



UNIVERSITÀ POLITECNICA DELLE MARCHE

FACOLTÀ DI INGEGNERIA

CORSO DI LAUREA IN: INGEGNERIA INFORMATICA E DELL'AUTOMAZIONE

DESIGN E SVILUPPO DI UN'INTERFACCIA FUNZIONALE ALLA
VISUALIZZAZIONE DI ALERT PER LA MANUTENZIONE
PREDITTIVA NELL'AMBITO DI INDUSTRY 4.0

DESIGN AND DEVELOPMENT OF A FUNCTIONAL INTERFACE
FOR VIEWING ALERTS FOR PREDICTIVE MAINTENANCE IN
THE CONTEXT OF INDUSTRY 4.0

Studente:
Marco Romanelli

Relatore:
PROF. EMANUELE FRONTONI

Correlatore:
ING. LUCA ROMEO

ANNO ACCADEMICO 2020/2021

Ai miei veri amici.
Alle persone che mi amano
e mi hanno amato davvero.
Non siamo fatti per stare da soli
ma nemmeno con chiunque
Ad Alessia, Andrés, Camilla.

SOMMARIO

ELENCO DELLE TABELLE.....	5
ELENCO DELLE FIGURE	6
ACRONIMI E ABBREVIAZIONI	7
CAPITOLO 1 INTRODUZIONE	9
1.1 Decision Support System.....	10
1.2 Predictive Maintenance.....	10
1.3 Caso d'uso Benelli.....	13
1.4 Obiettivi.....	13
CAPITOLO 2 RELATED WORK: DECISION SUPPORT SYSTEM FOR PREDICTIVE MAINTENANCE	14
2.1 Analisi dei Big Data.....	14
2.1.1 Latenza.....	14
2.1.2 Acquisizione dei dati.....	15
2.2 Machine Learning for Predictive Maintenance.....	15
CAPITOLO 3 MATERIALI.....	18
3.1 Broker MQTT	18
3.2 Sensori del macchinario.....	22
3.3 Parametri di input (Topic a livello 0).....	24
CAPITOLO 4 METODI.....	28
4.1 Workflow.....	28
4.1.1 Data Collection.....	29
4.1.2 Futures Extraction.....	29
4.1.3 Prediction.....	30
4.1.4 Cloud - Architecture.....	30
4.1.5 Interfaccia GUI.....	30
4.2 Estrazione delle Features: Data Analysis.....	31

4.3 Interfaccia GUI: Data Visualization.....	33
CAPITOLO 5 RISULTATI.....	35
CAPITOLO 6 CONCLUSIONI - FUTURE WORK	38
BIBLIOGRAFIA	4040

ELENCO DELLE TABELLE

Tabella 2-1: Tecniche ML e loro applicazione.....	17
Tabella 3-1: Topics related to the working step analyzer application embedded in the L0-Machine Agent.....	25
Tabella 3-2: Segnali acquisiti da CNC (ID: 1-13) e dall'accelerometro (ID: 14).....	25
Tabella 3-3: L2 CMM topics.....	26
Tabella 4-1: KPI relativi ai sensori.....	32
Tabella 5-1: Risultati computation time.....	35

ELENCO DELLE FIGURE

Figura 3-1: Flow chart of the data layer : jFMX MQTT Namespace.....	21
Figura 3-2: Macchinari usati.....	23
Figura 4-1: Schema della strategia proposta.....	29
Figura 4-2: Flow chart relativo al primo approccio.....	32
Figura 4-3: Flow chart relativo al secondo approccio.....	33
Figura 5-1: Vista dell'interfaccia grafica.....	36

ACRONIMI E ABBREVIAZIONI

DSS	DECISION SUPPORT SYSTEM
PDM	Predictive Maintenance
ML	Machine Learning
RUL	Vita utile rimanente
CPS	Cyberphysical System
KPI	Key Performance Indicator
ANN	Artificial Neural Network
RNN	Reti Neurali Ricorrenti
LSTM	Long Short Term Memory
GRU	Gated Recurrent Unit
SVM	Support Vector Machine
RF	Random Forest
BPNN	Back-Propagation Neural Network
RBFNN	Radial Basis Function Neural Network
ENN	Elman Neural Network
PNN	Probabilistic Neural Network
FNN	Fuzzy Neural Network
WNN	Wavelet Neural Network
BN	Rete Bayesana
DRF	Distributed Random Forest
XGBoost	Extreme Gradient Boosting
GBM	Gradient Boosting Machine

FMS	Sistema di Produzione Flessibile
CMM	Macchina di misure a coordinate
CNC	Controllo Numerico Computerizzato
TNI	Trapezoidal Number Integration

Capitolo 1

INTRODUZIONE

Le precedenti rivoluzioni industriali si sono basate sul miglioramento dei processi di produzione da un punto di vista fisico, meccanico ed energetico. La quarta rivoluzione industriale, chiamata anche Industria 4.0, si concentra principalmente su miglioramenti dal punto di vista digitale per ottenere informazioni su cosa succede sui processi fisici. Questo modello digitale ha come obiettivo fornire informazioni in tempo reale sullo stato dei macchinari consentendo al personale tecnico di affrontare un eventuale guasto o problema, passando così da un ripristino reattivo ad uno pianificato. Questo approccio intelligente spinge le aziende a dover affrontare le sfide proposte da questa nuova fase dell'industria. Nello specifico le aziende che si stanno spingendo in questa direzione si stanno trasformando letteralmente in fabbriche intelligenti. Tali fabbriche sono dotate di sensori, integrati in maniera opportuna, generando e archiviando una grande quantità di dati. I dati ottenuti attraverso sensori e macchinari sono alla base di questa rivoluzione industriale. Un'applicazione in questo ambito riguarda l'utilizzo dell'analisi dei dati al fine di progettare un sistema di supporto alle decisioni (Decision Support System, DSS) , consentendo un'ottimizzazione della produzione e un ripristino dei guasti più rapido. [8 ,9] Nel mondo dell'industria 4.0, lo sviluppo di queste applicazioni ha diversi scopi: possiamo trovare il Decision Support System, il quale funge da assistenza nello sviluppo progettuale, dove attraverso algoritmi di machine learning propone ad esempio una nuova versione del prodotto basandosi sulle caratteristiche finali del prodotto [10]. Un altro compito che può adempire l'utilizzo di questa mole di dati è la manutenzione predittiva (PdM) che consente di prevedere il degrado del prodotto e l'ottimizzazione autonoma del prodotto [11 , 12]. Un uso frequente del Machine Learning in questo campo permette lo sviluppo di applicazioni per la previsione dei guasti e la stima della

durata della vita utile del macchinario [13 , 14] , attraverso l'addestramento dell'algoritmo, sfruttando la grande quantità di dati a disposizione [15, 16]. In questo scenario, i sistemi di supporto alle decisioni di manutenzione (DSS) potenziati da IoT, Big Data e Machine Learning (ML) assumono un ruolo saliente per garantire la manutenibilità e l'affidabilità delle apparecchiature nelle industrie trasformando grandi set di dati in conoscenza e intelligenza fruibile [3, 4, 5]. Nei seguenti capitoli vengono esposti i concetti di DSS e PdM.

1.1 *Decision Support System*

Un sistema di supporto delle decisioni (DSS) è un sistema informativo che aiuta l'azienda in attività decisionali che richiedono giudizio, determinazione e una sequenza di azioni.

La letteratura più recente definisce i DSS come “sistemi interattivi che aiutano specialisti nel prendere decisioni in situazioni di incertezza”; in pratica un sistema automatico nel trattamento delle informazioni, con l'obiettivo di migliorare i processi decisionali non completamente strutturati.

L'implementazione del DSS industriale per monitorare lo stato dei processi industriali consente l'ottimizzazione dei programmi di manutenzione, facendo sì che si generino avvisi in tempo reale sui rischi operativi consentendo ai produttori di ridurre i costi di servizio, migliorandone la produttività.

Da un punto di vista strutturale, i DSS necessitano di tre componenti essenziali: a) un sistema per l'elaborazione dei dati con un'opportuna interfaccia utente-sistema; b) un database; c) un modello di analisi. [41]. Il DSS assiste la gestione di medio e alto livello di un'organizzazione analizzando enormi volumi di dati non strutturati e accumulando informazioni per la risoluzione di problemi e aiutare nel processo decisionale.

1.2 *Predictive Maintenance*

La Predictive Maintenance (PdM), chiamata anche “monitoraggio in linea” o “manutenzione basata sul rischio”, è oggetto di svariati documenti di ricerca recenti. Dai primi approcci basati sull'ispezione visiva, questa strategia si è evoluta, sfruttando tecniche avanzate di elaborazione del segnale incentrate

sul riconoscimento di modelli, apprendimento automatico, reti neurali e logica fuzzy.

La PdM è una strategia adottata quando si affrontano problemi di manutenzione. Il monitoraggio dei motori elettrici e delle apparecchiature utilizzate nell'industria, attraverso sensori integrati, consente di evitare sostituzioni non necessarie e di individuare la causa del principale guasto. Dunque, risulta essere un vantaggio influente sull'ottimizzazione dei programmi di manutenzione, fornendo in tempo reale avvisi sui rischi operativi e permettendo ai produttori di ridurre i costi di servizio e di massimizzare i tempi di attività, eliminando eventuali rallentamenti non previsti sul ciclo di produzione.

Le applicazioni della manutenzione predittiva si basano sui dati raccolti da sensori e algoritmi di analisi [3 ,4]. Nelle industrie di processo, i motori a induzione costituiscono circa il 70% di tutti i carichi elettrici azionati e proprio a questo proposito c'è stato molto interesse sui modi per diagnosticare meglio la condizione di benessere di questi motori [5]; il problema di manutenzione più comune è causato dal guasto del cuscinetto, una delle principali cause di danni al motore.

La manutenzione predittiva si basa essenzialmente su due aspetti:

- Risparmio energetico e ottimizzazione dell'efficienza energetica
- Riduzione dei tempi di fermo non previsti.

Alla base della manutenzione predittiva vi è una stima di parametri futuri che caratterizzano il sistema in questione (tipicamente una macchina, un impianto o un processo produttivo), attraverso modelli matematici al fine di identificare in anticipo le anomalie o potenziali guasti [6 ,7].

Lo schema di base è il seguente:

1. Misura di grandezze fisiche in tempo reale.
2. Stima dei parametri misurabili o non misurabili, al tempo $t + dt$ (misurazione all'istante 't' e stima del valore all'istante 'dt').
3. Identificazione dello stato dell'impianto ritenuto anomalo o difettoso.
4. Pianificazione delle attività preventive e correttive prima che il sistema raggiunga la condizione critica.

I dati di monitoraggio che vengono esposti negli approcci ML allo stato dell'arte per il PdM sono di diversa natura:

- le vibrazioni delle macchine, che possono essere utili per segnalare il deterioramento dei cuscinetti o deformazioni di parti meccaniche;
- la temperatura del motore e la corrente che assorbe, possono essere parametri che indicano possibili malfunzionamenti meccanici, i quali non consentono il corretto funzionamento;
- la misurazione attraverso opportuni sensori della composizione dell'olio lubrificante, nel quale la misurazione di particelle presenti in esso indica la degradazione delle parti a contatto con lo sfregamento.

Attraverso tali misurazioni, estrapolando i dati ed elaborandoli si può prevedere la vita utile rimanente dei componenti (RUL).

Il PdM, tuttavia, in ambienti di produzione implica il superamento di numerose sfide irrisolte, includendo la necessità di aggregare dati eterogenei raccolti da sensori di ambienti operativi dinamici e la necessità di informazioni sensibili mirate ad uno specifico dominio industriale [1, 2]. Una delle sfide alla base di questo approccio sta nella difficoltà di recuperare i dati di fallimento etichettati, non che il set di dati annotati fondamentale per l'addestramento del modello Machine Learning (ML) con il fine di identificare con precisione problemi e generare allarmi in maniera tempestiva.

L'annotazione dell'usura dei componenti, tuttavia, non è molto semplice: le misurazioni vengono tracciate attraverso diversi cicli di produzione e in base alle condizioni operative. Pertanto, una delle questioni ancora aperte include la difficoltà di ottenere tali dati, etichettarli in maniera qualitativamente corretta, e interpretarli. Una gran parte dei dati a disposizione non ha annotazioni, presenta valori mancanti ed è strutturata in maniera inadeguata. Tale problema comporta una forte necessità di avere a disposizione una quantità di set di dati annotati relativi ai vari guasti. Gli approcci proposti allo stato dell'arte vengono affrontati successivamente nel capitolo 2.

L'automazione, la stabilità e la robustezza della procedura per l'annotazione è una sfida tutt'ora ancora aperta e rappresenta il problema chiave per la progettazione di un approccio Machine Learning o Deep Learning (DL) per la soluzione accurata di un'attività PdM.

1.3 *Caso d'uso aziendale*

L'attività PdM proposta in questo lavoro ha avuto origine dalla richiesta di una società specifica (Benelli Armi Spa): l'approccio sperimentale rispecchia, dunque, un caso d'uso aziendale reale che include misurazioni avanzate e macchinari di elaborazione per la lavorazione di pezzi di fucili, attuando lavorazioni come fresatura, tornatura, alesatura ecc.

La richiesta dell'azienda è quella di creare e integrare un DSS basato sulla PdM per avere degli alert in tempo reale di eventuali anomalie o guasti.

1.4 *Obiettivi*

Attraverso l'industria 4.0 vengono introdotte delle modifiche rispetto all'approccio originario dell'automazione industriale; nello specifico le varie tecnologie Internet of Things (IoT) (ed anche Cyberphysical System, CPS) hanno fatto sì che venisse introdotto il concetto di automazione cognitiva e di conseguenza la produzione intelligente, il quale ci porta allo scopo di implementare prodotti e servizi intelligenti.

Gli obiettivi proposti in questo lavoro si basano sull'estrapolazione di Key Performance Indicator (KPI) attraverso approcci di feature extraction, i quali vengono usati come input per il modello Machine Learning (ML). Tali KPI vengono estratti dai dati acquisiti a livello inferiore e superiore attraverso il broker MQTT. Il secondo obiettivo si basa sullo sviluppo di un'interfaccia grafica che mostri degli alert generati attraverso i KPI. Tali alert consentono di monitorare in maniera tempestiva le procedure di manutenzione in modo tale da supportare il manutentore/operatore.

Capitolo 2

RELATED WORK

Lo stato dell'arte relativo al campo generale della manutenzione predittiva, secondo la letteratura proposta in [2], mette in evidenza diverse soluzioni per affrontare le sfide legate ad esso; nello specifico vengono divise in (i) problemi di integrazione, (ii) analisi dei Big Data, (iii) modelli di Machine Learning.

Il primo elemento fa riferimento al campo più generico della PdM, il quale ha come obiettivo quello di ridurre i costi della manutenzione e il miglioramento dell'efficienza produttiva prevedendo la manutenzione necessaria. L'approccio più comune è quello di integrare architetture che consentono la raccolta dei dati dai sensori, l'elaborazione di essi ed in fine la creazione di modelli di apprendimento automatico. Nello specifico, i relativi campi da affrontare per la risoluzione di queste sfide sono proposti nei successivi sotto capitoli.

2.1 *Analisi dei Big Data*

Una delle principali sfide correlata alla PdM è proprio il BigData, in quanto quest'ultimo deve garantire alcuni parametri fondamentali come latenza [17,18,19], larghezza di banda [17,18], scalabilità [20,18,21] e l'acquisizione dei dati [20,22,23].

2.1.1 *Latenza*

Per garantire una buona latenza vengono proposte soluzioni come l'edge computing, che consente di accorciare le distanze tra client ed esecutore di servizi e velocizzare tutte le operazioni di interconnessione [17,18], o l'utilizzo di un framework basato sul concetto di calcolo distribuito, dove il sistema gestisce il sovraccarico distribuendo l'elaborazione [19].

2.1.2 *Acquisizione dei dati*

Un'altra sfida molto importante è l'acquisizione dei dati, essendo necessario ottenere dati di qualità e la loro interpretazione. Spesso parte dei dati acquisiti ha valori mancanti, non è ben strutturata o sono assenti annotazioni relative a questi. In [20] vengono proposte tecniche per pre-elaborare i dati facendo sì che questi siano utilizzabili per la PdM. Oppure, come presente in [22,23], vengono proposti approcci attraverso framework per il monitoraggio e l'acquisizione di dati nei legacy machine o sfruttando i dati messi a disposizione del registro eventi forniti dai sistemi diagnostici.

2.2 **Machine Learning per Manutenzione Predittiva**

Tra le altre sfide elencate, stabilire una base per la creazione di modelli di machine learning è una di queste. Come per il problema relativo ai Big Data, è necessario anche in questo contesto una buona scalabilità e larghezza di banda e, a tal proposito, viene proposto l'apprendimento dei modelli ai margini della rete (edge of the networks) [18,21].

Viene definito 'run to fail' (R2F) l'approccio che mette in evidenza la tendenza di uno stato normale ad uno stato di fallimento [24,25]; in questo scenario viene proposta una strategia top-down che si basa in primo luogo sulla comprensione del funzionamento della macchina e, dopodiché affronta il problema, addestrando modelli con set di dati relativi agli errori con annotazioni [26].

Un altro punto da dover tenere in considerazione è l'eterogeneità del set dei dati, in quanto sia la mancanza che l'eccesso di eterogeneità rende più difficile l'addestramento dei modelli ML [27,28]. Da qui deriva che una delle cause di questa eterogeneità dipenda dal fatto che gli impianti di produzioni sono ambienti molto dinamici; dunque, il processo di una determinata macchina può cambiare dinamicamente [24].

Approcci alternativi per affrontare tali aspetti comprendono l'addestramento di più modelli di ML [25], l'uso di dati che provengono da macchinari che hanno lavorato in

condizioni simili [29], sfruttare data mining per generare un set di dati in un determinato contesto [30] e l'utilizzo di algoritmi di Deep Learning.

Un ultimo aspetto, non per importanza, è proprio il costo computazionale per l'addestramento di questi modelli di ML o DL [24, 28].

La letteratura insegna che sono presenti diverse tecniche di ML in base all'applicazione da sviluppare; in particolare sono evidenziate varie tecniche in funzione delle diverse fasi della manutenzione predittiva, ad esempio tecniche per la diagnosi, la prognosi o la stima della vita utile.

Una delle tecniche più utilizzate è l'utilizzo delle reti neurali artificiali (ANN): in [31] viene proposto questo approccio per il rilevamento e la previsione dei guasti, per eseguire la correzione degli errori a prescindere della macchina o del tipo di processo preso in considerazione.

È stato presentato anche un approccio attraverso Reti Neurali Ricorrenti (RNN) le quali sono un tipo di ANN che incorpora la memoria. Nello specifico viene proposto un modello di RNN Long Short Term Memory (LSTM) per identificare un possibile malfunzionamento: è previsto un modello per la classificazione che conta il numero di cicli di vita del motore ed un modello per la regressione per prevedere il numero rimanente di cicli [32]. La base di questo approccio è la capacità di considerare dati storici per prevedere il comportamento futuro. Viene usato LSTM anche insieme ad una seconda tecnica chiamata Gated Recurrent Unit (GRU) [33] e queste tecniche vengono confrontate anche con approcci simili come, ad esempio, Support Vector Machine (SVM) o Random Forest (RF) [17,34].

Nello specifico sono stati presi in studio anche algoritmi basati su reti neurali come, Back-Propagation Neural Network (BPNN), Radial Basis Function Neural Network (RBFNN), Elman Neural Network (ENN), Probabilistic Neural Network (PNN), Fuzzy Neural Network (FNN) e Wavelet Neural Network (WNN).

Vengono proposti anche modelli grafici utilizzando una rete bayesiana (BN): la scelta di questa tecnica è giustificata dal fatto che essa è ottima per le condizioni di incertezza ed è capace di suddividere problemi complessi in problemi più semplici usando predizioni condizionate (conditional probabilities).

Applicando algoritmi ad albero per la previsione di un guasto in [22] vengono proposti un insieme di algoritmi (Distributed Random Forest (DRF), Extreme Gradient Boosting (XGBoost), Gradient Boosting Machine (GBM)) di classificazione per prevedere la vita utile residua (RUL) delle macchine per la lavorazione del legno.

Di seguito una tabella di sintesi delle applicazioni basate su ML classificate in base alla tecnica ML e alla relativa applicazione.

Tabella 2-1: Tecniche ML e loro applicazione

PAPER	METODO ML	APPLICAZIONE
Li et al. [31]	ANN	Previsione dei guasti
Crespo et al. [19]	ANN	Rilevamento anomalie
Rivas et al. [32]	LSTM	Stima RUL
Cachada et al. [33]	LSTM/GRU	Previsione dei guasti
Schmidt et al. [34]	FFNN	Previsione delle serie temporali
Zhou et al. [17]	MLP	Classificazione dei guasti
Liu et al. [18]	ANN	Rilevamento anomalie
Liu et al. [35]	BPNN, ENN, RBFNN, PNN, FNN e WNN	Fusione dei dati
Cheng et al. [36]	ANN	Previsione delle condizioni
Daniyan et al. [37]	ANN	Stima RUL
Schmidt et al. [34]	kNN, FFNN, DT, RF, NB	Tempo di fallimento della classificazione delle classi
Zhou et al. [17]	kNN, FT, SVM	Classificazione dei guasti
Carbery et al. [38]	BN	Diagnosi e previsione dei guasti
Ansari et al. [39]	DBN	Previsione degli eventi di errore
Chukwuekwe, et al. [40]	ARMA	Previsione dei guasti
Adhikari et al. [25]	SVM	Rilevamento anomalie
Adhikari et al. [25]	ARIMA	RUL
Adhikari et al. [25]	DT, SVM, NB, RF	Classificazione dei guasti
Calabrese et al. [22]	GBM, DRF, XGBosot	RUL

Capitolo 3

MATERIALI

Di seguito sono esposti i mezzi usati per l'estrapolazione dei dati necessari, come vengono scambiati tra server e client, salvataggio degli stessi e richiesta del client che ha la necessità di usufruirne.

In particolare, il broker utilizzato per lo scambio/richiesta dei dati, i sensori presenti sui macchinari per raccogliere i dati e i relativi dati necessari per lo scopo del PdM.

3.1 Broker MQTT

MQTT è un protocollo di telemetria leggero, proveniente dal mondo M2M e ora ampiamente applicato nell'IoT. Il termine Internet of Things è stato utilizzato per la prima volta da Kevin Ashton nel 2009 per l'interconnessione di dispositivi fisici su Internet. L'idea di base è molto semplice: i dispositivi fisici possono scambiare dati tra loro o essere controllati da altri. Esempi di tali dispositivi potrebbero essere un frigorifero, un'auto, un edificio o praticamente qualsiasi altro dispositivo elettronico. Uno dei casi d'uso più comuni è la raccolta, la trasmissione, il consolidamento e la visualizzazione dei dati dei sensori. I risultati potrebbero essere una dashboard web con i valori aggregati o un allarme, quando viene superata una soglia. Offre un sistema di messaggistica publish-subscribe, che consente ai clienti di pubblicare messaggi senza preoccuparsi della loro eventuale destinazione. Pubblica/Sottoscrivi è basato sugli eventi e consente di inviare messaggi ai client. Il punto di comunicazione centrale è il broker MQTT, incaricato di inviare tutti i messaggi tra i mittenti e i legittimi destinatari. Ogni client che pubblica un messaggio al broker include un argomento nel messaggio. L'argomento sono le informazioni di instradamento per il broker. Ogni client che

desidera ricevere messaggi si iscrive a un determinato argomento e il broker consegna tutti i messaggi con l'argomento corrispondente al client.

Un broker MQTT famoso e leggero che è compatibile con tutti i sistemi Unix, Raspberry Pi compreso (vi permetterà ad esempio di usare Homebridge con aggiornamenti istantanei), è Mosquitto. Tale broker è il cardine della struttura utilizzata che consente il rilevamento e la comunicazione, si occupa di spedire tutti i messaggi tra mittenti e legittimi destinatari. L'implementazione del protocollo MQTT in JFMX è stata eseguita utilizzando la libreria Java Paho.

Questa architettura consente soluzioni di alto livello di scalabilità senza dipendenze tra i produttori di dati e i consumatori dei relativi dati. Questo è possibile in quanto ogni client che pubblica un messaggio al broker include un argomento nel messaggio, il quale rappresenta 'l'instradamento' per il broker. In maniera analoga ogni client che desidera ricevere messaggi sottoscrive un determinato argomento e i messaggi relativi all'argomento vengono consegnati dal broker al client. Dunque, i client non devono conoscersi, comunicano attraverso l'argomento.

Infatti, il concetto sul quale si basa MQTT per inviare messaggi sono gli argomenti. Un argomento è una stringa che può avere più livelli di gerarchia. Ad esempio, un argomento per l'invio dei dati di stato di mc2 di un FMS è il seguente: JFMX / L1 / fms / UNIT / mc1 / STATUS. Da un lato il cliente può iscriversi all'argomento esatto o, dall'altro, utilizzare un carattere jolly. Il carattere jolly (+) consente valori arbitrari per una gerarchia mentre il carattere jolly multilivello (#) consente di iscriversi a più di un livello (ad esempio l'intero sottoalbero).

L'organizzazione tematica MQTT consente di garantire la Quality of Service (QoS). Il protocollo MQTT gestisce la ritrasmissione e garantisce la consegna del messaggio, indipendentemente dall'affidabilità del trasporto sottostante. Inoltre, il cliente può scegliere il livello di QoS in base alla sua affidabilità di rete e alla logica dell'applicazione:

- QoS 0 - al massimo una volta: garantisce la massima resa possibile. Un messaggio non verrà riconosciuto dal destinatario o memorizzato e riconsegnato

dal mittente. Questo è spesso chiamato "spara e dimentica" e fornisce la stessa garanzia del protocollo TCP sottostante.

- QoS 1 - almeno una volta: è garantito che un messaggio verrà consegnato almeno una volta al destinatario. Il mittente memorizzerà il messaggio fino a quando non riceverà un riconoscimento sotto forma di messaggio di comando PUBACK dal destinatario. L'associazione di PUBLISH e PUBACK viene eseguita confrontando l'identificatore del pacchetto in ciascun pacchetto. Se il PUBACK non viene ricevuto in un ragionevole lasso di tempo, il mittente invierà nuovamente il messaggio PUBBLICA. Se un destinatario riceve un messaggio con QoS 1, può elaborarlo immediatamente, ad esempio inviandolo a tutti i clienti in abbonamento in caso di broker e quindi rispondendo con PUBACK.

- QoS 2 - esattamente una volta: garantisce che ogni messaggio venga ricevuto una sola volta dalla controparte. È il livello di servizio più sicuro e anche più lento. La garanzia è fornita da due flussi di andata e ritorno tra mittente e destinatario. Se un destinatario riceve un QoS 2 PUBLISH, elaborerà il messaggio di pubblicazione, di conseguenza, e lo riconoscerà al mittente con un messaggio PUBREC. Il destinatario memorizzerà un riferimento all'identificatore del pacchetto fino a quando non avrà inviato PUBCOMP. Questa procedura è importante per evitare di elaborare il messaggio una seconda volta. Quando il mittente riceve il PUBREC può tranquillamente eliminare la pubblicazione iniziale, perché sa che la controparte ha ricevuto con successo il messaggio. Memorizzerà PUBREC e risponderà con un PUBREL. Dopo che il destinatario ha ricevuto PUBREL, può scartare ogni stato memorizzato e rispondere con un PUBCOMP. Lo stesso vale quando il mittente riceve il PUBCOMP. Quando il flusso è completato, entrambe le parti possono essere sicure che il messaggio è stato consegnato e anche il mittente lo sa. Ogni volta che un pacchetto si perde per strada, il mittente è responsabile dell'invio dell'ultimo messaggio dopo un ragionevole lasso di tempo. Ciò è vero quando il mittente è un client MQTT e anche quando un broker MQTT invia un messaggio. Il destinatario ha la responsabilità di rispondere di conseguenza a ciascun messaggio di comando.

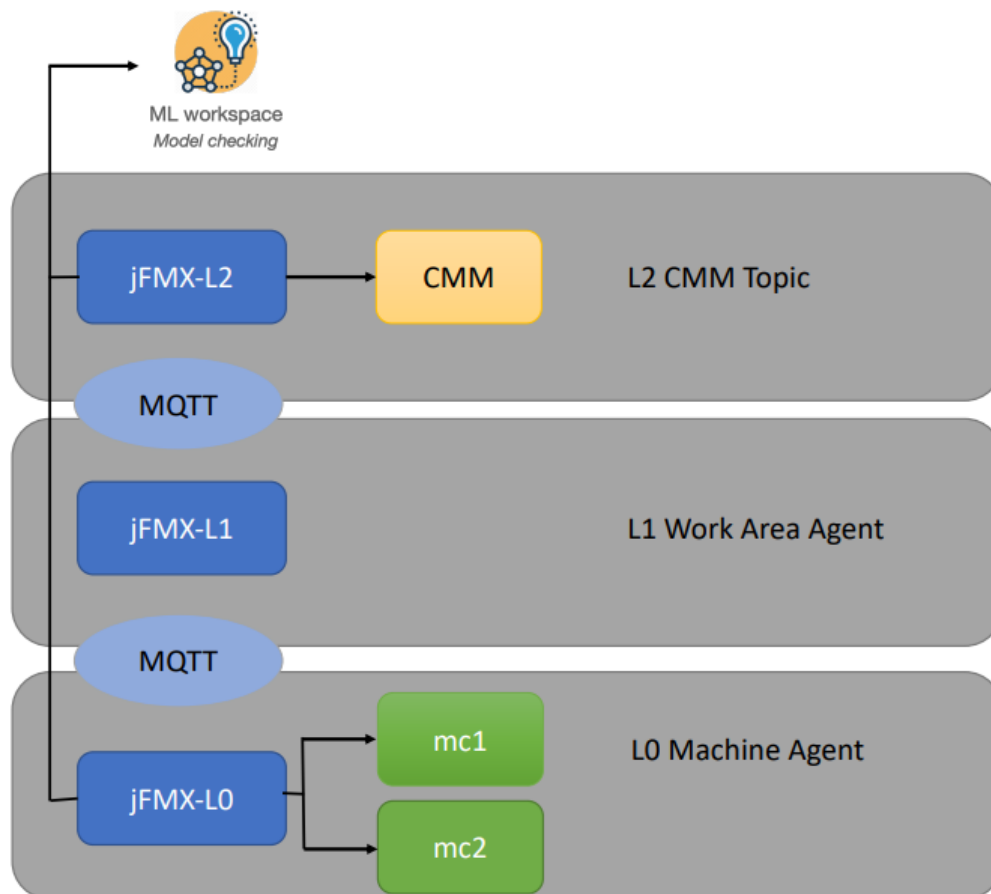


Figure 3: Flow chart of the data layer: jFMX MQTT Namespace

In base a questo criterio, la nostra applicazione IoT in esecuzione su un gateway IoT può essere visualizzata in termini di risorse che possiede e gestisce, nonché degli eventi non richiesti che segnala:

- `account_name`: identifica un gruppo di dispositivi e utenti. Può essere visto come partizione dello spazio dei nomi dell'argomento MQTT. Ad esempio, è possibile definire elenchi di controllo di accesso in modo che agli utenti venga concesso l'accesso solo agli argomenti figlio di un determinato `account_name`.
- `client_id`: identifica un singolo dispositivo gateway all'interno di un `account` (in genere l'indirizzo MAC dell'interfaccia di rete primaria di un gateway). `Client_id` si associa all'identificatore `client` (ID client) come definito nelle specifiche MQTT.

- `app_id`: identificatore di stringa univoco per l'applicazione (ad esempio, "L0" per argomenti mc, "L2" 285 per argomenti CMM).
- `resource_id`: identifica una o più risorse possedute e gestite da una particolare applicazione. La gestione delle risorse (ad es. Sensori, attuatori, file locali o opzioni di configurazione) include elencarli, leggere il valore più recente o aggiornarli a un nuovo valore. Un `resource_id` è un argomento gerarchico, dove, ad esempio, "fms / mc1 / mandrino / temp" può identificare un sensore di temperatura e "fms / sh / Y / pos" un sensore di posizione.

3.2 Sensori del macchinario

I dati delle lavorazioni dei macchinari sono stati acquisiti da diversi centri di lavoro (mc, nello specifico centri di lavoro a 5 assi mcm clock). Sull' mc a 5 assi, l'utensile da tagli si sposta su 3 assi lineari (X, Y, Z) e ruota su due assi (A, B) in tal modo da potersi avvicinare al pezzo da qualsiasi angolazione e direzione. In tal maniera l' mc può svolgere diverse funzioni multitasking, come la fresatura, tornitura, rettifica, alesatura ecc.



Machine center (mcm clock 5-axis machining center)



Hexagon 3D coordinate measuring machine

Figura 3-2: Macchinari usati.

In particolare, tutti gli mc sono predisposti per poter essere configurati con un singolo pallet, scambiatore di pallet, sistemi multi-pallet o integrati in un sistema di produzione flessibile (FMS).

I dati di monitoraggio delle condizioni vengono acquisiti da un sistema robotico di caricamento dei pezzi per la macchina di misura a coordinate (CMM). La CMM può essere utilizzata in maniera semplice anche da operatori con una minima formazione, come per il sistema di caricamento delle parti

automatico, rendendo il sistema più economico, aumentando la produttività della CMM e massimizza la capacità operativa.

3.3 Parametri di input (topic a livello 0)

Tutti gli argomenti pubblicati da LO Machine Agent eseguito da LO-Gateway, sono relativi ai parametri di elaborazione di mc con relativo app_id, per il nostro lavoro PdM vengono presi come riferimento gli argomenti relativi all'applicazione Working Step Analyzer relativi ai diversi mc (mc1, mc2) (Vedi tabella 3-1).

I parametri fondamentali per lo sviluppo di questo task PdM necessari per l'elaborazione sono presenti nel campo dataObj (controllo numerico computerizzato [CNC] e accelerometro).

Nella seguente tabella sono presenti i dati letti ed estrapolati attraverso il sistema flightrecorder dalle due sorgenti di dati accennati poco prima (CNC e accelerometro). I dati relativi alla sorgente CNC sono acquisiti ogni 24ms, mentre i relativi segnali dell'accelerometro sono campionati con un periodo di 1ms e viene e viene estrapolato il valore RMS su 100 campioni; dunque alla fine si ha un campione ogni 100 ms.

Tabella 3-1: Topics related to the working step analyzer application embedded in the L0-Machine Agent.

Field	Type	Description
Ts	Date	start date of the episode according to the clock of the mc
l2mc	String	mc code
l2wa	String	workArea code
ordNo	String	order number
ptType	String	identifier of the part type
opNo	String	operation number
dType	String	identifier of the workingstep message type ("wsEv": in case of sensor coming from mc, "accTrace": in case of sensor coming from Accelerometer)
slot	Int	sub section as indicated by the part program
pt	String	identifier of the part machined
tl	String	identifier of the tool formatted as <tooltype>/<tool serial number>
life	Int	life of the tool at the beginning of the step
sensor	String	name of the sensor
unit	String	measure unit for the specific sensor
sampling	Int	milliseconds sampling interval
dataObj	Object	complex object containing two elements: n[int] segment of the acquisition, data: Array [Double] data acquired
mc	String	machine name
wa	String	workArea name

Tabella 3-2: Segnali acquisiti da CNC (ID: 1-13) e dall'accelerometro (ID: 14)

ID	Sensor	Descrizione
1	BLOCK_NUM	Numero di blocco
2	SP_SPEED	Velocità rotazione mandrino
3	SP_POW	Potenza assorbita del mandrino
4	X_AXIS_CURR	Corrente assorbita asse X
5	Y_AXIS_CURR	Corrente assorbita asse Y
6	Z_AXIS_CURR	Corrente assorbita asse Z
7	B_AXIS_CURR	Corrente assorbita asse B

8	A_AXIS_CURR	Corrente assorbita asse A
9	X_AXIS_POS	Posizione asse X
10	Y_AXIS_POS	Posizione asse Y
11	Z_AXIS_POS	Posizione asse Z
12	B_AXIS_POS	Posizione asse B
13	A_AXIS_POS	Posizione asse A
14	velMOD_RMS	Modulo velocità vibratoria 3 direzioni

I relativi argomenti di L2 CMM e L0 Machine Center sono sincronizzati in base alle diverse mansioni che può svolgere il macchinario, attraverso diversi parametri fondamentali:

- ts: time stamp
- tl: definito anche come tooltype, ovvero il numero di serie dell'utensile
- pt: rappresenta il pezzo che il macchinario ha utilizzato
- frindex: rappresenta il tipo di lavorazione svolta dal macchinario.

I relativi argomenti L2 CMM una volta sincronizzati con gli argomenti L0 Machine Center sono salvati in un database SQL.

Gli argomenti L2 CMM rappresentano i dati di monitoraggio delle condizioni acquisiti dai macchinari, questi dati opportunamente elaborati rispecchiano la qualità della lavorazione avvenuta in termini di deviazione misurata rispetto ai valori ottenuti in condizioni ottimali. L'eventuale situazione di allarme si attiva nel momento in cui la deviazione misurata supera un certo livello di tolleranza.

Tabella 3-3: L2 CMM topics

Field	Type	Description
ts	Date	start date of the episode according to the clock of the mc
pt	String	identifier of the part machined
tl	String	identifier of the tool formatted as <tooltype>/<tool serial number>

FRINDEX	Int	type of processing
measured	Double	measured value
deviation	Double	measured deviation
tolerance	Double	admitted tolerance

CAPITOLO 4

METODI

In questo capitolo vengono esposti i vari pilastri per lo sviluppo di un DSS. Nello specifico vengono presentati i relativi studi precedenti ed è spiegato nel dettaglio come è stata sviluppata la parte delle estrazioni delle Features e la relativa interfaccia grafica (capitolo 4.2 e 4.3).

4.1 *Workflow*

La strategia proposta in tale lavoro consente di automatizzare completamente la raccolta dati, l'estrapolazione dei relativi KPI e il calcolo di previsione passando per i KPI salienti che rappresentano l'input del modello predittivo, correlati alla qualità dell'elaborazione e dunque alla riduzione delle perdite di produttività e ai problemi di manutenzione.

L'approccio è basato su cinque pilastri:

- a. Data Collection
- b. Feature Extraction
- c. Prediction
- d. Cloud-Architecture
- e. Data Analytics

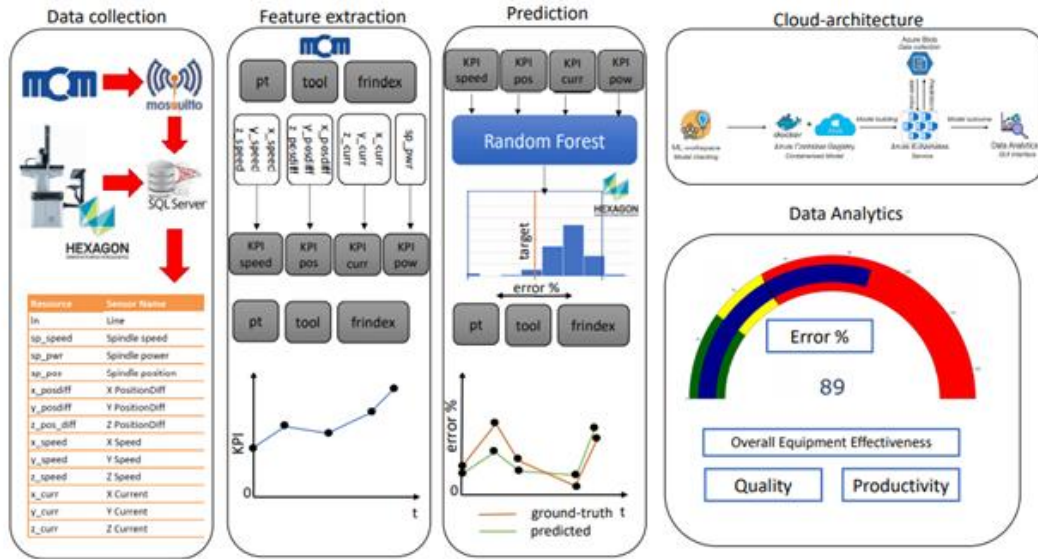


Figura 4-1: Schema della strategia proposta

4.1.1 Data Collection

La raccolta dei dati avviene per mezzo di una tecnologia di rilevamento IoT basata sul broker Message Queuing Telemetry Transport (MQTT). Il principio di base è l'utilizzo di argomenti che raccolgono i parametri di elaborazione (velocità, posizione, corrente) e tali dati vengono annotati quantitativamente nel set di dati da una macchina di misura a coordinate 3D. Sono presenti i parametri dei centri di lavoro a livello 0, mentre i dati di stato e condizionamento raccolti dalla macchina Hexagon rappresentano l'argomento a livello 2. Tutti i dati vengono sincronizzati tra loro e raccolti e archiviati in un database SQL.

4.1.2 Features Extraction

Per ogni parametro di lavorazione di ogni ciclo produttivo, in base all'utensile utilizzato, al pezzo di lavorazione e al tipo di lavorazione, viene estratto un Key Performance Indicator (KPI).

L'estrapolazione delle caratteristiche viene eseguito tramite integrazione trapezoidale (Trapezoidal Numerical Integration).

Tra gli obiettivi trattati in questo lavoro troviamo lo sviluppo dello script per l'estrapolazione dei KPI, il quale verrà approfondito nel capitolo successivo 4.2.

4.1.3 *Prediction*

I requisiti fondamentali per la scelta del modello ML sono le prestazioni predittive, l'interpretabilità e l'accuratezza predittiva. Per tali ragioni è stato scelto, per la risoluzione del compito di regressione, il modello Random Forest (RF).

Tale modello consiste in un insieme di alberi di regressione generati da vettori casuali indipendenti distribuiti in modo identico.

L'obiettivo del modello ML è quello di stimare i dati di stato e condizionamento utilizzando come parametri di input i KPI precedentemente estrapolati.

4.1.4 *Cloud – Architecture*

L'architettura del cloud si basa sulla logica del contenitore per impacchettare il modello ML e le sue dipendenze, consentendo di eseguire il tutto in modo affidabile su diversi ambienti. Un container è un'unità eseguibile di software in cui il codice dell'applicazione è pacchettizzato insieme alle relative librerie e dipendenze, in modo che possa essere eseguito ovunque, su desktop, locale e su cloud.

Allo stato dell'arte viene sfruttato il servizio Azure ML per fornire l'ambiente cloud per la distribuzione e l'aggiornamento del modello; nel dettaglio è sfruttata l'archiviazione BLOB di Azure per l'archiviazione dei pesi del modello ML e i relativi risultati.

Attraverso questo ambiente Docker si può garantire un isolamento hardware e software, flessibilità e interdipendenza tra i dispositivi IoT, la raccolta dati, le fasi di estrazione dei KPI e la previsione.

4.1.5 *Interfaccia GUI*

Tra i vari obiettivi raggiunti in questo lavoro c'è lo sviluppo di un'interfaccia GUI per la visualizzazione di errore prevista in base ai parametri fondamentali (tl, pt, frindex). Nello specifico tale interfaccia è finalizzata a fornire in modo tempestivo al manutentore la visualizzazione grafica del superamento di una certa soglia di tolleranza da parte della percentuale di errore.

Lo sviluppo di tale interfaccia è affrontato nel capitolo successivo (4.3)

4.2 Estrazione delle Features

L'estrazione dei parametri delle performance avviene attraverso il calcolo matematico dell'integrazione trapezoidale per calcolare i KPI per ogni parametro di lavorazione durante ogni ciclo produttivo.

Lo script per l'estrapolazione dei KPI è sviluppato in Python attraverso due librerie molto usate per i calcoli scientifici: nello specifico sono state usate Pandas e NumPy.

Pandas è fondamentale per la creazione di array multidimensionali, fornendo un oggetto tabella 2D chiamato Dataframe. Attraverso questa libreria è stato possibile interagire con i dati caricati attraverso un file csv. I dati presenti in questo file vengono estrapolati ed inseriti in un Dataframe.

I dati relativi ai parametri fondamentali per il calcolo dei KPI sono presenti nel campo 'dataObj' nel quale sono presenti i dati relativi al controllo numerico computerizzato e relativi all'accelerometro. Attraverso metodi forniti da tale libreria i dati sono stati 'ripuliti' eliminando quelli superflui e raggruppandoli in maniera consona, in base alla tripletta composta dal pezzo lavorato 'pt', dall'utensile 'tl' e dal tipo di lavorazione 'frindex'. Per ogni ciclo di lavorazione relativo al parametro 'ts' è stata applicata una funzione Lambda per il calcolo dell'integrale trapezoidale per mezzo della libreria NumPy.

La strategia di estrapolazione dei KPI si basa sull'analisi dell'area geometrica e la stima dell'area trapezoidale, frequentemente utilizzata per le misurazioni di anomalie.

I relativi KPI vengono calcolati normalizzando temporalmente la stima dell'area trapezoidale attraverso la seguente formula:

$$\begin{aligned} KPI &= \frac{1}{t_d - t_1} \int_{t_1}^{t_d} x(t) dt \\ &= \frac{1}{t_d - t_1} \sum_{i=1}^d \int_{t_{i-1}}^{t_i} x(t) dt \\ &\approx \frac{1}{2} \frac{1}{t_d - t_1} \sum_{i=1}^d (t_i - t_{i-1}) [x_i - x_{i-1}] \end{aligned}$$

I KPI relativi alla posizione e alla corrente sono stati calcolati estraendo un KPI globale calcolando la distanza euclidea degli assi X,Y,Z. Nella seguente tabella sono rappresentati i KPI estrapolati dai relativi sensori.

SENSORI (CNC)	KPI
SP_SPEED	Velocità
SP_POW	Potenza
X, Y, Z_AXIS_POS	Posizione
X, Y, Z_AXIS_CURR	Corrente

Tabella 4-1 : KPI relativi ai sensori

Lo sviluppo dello script per l'estrazione delle features ha avuto diverse sperimentazioni, per quanto riguarda il primo approccio proposto tale script è stato sviluppato su una struttura basata su diversi cicli for annidati. Di seguito è mostrato il flow chart del primo approccio.

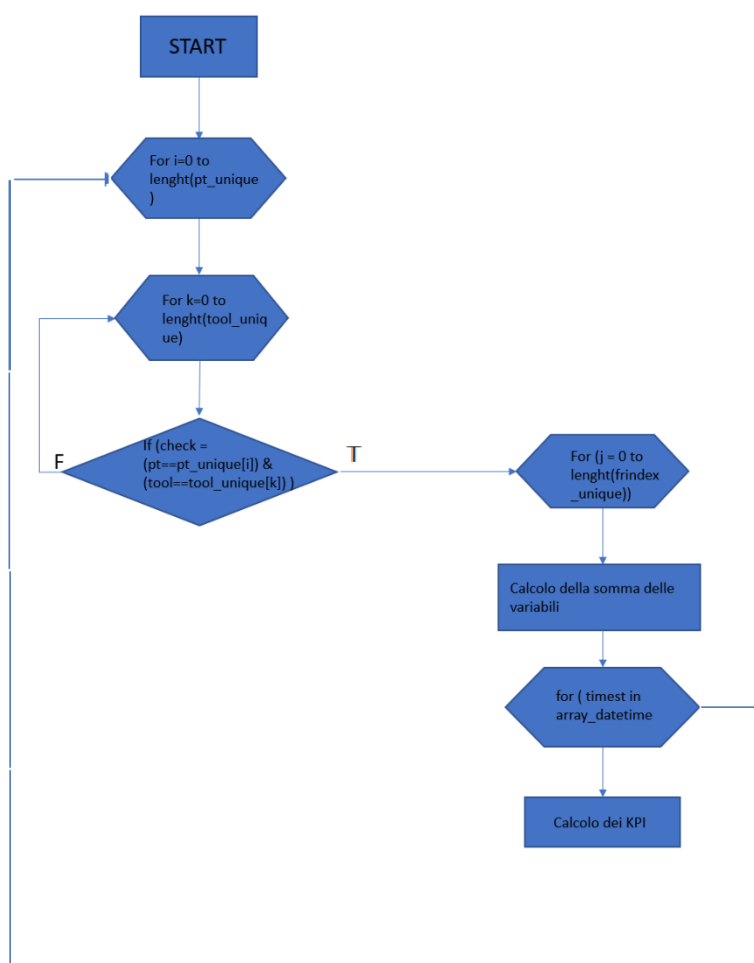


Figura 4-2: Flowchart relativo al primo approccio usato.

Mentre attraverso un secondo approccio, lo script è stato reso più pulito e da un punto di vista computazionale più leggero. Di seguito il relativo flow chart.

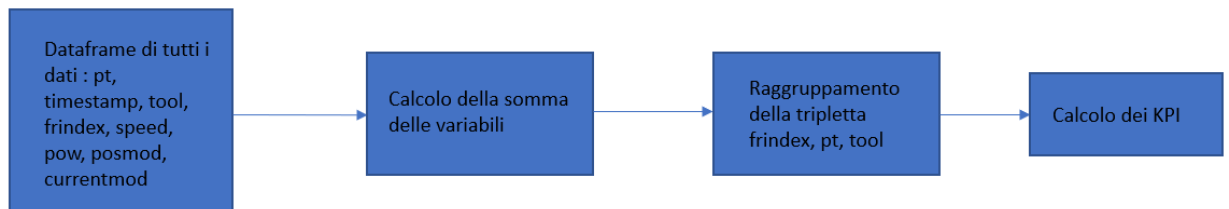


Figura 4-3: Flowchart relativo al secondo approccio usato.

4.3 *Interfaccia GUI*

Lo sviluppo dell'interfaccia grafica si basa sull'utilizzo di due librerie 'Plotly' e 'Dash'.

La prima libreria e le sue derivate sono state utilizzate per mettere a grafico i KPI calcolati dai vari parametri fondamentali.

La seconda libreria consente di astrarre le tecnologie e i protocolli per creare una Web app completa per la visualizzazione interattiva dei dati attraverso un'interfaccia utente personalizzata.

Le app Dash sono composte da due parti, la prima parte è il "Layout" e descrive come appare l'applicazione, mentre la seconda descrive l'interattività dell'applicazione.

Nella prima parte sono presenti tre componenti principali, ovvero 'dcc.Upload' che consente di poter caricare il file csv sulla pagina web, il secondo componente è 'dcc.Dropdown' il quale è un menù a tendina che consente all'operatore di poter scegliere il 'pt' e i relativi KPI da visualizzare, ed in fine sono presenti quattro componenti 'dcc.Graph' relativi ai grafici dei quattro KPI che vengono visualizzati. La seconda parte di codice attraverso i tag '@app.callback' è possibile creare l'interazione tra le varie funzioni presenti nello script. Nello specifico è stata implementata una funzione che ha come input il file csv che viene caricato nella

pagina web, al quale viene fatto il parse dei relativi dati, dove attraverso un menù a tendina vengono mostrati i 'pt' presenti nel file, consentendo all'operatore di avere una scelta su quali KPI mostrare relativi al 'pt' scelto.

Una volta scelto il pt preso in esame, dopo un breve lasso di tempo per calcolare i relativi KPI vengono esposti sotto forma di grafico i valori minimi, la media e il massimo.

Il primo approccio prevedeva l'utilizzo di quattro '@app.callback' , ognuno per ogni KPI da elaborare e mettere a grafico. Si è notato che questo approccio faceva sì che la pagina web reagiva in un lasso di tempo eccessivo, per tale motivo si è passato ad un approccio con un singolo '@app.callback' per mostrare i quattro KPI.

Attraverso la libreria Dash, una volta avviata l'applicazione, è possibile accedere all'interfaccia collegandosi al link <http://127.0.0.1:8050/>, il quale è un link generato in locale che funge da mezzo per la visualizzazione grafica.

CAPITOLO 5 RISULTATI

I risultati ottenuti in questo lavoro riguardano l'estrazione delle features e l'interfaccia grafica. Per quanto concerne l'estrazione delle features, si sono ottenuti risultati notevoli soprattutto dal punto di vista del computational time, ovvero sul tempo di elaborazione dei vari dati per l'estrazione dei valori dei KPI. Il tempo di elaborazione è un aspetto fondamentale per questo lavoro, in quanto c'è da considerare che il file sul quale elaborare i dati supera diverse Gb, il che significa il dover aver bisogno di programmi ottimizzati da questo punto di vista per consentire tempi di attesa accessibili per poter avere valutazioni sul processo lavorativo in real time. Come spiegato nel capitolo 4 per l'estrazione delle features c'è stato un primo approccio che prevedeva l'utilizzo di quattro cicli for che iteravano le triplette delle variabili 'pt', 'tool, e 'frindex'; attraverso questo approccio il tempo per l'estrazione dei KPI risultava troppo elevato. In particolare, per l'elaborazione di 5000 righe del file csv caricato, il calcolo per l'estrazione delle features era pari a 2 minuti e 40 secondi. Diversamente con il secondo approccio che eliminava i vari cicli for, i risultati sperimentali per l'estrapolazione dei KPI sulle prime 5000 righe risulta molto inferiore, si è arrivati ad avere tempi di attesa pari a 7,8 s.

Approccio	Numero righe analizzate	Data primo valore	Data ultimo valore	Tempo di elaborazione
<i>Primo approccio con 4 cicli for</i>	5000 righe	28/11/2020 13:53:04	03/12/2020 01:10:40	2 minuti e 40 secondi
<i>Secondo approccio senza cicli for</i>	5000 righe	28/11/2020 13:53:04	03/12/2020 01:10:40	7,8 secondi

Tabella 5-1 : Risultati computation time.

Per quanto riguarda l'interfaccia grafica, è presente un riquadro che consente di caricare il file csv del data collection; un menù a tendina che consente al manutentore di selezionare il 'pt' sul quale estrapolare i KPI, ed infine nella sezione sottostante vengono visualizzati i KPI (Speed, Pow, Posmod, Currentmod) relativi al 'pt' selezionato. Nell'immagine successiva è esposto un esempio di l'interfaccia grafica precedentemente commentata.



Figura 5-1: Vista dell'interfaccia grafica.

Lo sviluppo dell'interfaccia analitica dei dati è una parte fondamentale del lavoro, in quanto per mezzo di essa è stato possibile creare un supporto concreto per il manutentore/operatore. Attraverso di essa è possibile fornire un'indicazione tempestiva che mostri il valore medio, massimo e minimo per uno specifico pt.

Questa interfaccia GUI può potenziare, da un punto di vista qualitativo, il processo di lavorazione, consentendo all'operatore di prevedere situazioni di anomalia (le quali hanno valenza di allarme in quanto è una rappresentazione di un errore di lavorazione significativo) e per mezzo della visualizzazione grafica dei diversi parametri fondamentali poter localizzare la fonte dell'errore.

CAPITOLO 6: CONCLUSIONI E FUTURE WORK

In una valutazione riassuntiva dell'intero lavoro è possibile concludere che tale compito PdM è efficace per un approccio DSS. I pilastri portanti dello sviluppo di questo lavoro si basano sulla raccolta dei dati, nell'estrazione delle caratteristiche fondamentali, nella previsione, nell'archiviazione nel cloud e nella rappresentazione grafica dei risultati. Attraverso la collaborazione tra università e azienda l'approccio proposto presenta un buon compromesso per lo sviluppo di un DSS applicabile in un caso d'uso aziendale reale.

Per mezzo di questo lavoro è stato fatto un passo in avanti per lo sviluppo di un'applicazione funzionante e automatizzata per il supporto del ciclo produttivo dell'azienda. Attraverso gli studi precedenti è stato possibile integrare il compito PdM in un'interfaccia grafica che supporti in maniera diretta e tempestiva il manutentore. Tale implementazione rappresenta il vero vantaggio per l'ottimizzazione della manutenzione dando la possibilità di rendere più efficace la produttività e riducendo i costi di fermo macchina.

Tuttavia, l'implementazione di questo DSS proposto, avrà una totale efficacia con i lavori futuri che consentiranno di integrare nel miglior modo opportuno un sistema completamente automatizzato. Il che sarà reso possibile attraverso sistemi proposti da AWS. Per tali lavori futuri è stato pensato l'utilizzo dei servizi Bucket S3 di AWS che consente di poter conservare e aggiornare gli input e i pesi del modello ML. Il servizio AWS citato mette a disposizione anche 'Object Lambda', ovvero una funzione Lambda dove è possibile aggiungere del proprio codice per modificare ed elaborare i dati, il che consentirebbe il processo di automazione delle estrazione delle features, eliminando i dati grezzi e archiviando solamente i dati fondamentali per il modello ML e per la visualizzazione grafica di essi.

Una valida alternativa proposta da AWS è IoT SiteWise, tale servizio è stato creato proprio per questo tipo di esigenze. Tale supporto ha come obiettivo quello di semplificare tali processi industriali, mettendo a disposizione storage per la raccolta dei dati e l'uso del broker MQTT o attraverso API; modellare le risorse, creare applicazioni web completamente gestite per analizzare i dati in apparecchiature industriali. I lavori successivi consentiranno di apportare dei benefici concreti all'azienda per mezzo del nostro approccio sull'intero processo produttivo.

BIBLIOGRAFIA

- [1] M. Compare, P. Baraldi, E. Zio, Challenges to iot-enabled predictive maintenance for industry 4.0, *IEEE Internet of Things Journal* 7 (5) (2019) 4585–4597.
- [2] J. Dalzochio, R. Kunst, E. Pignaton, A. Binotto, S. Sanyal, J. Favilla, J. Barbosa, Machine learning and reasoning for predictive maintenance in industry 4.0: Current status and challenges, *Computers in Industry* 123 (2020) 103298. doi:<https://doi.org/10.1016/j.compind.2020.103298>. 560 URL <http://www.sciencedirect.com/science/article/pii/S0166361520305327>
- [3] S.-j. Wu, N. Gebraeel, M. A. Lawley, and Y. Yih, “A neural network integrated decision support system for condition-based optimal predic[1]tive maintenance policy,” *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 37, no. 2, pp. 226–236, 2007.
- [4] E. Frontoni, R. Pollini, P. Russo, P. Zingaretti, and G. Cerri, “Hdomo: Smart sensor integration for an active and independent longevity of the elderly,” *Sensors*, vol. 17, no. 11, p. 2610, 2017.
- [5] B. Lu, D. B. Durocher, and P. Stemper, “Predictive maintenance techniques,” *IEEE Industry Applications Magazine*, vol. 15, no. 6, 2009.
- [6] R. K. Mobley, *An introduction to predictive maintenance*. Butterworth-Heinemann, 2002.
- [7] A. Grall, L. Dieulle, C. Bérenguer, and M. Roussignol, “Continuous-time predictive-maintenance scheduling for a deteriorating system,” *IEEE transactions on reliability*, vol. 51, no. 2, pp. 141–150, 2002.
- [8] A. Cachada, J. Barbosa, P. Leitño, C.A. Grcaldcs, L. Deusdado, J. Costa, C. Teixeira, J. Teixeira, A.H. Moreira, P.M. Moreira, et al. “Maintenance 4.0: intelligent and predictive maintenance system architecture” 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), vol. 1 (2018), pp. 139-146
- [9] M.I. Ali, P. Patel, J.G. Breslin “Middleware for real-time event detection and predictive analytics in smart manufacturing” 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS) (2019), pp. 370-376
- [10] L. Romeo, J. Loncarski, M. Paolanti, G. Bocchini, A. Mancini, E. Frontoni “Machine learning-based design support system for the prediction of heterogeneous machine parameters in industry 4.0” *Expert Syst. Appl.*, 140 (2020), p. 112869
- [11] P. O’Donovan, C. Gallagher, K. Leahy, D.T. O’Sullivan “A comparison of fog and cloud computing cyber-physical interfaces for industry 4.0 real-time embedded machine learning engineering applications” *Comput. Ind.*, 110 (2019), pp. 12-35 <http://www.sciencedirect.com/science/article/pii/S016636151830366X>
- [12] H. Boyes, B. Hallaq, J. Cunningham, T. Watson “The industrial internet of things (iiot): an analysis framework” *Comput. Ind.*, 101 (2018), pp. 1-12 <http://www.sciencedirect.com/science/article/pii/S0166361517307285>
- [13] P. ADHIKARI, H.G. RAO, D.-I.M. BUDERATH “Machine Learning Based Data Driven Diagnostics & Prognostics Framework for Aircraft Predictive Maintenance” (2018)

- [14] C. Zhou, C.-K. Tham “Graphel: a graph-based ensemble learning method for distributed diagnostics and prognostics in the industrial internet of things” 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS), IEEE (2018), pp. 903-909
- [15] B. Schmidt, L. Wang “Predictive maintenance of machine tool linear axes: a case from manufacturing industry” *Proc. Manuf.*, 17 (2018), pp. 118-125
- [16] D. O. Chukwuekwue, T. Glesnes, P. Schjøberg, Condition monitoring for predictive maintenance-towards systems prognosis within the industrial internet of things.
- [17] C. Zhou, C.-K. Tham, Graphel: a graph-based ensemble learning method for distributed diagnostics and prognostics in the industrial internet of things, 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS), IEEE (2018), pp. 903-909
- [18] Z. Liu, C. Jin, W. Jin, J. Lee, Z. Zhang, C. Peng, G. Xu, Industrial ai enabled prognostics for high-speed railway systems, 2018 IEEE International Conference on Prognostics and Health Management (ICPHM), IEEE (2018), pp. 1-8
- [19] A. Crespo Márquez, A. de la Fuente Carmona, S. Antomarioni, A process to implement an artificial neural network and association rules techniques to improve asset performance and energy efficiency, *Energies*, 12 (18) (2019), p. 3454
- [20] C. Hegedüs, P. Varga, I. Moldován, The mantis architecture for proactive maintenance, 2018 5th International Conference on Control, Decision and Information Technologies (CoDIT), IEEE (2018), pp. 719-724
- [21] T. Cerquitelli, D. Bowden, A. Marguglio, L. Morabito, C. Napione, S. Panicucci, N. Nikolakis, S. Makris, G. Coppo, S. Andolina, et al. A fog computing approach for predictive maintenance, *International Conference on Advanced Information Systems Engineering*, Springer (2019), pp. 139-147
- [22] M. Calabrese, M. Cimmino, F. Fiume, M. Manfrin, L. Romeo, S. Ceccacci, M. Paolanti, G. Toscano, G. Ciandrini, A. Carrotta, et al. ,Sophia: an event-based iot and machine learning architecture for predictive maintenance in industry 4.0, *Information*, 11 (4) (2020), p. 202
- [23] P. Strauß, M. Schmitz, R. Wöstmann, J. Deuse, Enabling of predictive maintenance in the brownfield through low-cost sensors, an iiot-architecture and machine learning, 2018 IEEE International Conference on Big Data (Big Data), IEEE (2018), pp. 1474-1483
- [24] Y. Xu, Y. Sun, X. Liu, Y. Zheng, A digital-twin-assisted fault diagnosis using deep transfer learning, *IEEE Access*, 7 (2019), pp. 19990-19999
- [25] P. ADHIKARI, H.G. RAO, D.-I.M. BUDERATH, Machine Learning Based Data Driven Diagnostics & Prognostics Framework for Aircraft Predictive Maintenance (2018)
- [26] C.P. Gatica, M. Koester, T. Gaukstern, E. Berlin, M. Meyer, An industrial analytics approach to predictive maintenance for machinery applications, 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE (2016), pp. 1-4
- [27] S. Selcuk, Predictive maintenance, its implementation and latest trends, *Proc. Inst. Mech. Eng. Part B: J. Eng. Manuf.*, 231 (9) (2017), pp. 1670-1679
- [28] D.L. Nuñez, M. Borsato, Ontoprog: an ontology-based model for implementing prognostics health management in mechanical machines, *Adv. Eng. Inform.*, 38 (2018), pp. 746-759
- [29] B. Schmidt, L. Wang, Cloud-enhanced predictive maintenance, *Int. J. Adv. Manuf. Technol.*, 99 (1–4) (2018), pp. 5-13

- [30] R.S. Peres, A.D. Rocha, P. Leitaó, J. Barata, Idarts-towards intelligent data analysis and real-time supervision for industry 4.0, *Comput. Ind.*, 101 (2018), pp. 138-146
- [31] Z. Li, Y. Wang, K.-S. Wang, Intelligent predictive maintenance for fault diagnosis and prognosis in machine centers: Industry 4.0 scenario, *Adv. Manuf.*, 5 (4) (2017), pp. 377-387
- [32] A. Rivas, J.M. Fraile, P. Chamoso, A. González-Briones, I. Sittón, J.M. Corchado, A predictive maintenance model using recurrent neural networks, *International Workshop on Soft Computing Models in Industrial and Environmental Applications*, Springer (2019), pp. 261-270
- [33] A. Cachada, J. Barbosa, P. Leitño, C.A. Gcraldcs, L. Deusdado, J. Costa, C. Teixeira, J. Teixeira, A.H. Moreira, P.M. Moreira, et al., Maintenance 4.0: intelligent and predictive maintenance system architecture, *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1 (2018), pp. 139-146
- [34] B. Schmidt, L. Wang, Predictive maintenance of machine tool linear axes: a case from manufacturing industry, *Proc. Manuf.*, 17 (2018), pp. 118-125
- [35] M. Huang, Z. Liu, Y. Tao, Mechanical fault diagnosis and prediction in iot based on multi-source sensing data fusion, *Simul. Model. Pract. Theory* (2019), p. 101981
- [36] J.C. Cheng, W. Chen, K. Chen, Q. Wang, Data-driven predictive maintenance planning framework for mep components based on bim and iot using machine learning algorithms, *Autom. Constr.*, 112 (2020), p. 103087
- [37] I. Daniyan, K. Mpofu, M. Oyesola, B. Ramatsetse, A. Adeodu, Artificial intelligence for predictive maintenance in the railcar learning factories, *Proc. Manuf.*, 45 (2020), pp. 13-18
- [38] C.M. Carbery, R. Woods, A.H. Marshall, A bayesian network based learning system for modelling faults in large-scale manufacturing, *2018 IEEE International Conference on Industrial Technology (ICIT)* (2018), pp. 1357-1362
- [39] F. Ansari, R. Glawar, W. Sihn, Prescriptive maintenance of cpps by integrating multimodal data with dynamic bayesian networks, *Machine Learning for Cyber Physical Systems*, Springer (2020), pp. 1-8
- [40] D. O. Chukwuekwe, T. Glesnes, P. Schjølborg, Condition monitoring for predictive maintenance-towards systems prognosis within the industrial internet of things.
- [41] *Analisi dello stato dell'arte sui decision support system (DSS)*, 2008, Enrico Massaroni, Bernardo Quattrococchi, Antonio Renzi, Luca Proietti.