ad ogni passo)	<b>200</b>
<b>Dimensione sequenza di tuple</b> (numero di tuple selezionate ad ogni passo)	<b>15</b>
<b>Numero di epoch</b> (rappresenta il numero di passaggi completi attraverso il dataset di training)	<b>10</b>

Tabella 6 Parametri utilizzati per l'addestramento della rete neurale

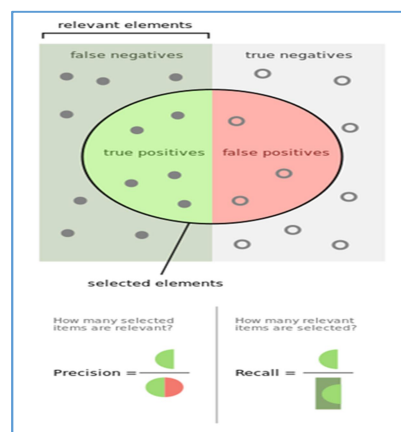
### Validation and testing

La misura utilizzata per validare il modello durante la fase di addestramento sarà la **F1-measure** (o F1-score)<sup>6</sup>, data dalla seguente formula:

$$2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

**Precision** è il numero di risultati positivi corretti diviso il numero di tutti i risultati positivi restituiti dal classificatore

**Recall**, è il numero di risultati positivi corretti diviso il numero di tutti i campioni pertinenti (tutti i campioni che avrebbero dovuto essere identificati come positivi).



<sup>6</sup> F1 Score, [https://en.wikipedia.org/wiki/F1\\_score](https://en.wikipedia.org/wiki/F1_score)



Le misure risultanti dalla validazione applicata rispettivamente ai dataset di training, al dataset di test e alla baseline sono rappresentate nelle tabelle seguenti:

Tabella 7 – Misure sul dataset di training

Accuracy	Precision	Recall	F1-score
0.99701666163141994	0.971379011274935	0.8719346049046321	0.918974358974359

Tabella 8 - Misure sul dataset di test

TEST Accuracy	Precision	Recall	F1-score
0.9917446808510638	0.731006160164271	0.8496420047732697	0.7858719646799117

Tabella 9 - Misure sulla baseline

BL 01 Accuracy	Precision	Recall	F1-score
0.5036170212765958	0.018248800548320766	0.5083532219570406	0.03523281779836241

Il modello generato è un **modello di classificazione binaria** in cui la categoria da apprendere è la feature ‘Failure’ (che può assumere valore 0 o 1). Il funzionamento prevede che a valle dell’arrivo di una sequenza di 15 tuple di dati (con una distanza temporale di 10 s l’una dall’altra) il sistema predica il verificarsi di un malfunzionamento del macchinario a distanza di 5 lassi temporali dall’ultima ricezione. In Figura 38 si mostra tutto il processo che è necessario per giungere alla creazione addestramento e memorizzazione del modello.

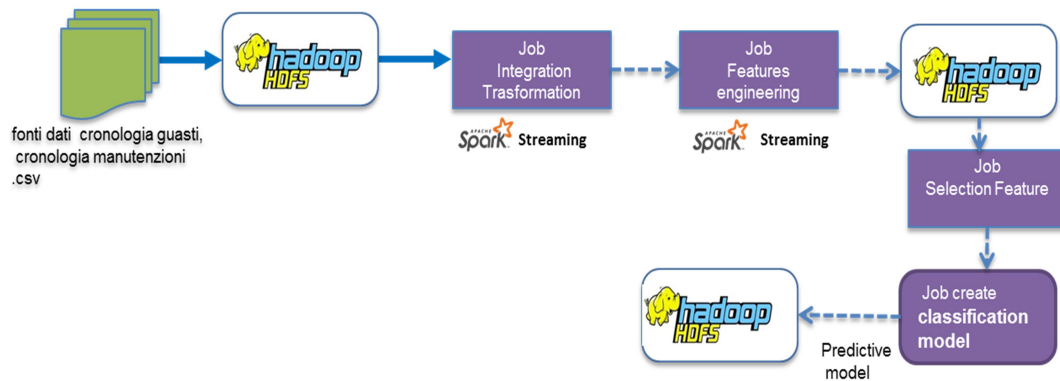


Figura 38: Pipeline di modelling for classification

#### 4.4. Servizi per la rilevazione di anomalie (anomaly detection)

Un'ulteriore possibile rappresentazione per problemi di diagnostica nel ambito Industry è quella di considerarlo il problema di anomaly detection. Ciò significa che il modello deve essere in grado di stabilire se il funzionamento della macchina rientra in uno stato normale o se si discosta da esso, rientrando cioè in un caso di anomalia. L'interpretazione del problema è dunque molto simile a quella della classificazione

binaria. Tuttavia questa metodologia si differenzia dalla classificazione dato che rientra nei casi di learning semi-supervisionato (a differenza dei caso precedente, che è supervisionato), in quanto il modello necessita solamente di imparare da input che rappresentano stati di funzionamento corretti e deve, in seguito alla fase di training, riconoscere stati anomali non noti, ossia di cui non conosce le caratteristiche. Ovviamente si perde la possibilità di distinguere i differenti guasti, ma questo approccio permette di risolvere le criticità che possono esistere con l'adozione dei modelli di classificazione. Poiché quest'ultimo necessita che la macchina va effettivamente portata in stato di guasto per la creazione del dataset di training in cui devono essere presenti degli esempi classi di guasti che si vorranno poi riconoscere in fase di testing. Quindi questa tipologia di tecnica può andare ben in situazione in cui ci sono storici e in ambienti controllati, ma i casi d'uso reali in cui si vuole applicare il sistema a macchine già in produzione. E' un procedimento troppo lungo e costoso per essere effettuato in linee produttive reali.



Pertanto analizzando in un contesto reale si è possibile l'obiettivo di riconoscimento di eventi rari non prevedibili. Quindi si provvederà all'analisi di un sistema di magazzino automatico. Il magazzino automatico da esaminare è dotato di scaffalature simili a quelle utilizzate nei magazzini manuali, ovvero dei reticoli in alluminio o in acciaio formati da montanti e correnti. I montanti sono profilati verticali, mentre i correnti sono profilati orizzontali. Le macchine che trasportano le UdC sono chiamate trasloelevatori, ed ogni corsia è dotata di almeno uno di essi. Un traslo è una macchina completamente automatica costituita da una colonna portante che si muove lungo l'asse x, un carrello che si muove lungo l'asse y e delle forche che si muovono lungo l'asse z. I movimenti in x e in y del traslo sono congiunti, ovvero avvengono in contemporanea, cosa che lo rende molto più efficiente di un semplice carrello elevatore, dal punto di vista cinematico. Il traslo a sua volta contiene un **dispositivo di prelievo**, cioè un apparato in grado di muoversi lungo l'asse Z per il prelievo e il deposito dei vassoi dagli armadi, e di scorrere nel traslo lungo l'asse Y per raggiungere un determinato vassoio presente nell'armadio.

Per permettere le movimentazioni del traslo e del dispositivo di prelievo lungo i tre assi di riferimento che compongono lo scenario, vi sono tre motori che garantiscono l'erogazione di potenza per movimentare l'apparato. In particolare, il motore che permette al traslo di muoversi lungo l'asse X è solidale col macchinario e posizionato in un estremo dell'asse X, il motore che consente il movimento lungo l'asse Y è solidale con il traslo e si muove con esso, mentre il motore che consente al dispositivo di prelievo di muoversi lungo l'asse Z è contenuto nel dispositivo di prelievo stesso.

La movimentazione lungo gli assi avviene mediante l'utilizzo di catene e cinghie collocate in particolari spazi. In particolare, per garantire la movimentazione lungo l'asse X delle cinghie orientate in maniera orizzontale sono presenti alle estremità verticali del macchinario, quindi alla base e al tetto del macchinario; per muoversi lungo l'asse Y si prende in esame il caso in cui vi siano quattro catene orientate verticalmente e disposte agli spigoli del traslo; per muoversi lungo l'asse Z sono presenti due catene orientate in profondità e disposte ai lati del dispositivo di prelievo. Il tutto è gestito da encoder sensori ottici rilevano i movimenti del traslo pre-analisi sul campo il caso che risultato la più facile da riconoscere e quella che si distanzia di più è l'allentamento delle catene del dispositivo di prelievo in quanto è il guasto che ha i maggiori effetti sui dati raccolti



dalla sensoristica. Le cause che possono determinare problemi sulle catene, possiamo essere:

- Sovraccarico: la catena è sottoposta a un eccessivo carico che può causare una deformazione permanente del materiale e porta a una vita utile della catena molto corta.
- Fatica: la catena è sottoposta a carichi elevati ripetutamente oltre al limite della resistenza.
- Usura: il carico a cui la catena normalmente è sottoposta consuma il materiale in modo tale che la catena si allunghi oltre il limite utilizzabile.

Tra le ipotesi di cause l’attenzione sarà focalizzata a situazione di usura dovuta ai ripetuti cicli di carico e deposito dei vassoi da e verso il magazzino o la baia, che possono contenere pesi a volte anche molto elevati.

In particolare ci si pone nell’ottica di andare a misurare e basare le proprie idee sugli algoritmi predittivi sulle quantità rilevate dai due encoder che monitorano l’asse Y. In condizioni ottimali, prima della fase di prelievo di un carico da parte del dispositivo di prelievo, con una buona probabilità l’encoder a filo che risente del peso del vassoio e l’encoder calettato sul motore che eroga potenza lungo l’asse Y e che non risente del peso del vassoio sono in uno stato di consistenza, cioè il valore misurato da entrambi gli encoder coincide o è molto vicino. Nel momento in cui il dispositivo di prelievo si accinge a prendere il vassoio, appena il baricentro del vassoio supera il magazzino e si sposta sopra il dispositivo, scarica su di esso tutto il proprio peso. Il dispositivo di prelievo si trova quindi all’improvviso a sostenere un peso non più nullo, il che lo fa affondare leggermente verso il basso. Tale spostamento è ignorato dall’encoder calettato sul motore, ma viene misurato dall’encoder a filo verticale posto a un lato del traslo. Il sistema in retroazione cerca quindi rapidamente di bilanciare la potenza del motore in modo da riportare la differenza tra i due encoder quanto più prossima allo zero.

A valle di quanto spiegato fino ad ora, la differenza che viene riscontrata tra le due misurazioni di posizione dell’asse Y, rappresenta la variabile principale nell’analisi del problema di sovrallungamento delle catene lungo l’asse Y. È infatti proprio su questa variabile che le analisi effettuate si concentrano, ed in questo documento faremo



riferimento a tale valore con il termine spread. Data la misura della posizione del dispositivo di prelievo dell’encoder a filo (YActualPosition), e data la misura di tale posizione dell’encoder calettato sul motore (YPositionFeedBack1Value), lo spread è definito in questo modo:  $spread = YActualPosition - YPositionFeedBack1Value$

#### *Data Preparation e feature engineering*

Per questo specifico caso d’uso, il dataset oggetto di analisi del dispositivo di prelievo è stato fornito da ICAM azienda produttrice .Come spiegato nel paragrafo precedente il macchinario è disposto di una serie di sensori, di natura anche diversa. I due tipi principali sono gli Encoder. Tutte le rilevazioni prodotte dai sensori, vengono indirizzate ad un Raspberry, ovvero una unità di calcolo che si occupa di rimappare e assemblare i dati che vengono trasmessi con il medesimo protocollo mqtt e pubblicati sul topic kafka, Data la moltitudine di variabili a disposizione, nell’analizzare tali dati è stato limitato il perimetro di variabili da prendere in considerazione, tenendo conto del problema di manutenzione che è stato preso in considerazione, ovvero la sovralongazione delle catene. Nonostante siano state analizzate molte variabili, quelle che sono risultate essere più significative ed interessanti rispetto al caso di manutenzione individuato sono le posizioni lungo l’asse Y del dispositivo di prelievo, riportate dagli encoder indicati nei capitoli precedenti, e lo spread. In Figura 39 viene mostrato un plot che fa riferimento ad un breve lasso di tempo di operazione della macchina, e che mostra l’andamento delle due misurazioni. Come l’immagine mostra, e come lo scatter plot mostrato in Figura 39 conferma, le due misurazioni sono apparentemente sempre uguali, o comunque molto vicine tra di loro, ed effettivamente questo è il genere di comportamento che ci si aspetta nel momento in cui i pesi caricati sul dispositivo di prelievo non siano esagerati.

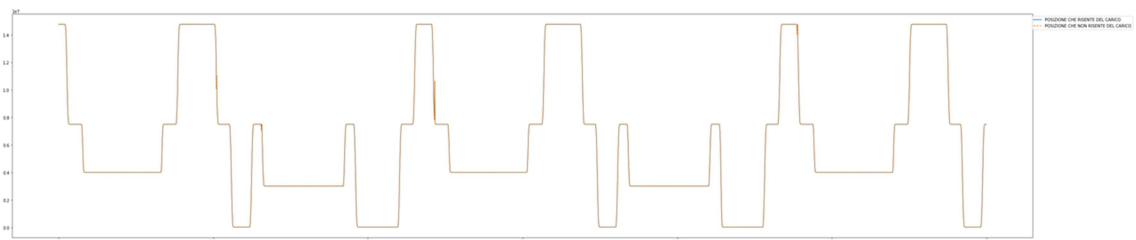


Figura 39 Plot delle misurazioni macchinario ICAM

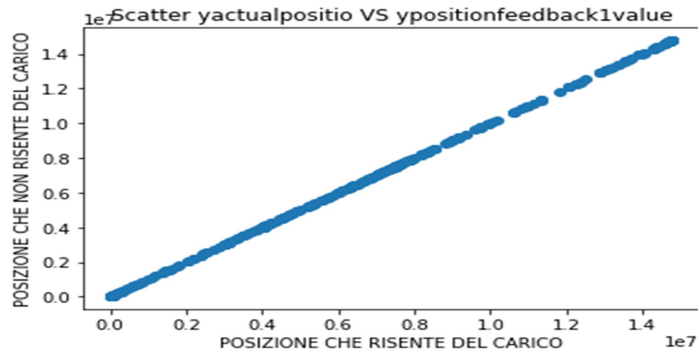


Figura 40 Differenze rilevazioni YActualPosition e YPositionFeedback1Value

A livello microscopico tuttavia, le due misurazioni si discosteranno sempre, per effetto della gravità che sarà sempre una forza presente. Per questo caso specifico per effettuare le analisi sarà necessario eseguire una fase di preprocessing ritenuta utile per mitigare la poca precisione del sensore dal momento che una porzione di essi presentava dei valori del tutto anomali, dovuti probabilmente ad un errore di misurazione dei sensori. Inoltre per l'estrazione di feature quindi individuare dati anomali saranno necessari calcolare una serie di misure statistiche. Le feature che si è scelto di utilizzare sono prese dalla letteratura e risultano essere fra le più utilizzate in analisi vibrazionale o analisi dei segnali in generale. L'elenco delle feature estratte, con nome e descrizione, è il seguente:

- Mean - La media di tutti i valori di accelerazione del segnale
- Max - Il massimo fra i valori di accelerazione del segnale
- Min - Il minimo fra i valori di accelerazione del segnale
- Std - La deviazione standard dei valori di accelerazione

Un'altra operazione necessaria è l'uniformazione della frequenza del dataset. Infatti il dataset Ethercat originale ha una frequenza di rilevazione di circa 200 millisecondi, generando quindi molti dati ridondanti e che aumentano il volume del dataset ma senza aggiungere informazioni importanti. Quindi una volta selezionate le variabili individuate come significative rispetto allo scenario (ovvero le due misurazioni di posizione Y e la loro differenza), tale dataset saranno necessario portalo ad una frequenza di 1 minuto.

Una volta analizzati selezionati, e puliti i dati e fatte delle caratterizzazioni delle variabili dal punto di vista statistico per capire il funzionamento del macchinario si passerà .



Per fronteggiare questa analisi preliminare e quindi tutta la fase di analisi esplorativa dei dati e di sperimentazione iniziale, gestione e manipolazione dei dataset si farà uso dei servizi infrastrutturali resi disponibili dalla piattaforma di Data Exploration in particolare la libreria Pandas e Matplotlib per la visualizzazione grafica dei segnali di vibrazione nel dominio del tempo e delle frequenze.

### *Anomaly Detection*

Alla luce delle analisi e delle manipolazioni sui dati lo scenario che si prevede di attuare per il caso ICAM è quello di rilevazione delle anomalie su base statistica.

L'approccio più semplice per il rilevamento delle anomalie per ogni serie temporale calcola il valore medio del segnale passato e monitora ogni deviazione da esso, andando avanti.

A tal proposito per definire statisticamente un comportamento anomalo, dal momento che i dati a disposizione non erano rappresentativi di una anomalia, essi sono stati considerati come una baseline da cui partire per poter poi considerare anomalo tutto ciò che non rispecchiasse le loro caratteristiche.

In pratica, presa la distribuzione del valore assoluto di spread registrato tra il 12 Dicembre 2018 e il 26 Febbraio 2019, sarà definito un limite di numero di deviazioni standard entro cui la media dello spread registrato in real time deve mantenersi, al di sopra della media di tale distribuzione, classificata come ‘normale’.

Il numero limite di deviazioni standard, è un dato da definire in maniera empirica con l'aiuto del personale tecnico specializzato, che conosca il funzionamento del macchinario di scaffalature. Per maggior dettagli sia sullo scenario e sia sulle scelte implementate si rimanda al TR 5.2 [18]

Dunque, data la distribuzione dello spread che rispecchia un comportamento normale del macchinario e quindi di stato di salute delle catene, viene eseguito un controllo a frequenza predefinita (ad esempio ogni 30 minuti) della media dello spread calcolata in tale lasso di tempo.

Se la media dello spread supera di un determinato numero di deviazioni standard (definito empiricamente) la media della distribuzione normale dello spread (che riflette





le catene sane) allora viene inviato avviato un servizio di Alerting al componente **Alert Controller** del Data Monitoring .Dovendo formalizzare la logica di rilevazione delle anomalie, dato  $n$ , il numero di deviazioni standard entro cui la media dello spread (in valore assoluto e mediato al minuto) deve mantenersi rispetto alla media della distribuzione ‘normale’ di tale spread, data  $m$  tale media di distribuzione normale, data  $m1(t)$  la media dello spread che si riscontra in streaming, calcolata ogni  $t$  istanti, ed infine data  $std$  la deviazione standard della distribuzione normale di spread, possiamo definire la regola di rilevazione di una anomalia riguardo lo stato di sovralongazione delle catene sarà così definita:

$$m1(t) > m + n*std$$

Mentre il servizio Infrastrutturale messo a disposizione dalla “Data Governance” e quindi Dalla Piattaforma DF per apprendimento automatico volto a rilevamento di anomalie nei dataset sarà Scikit-Learn.

#### *Real time Processing*

Dal momento che nell’implementare tale meccanismo non sono stati messi a disposizione dei sensori reali da cui prendere dati in real-time, sarà necessario un modulo quindi un servizio client che simulerà ingestion dei dati attraverso il protocollo MQTT. Tale simulazione deve ovviamente rispecchiare le caratteristiche reali dello spread, e nel fare questo deve anche ricreare uno scenario in cui i record generati possano essere significativi dal punto di vista del riconoscere una anomalia, e quindi poter dimostrare l’avvio di un workflow di manutenzione.

A tal proposito la logica di simulazione si basa su un principio molto semplice: data la distribuzione del valore assoluto dello spread, mediato al minuto, considerata ‘normale’, e potendo qualitativamente assumere una distribuzione normale di tale spread, per poter simulare i dati verranno casualmente ricampionati da tale distribuzione a frequenza fissa (ovvero sempre ogni minuto).

Man mano che i record verranno ricampionati, saranno inviati in un topic di Kafka, simulando così il processo di streaming dei dati sensoristici di dati ‘normali’ di spread.



Tuttavia, per poter completare lo scenario è stato necessario anche simulare dei dati anomali, in modo tale da poter dimostrare e verificare il comportamento corretto dell’implementazione proposta e progettata. Per l’elaborazione dei dati in real-time E valutare il delta si utilizzerà un simulatore realizzato con il framework node.js . Quindi i dati giungeranno dal simulatore come mostrato in Figura 41 mentre in fase di sperimentazione sia avvierà lo scenario con il sistema ICAM il sistema reale come mostrato in Figura 42 ma per dettagli in TR 5.2 [18] .

Mentre per l’elaborazione dei dati streaming avverrà con spark streaming e si prevederà l’implementazione di un servizio Alerting deputato all’invio dei dati delle serie temporali al datastore della soluzione Smart Factory per la visualizzazione in tempo reale mediante data visualization le rilevazione in tempo reale normali e anomali. Se ci sono anomalie se si vedrà nel paragrafo successivo l’avvio del processo di manutenzione per dettagli si rimanda paragrafo 0 .

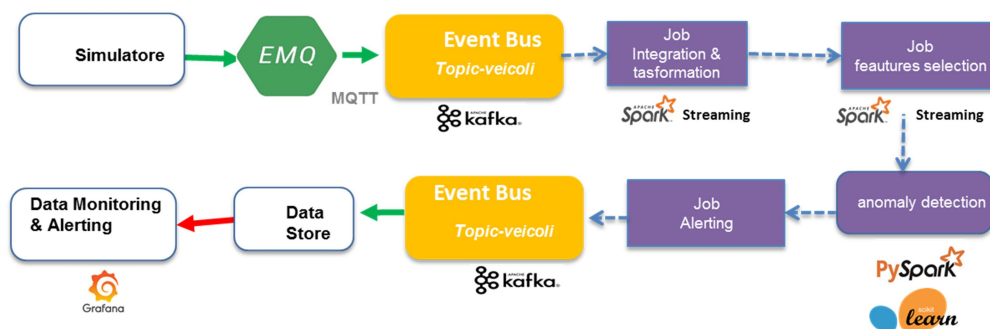


Figura 41 :Real-time Data Processing and Monitoring caso simulato

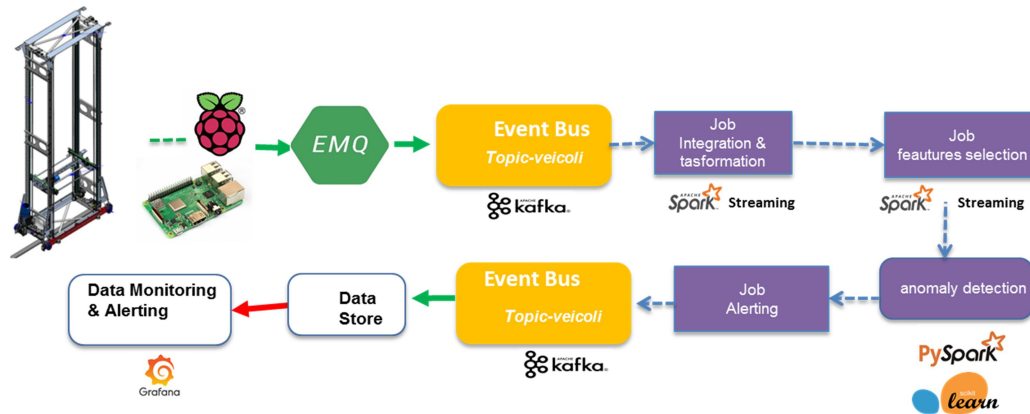


Figura 42 :Real-time Data Processing and Monitoring contest reale

### Alerting

Al fine di poter integrare il meccanismo e i servizi di anomaly detection rimpementati nella Piattaforma Digital Future con il modulo di Asset e Workflow Management, verranno registrati in quest’ultimo i KPI che permetteranno l’esecuzione vera e propria dello scenario di Anomaly Detection. Nel cruscotto dell’anagrafica dei KPI verranno infatti registrati 3 KPI fondamentali:

- Absolute Value Spread Normal Mean
- Absolute Value Spread Normal Standard Deviation
- Number of Standard Deviations Above the Mean.

Ad ogni esecuzione del Job in Streaming, il “Middleware Interface” fornirà i KPI configurati nel modulo “Workflow Management” attraverso la definizione di API. Successivamente verrà calcolata la media dei valori di spread ricevuti e confrontato con il KPI composto calcolato a partire dai KPI forniti dal Middleware Interface sarà definito r dalla seguente formula:

$$KPI = Mean_k + Stds_k * Dev_k$$



### Start process mantanance

Al superamento della media dello spread calcolato nella finestra temporale fissata, il servizio Alerting invocherà un endpoint specifico del “Middleware Interface” come mostrato in Figura 43 permettendo così l’avvio di un processo di manutenzione che sarà simile a quello descritto nel paragrafo 0 e per dettaglio sul caso sperimentale contesto con ICAM si rimanda a TR 5.2

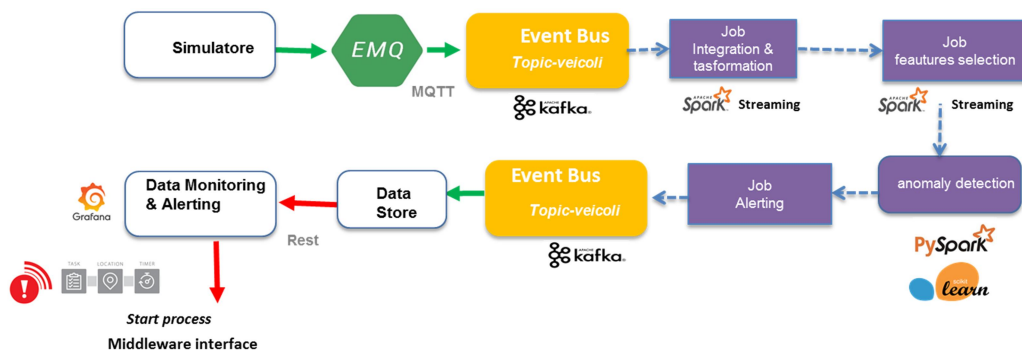


Figura 43: Start process

### Research Document

Nelle attività di manutenzione per macchinari più o meno complessi come il caso dei magazzini automatici la mole di dati ed informazioni legate allo svolgimento delle attività possono essere numerose. Inoltre, la disponibilità di tecnici esperti è esigua e la dislocazione dei clienti in diversi non aiuta. Per questo motivo, risulta necessario sfruttare tecniche di analisi semantica messe a disposizione del componente Multimedia Manager in particolare del componente Media Search Manager della Piattaforma DF che permettono di estrarre informazioni da qualsiasi genere di documento come ad esempio PDF, documenti per l’automazione d’ufficio, fogli di calcolo e presentazioni multimediali.

In particolare si utilizza il repository della soluzione Smart Factory per archiviare informazioni in modo organizzato per sfruttando la soluzione Alfresco e la console di ricerca resa fruibile direttamente ai dipendenti del cliente e/o agli operatori. Quest’ultimi possono ricercare la documentazione per tag oppure possono avere a diposizione il materiale in allegato al workorder. Se nello specifico l’operatore deve eseguire delle ricerche istruzioni più di dettagliate inserira dei tag di riferimento, il la soluzione Smart Factory invierà la richiesta alla Piattaforma DF la quale attiverà la



componente Media Search Manager che procederà alla ricerca e a restituire le procedure manutentive necessarie per maggiore dettagli si rimanda al Allegato di riferimento D 4.3.

#### 4.5. Servizi Monitoraggio della linea produttiva

Come espresso all’inizio nel Capitolo 3, nel presente progetto sono stati analizzati integrabili con soluzione SAP per:

- **Caso 1:** relativo al monitoraggio e predizione del numero di pezzi prodotti dalla macchina fresatrice in un dato lasso di tempo al fine di rilevare eventuali criticità nel processo produttivo, causati dal non raggiungimento della quantità target di pezzi stabiliti nell’ordine di produzione;
- **Caso 2:** relativo al monitoraggio e predizione stato di salute dei sensori stessi, nello specifico della velocità di rotazione della punta della macchina fresatrice al fine di rilevare eventuali anomalie nel processo produttivo, causati ad esempio dalla errata configurazione del macchinario non idonea al raggiungimento del target stabilito nell’ordine di produzione o da un rallentamento della velocità dovuta ad un guasto del macchinario;

I modelli di funzionamento relativi ai 2 scenari appena citati sono rappresentati rispettivamente nelle Figura 44 e Figura 45



Figura 44 - Modello di funzionamento scenario predizione numero di pezzi prodotti



Figura 45 - Modello di funzionamento scenario predizione velocità di rotazione

Nel seguito descriviamo i passi principali e flussi operativi e job che posso portato all'implementazione dei casi.

#### 4.5.1. *Servizi per prevedere i valori dei sensori in istanti futuri*

Il caso preso in esame è dato da una macchina che aziona una fresatrice in grado di produrre varie tipologie di manufatti (Linea di Produzione), ad esempio anelli di diametro diverso, connessa a sensori che misurano grandezze di interesse per il processo quali la potenza elettrica istantanea, la velocità di rotazione della punta e lo stato di led multicolore di colore diverso (rosso, verde, blu), associati ciascuno ad uno specifico ordine di produzione, che indicano con il loro stato (acceso/spento) l'avvenuta produzione di un singolo pezzo. Inoltre altri sensori segnalano le anomalie che si possono verificare nel corso della fase di produzione del singolo lotto identificate da un codice d'errore; il sistema comanda in tale eventualità il fermo macchina (l'operatore deve premere un pulsante lampeggiante per interrompere la produzione e la velocità di rotazione della punta si azzerà) e una volta effettuato l'intervento di manutenzione viene comandato il riavvio (l'operatore deve premere un altro pulsante lampeggiante per riprendere la produzione). Inoltre nel corso della produzione e fino al termine della



stessa viene anche simulato un rallentamento della macchina con conseguente dimezzamento della velocità angolare della punta. Nella Figura 46 che segue vengono descritti i vari elementi che compongono lo scenario e il flusso delle informazioni.

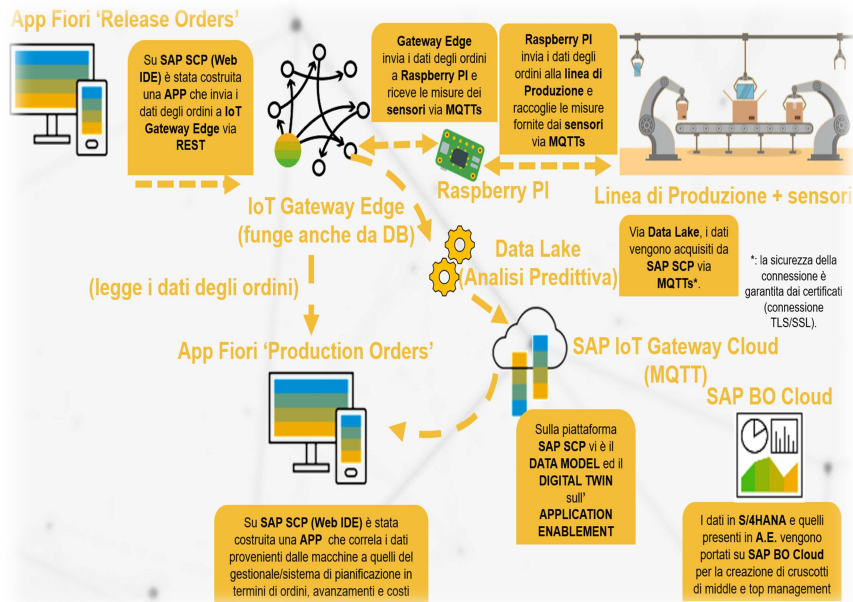


Figura 46: componenti di avvio ordine su sap

Si Prevede anche in questo caso l’impiego di un dispositivo Raspberry PI Model 3B+ avente le seguenti funzioni:

- controllo led X (accensione/spengimento) (X=1,2,3)
- acquisizione stato (acceso/spento) del led X indicante l’avvenuta produzione del singolo pezzo del lotto X (X=1,2,3)
- acquisizione misura potenza istantanea consumata fornita da energy meter
- acquisizione misura velocità di rotazione istantanea della punta rotante
- controllo pulsanti per il controllo della produzione (Avvio, Stop, Riavvio, Incremento produzione)
- controllo motore collegato alla punta rotante
- controllo display LCD a 16 bit per la visualizzazione andamento produzione del lotto X (X=1,2,3)

Si prevede inoltre l’utilizzo dei seguenti componenti hardware:

- un misuratore di potenza elettrica istantanea (energy meter)



- un sensore per la misura della velocità istantanea di rotazione della punta
- un sensore per la rilevazione delle anomalie
- un led multicolore (rosso, blu, verde)
- quattro pulsanti (verde, rosso, blu, bianco)
- scatola contenente Raspberry PI e su cui sono montati i quattro pulsanti
- motore elettrico che aziona una punta rotante che simula una fresatrice
- modulo relè a 4 vie per il controllo del motore e dei pulsanti
- cartonato che simula macchina a controllo numerico (PLC)
- monitor su cui vengono visualizzate le applicazioni SAP Web IDE disponibili su Cloud

L'energy meter, il sensore di velocità e il led multicolore sono collegati a porte GPIO output del Raspberry PI e forniscono rispettivamente la misura della potenza istantanea utilizzata per la realizzazione del singolo pezzo, la velocità di rotazione della punta e l'indicazione tramite accensione dell'avvenuta produzione di un singolo pezzo. Il colore del led identifica l'ordine di lavoro in corso di esecuzione. Il motore elettrico e i quattro pulsanti sono collegati a porte GPIO input del Raspberry PI: all'atto del rilascio

dell'ordine di produzione da parte dell'applicazione SAP Fiori “Release Orders” sul display LCD appare la scritta “Order ready to go – Press green button to start” e il pulsante verde si illumina; dopo aver premuto il pulsante viene avviata la produzione dell'ordine e il motore aziona la punta che comincia a ruotare; nel contempo il led si accende ogni volta che viene prodotto un pezzo e su display LCD viene visualizzata l'informazione sull'ordine che si sta eseguendo e il numero di pezzi fatti sul totale. Si prevede infine l'utilizzo della componente di EDGE computing (estrazione features e similarity scoring).

La serie temporale utilizzata per l'implementazione questo scenario sono ottenuti come per gli altri casi da una fase di preparazione dei dati relativa alla velocità di rotazione della punta (misura rotate). Dopo questa passo preliminare per la creazione del modello ARIMA segue la determinazione della stazionarietà della serie temporale, l'eventuale determinazione del parametro d del modello ARIMA, per poi passare alla





determinazione dei parametri AR(p) e MA(q). I dati sono dai i indicatori e misure indicate in Tabella 10.

Indicatori	Misure
<b>pieces_DF_actual:</b> numero di pezzi realmente prodotti dalla fresatrice	<b>pieces_actual:</b> numero reale di pezzi prodotti <b>time_forecast:</b> data e ora di rilevazione <b>id_ordine:</b> identificativo ordine di produzione <b>target_quantity:</b> numero di pezzi previsti dall’ordine di produzione
<b>pieces_DF_forecast:</b> numero di pezzi predetto	<b>pieces_forecast:</b> numero di pezzi predetto <b>time_forecast:</b> Data e ora di riferimento <b>id_ordine:</b> identificativo ordine
<b>pieces_DF_threshold:</b> soglie di tolleranza inferiore e superiore numero pezzi predetto	<b>pieces_forecast_lowerbound:</b> soglia di tolleranza inferiore numero di pezzi predetto <b>pieces_forecast_upperbound:</b> soglia di tolleranza superiore numero di pezzi predetto <b>time_forecast:</b> data e ora di riferimento <b>id_ordine:</b> identificativo ordine
<b>rotate_DF_actual:</b>	<b>rotate_actual:</b> velocità di rotazione reale rilevata <b>time_forecast:</b> data e ora di riferimento <b>id_ordine:</b> identificativo ordine
<b>rotate_DF_forecast:</b>	<b>rotate_forecast:</b> velocità di rotazione predetta <b>time_forecast:</b> data e ora di riferimento <b>id_ordine:</b> identificativo ordine
<b>rotate_DF_thresholds</b>	<b>rotate_forecast_lowerbound:</b> soglia inferiore di tolleranza velocità di rotazione predetta <b>rotate_forecast_upperbound:</b> soglia superiore di tolleranza velocità di rotazione predetta <b>time_forecast:</b> data e ora di riferimento <b>id_ordine:</b> identificativo ordine di produzione

Tabella 10 Indicatori e misure

*Data Preparation e feature engineering da dati storici*

Il processo preliminare di preparazione dei dati è lo stesso già presentato nel paragrafo 7.2.1 relativamente allo scenario di predizione dello stato di salute del macchinario. I

Date	rotate	pieces	tot_pieces
01/01/2015 05:00:10	398,1191317	1	1
01/01/2015 05:00:20	412,3476603	2	3
01/01/2015 05:00:30	428,8524735	2	5
01/01/2015 05:00:40	429,4540881	2	7
01/01/2015 05:00:50	411,4737903	2	9
01/01/2015 05:01:00	411,1166697	2	11
01/01/2015 05:01:10	413,5895092	2	13
01/01/2015 05:01:20	409,5985793	2	15
01/01/2015 05:01:30	426,3010236	2	17
01/01/2015 05:01:40	419,095329	2	19
01/01/2015 05:01:50	410,5433289	2	21
01/01/2015 05:02:00	414,5127504	2	23
01/01/2015 05:02:10	398,0629857	1	24
01/01/2015 05:02:20	426,0156232	2	26
01/01/2015 05:02:30	395,3409278	1	27
01/01/2015 05:02:40	424,4778755	2	29
01/01/2015 05:02:50	433,4149921	2	31
01/01/2015 05:03:00	423,575889	2	33
01/01/2015 05:03:10	448,7006504	2	35
01/01/2015 05:03:20	412,7212873	2	37
01/01/2015 05:03:30	430,7975527	2	39
01/01/2015 05:03:40	401,3648539	2	41

Exprivi

verticali  
 fabbrica  
 ligente 88



dati simulati vengono reperiti da un CSV preventivamente memorizzato su HDFS (Hadoop Distributed File System ), file system distribuito presente sulla piattaforma Digital Future, che consente di archiviare in maniera trasparente grandi quantità di dati su di replicare le informazioni archiviate e comprimerle; A partire dai dati di ottenuti dalla prima fase di pre-processing e raffigurati in Figura 47 per l’analisi dello scenario in oggetto è stato generato un nuovo dataset contenente i dati relativi alla serie temporali con intervalli temporali pari a 10s relativa solo alla velocità di rotazione le cui misure sono state ottenute mediante l’utilizzo di una funzione generatrice di numeri random con distribuzione gaussiana; Infine si è generato una vista del dataset finale come in **Errore. L'origine riferimento non è stata trovata.** contenente i dati delle serie temporali per la velocità di rotazione e il numero di pezzi prodotti.

Tabella 11: Dataset finale delle serie temporali fresatrice

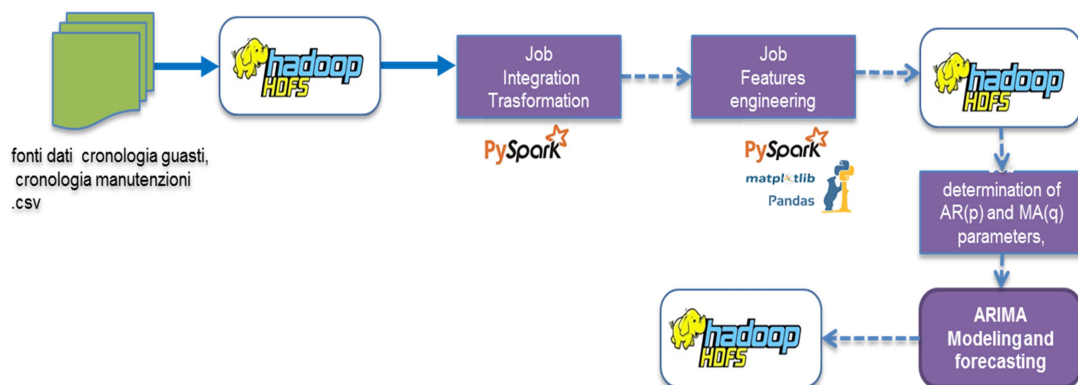


Figura 47 : Pipeline create modello ARIMA



### *Creating ARIMA models*

Come ampiamente descritto nel TR 4.2 [1], Un altro modello che viene utilizzato spesso in letteratura è l’Auto Regressive Integrated Moving Average (ARIMA), utilizzato per modellare il comportamento di timeseries e poterne poi predire l’andamento.

In [19] viene usato ARIMA per monitorare lo stato di salute dei sensori stessi, effettuando anomaly detection. Si utilizza lo storico dei valori raccolti per allenare il modello, che viene poi utilizzato per fornire previsioni dei prossimi punti. Si confrontano poi i valori effettivamente forniti dal sensore con quelli stimati, e se la differenza supera una soglia prefissata si segnala una anomalia. Anche in [20] ARIMA viene utilizzato per la previsione dei valori dei sensori in istanti futuri, anche se si differenzia dal lavoro precedente in quanto non si propone un sistema a soglia fissa per rilevare le anomalie, ma si usa un sistema di learning supervisionato per effettuare classificazione di difetti di macchine tagliatrici. In questo caso la fase di training della classificazione avviene utilizzando dati reali raccolti da sensori in uno stato di guasto, mentre in fase di testing i valori sono quelli previsti da ARIMA. Analizzando quest’ultimo caso per la creazione di un modello predittivo ARIMA è necessaria la definizione dei parametri del modello ed in particolare:

- ✓ **p – numero di termini autoregressivi;**
- ✓ **d – numero di differenze non stagionali;**
- ✓ **q- numero di termini per la media mobile (mobile-average)**

e da un valore costante da aggiungere secondo necessità. La definizione dei parametri del modello passa attraverso le seguenti fasi principali:

- Rendere la serie temporale stazionaria, qualora necessario, mediante un processo di differenziazione; al termine di questa fase sarà determinato il parametro d;
- Studiare i pattern di **autocorrelation** e **partial autocorrelation** al fine di determinare se i *lags* (ritardi) della serie stazionaria e/o i ritardi degli errori di predizione devono essere inseriti nell’equazione per il forecasting; questa fase serve per determinare i parametri p e q;



- **Addestrare** il modello in base ai parametri sopra determinati e controllare i grafici PACF e ACF dei residui per verificare che tutti i pattern siano stati inclusi nel modello;
- I pattern rimanenti in PACF e ACF possono suggerire la necessità di aggiungere ulteriori termini AR e MA;

#### 4.5.1.1.1. Criteri generali di determinazione dei termini AR(p) e MA(q)

Per la definizione dei parametri AR(p) e MA(q) del modello vengono utilizzati:

- **Il plot ACF (AutoCorrelation Function)**, che mostra la correlazione tra una serie temporale con se stessa a differenti ritardi: L'autocorrelazione di Y al ritardo k è la correlazione tra Y e il ritardo (Y, k);
- **Il plot PACF (Partial Autocorrelation Function)**, che mostra la «parziale» correlazione di una serie temporale con se stessa al ritardo k, che non è spiegata dalle autocorrelazioni di ordine più basso.

La Figura 48 mostra i criteri generali di definizione dei parametri AR(p) e MA(q):

- Nel caso in cui ACF decresce gradualmente e PACF si interrompe bruscamente dopo un certo numero di ritardi è necessario aggiungere dei termini AR per la creazione del modello predittivo: una serie temporale AR solitamente presenta una autocorrelazione positiva al ritardo 1;
- Nel caso in cui ACF si interrompe bruscamente dopo alcuni ritardi e PACF decresce gradualmente è necessario aggiungere dei termini MA per la creazione del modello predittivo: una serie temporale MA presenta generalmente una autocorrelazione negativa al ritardo .

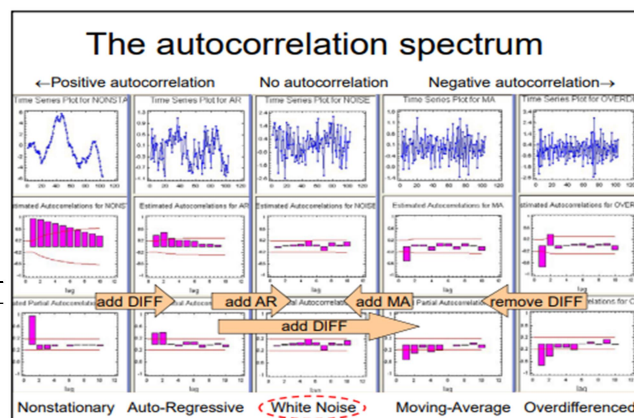




Figura 48 - Spettro con criteri di definizione dei termini AR e MA

#### 4.5.1.1.2. Determinazione della stazionarietà della serie temporale

Per la determinazione della stazionarietà della serie temporale si utilizzano due approcci:

- Visualizzazione grafica della serie temporale al fine di rilevare eventuali trend o elementi di stagionalità che denotino una non stazionarietà della serie temporale; nella figura che segue viene mostrato l'andamento della serie temporale utilizzando le apposite librerie grafiche di python;

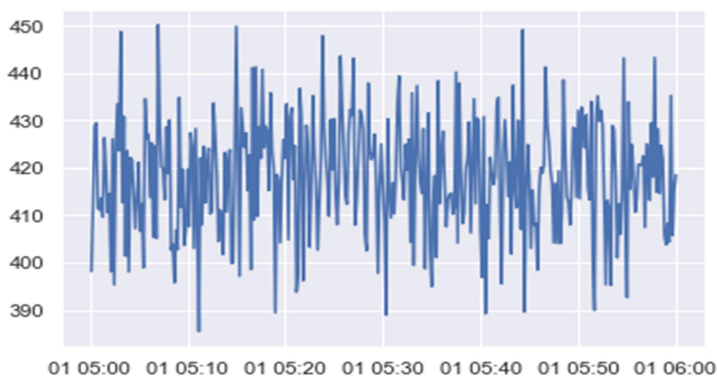


Figura 49 - Serie temporale relativa alla velocità di rotazione

- Metodo basato sul test statistico «Augmented Dickey-Full Test», utilizzando una specifica funzione python generata ad-hoc:
  - ✓ Null-Hypothesis, la serie ha una radice unitaria il che implica che non è stazionaria ossia presenta una struttura dipendente dal tempo;
  - ✓ Alternate Hypothesis, l'ipotesi nulla viene scartata. La serie temporale non ha radice nulla e quindi è stazionaria ossia non ha una struttura dipendente dal tempo.

```
Result of Dickey-Fuller Test
Test Statistic      -19.412753
p-value             0.000000
#Lags used          0.000000
Number ofObservations Used  359.000000
Critical Value (1%)  -3.448697
Critical Value (5%)  -2.869625
Critical Value (10%) -2.571077
dtype: float64
```



Infine l'interpretazione del risultato del test viene effettuata osservando il valore p-value: un p-value sotto una soglia definita (5% o 1%) suggerisce che la serie è stazionaria altrimenti è non stazionaria.

Nel caso specifico dalla visualizzazione grafica della serie (media e varianza sono costanti) e dal test di Dickey-Full (il valore di p-value è inferiore a 0,05) si evince che la **serie risulta essere stazionaria**: non è necessario procedere con la differenziazione per renderla stazionaria e il **valore di differenziazione d del modello ARIMA è posto pari a 0**.

#### 4.5.1.1.3. Determinazione dei termini AR(p) e MA(q)

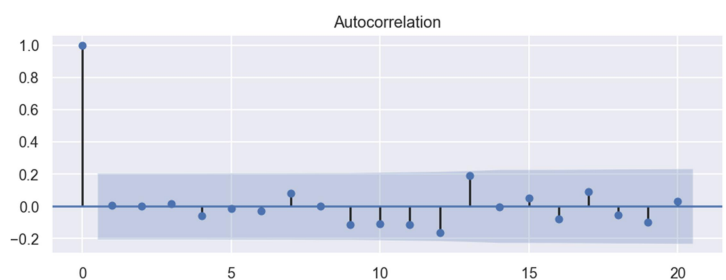
Per la determinazione dei parametri AR(p) e MA(q), è necessario procedere nel seguente modo:

- Osservando il grafico di autocorrelazione è possibile evincere che la correlazione tra una osservazione della serie temporale al tempo  $t$  e



le precedenti è randomica, non segue un andamento ben preciso;

- La correlazione al ritardo 1 è «leggermente» negativa pertanto applichiamo un modello MA. Per individuare il numero di termini MA è necessario vedere a quale ritardo il grafico



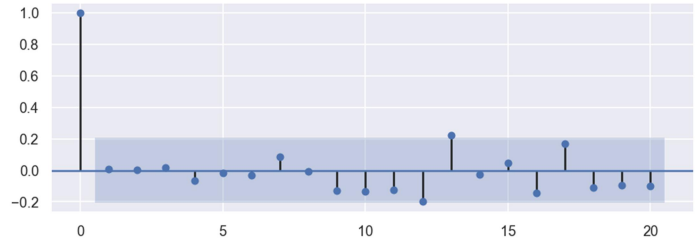
taglia la soglia (la regione blu). Nel caso specifico i lag (ritardi) di rilievo sono 3 e 13 pertanto scegliamo di selezionare un modello

MA (3) e quindi un parametro  $q=3$ ;

- Una volta determinato il parametro  $q$  addestriamo il modello ARIMA(0,0,3) e osserviamo cosa succede alla componente residua nuovamente attraverso la visualizzazione dei plot PACF e ACF;



**Digital Future – POR Regione Puglia**  
**Fondo Europeo di Sviluppo Regionale 2014-2020**  
**Titolo II – Capo 1 “Aiuti ai programmi di intervento delle Grandi Imprese”**



A questo punto si evidenzia una normalizzazione del plot ACF mentre nel plot PACF il grafico taglia la soglia in corrispondenza dei lag 12 e 13. Scegliamo quindi di addestrare un modello ARIMA (12,0,3).

*Validation model ARIMA*

Il modello ottenuto dalle fasi definite nei precedenti paragrafi, è quindi un modello ARIMA (12,0,3).

La Figura 50 mostra le caratteristiche del modello, da cui si evince che I valori del P-value nella colonna 'P>|z|' per AR(12) e MA(3) sono altamente significativi in quanto  $\ll 0.05$ ; Dplot degli errori residui (Figura 51) si evince che la media è molto vicina allo zero e la

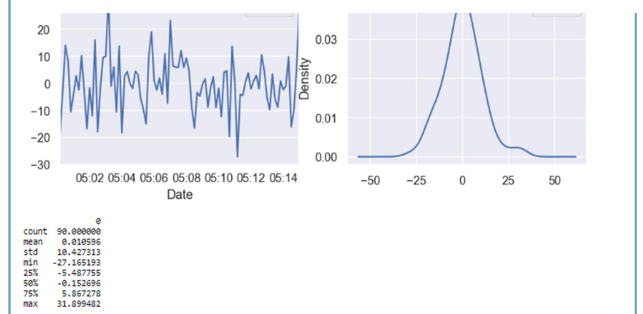
ARIMA Model Results						
Dep. Variable:	rotate	NO. Observations:	90			
Model:	ARMA(12, 3)	Log Likelihood	-338.112			
Method:	css-mle	S.D. of innovations	9.733			
Date:	Tue, 12 Nov 2019	AIC	710.224			
Time:	12:00:30	BIC	752.720			
Sample:	01-01-2015	HQIC	727.361			
	- 01-01-2015					
	coef	std err	z	P> z	[0.025	0.975]
const	416.3847	0.829	502.154	0.000	414.760	418.010
ar.L1.rotate	-0.6793	0.110	-6.155	0.000	-0.896	-0.463
ar.L2.rotate	-0.5458	0.137	-3.991	0.000	-0.814	-0.278
ar.L3.rotate	-0.7958	0.145	-5.475	0.000	-1.081	-0.511
ar.L4.rotate	-0.1155	0.162	-0.711	0.480	-0.434	0.203
ar.L5.rotate	-0.1227	0.162	-0.756	0.452	-0.441	0.195
ar.L6.rotate	-0.3495	0.165	-2.113	0.038	-0.674	-0.025
ar.L7.rotate	-0.0358	0.163	-0.220	0.827	-0.355	0.283
ar.L8.rotate	0.0654	0.161	0.407	0.685	-0.250	0.381
ar.L9.rotate	-0.1056	0.162	-0.653	0.516	-0.423	0.212
ar.L10.rotate	-0.1590	0.142	-1.116	0.260	-0.438	0.120
ar.L11.rotate	-0.2783	0.134	-2.064	0.041	-0.540	-0.017
ar.L12.rotate	-0.3955	0.109	-3.641	0.000	-0.608	-0.183
ma.L1.rotate	0.7749	nan	nan	nan	nan	nan
ma.L2.rotate	0.7749	0.066	11.737	0.000	0.645	0.904
ma.L3.rotate	1.0000	nan	nan	nan	nan	nan

varianza costante;

la curva della densità è gaussiana e centrata sul valore 0;

La Figura 52 mostra le metriche di valutazione del modello, mentre la Figura 53 offre una rappresentazione dei valori di forecast rispetto ai valori reali ottenuti applicando il modello;

**Figura 50 - Rappresentazione del modello ARIMA per la predizione della velocità di rotazione**



```
{'mape': 0.018633171068748407,
'me': -0.5786887355298177,
'mae': 7.827524778504324,
'mpe': -0.0006971024257315008,
'rmse': 10.673822470686806,
'acf1': 0.16854180731944846,
'corr': 0.5140398763115386,
'minmax': 0.018388647653367562}
```

**Figura 51 - Plot degli errori residui**

**Exprivia**



Figura 52 - Metriche di valutazione del modello

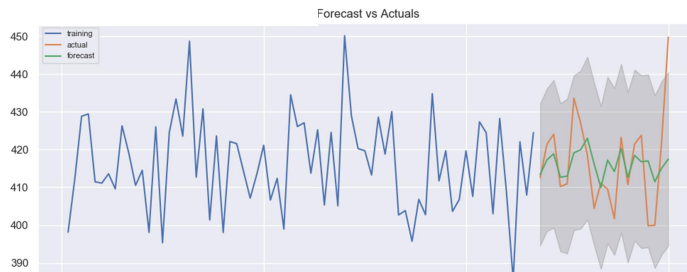


Figura 53 - Forecast vs Actual modello ARIMA predizione velocità di rotazione

### *Real-time Processing*

Per il processo produttivo della fresatrice, sono stati generati i relativi device e i topic sulla piattaforma Digital Future, utilizzando la stessa procedura descritta nel paragrafo 4.1.2 e di seguito si descriviamo i passi per predizione della velocità di rotazione della punta rotante di una fresatrice.

Il **modello ARIMA** generato per la predizione della velocità di rotazione potrà essere utilizzato per predire la velocità di rotazione della punta rotante della fresatrice nell’arco di 1 ora al fine di rilevare eventuali anomalie nel processo produttivo o nel macchinario ad esempio si potrà verificare che la configurazione della velocità di rotazione non è adeguata al target stabilito di un ordine di produzione.

Pertanto si potrà simulare l’arrivo di un ordine da un sistema di gestione ordine in particolare si prevede di recuperare attraverso un servizio Rest dedicato al reperimento dei dati relativi al singolo ordine di produzione provenienti dall’applicazione SAP Fiori Release Orders. La simulazione viene alimentata mediante di «Rilascio Ordine» contenente le informazioni necessarie per l’avvio dell’ordine di produzione (numero di pezzi da produrre, id dell’ordine, lasso temporale entro cui i pezzi dovranno essere prodotti);

Da un punto di vista prettamente architetturale si assume che i realizzazione di moduli che verranno no installati su un Raspberry PI e che avranno il compito di:

- recepire gli ordini provenienti dall’applicazione Release Orders,
- inviare gli ordini di produzione alla macchina fresatrice





- recepire le misure provenienti dai sensori via protocollo MQTTs e invio delle misure provenienti dai sensori alla piattaforma

In particolare attraverso un servizio MQTTOrderIngestion, si occupare di simulare er il reperimento dei dati provenienti dai sensori in tempo reale e la normalizzazione dei dati nel formato JSON conforme al data model e si occuperà anche di inviare i dati normalizzati alla piattaforma DF attraverso il mdoluo **MQTTOrderPublisher** .

### *Alerting*

Una volta creato il modello addestrato, testato ,salvato sulla Piattaforma, in fase di erogazione arrivano i dati in Piattaforma un job esegue l’elaborazione su menzionata e successivamente il job alerting invia le serie temporali rielaborate e i dati di predizione al servizio di sottoscrizione specifico topic kafka MQTTSubscriber a su volta un servizio invia i dati al datastore INFLUXDB. Il datastore è un componente dell’ applicazione SMART Factory (Fabbrica Intelligente) deputato alla memorizzazione dei dati delle serie temporali provenienti dai sensori, e quelli prodotti dai modelli predittivi; Mentre attraverso il servizio Alert si invia i dati al sistema di Data Monitoring (Grafana) deputato alla visualizzazione delle serie temporali . In sostanza nel caso specifico ogni 15 minuti, viene costruito il modello ARIMA utilizzando la parametrizzazione stabilita a priori durante la fase di analisi, e «predetta» la velocità di rotazione della macchina fresatrice da li a 1 ora; Il componente Alert Controller del Data Monitoring allo scadere di un lasso temporale stabilito (5s), controlla se la velocità di rotazione prevista allo scadere dell’ora non è adeguata alle richieste di produzioni “Alert Controller” invierà una mail di alert ai referenti preposti e un servizio attiva il processo di intervento e quindi ispezione;

#### **4.5.2. Servizi per predire il numero di pezzi prodotti**

La serie temporale utilizzata per l’implementazione di questo scenario è quella ottenuta nella fase di preparazione dei dati e relativa al numero di pezzi prodotti dalla fresatrice (misura tot\_pieces). Come descritto per lo scenario relativo alla predizione della velocità di rotazione anche per questo scenario occorre per prima procedere preparazione de dati e a partire dai dati di telemetria ottenuti dalla prima fase di pre-processing e



raffigurati in Tabella 5 per l’implementazione anche questo scenario in oggetto è stato generato un nuovo dataset contenente i dati relativi a 2 distinte serie temporali con intervalli temporali pari a 10s:

- ✓ **Una serie temporale relativa alla velocità di rotazione** della punta rotante della fresatrice, le cui misure sono state ottenute mediante l’utilizzo di una funzione generatrice di numeri random con distribuzione gaussiana;
- ✓ **Una serie temporale relativa al numero di pezzi prodotti** dalla fresatrice, ottenuta a partire dalla prima serie temporale sopra menzionata definendo delle relazioni tra le misure di rotazione e il numero di pezzi prodotti ad ogni istante  $t$ ;

Successivamente anche per questo caso per creare il modello predittivo ARIMA si prevede la determinazione della stazionarietà della serie, l’eventuale differenziazione della stessa e infine la determinazione dei parametri AR(p) e MA(q).

#### *Creating ARIMA models*

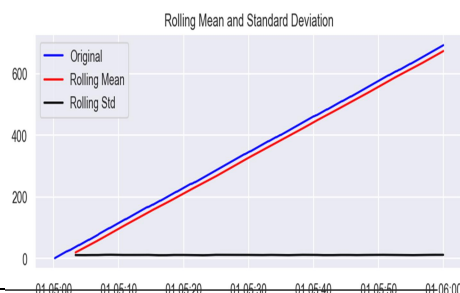
Come ampiamente *descritto* per la creazione di un modello predittivo ARIMA è necessaria la definizione dei parametri del modello ed in particolare che si descrive nei paragrafi successivi:

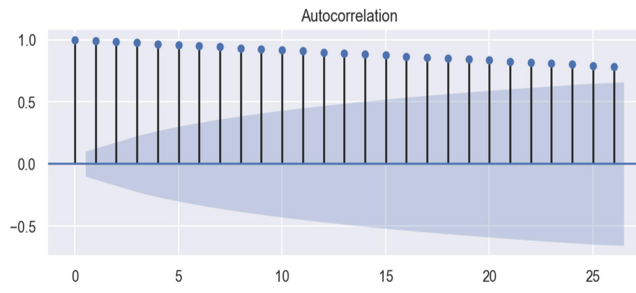
- ✓ **p** – numero di termini autoregressivi;
- ✓ **d** – numero di differenze non stagionali;
- ✓ **q**- numero di termini per la media mobile (mobile-average)

#### **4.5.2.1.1. Determinazione della stazionarietà della serie temporale**

Dalla visualizzazione della serie temporale si evince che l’andamento del numero di pezzi prodotti nel tempo è crescente (**pertanto la media non è costante**); dal test di Dickey-Full il valore di p-value risulta superiore a 0,05: pertanto **la serie non è stazionaria ed è necessario procedere con la differenziazione per renderla stazionaria**

Dalla visualizzazione dei plot di autocorrelazione per la serie originaria, la serie differenziata di 1 e la serie differenziata di 2 si evince che il parametro di differenziazione corretto è 1.

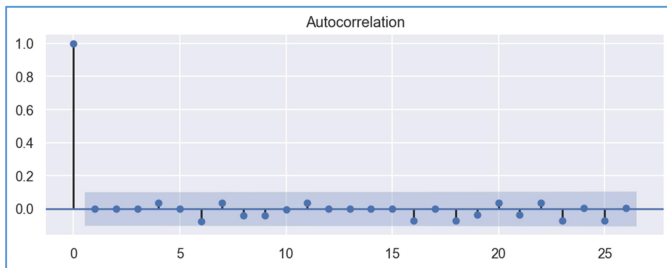




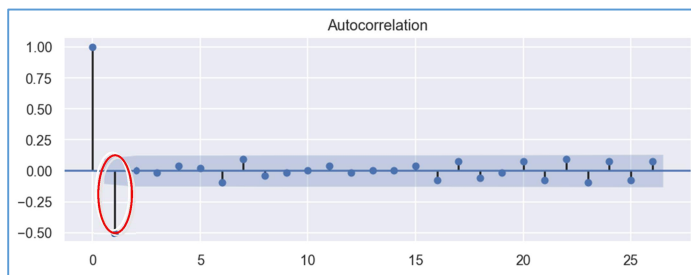
```

Result of Dickey-Fuller Test
Test Statistic      0.597940
p-value            0.987563
#Lags used         0.000000
Number of Observations Used 359.000000
Critical Value (1%) -3.448697
Critical Value (5%) -2.869625
Critical Value (10%) -2.571077
dtype: float64
  
```

Serie originaria: andamento decrescente



Ordine di differenziazione 1: la serie rimane dentro i confini



Ordine di differenziazione 2: la correlazione al lag 1 è negativa, la serie è sovra-differenziata

ARIMA Model Results

---

Dep. Variable:	D.tot.pieces	No. Observations:	359
Model:	ARIMA(0, 1, 0)	Log Likelihood	-44.741
Method:	css	S.D. of innovations	0.274
Date:	Tue, 12 Nov 2019	AIC	93.483
Time:	16:02:38	BIC	101.250
Sample:	01-01-2015	HQIC	96.571
	- 01-01-2015		

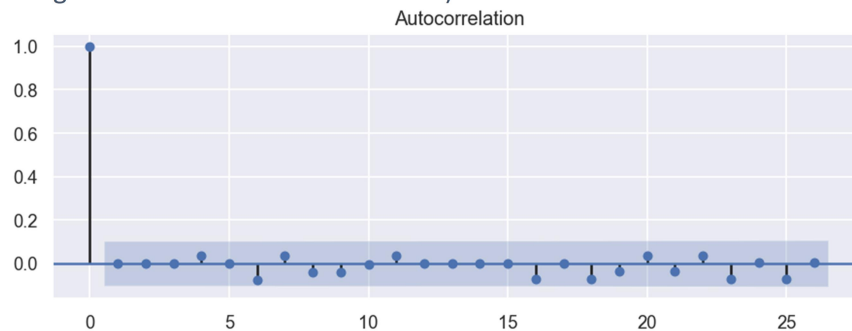
---

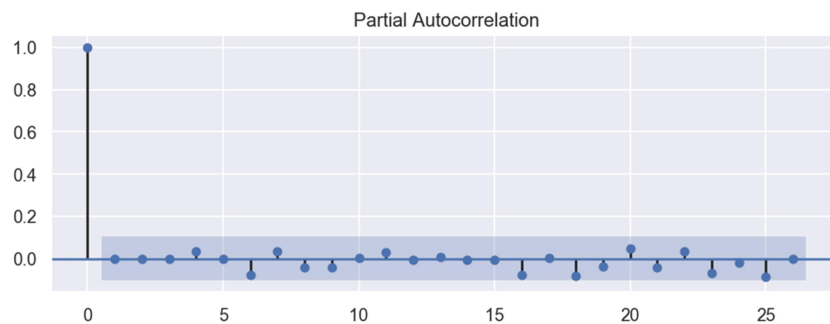
	coef	std err	z	P> z	[0.025	0.975]
const	1.9248	0.014	133.059	0.000	1.896	1.953

---

#### 4.5.2.1.2. Determinazione dei termini AR(p) e MA(q)

Dai plot PACF e ACF della serie temporale differenziata (d=1) e rappresentati nel seguito, si evidenzia che non è necessario aggiungere termini di AR e MA (i valori restano nelle soglie ossia all'interno dell'area blu)





Viene pertanto addestrato un modello ARIMA (0,1,0). Dalla rappresentazione del modello si evince che i valori del P- value nella colonna ‘P>|Z|’ per AR(12) e MA(3) sono altamente significativi in quanto  $\ll 0.05$

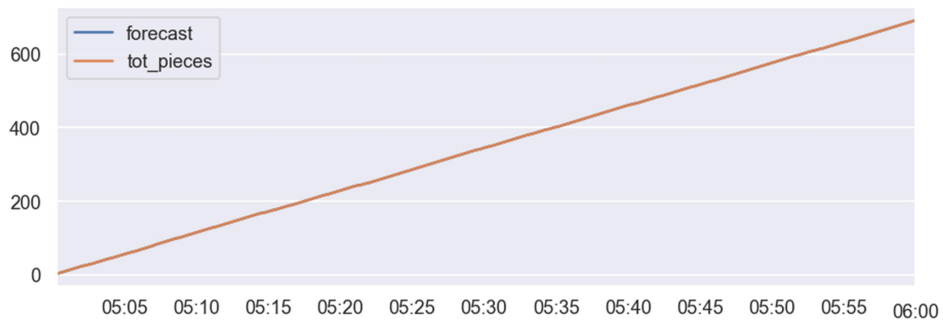


Figura 54 - Forecast vs Actual modello ARIMA predizione numero di pezzi

### Real time Processing

I dati che arrivano in streaming come menzionato nel paragrafo 0 come mostrato attraverso job si recuperano i dati e successivamente con un successivo job subiranno trasformazione al fine definire i parametri di input al modello

- p – numero di termini autoregressivi;
- d – numero di differenze non stagionali;
- q- numero di termini per la media mobile (mobile-average)



successivamente vengono inviati al Data Governance (DataLake) componente della piattaforma Digital Future deputato all’addestramento delle rete neurale tramite le librerie python di Tensorflow<sup>7</sup>, per la rilevazione predittiva dello stato di salute del macchinario e la gestione preventiva di future anomalie e mediante Apache Zeppelin, notebook web-based completamente open-source che offre funzionalità di inserimento ed esplorazione dei dati, visualizzazione, condivisione e collaborazione con Hadoop<sup>8</sup> e Spark per le pre-elaborazione.

### *Alerting*

Il modello ARIMA generato si prevede che di utilizzarlo anche p predire il numero di pezzi prodotti dalla macchina fresatrice nell’arco di 1 ora o meglio al fine di rilevare che il numero di pezzi prodotti non è adeguato al target stabilito nell’ordine di produzione . La simulazione anche per questo scenario verrà alimentata mediante una dashboard di «Rilascio Ordine» contenente le informazioni necessarie per l’avvio dell’ordine di produzione (numero di pezzi da produrre, id dell’ordine, lasso temporale entro cui i pezzi dovranno essere prodotti); I dati verranno inviati in tempo reale alla piattaforma IOT Digital Future e successivamente risultati del predizione saranno immagazzinati nel database per serie temporali e visualizzati su dashboard dal sistema Data Monitoring .Il processo in genarle ogni 15 minuti, si invocherà il modello ARIMA il quale utilizzando la parametrizzazione stabilita a priori durante la fase di analisi, predirà » il numero di pezzi prodotti dalla macchina fresatrice da li a 1 ora; Ed Il componente **Alert Controller** del Data Monitoring allo scadere di un lasso temporale stabilito (5s), controlla se vi sono anomalie associate al processo ossia verifica numero finale di pezzi prodotti previsti allo scadere dell’ora è inferiore a quello previsto. In caso di anomalia si il modulo **Alert**

---

<sup>7</sup> TensorFlow, <https://www.tensorflow.org/>

<sup>8</sup> Apache Hadoop, <https://hadoop.apache.org/>



Controller invierà una mail di alert ai referenti. Di seguito in Figura 55 si riporta il flusso completo dall’invio ordine alla visualizzazione dell’alert.

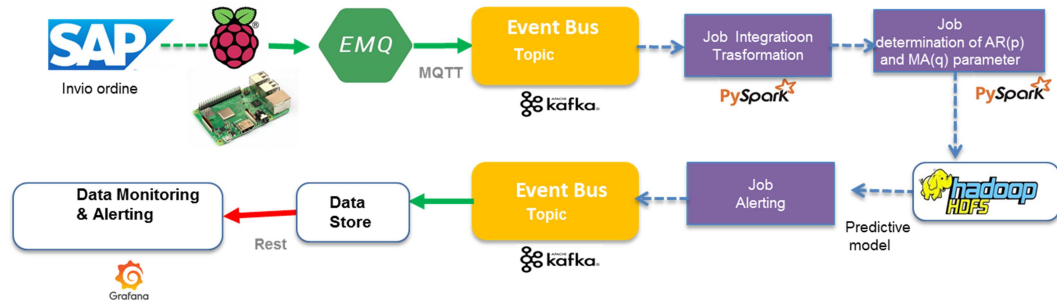


Figura 55 : pipeline real time



## 5. Bibliografia

- [1] Exprivia, « OR 4-TR 4.2 Metodi e tecniche per la Fabbrica Intelligente».
- [2] Exprivia, «Allegato TR 4.3 -Analisi servizi di Piattaforma per Product Quality Management».
- [3] Y. S. X. L. Z. YAN XU 1, «A Digital-Twin-Assisted Fault Diagnosis Using,» IEEE, i, 22 Febbraio 2019..
- [4] L. Y. X. G. L. & Z. Jinjiang Wang, «Digital Twin for rotating machinery fault diagnosis in smart manufacturing,» *International Journal of Production Research*, n. 57(12):3920-3934, Luglio 2019.
- [5] G. Electric, «GE Digital Twin,» 2018. [Online]. Available: [https://www.ge.com/digital/sites/default/files/download\\_assets/Digital-Twin-for-the-digital-power-plant-.pdf](https://www.ge.com/digital/sites/default/files/download_assets/Digital-Twin-for-the-digital-power-plant-.pdf).
- [6] N. M. a. N. M. Zheng Liu, «The role of data fusion in predictive maintenance using digital twin,» in *AIP Conference Proceedings 1949, 020023 (2018)*, 20 Aprile 2018.
- [7] M. Zubani, «Manutenzione Predittiva: tipi di manutenzioni a confronto e vantaggi,» Giugno 5 2018. [Online]. Available: <https://www.toolsforsmartminds.com/it/insight/blog/178-manutenzione-predittiva-tipi-di-manutenzioni-a-confronto-e-vantaggi>.
- [8] «Anomaly Detection, A Key Task for AI and Machine Learning, Explained,» [Online]. Available: <https://www.kdnuggets.com/2019/10/anomaly-detection-explained.html>.
- [9] Azure, «Studio del modello di soluzione Cortana Intelligence per la manutenzione predittiva nel settore aerospaziale e in altri campi,» [Online]. Available: Studio del modello di soluzione Cortana Intelligence per la manutenzione predittiva nel



settore aerospaziale e in altri campi” (<https://docs.microsoft.com/it-it/azure/machine-learning/team-data-science-process/cortana-analytics-playbook-predictive-mainten>).

- [10] «Accuratezza,» [Online]. Available: <https://it.wikipedia.org/wiki/Accuratezza>.
- [11] «Precisione,» [Online]. Available: <https://it.wikipedia.org/wiki/Precisione>.
- [12] «F1\_score,» [Online]. Available: [https://it.wikipedia.org/wiki/F1\\_score](https://it.wikipedia.org/wiki/F1_score).
- [13] «ANALISI PREDITTIVA: PERCHÈ CI INTERESSA E CHI LA USA?,» marzo 2019. [Online]. Available: <https://www.dataskills.it/analisi-predittiva-dei-dati/#gref>.
- [14] J. K. Gill, «Anomaly Detection with Time Series Forecasting,» Novembre 2018. [Online]. Available: <https://www.xenonstack.com/blog/time-series-deep-learning/>.
- [15] V. Flovik, «How to use machine learning for anomaly detection and condition monitoring,» 31 Dicembre 2018. [Online]. Available: <https://towardsdatascience.com/how-to-use-machine-learning-for-anomaly-detection-and-condition-monitoring-6742f82900d7>.
- [16] H. K. E. a. A. R. H. M. Elattar, Prognostics: a literature review,” *Complex & Intelligent Systems*, vol. 2, vol. 2, no. 2, pp. 125–154, 2018.
- [17] «Spark Streaming Programming Guide,» [Online]. Available: <https://spark.apache.org/docs/latest/streaming-programming-guide.html>.
- [18] Exprivia, OR 5 -TR 5.2 Esecuzione dell'esperimento.
- [19] T. K. a. T. Borovicka, Early failure detection for predictive in ITAT, pp. 123–130, 2016..
- [20] A. K. a. A. Sane, «Machine learning for predictive maintenance of of industrial machines using iot sensor data,» in *Software Engineering and Service Science (ICSESS) 2017 8th IEEE International Conference on*, pp. 87–90, IEEE, 2017.
- [21] B. G. Diego Galar, «Fusion of CMMS Data and CM Data - A Real Need for





Maintenance,» 2 Ottobre 2012. [Online]. Available:  
<https://www.maintworld.com/Applications/Fusion-of-CMMS-Data-and-CM-Data-A-Real-Need-for-Maintenance>.

[22] «<https://github.com/microsoft/AMLWorkshop>,» [Online].