POLITECNICO DI TORINO

Facoltà di Ingegneria

Corso di Laurea Magistrale in Ingegneria Informatica
Data Science

Progettazione e sviluppo di una metodologia automatica per l'analisi di serie temporali in un contesto di Industria 4.0

Relatore:
Prof.ssa Tania CERQUITELLI

Candidato:
Andrea BARBERO

Anno Accademico 2019/2020

Oggi non è che un giorno qualunque di tutti i giorni che verranno,
ma ciò che farai in tutti i giorni che verranno dipende da quello che farai oggi.
23 Ottobre 2020

Ringraziamenti

Vorrei ringraziare tutte le persone che mi hanno aiutato e sostenuto durante la mia carriera universitaria e nella stesura della tesi.
Innanzitutto ringrazio la mia famiglia, in particolare i miei genitori e mia sorella, che mi sono stati vicini e mi hanno sempre motivato durante il corso degli studi, dandomi tutto il necessario per proseguire questo percorso al meglio.
Un ringraziamento particolare va alla Professoressa Tania Cerquitelli, che ha accettato il ruolo di relatore per questo progetto di tesi, dandomi sempre ottime idee e consigli per sviluppare questo lavoro. Un grazie va anche a Paolo, che mi ha aiutato nel trovare le metodologie migliori per realizzare l'elaborato e mi ha dato manforte nei momenti di difficoltà.
Inoltre, un ringraziamento va anche ai miei compagni universitari, in particolare ad Antonino, per l'aiuto e il supporto reciproco nella preparazione degli esami e la realizzazione di progetti.
Infine, un ringraziamento va anche a tutti i miei amici, in particolare i miei due coinquilini Chry e Fanta, per avermi fatto passare momenti divertenti e di spensieratezza, rendendo il periodo di permanenza a Torino più piacevole.

Indice

Elenco delle figure

Introduzione

L'Industria è entrata in una nuova era, quella del 4.0.
Lo sviluppo industriale ha portato alla nascita di fabbriche completamente automatizzate e interconnesse e che pare essere la direzione verso cui l'intero sistema produttivo mondiale si sta, velocemente, muovendo.
Industria 4.0 significa cambiamento, in quanto porta con se nuove esigenze in termini di risorse, competenze e lavoro.
Le tecnologie su cui si basa l'Industria 4.0 generano una mole immensa di dati, che ha un valore inestimabile se opportunamente analizzata e interpretata.
L'implementazione di un impianto di produzione moderno richiede una raccolta di dati continua in cui il tempo è l'elemento fondamentale per gestire il flusso di informazioni nel sistema di produzione. Per questo motivo, database contenenti serie temporali sono sempre più diffusi e permettono di mantenere la precisione richiesta per il corretto mantenimento della linea di produzione e per il controllo degli eventi.
Da qui nasce l'idea di sviluppare all'interno di questo progetto di tesi una metodologia d'analisi automatica per le serie temporali. Questa metodologia è stata applicata all'interno di un'estensione del framework ADESCA, focalizzandosi sul rilevamento e analisi di serie temporali in modo automatico, con l'obiettivo di minimizzare lo sforzo dell'esperto di dati e creare un framework utilizzabile da tutti gli utenti, andando incontro al concetto di democratizzazione della scienza dei dati.

La metodologia introdotta è stata implementata in Python, uno dei linguaggi di programmazione più utilizzati per l'applicazione di tecniche di Data Mining e Machine Learning. All'interno dell'estensione, così come nel framework originale, si è fatto uso principalmente delle seguenti librerie:

- Pandas, una libreria per la manipolazione dei dati in formato tabellare o sequenziale, molto utile per caricare e salvare i formati standard per i dati tabulari, come i file csv e xls
- Matplotlib, una libreria molto potente e flessibile per la creazione di grafici di diverso tipo in modo semplice e intuitivo
- Seaborn, una libreria di visualizzazione dati basata su matplotilib. Seaborn fornisce un'interfaccia di alto livello per disegnare grafici statistici ricchi di informazioni
- Sklearn, una libreria di apprendimento automatico che contiene tutti i più importanti algoritmi di classificazione, regressione e clustering

Per costruire l'architettura dell'applicazione web è stato utilizzato Flask, un framework per applicazioni web in Python. La parte grafica delle pagine web è stata realizzata tramite CSS e Javascript. Di particolare interesse è stata HighCharts, una libreria scritta in puro Javascript che ha permesso la creazione di grafici tridimensionali interattivi di alto livello.

Il seguente elaborato si articola in 5 capitoli.
Il primo capitolo presenta lo stato dell'arte dell'Industria 4.0, descrivendo le nuove tecnologie introdotte e il loro sviluppo all'interno delle aziende in Italia e all'estero.
Il secondo capitolo descrive la metodologia introdotta per l'analisi automatizzata delle serie temporali e riporta una spiegazione teorica degli algoritmi utilizzati.
Il terzo capitolo mostra l'architettura del framework implementato e il workflow, cioè il possibile flusso di esecuzione che l'utente può svolgere.
Il quarto capitolo analizza l'applicazione della metodologia introdotta nel framework utilizzando 3 dataset industriali diversi, contenenti serie temporali, mostrando i risultati ottenuti e facendone un confronto finale.
Il quinto capitolo è il capitolo conclusivo e contiene un resoconto sull'elaborato svolto e propone eventuali sviluppi futuri per il miglioramento e ampliamento del progetto.

Capitolo 1

Industria 4.0 - Stato dell'arte

Il termine 'Industry 4.0' indica una tendenza dell'automazione industriale che integra nuove tecnologie produttive per migliorare le condizioni di lavoro, creare nuovi modelli di business e aumentare la produttività e la qualità produttiva degli impianti.
L'espressione 'Industry 4.0' nasce in Germania, durante la fiera di Hannover nel 2011. Nello specifico, la paternità del termine viene attribuita a Henning Kagermann, Wolf-Dieter Lukas e Wolfgang Wahlster che presentarono al governo tedesco una serie di raccomandazioni per la sua implementazione. [1]

L'Industry 4.0 nasce con l'avvento della quarta rivoluzione industriale, con la quale si sta procedendo verso una produzione industriale del tutto automatizzata e interconnessa.
Le nuove tecnologie digitali impattano principalmente su quattro direttrici di sviluppo [1]:

- L'utilizzo dei dati, in particolare la potenza di calcolo e la connettività. Si declina in Big Data, Open Data, Internet of Things (IoT), Machine-to-machine e Cloud Computing.

- L'Analytics, cioè l'analisi dei dati raccolti in modo tale da ottenere un risultato. Il *Machine Learning*, cioè l'adattamento e perfezionamento della resa delle macchine industriali basandosi sui risultati dedotti processando i dati, è anco-

ra oggi poco utilizzata dalle imprese, infatti viene sfruttato solo l'1% dei dati raccolti.

- L'interazione tra uomo e macchina, con lo sviluppo della realtà aumentata.
- Il passaggio dal digitale al reale, che comprende la manifattura additiva, la robotica, le comunicazioni, le interazioni tra le macchine, le nuove tecnologie per immagazzinare i dati e l'utilizzo dell'energia, con l'obiettivo di minimizzare i costi e ottimizzare le prestazioni.

Le nuove tecnologie permettono la raccolta e l'analisi dei Big Data da cui trarre informazioni utili basate su misure. Già nell'ultimo decennio del secolo scorso, venne affermato che il governo dei fenomeni dipende dalla loro misurazione e comprensione, basandosi sullo slogan di W. Edwards Deming "If You Can't Measure It, You Can't Manage It". [2]

I principali strumenti dell'Industria 4.0 possono essere raggruppati in [3]:

- Middleware: software di connessione che mette in comunicazione servizi e ambienti di sviluppo di applicazioni distribuite, permettendone l'interazione. Oggi è la base tecnologica di tutte le integrazioni delle applicazioni aziendali, garantisce una migliore archiviazione, gestione e ripristino dei dati e migliora lo sviluppo delle app.
- Open Technologies: permettono di ricostruire il modello digitale tridimensionale di ogni oggetto reale e sono oggi alla base di Industria 4.0, rivoluzionando la produzione di molte aziende. Includono gli scanner 3D, i quali possono essere robotizzati e inseriti nelle linee di produzione per un controllo automatizzato.
- Automation: riduzione significativa delle tempistiche e le spese operative. È il prossimo grande salto della produttività: le nuove tecnologie intervengono per automatizzare la raccolta dei dati della produzione e creare report, analisi, grafici e trasferire i dati tra i diversi sistemi.

1.1 Smart Factory e gestione dei dati

Nell'Industria 4.0 è basilare il concetto di Smart Factory, cioè rendere l'azienda intelligente tramite l'ingresso dei Cyber-Phycal System (CPS) in cui l'uomo è ancora necessario, ma non più fondamentale. Ogni azienda dovrà fornirsi di macchine robotizzate e intelligenti, in grado di prendere decisioni autonome interagendo con l'ambiente.
La Smart Factory è un sistema di produzione flessibile e efficiente in grado di soddisfare le esigenze del mercato moderno e raggiungere l'integrazione tra i vari partner industriali e non industriali. [4]
I CPS sono sistemi di entità computazionali in stretta connessione con il mondo fisico circostante. I componenti fisici e software sono profondamente intrecciati, perché nonostante operino su diverse scale spaziali e temporali, interagiscono tra loro in diversi modi a seconda del contesto. [5]
L'idea principale alla base dei CPS è l'integrazione delle capacità di autogestione.
La capacità di interagire e la capacità di comunicazione consentono ai CPS di migliorare la produzione, effettuare cicli di feedback e fornire supporto ottimale alle persone nelle loro decisioni. Utilizzando i sensori, i CPS sono in grado di ricevere dati fisici e convertirli in segnali digitali, condividendo queste informazioni alla rete digitale a cui sono connessi. [4]
Le apparecchiature di produzione si trasformeranno in Cyber-Physical Production Systems (CPPS), cioè macchinari potenziati dal software, dotati di una propria potenza computazionale, con una vasta gamma di sensori e attuatori integrati. I CPPS conoscono il loro stato, la loro capacità e le loro diverse opzioni di configurazione e saranno quindi in grado di prendere decisioni autonomamente. La produzione di massa verrà rimpiazzata dalla personalizzazione di massa, in cui ogni prodotto finale presenterà caratteristiche uniche, in base alle scelte fatte dal consumatore.
La combinazione di CPS e CPPS provocherà dei cambiamenti nella produzione manifatturiera e di controllo, e si sposterà verso sistemi decentralizzati, con obiettivi contrastanti, in cui solo una gestione efficiente delle operazioni porterà a risultati ottimali. [6]

Monitoraggio, configurazione, acquisizione di dati e controllo sono di assoluta importanza all'interno delle Smart Factory. Per questo è necessario analizzare e avere una visione d'insieme della situazione, azioni rese possibili solo tramite particolari tecnologie in grado di mettere in comunicazione costante i macchinari e i software gestionali.
La gestione dei dati è effettuata tramite un'accurata attività di Big Data Management. I dati rappresentano oggi il principale patrimonio informativo e imprenditoriale, in quanto su di essi si basa la possibilità di prendere decisioni e di migliorare il processo di gestione strategica dell'economia che caratterizza l'attività quotidiana di ogni azienda. Gli attuali sistemi offrono capacità di esplorazione e di analisi dei Big Data basata su processi stocastici e modelli dinamici, teorie dei sistemi a dinamica non lineare, nuove idee dalle scienze cognitive, capacità di elaborazione dei dati con ampiezze di storage e potenze di calcolo virtualmente illimitate. [2]
Le aziende aumentano la produttività, sfruttando le risorse in modo più efficiente per ridurre l'impatto ambientale e ponendo le basi per migliorare le qualità della vita. L'industria 4.0 richiede flessibilità, resilienza e reattività. Queste necessità portano allo sviluppo di sistemi in grado di ridurre i tempi, ridurre i costi, ridurre gli sprechi e di conseguenza le risorse. La disponibilità in azienda di dati e misure puntuali ha effetti positivi anche in termini di efficienza e di flessibilità dei programmi di produzione attraverso il recepimento dei frequenti cambi, la gestione dinamica degli ordini ai fornitori e la previsione continua dei trasporti.
La Data Visualization è utile in quanto permette di avere una visione complessiva dei dati e dell'informazione contenuta in essi, rendendola chiara e utilizzabile da tutti i membri di un'azienda. Consente la sperimentazione di diversi scenari industriali ed è per questo che sono stati sviluppati parecchi *data visualization tool*, con diversi formati in base alle necessità delle aziende.
Due elementi importanti nella Smart Factory sono l'interazione da macchina a macchina (M2M) e la collaborazione Human-To-Machine (H2M).
L'M2M consiste nella connessione tra macchine e prodotti per la comunicazione tramite IoT industriale, basata principalmente su reti wireless.
L'H2M è invece necessaria quando le attività di produzione sono troppo destruttu-

rate per essere completamente automatizzate. Attualmente molto sforzo di ricerca è investito nella cosiddetta robotica collaborativa, che consiste nella cooperazione tra uomo e robot appositamente progettati per svolgere mansioni di lavoro non strutturate nella linea di produzione manifatturiera, compiti che precedentemente venivano svolti solo manualmente. [7]

1.2 Decentralizzazione e integrazione verticale

La nozione di decentralizzazione è un pilastro fondamentale nell'Industry 4.0. Questo decentramento può essere fisico, ma deve essere soprattutto logico. Dal punto di vista fisico un CPS può identificarsi e connettersi ad un sistema fisicamente centralizzato, fornendo la propria posizione e il proprio stato ovunque si trovi.
Materiali intelligenti e risorse intelligenti saranno connesse da un legame logico, che permetterà l'elaborazione del materiale ad ogni risorsa, consentendo la creazione di prodotti unici. [6]
Strettamente legato al concetto di decentralizzazione è il concetto di integrazione verticale, che garantisce conformità, controllo o realizzazione di qualsiasi processo aziendale. L'integrazione verticale consiste nell'integrazione dei vari sistemi IT a diversi livelli gerarchici, come la pianificazione aziendale, la pianificazione della produzione o la gestione.
La produzione deve essere modulare e decentralizzata, in modo che tutti i servizi e le funzioni possano essere utilizzate dai diversi CPS. [6]
Le unità di produzione modulari devono cooperare tra loro per svolgere compiti comuni, sfruttando la percezione reciproca e la comunicazione tra moduli intelligenti. Occorre fare attenzione alla ridondanza, sfruttando la combinazione ottimale delle diverse funzioni dei moduli di produzione. Ogni unità di produzione può non solo soddisfare i requisiti di produzione dei prodotti, ma anche migliorare l'efficienza della fabbrica in modo auto-organizzato. [8]
L'unità produttiva è suddivisa in moduli da apparecchiature di produzione (ad es. robot industriale, braccio meccanico e centro di lavoro), migliorando sia la programmazione dinamica, sia la capacità di produzione flessibile. Piranda e Bourgeois han-

no proposto un algoritmo distribuito per la riconfigurazione del reticolo dei robot modulari auto-configurabili, che semplificano drasticamente la complessità della configurazione dei robot basato su un approccio iterativo. [8]
Sono stati introdotti dei robot cognitivi, integrati verticalmente nell'industria manifatturiera e coordinati con il sistema di esecuzione della produzione. Essi sono in grado di percepire le informazioni, di modificare la gestione della pianificazione e adeguare il comportamento di produzione per far fronte in modo indipendente a un problema di fabbricazione.
Il livello di sviluppo di una Smart Factory è strettamente correlato all'unità di produzione modulare. [8]

1.3 Le tecnologie dell'Industria 4.0

Le tecnologie avanzate su cui è basata l'Industria 4.0 sono:

- IoT e IIoT: componenti, sensori, microprocessori, piattaforme software, che permettono la comunicazione tra oggetti e macchinari. Inizialmente si trattava solo di tecnologie di identificazione basate su frequenze radio (RFID). In seguito, la tecnologia IoT è stata ampliata con sensori, sistemi GPS e dispositivi mobili che sfruttando le connessioni Wi-Fi, Bluetooth e le reti telefoniche, hanno permesso la creazione di reti di comunicazione;
- Cloud: tecnologie per l'archiviazione, l'elaborazione e la trasmissione di dati;
- Big Data: mole di dati generati da ogni macchinario da analizzare dando origine alla Big Data Analytics.

Queste tecnologie permettono la realizzazione di un *gemello digitale*, cioè di un modello del processo fisico che viene utilizzato per svolgere test, con il fine di migliorare le funzionalità, gestire i malfunzionamenti e prevenire gli errori di progettazione. La simulazione tramite gemello digitale assume oggi un ruolo chiave nell'Industria 4.0. [3]

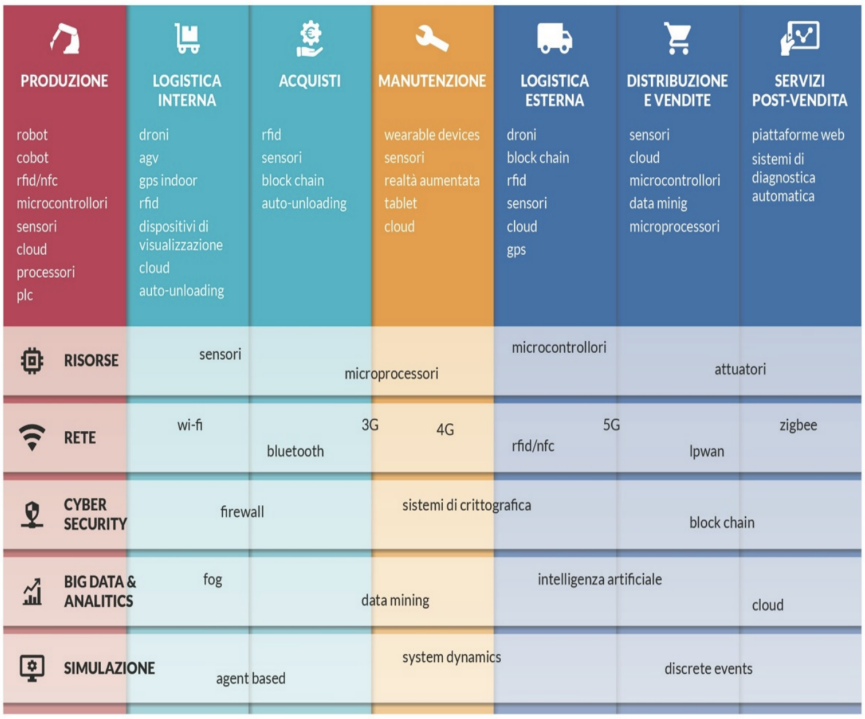


Figura 1.1: *Tecnologie per l'Industria 4.0*

Le tecnologie su cui si basa la nuova Industria 4.0 generano una mole immensa di dati che devono essere archiviati e gestiti in modo appropriato per assicurare un utilizzo vantaggioso all'azienda. Oggi è preferibile l'utilizzo dei *time series DB*, cioè database ottimizzati per le serie temporali: possiamo dire che questi tipi di dati rappresentano collettivamente il modo in cui un sistema o un processo cambiano nel tempo e comprendono dati dei sensori, monitoraggio delle prestazioni, metriche di server e altro ancora. [3] A ciascun data point in una serie storica è associato un timestamp, che viene registrato ogni volta su una nuova riga. È proprio questa caratteristica che li rende così potenti, perché permette di tracciare il cambiamento di un sistema nel tempo, di monitorare l'andamento nel presente e arrivare anche a predire i cambiamenti nel futuro.

I sensori IoT vengono oggi utilizzati per avere accesso ai dati storici e per ottenere approfondimenti e informazioni da applicare alla situazione attuale. Questi sensori permettono la realizzazione di " wireless sensor networks (WSN)", cioè di reti in grado di generare dati che le aziende più evolute sono in grado di gestire in tempo

reale e senza alcuna interazione umana, soprattutto nel prevenire guasti ai macchinari e fermo macchina. [5]
Il sensore è definito come un dispositivo che fornisce un output appropriato, in risposta ad un specifico valore misurato. La maggior parte dei sensori si comporta come un dispositivo passivo, in cui i valori cambiano a seconda dell'eccitazione esterna. I sensori IoT sono attivi, infatti sono in grado di elaborare il segnale di ingresso a livello logico, al fine di aumentare il livello di elaborazione delle informazioni. Il sensore intelligente è in grado di prendere una decisione logica ed eseguire un'azione a seconda delle informazioni rilevate. Altre caratteristiche sono la capacità di autotest, una calibrazione variabile e la migliore gestione di falsi input (rumore). Questa migrazione da sensori analogici a sensori intelligenti ha semplificato il lavoro degli utenti.
I produttori di sensori IoT stanno creando nuovi sensori con un costo ridotto e in grado di soddisfare le esigenze di applicazioni sempre più complesse. [4]
L'evoluzione tecnologica di diversi sistemi software e hardware ha portato allo sviluppo di WSN sempre più efficienti: l'Internet Protocol version 6 (IPv6) ha permesso la connessione di un numero illimitato di device, consentendone l'identificazione; Wi-Fi e Wimax garantiscono una maggior velocità di comunicazione a costo ridotto; le piattaforme mobili offrono comunicazioni di ogni genere tra diversi tipi di apparecchi.
Le WSN rappresentano l'espansione della tecnologia di comunicazione wireless esistente destinata ad applicazioni industriali. L'applicazione delle reti industriali è complessa ed è difficile introdurre uno standard di comunicazione di rete wireless generico. Le caratteristiche di questa rete dovrebbero essere bassa latenza, alta affidabilità, alta precisione nella sincronizzazione e dovrebbe permettere una quantità di accessi elevati e basso consumo energetico nell'acquisizione dei dati. Occorre eliminare i problemi di sincronizzazione nella trasmissione dei dati, con la conseguente perdita di pacchetti o ritardi, e limitare i fattori di interferenza che influenzano negativamente la qualità del servizio. [8]
La loro applicazione è davvero ampia e va dal settore automobilistico al controllo industriale, dall'assistenza sanitaria all'esplorazione petrolifera, grazie soprattutto

alla capacità di rilevare e trasmettere dati a basso costo.
Un'altra novità importante è l'utilizzo dei *chatbot* e degli assistenti intelligenti, i quali permettono di fornire assistenza agli operatori di produzione, fornendo loro una conoscenza base dei dati ottenuti dai sensori utili per effettuare scelte per l'assistenza e la manutenzione. [5]

Il Cloud Computing (CC) è una tecnologia per le aziende che intendono investire in risorse di outsourcing IT. È definito come un modello per consentire l'accessibilità a qualsiasi tipo di risorsa on-demand, in qualsiasi momento, e che possono essere rese disponibili facilmente a costi ridotti con un elevato grado di automazione.
L'adozione del CC presenta notevoli vantaggi legati alla riduzione dei costi, come ad esempio i costi sulla rimozione dell'infrastruttura IT nell'organizzazione, il servizio di razionalizzazione delle risorse da parte degli utenti che consumano solo le risorse informatiche effettivamente utilizzate o la portabilità nell'utilizzo di qualsiasi dispositivo collegato a Internet accedendovi da qualsiasi località del mondo.
Sono permessi quattro tipi di accesso: pubblico, privato, ibrido (combinazione di cloud pubblici e privati) e comunità (condivisi da più organizzazioni). [9]
L'infrastruttura del CC può essere suddivisa in:

- Infrastructure as a Service (IaaS): è il luogo in cui i produttori di servizi cloud forniscono agli utenti risorse informatiche fondamentali, con infrastrutture virtuali, come server virtuali, reti o archivi e dove gli utenti possono distribuire ed eseguire software arbitrario come, ad esempio, applicazioni di sistemi operativi;

- Platform as a Service (PaaS): è il luogo in cui gli utenti sviluppano ed eseguono applicazioni che utilizzano linguaggi di programmazione sul cloud. I vantaggi sono la scalabilità, server ad alta velocità e archiviazione. Gli utenti possono creare, eseguire e distribuire le proprie applicazioni con l'uso di piattaforme IT remote;

- Software as a Service (SaaS): è il luogo in cui risiedono e funzionano le applicazioni in un'infrastruttura cloud. Presentano un'interfaccia, come un brow-

ser Web e programmi, accessibile da vari dispositivi. L'obiettivo è eliminare le applicazioni di servizio sui dispositivi locali del singolo utente, ottenendo un'elevata efficienza per gli utenti.

Lo sviluppo di tecnologie avanzate ha permesso la nascita di una nuova modalità di produzione orientata al servizio, che consente agli utenti di richiedere servizi in tutte le fasi del ciclo di vita di un prodotto che vanno dalla progettazione, fabbricazione, gestione ecc. Si crea così una cooperazione tra fornitori, operatori e consumatori per mantenere il funzionamento di un sistema. [9]

La simulazione è uno strumento indispensabile e potente che permette di affrontare la complessità dei sistemi, risolvendo problemi che non possono essere gestiti con i soliti modelli matematici.
In un ambiente di produzione di prodotti personalizzati, il valore della simulazione è notevole ed evidente, in quanto consente esperimenti per la validazione di prodotti, processi o sistemi di progettazione e configurazione. La modellazione tramite simulazione aiuta a ridurre i costi, a ridurre i cicli di sviluppo e ad aumentare la qualità del prodotto.
La simulazione è definita come un'imitazione operativa, nel tempo, di un sistema o di un processo del mondo reale. Utilizza la storia artificiale di un sistema reale e aiuta a comprenderlo tramite la sua analisi comportamentale. Questa tecnologia consente l'approfondimento su sistemi complessi attraverso lo sviluppo di prodotti complessi e versatili e rende possibile testare nuovi concetti o sistemi, prima della loro reale implementazione.
Rispetto alla simulazione convenzionale, la simulazione real-time, on-line, può analizzare il comportamento dell'utente e del sistema in millisecondi, consentendo all'utente di sviluppare e produrre "virtualmente" un prototipo per il servizio di produzione di un prodotto. [9]

La realtà aumentata permette una migliore manutenzione dei macchinari industriali e l'assistenza da remoto in base ai dati generati dal macchinario stesso. I vantaggi derivanti sono la riduzione dei costi di assistenza e dei tassi di errore, ma soprattut-

to un aumento della qualità degli interventi. Infatti, con la realtà virtuale l'operatore può visualizzare i dati di pianificazione dal punto di vista dell'utente, riducendo il margine d'errore. Questo garantisce importanti vantaggi in termini di produttività aziendale, riduzione di costi, possibilità di errore e analisi dei dati, grazie all'integrazione con le strutture cloud esistenti.
La realtà aumentata permette la riduzione dei difetti, rilavorazioni e ispezioni ridondanti, sfruttando informazioni intuitive e combinando intelligenza e flessibilità dell'operatore con sistemi in grado di aumentare l'efficienza delle fasi di lavoro manuale. A livello di macchina, può ridefinire la manutenzione e la riparazione delle apparecchiature tramite una diagnostica intelligente fondata sui dati raccolti dai sensori in tempo reale; a livello aziendale, invece, può consentire ai responsabili di produzione di avere una visione d'insieme delle stazioni di lavoro, visualizzare gli indicatori di produzione e analizzare, diagnosticare e risolvere problemi e difetti. [10]

1.4 Limiti dell'Industria 4.0 nelle fabbriche moderne

Per molte industrie, le infrastrutture esistenti non sono ancora adatte a supportare la trasformazione tecnologica introdotta con l'Industry 4.0. Fondamentale sarà l'integrazione dei CPS, garantendo la connessione di componenti e metodi eterogenei. La sfida include la progettazione di interfacce in grado di adattarsi ai diversi componenti e riuscire a testarli. L'Industry 4.0 impone lo sviluppo di ambienti intelligenti, i quali richiedono dispositivi smart più avanzati, in grado di auto-configurarsi, autoproteggersi e di lavorare in modo ottimale. [5]
L'Industry 4.0 è caratterizzata da elevate velocità di flusso di dati e requisiti di elaborazione intensivi. Una fabbrica può avere risorse di sistema insufficienti per mantenere un'alta affidabilità e garantire tolleranze alle interruzioni. Nuove tecnologie dovrebbero contribuire alla resilienza dell'Industry 4.0 [5]. Tra queste la blockchain, una tecnologia che permette lo scambio di dati e di prodotti in maniera sicura. La blockchain permette la realizzazione di un database distribuito, strutturato in bloc-

chi, in grado di gestire le transazioni condivisibili tra più nodi di una rete. Ogni nodo deve controllare e approvare tutte le transazioni creando una rete che condivide su ciascun nodo l'archivio di tutta la blockchain. Le transazioni dei dati sono marcate e raggruppate in blocchi, ognuno dei quali è identificato dalla propria chiave crittografica. Il concetto di immutabilità sta nel fatto che tutte le transazioni possono essere considerate non modificabili. [11]

La scalabilità assume un ruolo fondamentale. Dato che le reti di produzione possono essere utilizzate con un'ampia varietà di dati ad alta velocità, la crescita del volume di informazioni porterà a problemi di scalabilità.

L'enorme mole di dati raccolti real-time, in continua crescita, necessita lo sviluppo di nuove tecniche e algoritmi di analisi, in quanto la presenza di dati non elaborati non fornisce un valore significativo al processo decisionale in una rete di produzione. Consentire l'elaborazione dei dati raccolti da un gran numero di dispositivi in modo efficiente e senza intoppi all'interno delle reti IoT è un compito impegnativo. La sfida consiste nel superare gli ostacoli nella gestione della connessione delle diverse entità per favorirne la comunicazione attraverso una piattaforma comune.

Per permettere la collaborazione tra diverse fabbriche o all'interno della stessa, è necessario un processo di standardizzazione dell'Industria 4.0. [5] Ad esempio, il "Reference Architecture Model for Industry 4.0 (RAMI 4.0)" , introdotto dalla German Electrical and Electronic Manufacturers' Association, introduce un sistema di coordinate tridimensionali che descrive i componenti essenziali dell'Industria 4.0. Questo sistema si basa sulla scomposizione delle interrelazioni complesse in sottosistemi, cluster o moduli. Un altro esempio è "The Industrial Internet Reference Architecture (IIRA)", il cui scopo è creare un sistema capace di gestire l'interoperabilità, mappare le tecnologie applicabili e guidare la tecnologia verso una standardizzazione. Standard di comunicazione, standard di identificazione e gli standard di sicurezza utilizzati nell'IoT potrebbero essere i principali driver per la diffusione delle tecnologie IoT. [5]

L'integrazione dello spazio virtuale con il mondo fisico ha fatto emergere numerosi problemi legati alla privacy e alla sicurezza. Una tremenda quantità di informazioni personali e private vengono raccolte automaticamente quando si applica l'IoT, ma

questo provoca di conseguenza l'aumento della superficie d'attacco. Molte tecnologie di crittografia esistenti sono state impiegate nelle WSN, ma la complessità e la mobilità dei servizi IoT, fanno si che la sicurezza delle informazioni e la protezione della privacy dei dati siano aspetti fondamentali che la ricerca futura dovrebbe considerare [5]. Le tecnologie 4.0 devono garantire la creazione di un ambiente sicuro all'interno degli ambienti industriali durante tutto il processo di produzione. [9]
La società di ricerca Gartner nello studio "Worldwide IoT security spending forecast 2018-2021 per segment" ha scoperto che negli ultimi tre anni un'azienda su 5 ha subito attacchi informatici. Il rischio è legato all'enorme quantità di dispositivi intelligenti che raccolgono dati e controllano i nostri edifici, la nostra produzione e i servizi di mobilità. Ecco perché Gartner stima una crescita negli investimenti in sicurezza informatica nell'ottica dell'IoT. [12]

1.5 Il mondo del lavoro

L'Industry 4.0 cambierà notevolmente il mondo del lavoro, con la necessità di nuove professionalità e la scomparsa di altre. Dalla ricerca 'The Future of the Jobs', presentata al Word Economic Forum 2016, è emerso che fattori tecnologici e demografici saranno determinanti nell'evoluzione del mondo lavorativo. La previsione stima la creazione di oltre 2 milioni di posti di lavoro, ma sarà la conseguenza della perdita di 7 milioni, con un saldo negativo di 5 milioni di posti, concentrati soprattutto nel campo dell'amministrazione e nella produzione. I settori lavorativi coinvolti positivamente saranno l'area finanziaria, il management, l'informatica e l'ingegneria. Cambieranno di conseguenza le competenze necessarie, tra cui la più ricercata rimarrà il problem solving, ma anche il pensiero critico e la creatività diventeranno qualità essenziali. La rapida evoluzione dello scenario porta la necessità di attrezzarsi per cogliere i benefici dello Smart Manufacturing, l'innovazione digitale nei processi dell'industria. [1]
L'Industry 4.0, richiede una coerente preparazione professionale dei soggetti interessati, relazionata alla continua evoluzione tecnologica. Emerge la necessità di valutare il fabbisogno di competenze professionali per lo sviluppo della logistica 4.0,

cercando di adattare dei programmi formativi che tengano in considerazione sia le persone che già operano in azienda e sia i giovani che si avvicinano al mercato del lavoro nel settore. La sfida è quella di riuscire a sviluppare competenze adeguate per rispondere alle necessità di un mondo produttivo in rapido cambiamento.
L'utilizzo di sistemi sempre più digitalizzati richiede figure professionali, sia a livello direttivo che operativo, dotate di specifiche conoscenze e adeguatamente formate, in grado di trasformare le opportunità tecnologiche in nuove opportunità di business. La persona resta il fulcro di ogni cambiamento, ma dovrà imparare a convivere con una fase di transizione nella quale le varie professioni e competenze continueranno a mutare in modo rapido.
L'evoluzione dei profili professionali richiesti dalle aziende si riflette sulle competenze che le aziende richiedono anche ai giovani laureati. Ad un'adeguata formazione di base si affiancano conoscenze linguistiche, informatiche e digitali, nonché la capacità di confrontarsi con la nuova realtà lavorativa (capacità comunicative e relazionali, negoziali, di lavorare in team, di risolvere eventuali situazioni di conflitto, spirito di iniziativa, ecc.).
Il rischio dell'esistenza di un distacco tra quanto viene richiesto dalle aziende e le conoscenze possedute dai giovani al loro ingresso nel mondo del lavoro, evidenzia che la chiave è trovare un'integrazione più stretta tra il mondo delle aziende e quello scolastico-accademico. [13]
La Deloitte Millennial Survey 2018 ha evidenziato la percezione dei giovani nei confronti delle opportunità lavorative che scaturiscono da un mondo sempre più complesso e in fase di trasformazione. Secondo lo studio, è emerso come i millenial ritengono che le imprese si concentrino troppo sul proprio business, dando priorità alla creazione di nuovi prodotti e servizi, piuttosto che pensare all'equilibrio tra lavoro e vita privata o all'impatto ambientale.
Fondamentale per le imprese sarà la possibilità di creare modelli di business innovativi, basati sulla tecnologia 4.0, tenendo in considerazione l'impatto sociale, con l'obiettivo di creare nuovi mercati, nuovi prodotti e cercando di attirare l'attenzione dei migliori talenti sul mercato. [14]

1.6 L'Industry 4.0 in Italia

L'Industria 4.0 nelle aziende italiane non è ancora stata accolta completamente, risultando in uno stato arretrato rispetto ad altri paesi.
L'indagine "EY Digital Manufacturing Maturity Index 2019", evidenzia che solo il 14% delle industrie italiane ha raggiunto uno stato digitale più avanzato, classificabile come 4.0, cioè con sistemi informativi in grado di scambiare informazioni verticalmente dalle macchine al cloud e presentando una buona integrazione delle informazioni lungo tutto il processo produttivo.
Il 49% delle aziende sta mettendo le basi per una gestione digitale dei processi, mentre il restante 37% si trova in una fase sperimentale di trasformazione digitale.
È evidente il divario tra piccole e grandi imprese, soprattutto nello sviluppo tecnologico. Il 70% delle grandi aziende ha introdotto nuove tecnologie, sfruttando le agevolazioni fiscali introdotte dal governo. Le piccole e medie aziende, invece, non sono riuscite a sfruttare pienamente questi incentivi. [1]
Numerose indagini hanno mostrato che i dirigenti di azienda sono consapevoli dei cambiamenti necessari ed esprimono un giudizio positivo per il futuro, nonostante ci sia ancora molta incertezza sulle strategie da seguire durante questa fase di trasformazione. Malgrado la situazione in bilico tra speranza e incertezza, le statistiche dimostrano che l'Italia è all'avanguardia per quanto riguarda lo sviluppo tecnologico. [14]
Il nostro Paese è ricco di risorse e potenzialità, che vanno però sfruttate e valorizzate tramite collaborazioni fra imprese, istituzioni ed enti pubblici e privati. Nel 2017 il Governo italiano ha emanato il "Piano Nazionale Impresa 4.0", contenuto all'interno della legge di bilancio 2017. Il provvedimento prevedeva incentivi fiscali per mobilitare gli investimenti privati nel campo dello sviluppo e innovazione con focus sulle tecnologie dell'Industria 4.0, diffusione della banda ultralarga, formazione dalle scuole all'università con lo scopo di incentivare le imprese ad adeguarsi e aderire pienamente alla quarta rivoluzione industriale. Queste manovre hanno riscosso risultati positivi, come dimostra la crescita degli investimenti nelle tecnologie digitali e l'acquisto di nuovi macchinari industriali.

Il 21 settembre 2017 il ministro Calenda ha presentato la fase due del Piano Nazionale, passando da Industria 4.0 a Impresa 4.0, ponendo l'attenzione sulla digitalizzazione dei servizi. Sono stati definiti nuovi investimenti, le strategie nazionali sulla tecnologia Blockchain e sull'Intelligenza Artificiale e la sperimentazione del 5G. [1] Il progetto Impresa 4.0 è stato arricchito nel 2019, con l'introduzione di incentivi per il cloud computing e la conferma dell'iperammortamento. In breve è stabilita la maggiorazione al 140% della deducibilità dei canoni per servizi e software utilizzati in cloud, mentre l'iperammortamento introduce aliquote differenziate sulla base del tetto degli investimenti. È stato introdotto il voucher per l'Innovation Manager per sostenere i processi di trasformazione digitale secondo quanto previsto dal Piano Nazionale Impresa 4.0.

1.7 Applicazioni moderne di Industry 4.0

L'Industria 4.0 è oggi parte integrante nella creazione di sistemi produttivi e ha avuto applicazione in moltissimi campi.

Alcune applicazioni concrete sono le Smart City, cioè le città che sfruttano l'IoT come supporto innovativo degli ambiti di gestione e nell'erogazione di servizi pubblici, con il fine di migliorare la vivibilità dei propri cittadini. Vengono sfruttate le informazioni in tempo reale per adattarsi al bisogno degli utenti. Alcuni esempi possono essere i semafori o i lampioni intelligenti in grado di funzionare solo in presenza di persone.

La Smart Farm consiste nell'agricoltura di precisione e l'automazione di attività produttive, ovvero l'uso interconnesso di più tecnologie che consentono di ottenere coltivazioni sostenibili, di determinare lo stato di salute e di sviluppo di piante e frutti in tempo reale. Sono state create numerose applicazioni che in base alle condizioni climatiche permettono la gestione automatizzata di acqua, fertilizzanti e concimi.

Molto rilevante è l'introduzione di una nuova criptovaluta, denominata IOTA, uti-

lizzata per il pagamento tra macchine e oggetti. IOTA presenta dei vantaggi rispetto ad altre criptovalute in quanto permette la realizzazione di transazioni libere e senza commissioni, ed è caratterizzata da una elevata velocità in ogni singola operazione. Le transazioni, infatti, avvengono parallelamente ed hanno peculiarità del tutto differenti. Non è soggetta a variabilità di valore in base all'utilizzo, ma è legata all'andamento del mercato come ogni altra moneta.

La quinta generazione della rete di comunicazione mobile, il 5G, è considerata comunemente come il fattore che consentirà all'IoT di decollare definitivamente per raggiungere i grandi numeri previsti, consentendo la connessione di 76 miliardi di dispositivi nel 2025 (oggi sono circa 20 miliardi).
Il 5G permette di raggiungere prestazioni, tempi di latenza e affidabilità che fino a oggi non erano possibili. Con la connessione mobile, inoltre, ogni genere di dispositivo potrà connettersi a Internet senza dover necessariamente utilizzare il wifi, quindi anche in assenza di una linea fissa. Questa evoluzione consentirà la connessione anche in stabilimenti produttivi situati in aree non serviti dalla rete fissa.
Lo standard 5G consente di tracciare costantemente e identificare in tempo reale tutti gli oggetti collegati alla rete, garantendo la comunicazione anche tra oggetti che utilizzano protocolli differenti. [12]

Capitolo 2

Metodologia di analisi automatizzata di serie temporali

In questo capitolo descriverò la metodologia e l'insieme di algoritmi utilizzati per l'implementazione di un framework in grado di svolgere un processo di analisi automatico di un qualsiasi dataset contenente serie temporali, cercando di minimizzare il contributo dell'esperto di dati.
La metodologia proposta mette a confronto due approcci: *Features based Analysis* e *Time-series based Analysis*. In entrambi i casi, la soluzione è stata studiata per serie temporali univariate con la stessa lunghezza. Nel caso in cui il dataset esaminato presentasse time-series di diversa dimensione, si è deciso di tagliarle alla lunghezza minima fra di esse.

2.1 Cos'è una serie temporale?

Una serie temporale consente di rappresentare l'evoluzione nel tempo di una misurazione di uno specifico sensore. A ciascun dato è associato come chiave primaria il timestamp di raccolta del dato stesso, ovvero l'istante temporale in cui è letto dall'architettura. Ad ogni timestamp è abbinata una time-series, cioè un vettore di dati raccolti sempre dallo stesso sensore in brevissimi istanti di tempo successivi.
Questo rende l'analisi di serie temporali diversa da altri dataset soliti, dove non c'è un ordine naturale delle osservazioni.

Una serie temporale x_i di lunghezza n è una sequenza di dati, misurati a intervalli di tempo regolari, denotati:

$$x_i = x_{i,1}, x_{i,2}, ..., x_{i,n}$$

dove $x_{i,j}$ rappresenta il j-esimo elemento della serie temporale x_i.
Le serie temporali possono essere univariate o multivariate. Una serie temporale è univariata se contiene record di un'unica osservazione.[15]
All'interno della tesi si fa riferimento a questo tipo di serie temporali.

L'implementazione di un impianto di produzione Industry 4.0 richiede una raccolta di dati in grado di garantire un flusso continuo di informazioni. Poiché il tempo è l'elemento critico del suo funzionamento, un database di serie temporali offre di gran lunga la soluzione migliore per fornire questa precisione richiesta.
Durante il funzionamento, le applicazioni di processo supportate da un database di serie temporali forniscono servizi per mantenere la linea di produzione efficiente e ridurre al minimo i tempi di fermo macchina.
L'efficienza della linea di produzione consiste nel controllo della sequenza degli eventi del processo di produzione. Questo controllo richiede l'ingestione di enormi quantità di dati provenienti da sensori, in modo che le istruzioni in tempo reale possano essere fornite ai sistemi cyber-fisici e ad altri aspetti della linea.
La minimizzazione dei tempi di fermo della linea di produzione è garantita attraverso l'analisi dei dati per prevedere i problemi e i guasti delle apparecchiature prima che si verifichino effettivamente.
Il ruolo di un database di serie temporali è quello di fornire un monitoraggio di precisione degli eventi e di fornire informazioni sui dati in modo che gli enormi volumi di dati ad alta precisione possano essere conservati per brevi periodi mentre i dati a bassa precisione siano conservati più a lungo o a tempo indeterminato.
Lo scambio di dati è fondamentale per il buon funzionamento di qualsiasi processo di produzione avanzato, soprattutto in un ambiente industriale 4.0. Qui l'intero processo si basa sull'ampio uso di sensori per fornire dati che portano alla regolazione e ottimizzazione del processo in tempo reale. [16]

Per questi motivi, l'implementazione di una metodologia automatica per l'analisi di serie temporali può risultare utile, soprattutto in un contesto di Industria 4.0.

2.2 Features based Analysis

Il primo approccio proposto è stato pensato e studiato durante il percorso di tesi, con l'obiettivo di ottenere una trasformazione dell'insieme di dati che portasse a un'analisi più semplice e veloce. La rappresentazione di dataset contenenti serie temporali, tramite una colonna contenente il timestamp e una colonna contenente il vettore di dati raccolti ad ogni preciso istante di tempo, non è di facile interpretazione per l'utente, soprattutto a causa delle grandi dimensioni del vettore contenente la serie temporale, nell'ordine delle migliaia. Inoltre, il legame con il tempo non permette l'applicazione degli algoritmi classici.

2.2.1 Estrazione e selezione delle features

Per rendere il dataset più maneggevole e per poterne usufruire meglio, la soluzione proposta consente di suddividere le serie temporali in un numero preciso di segmenti, definiti 'split'. Nella figura 2.1 è rappresentato l'andamento nel tempo di tre serie temporali e una possibile suddivisione di ogni segnale in parti uguali.

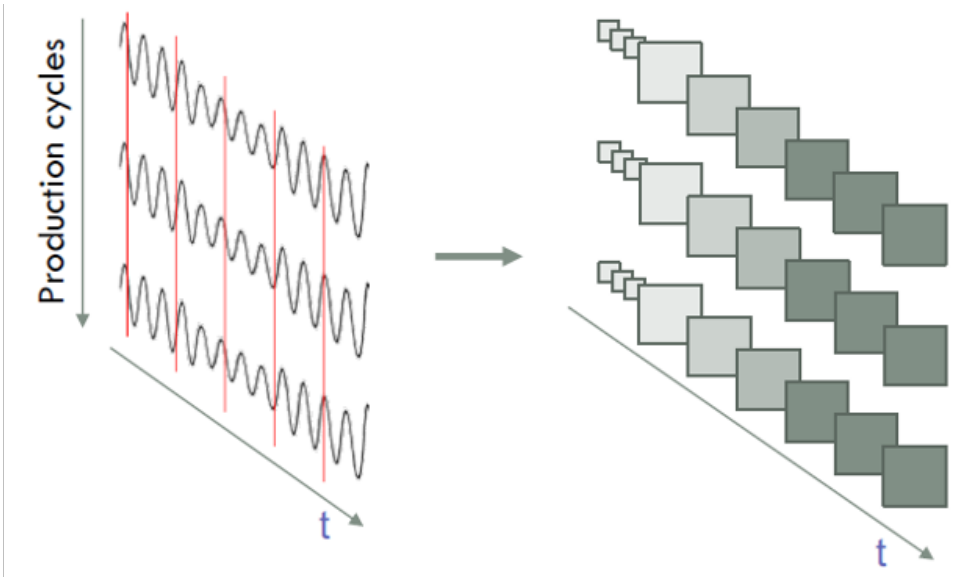


Figura 2.1: *Suddivisione delle serie temporali in 'split'*

Per ognuna di queste divisioni vengono successivamente calcolate delle features, definite Smart Data, che rappresentano le caratteristiche principali relative al segmento di segnale in esame.
Nel framework sono state utilizzate 14 features principali che sono:

- la media
- la deviazione standard
- il minimo
- il primo quartile
- la mediana
- il terzo quartile
- il massimo
- il coefficiente di Curtosi (*kurtosis*)
- l'indice di asimmetria (*skewness*)
- la radice quadrata della media dei quadrati dei valori (*root_mean_square*)
- la somma dei valori assoluti (*abs_values_sum'*)
- il numero di valori sopra la media diviso il numero totale di valori (*elem_over_mean*)
- la somma dei quadrati dei valori (*abs_energy*)
- la media dei valori assoluti delle differenze tra due valori consecutivi (*mean_abs_change*)

Ad esempio, se si decidesse di dividere le serie temporali in 10 parti, calcolando le 14 features introdotte precedentemente per ogni split, si otterrebbero 140 features per ogni timestamp.
In seguito, volendo ridurre ulteriormente la mole di dati presente, viene eseguita l'estrazione delle features più significative. La scelta viene effettuata calcolando la matrice di correlazione tra tutte le feauteres ottenute e tenendo in considerazione

solo quelle colonne la cui media dei valori assoluti è inferiore ad una determinata soglia, cioè mantenendo solo quelle meno correlate tra di loro.
La matrice di correlazione [17] permette di osservare la relazione di una variabile con ogni altra variabile.
Il coefficiente di correlazione di Pearson $\rho_{X,Y}$ è definito come la covarianza di X e Y divisa per il prodotto della deviazione standard di X e Y. Se $\rho_{X,Y} > 0$, le due variabili sono direttamente proporzionali, se $\rho_{X,Y} < 0$ le due variabili sono inversamente proporzionali.
Il coefficiente di Pearson è compreso nel range [-1,1]:

$$\rho_{X,Y} = \frac{Cov(X,Y)}{\sqrt{Var(X) * Var(Y)}} = \frac{\sum_{i=1}^{n}(x_i - \mu_x)(y_i - \mu_y)}{\sqrt{\sum_{i=1}^{n}(x_i - \mu_x)^2}\sqrt{\sum_{i=1}^{n}(y_i - \mu_y)^2}} \quad , \quad \rho_{X,Y} \in [-1, 1]$$

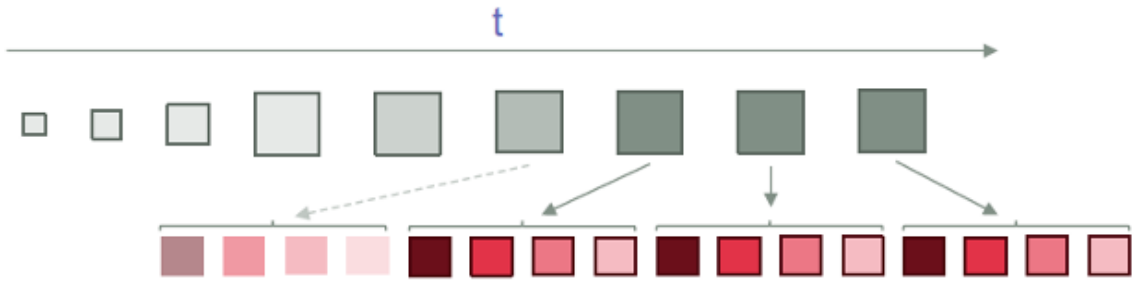


Figura 2.2: *Calcolo ed estrazione delle features per ogni split*

2.2.2 Scelta del numero di split

Per rendere il processo di analisi del dataset il più automatizzato possibile, il framework propone all'utente una possibile suddivisione automatica delle serie temporali in un numero K di split con la stessa dimensione.
La scelta di K viene effettuata applicando ripetutamente l'algoritmo DBSCAN al dataset ottenuto tramite il calcolo delle features, aumentando di volta in volta il numero di split, e confrontando 5 indici relativi alla bontà del clustering.

Algoritmi di clustering: DBSCAN

Gli algoritmi di clustering [17] permettono di raggruppare i dati in classi omogenee e risultano molto utili quando il dataset in esame non presenta un'etichetta di classe. Quando si raggruppano le osservazioni di un dataset si cerca di dividerle in gruppi distinti, definiti cluster, in modo tale che le osservazioni all'interno di ogni gruppo siano abbastanza simili tra loro e siano abbastanza diverse da quelle presenti in altri gruppi.
I diversi algoritmi di clustering basano il concetto di similarità sul calcolo delle distanze tra i punti ed è quindi necessario che i dati siano in formato numerico.
Questo permette l'applicazione diretta di queste tecniche sia ai dataset contenenti delle time-series sia a quelli contenenti le relative features.

L'algoritmo DBSCAN [18] è un metodo di clustering basato sul concetto di densità, in quanto connette regioni di punti con densità sufficientemente alta.
Il DBSCAN stima la densità attorno a ciascun punto contando il numero di punti in un intorno *eps* ed applica delle soglie definite *minPts*. Entrambi i parametri vengono definiti a priori, prima dell'applicazione dell'algoritmo.
In base alla scelta di *eps* e *minPts*, ogni punto p viene classificato come:

- *Core Point*, se almeno *minPts* punti sono ad una distanza inferiore di *eps* da esso, p incluso;
- *Border Point*, se non è un Core Point, ma cade nell'intorno *eps* di un altro Core Point;
- *Noise Point*, se è un punto non raggiungibile da altri punti, e non è quindi classificabile come Core Point o Border Point;

Fondamentale è il concetto di raggiungibilità:

- Un punto q è *direttamente raggiungibile* da p se il punto q è contenuto nell'intorno *eps* di p e lo stesso p è un Core Point.
- Un punto q è *raggiungibile* da p se esiste un percorso $p_1 ... p_n$ da p a q, dove ogni $p_i + 1$ è direttamente raggiungibile da p_i.

La raggiungibilità [19] non è una relazione simmetrica in quanto un punto non-core può essere raggiungibile, ma nulla può essere raggiunto da esso.
Pertanto, viene introdotta la *density-connectedness*:

- Due punti p e q sono *density-connected* se esiste un punto t tale che sia p che q siano raggiungibili da t.

La density-connectedness è simmetrica. Un cluster quindi soddisfa due proprietà:

- Tutti i punti del cluster sono mutuamente *density-connected*.
- Se un punto è *density-connected* ad un qualunque punto del cluster, allora è parte del cluster stesso.

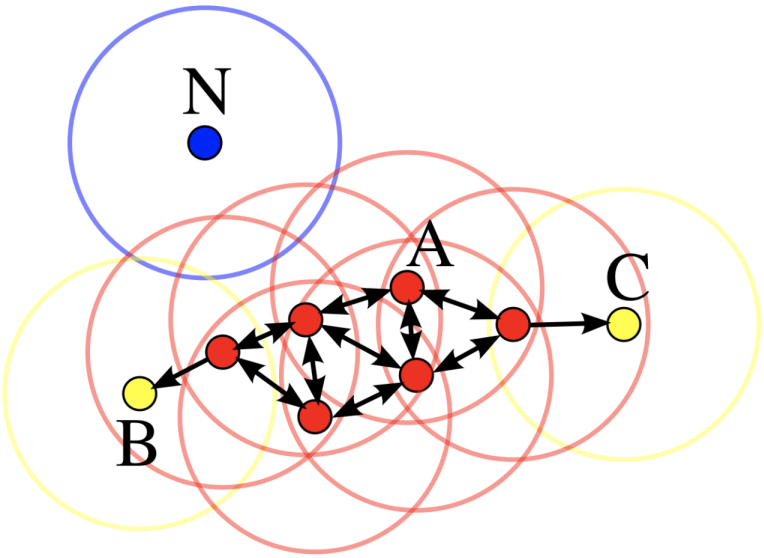


Figura 2.3: *Esempio di Core, Border e Noise Points*

Nell'esempio in figura 2.3 con *minPts* = 4, i punti rossi sono dei Core Points, perché ognuno di essi ha nel proprio intorno di raggio *eps* almeno 4 punti. Essendo tutti raggiungibili tra di loro, questi punti rossi formano un singolo cluster.
I punti B e C sono dei Border Points, in quanto non sono dei Core Points, ma sono raggiungibili da essi e quindi appartengono a loro volta al cluster.
Il punto N non è né un Core Point né un Border Point, ed è quindi classificato come Noise Point.[19]

L'algoritmo procede in questo modo: [18]

- etichetta ogni punto come Core, Border o Noise Point;
- elimina i Noise Points;
- mette un bordo tra tutti i punti che sono dentro l'intorno di raggio *eps* l'uno dell'altro;
- crea per ogni gruppo di Core Points connessi un cluster separato;
- assegna ogni Border Points ad uno dei cluster dei Core Points associati.

Scelta automatica dei parametri

La scelta dei parametri *minPts* e *eps* è determinante per il comportamento dell'algoritmo DBSCAN, visto che valori troppo bassi portano ad avere molti cluster e classificare molti punti come Noise Points. Invece, valori troppo alti portano a unire diversi cluster e far si che molti punti appartengano allo stesso cluster.
Data l'importanza di questi due parametri, è stata introdotta una metodologia per il calcolo di *minPts* e *eps* in modo automatico, escludendo la scelta da parte dell'utente.

MinPts
Per il calcolo di *minPts* viene preso un intervallo di valori di k, da 2 a 50, e partendo da k = 2 si calcola per ciascun punto la distanza dal suo k-esimo punto più vicino.
Si ordinano le distanze ottenute in ordine decrescente e si rappresentano graficamente come una curva.
Dopodiché, si ripete lo stesso procedimento per k+1 e plottando nuovamente la curva delle distanze viene fatto il confronto con la curva precedente tramite una misura di distanza definita MAPE (*mean absolute percentage error*), la cui formula è:

$$M = \frac{100}{n} \sum_{t=1}^{n} \frac{A_t - F_t}{A_t}$$

dove A_t è un punto della k-esima curva, F_t è un punto della k+1-esima curva e n è il numero di punti nel dataset.

Se il valore di M è inferiore ad una soglia, precedentemente prefissata, allora le due curve sono simili e si stoppa l'algoritmo fissando il valore di *minPts* = k.

Eps

Definito il valore di *minPts* = k, si calcola per ogni punto la distanza dal suo k-esimo più vicino e si ordinano le distanze ottenute in ordine decrescente.
Plottando le distanze graficamente si ottiene una curva, e l'obiettivo è trovare il punto di gomito della curva stessa. Per trovare questo punto in modo automatico si traccia una retta tra il primo punto e l'ultimo della curva e si trova il punto della curva più distante dalla retta tracciata, come mostrato in figura 2.4.
Il punto di gomito corrisponde al valore di *Eps*. Di conseguenza, tutti i punti che lo precedono verranno etichettati come Noise Points, mentre i punti successivi verranno assegnati ai cluster.

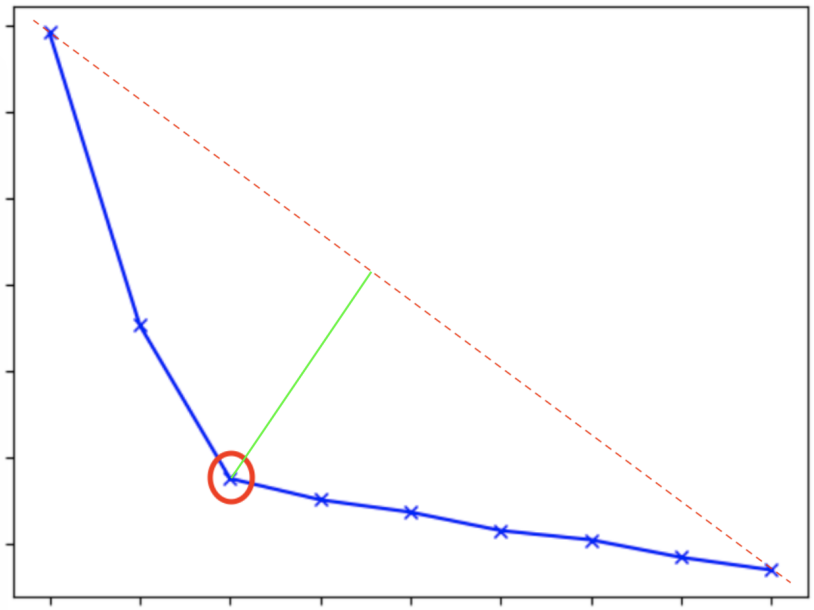


Figura 2.4: *Punto di gomito*

Indici per misurare la qualità del clustering

Per capire con quale numero di split l'algoritmo DBSCAN fornisce risultati migliori, vengono calcolati 5 indici relativi alla bontà del clustering ottenuto:

- *Average Silhouette Index (ASI)*: è definito come

$$ASI = \frac{1}{N}\sum_{k=1}^{K}\sum_{i \in C_k} s_i$$

dove N è il numero di elementi nel dataset, C_k è l'insieme degli elementi appartenenti al cluster k = 1,..,K, e s_i è l'indice di Silhouette.
La Silhouette è una misura di quanto un oggetto sia simile nel proprio cluster, rispetto ad altri cluster. È compreso tra -1 e 1, dove un valore alto indica che l'oggetto è ben abbinato al proprio cluster e scarsamente abbinato ai cluster vicini;

- *Global Silhouette Index (GSI)*: è definito come

$$GSI = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{|C_k|}\sum_{i \in C_k} s_i$$

dove $|C_k|$ è la cardinalità del cluster k. È compreso tra -1 e 1;

- *Davies-Bouldin*: è definito come la misura di somiglianza media di ogni cluster con il suo cluster più simile, dove la somiglianza è il rapporto tra le distanze all'interno del cluster e le distanze tra i cluster. Più i cluster sono separati e meno sono dispersi, più il punteggio sarà migliore.
Il punteggio minimo è zero, con valori inferiori che indicano migliore clustering;

- *Dunn Index*: è definito come la minima distanza inter-cluster divisa per la dimensione massima del cluster. Grandi distanze tra i cluster (migliore separazione) e minori dimensioni dei cluster (cluster più compatti) portano ad un punteggio migliore.
È compreso tra 0 e $+\infty$, e più il valore è alto migliore è il clustering;

- *Calinski-Harabasz*: è definito come il rapporto tra la dispersione all'interno del cluster e la dispersione tra i cluster, dove la dispersione è definita come la somma delle distanze al quadrato.
È compreso tra 0 e $+\infty$, e più il valore è alto migliore è il clustering;

2.2.3 Procedura per la scelta automatica del numero di split

Partendo dal dataset originale contente le time-series e dato k = 1, il framework:

- calcola il dataset trasformato tramite estrazione delle features con numero di split = k della stessa dimensione;
- applica l'algoritmo di DBSCAN sul dataset trasformato;
- calcola i 5 indici relativi alla bontà del cluster;
- ripete la procedura incrementando k di 1.

Il ciclo finisce quando almeno 4 su 5 indici relativi alla bontà del clustering sono peggiori rispetto alle due iterazioni precedenti oppure si è raggiunto il valore limite di k, pari a 24.
Al termine del ciclo, si guardano i valori di ogni indice per tutti i possibili numeri di split provati, si estraggono i 5 valori migliori dell'indice e si attribuisce un punteggio da 5 a 1 (5 al primo valore migliore, 4 al secondo e così via). Alla fine si sommano per ogni numero di split il punteggio ottenuto dai relativi indici e si guarda quale numero di split ha ottenuto il punteggio complessivo maggiore.
Questo numero di split sarà quello consigliato all'utente.

Esempio di scelta automatica del numero di split

Partendo da k = 1, vengono calcolati i 5 indici relativi alla bontà del clustering, in questo caso dell'algoritmo DBSCAN, sul dataset ottenuto tramite il calcolo e l'estrazione delle feauteres calcolate con numero di split pari a k.
Ripetendo la procedura, aumentando ad ogni ciclo il valore di k, si ottiene la seguente tabella:

N_split	ASI	GSI	Davies - Bouldin	Dunn Index	Calinski - Harabasz
1	0.82749	0.54430	1.65639	0.00285	5614.154
2	0.51150	0.55502	2.16739	0.01346	1492.465
3	0.68218	0.62620	1.74041	0.00702	2324.846
4	0.55195	0.57699	1.64444	0.00711	1067.738
5	0.58110	0.60832	1.57891	0.01309	1536.741
6	0.59192	0.52477	1.45983	0.01537	2452.284
7	0.60342	0.60034	1.36343	0.01249	2075.963
8	0.41984	0.59501	1.33643	0.01116	1402.361
9	0.45163	0.47696	1.32854	0.01786	705.289
10	0.45325	0.48623	1.49490	0.01475	500.220
11	0.33127	0.37243	1.65998	0.02771	270.728

Figura 2.5: *Tabella contenente gli indici per ogni numero di split*

La tabella termina con numero di split = 11, perché con questo valore 4 indici su 5, quelli evidenziati in blu, sono peggiori rispetto agli stessi indici calcolati nelle due iterazioni precedenti, evidenziati in rosso.
Ricavati 11 possibili valori per il numero di split, bisogna trovare il numero di split migliore. Per fare ciò, si prende ogni singolo indice (colonna) e si attribuisce un punteggio, da 5 a 1, ai 5 valori migliori di quell'indice. Agli altri verrà assegnato punteggio zero. Si ricorda che per il Davies-Bouldin i valori minori sono migliori.

ASI		GSI		Dav-Bou		Dunn		Cal - Har	
0.82749	5	0.54430	0	1.65639	0	0.00285	0	5614.154	5
0.51150	0	0.55502	0	2.16739	0	0.01346	1	1492.465	0
0.68218	4	0.62620	5	1.74041	0	0.00702	0	2324.846	3
0.55195	0	0.57699	1	1.64444	0	0.00711	0	1067.738	0
0.58110	1	0.60832	4	1.57891	0	0.01309	0	1536.741	1
0.59192	2	0.52477	0	1.45983	2	0.01537	3	2452.284	4
0.60342	3	0.60034	3	1.36343	3	0.01249	0	2075.963	2
0.41984	0	0.59501	2	1.33643	4	0.01116	0	1402.361	0
0.45163	0	0.47696	0	1.32854	5	0.01786	4	705.289	0
0.45325	0	0.48623	0	1.49490	1	0.01475	2	500.220	0
0.33127	0	0.37243	0	1.65998	0	0.02771	5	270.728	0

Figura 2.6: *Calcolo dei punteggi*

Al termine, si sommano i punteggi ottenuti per ogni numero di split e si osserva quale numero di split ha ottenuto punteggio migliore.

N_split	Punteggio
1	5 + 0 + 0 + 0 + 5 = 10
2	0 + 0 + 0 + 1 + 0 = 1
3	12
4	1
5	6
6	11
7	11
8	6
9	9
10	3
11	5

Figura 2.7: *Calcolo punteggio complessivo*

In questo esempio, il punteggio migliore si ottiene con numero di split pari a 3, che sarà quello consigliato all'utente.

2.3 Time-series based Analysis

Il secondo approccio consiste nell'utilizzo di un algoritmo tradizionale allo stato dell'arte per l'analisi di serie temporali. È un approccio *raw-data based*, cioè opera direttamente sulle time-series, e permette di valutare il grado di similarità tra diverse serie temporali calcolando la distanza tra ognuna di esse.
Il concetto di distanza euclidea porta a risultati non corretti in presenza di serie temporali.

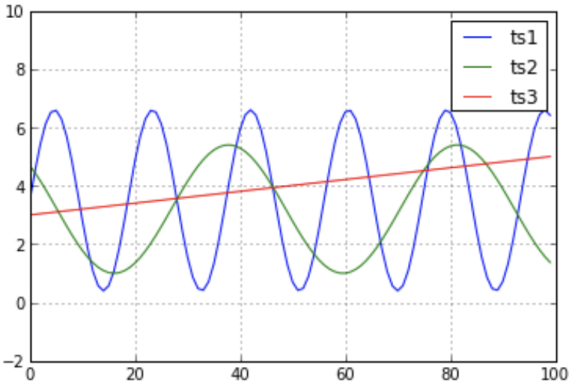


Figura 2.8: *Andamento di tre serie temporali*

Osservando la figura 2.8, è evidente che la serie temporale *ts1* è più simile alla serie temporale *ts2*, rispetto a *ts3*. Calcolando però la distanza euclidea tra *ts1* e *ts2* e tra *ts1* e *ts3*, si ottiene che *ts1* è più vicina a *ts3*.
Di conseguenza, la distanza euclidea non può essere utilizzata per misurare la similarità tra serie temporali, in quanto è influenzata dalla distorsione lungo l'asse x.
Mentre l'essere umano è in grado intuitivamente di valutare il grado di somiglianza, questo compito diventa molto complesso per un computer. Non è banale estrarre da serie temporali un modello statistico, poiché possono mostrare proprietà variabili con il tempo ed essere non stazionarie.
Le tecniche esistenti possono essere divise in *shape-based* e *structure-based*. Le tecniche *shape-based* utilizzano una misura di somiglianza in combinazione con la ricerca di 1-NN. Sono competitive su set di dati preelaborati, ma falliscono su dati lunghi o rumorosi. Le tecniche *structure-based* trasformano la rappresentazione della serie temporali o estraggono da esse delle caratteristiche rappresentative creando nuovi modelli caratteristici. Ciò comporta un costo computazionale più elevato. [20]
Diversi algoritmi sono stati introdotti per poter misurare la similarità tra serie temporali e la mia scelta è ricaduta sull'utilizzo dell'algoritmo BOSS.

2.3.1 BOSS

L'algoritmo Bag-of-SFA-Symbols (BOSS) [20] è una misura di somiglianza *structure-based* che applica la riduzione del rumore alle serie temporali non elaborate.
Prima estrae le sottostrutture di una serie temporale e, successivamente, utilizza un filtro passa basso e la quantizzazione delle sottostrutture per ridurre il rumore. Questo consente l'applicazione di algoritmi di corrispondenza tramite stringhe. Due serie temporali vengono confrontate in base all'insieme delle differenze del rumore presente nei modelli ridotti.
I vantaggi del BOSS sono la velocità, la riduzione del rumore e l'invarianza agli offset trattata come parametro.

Background

Prima di entrare nei dettagli del BOSS, occorre introdurre una serie di operazioni preliminari per la realizzazione dell'algoritmo.
Data una serie temporale $T = (t_1, ..., t_n)$ con $n \in \mathbb{N}$:

- viene applicata una funzione di *windowing* per suddividerla in finestre di lunghezza fissa;
- per ogni finestra viene applicata una rappresentazione simbolica, chiamata Symbolic Fourier Approximation (SFA), che permette di applicare un filtro passa basso e una quantizzazione per ridurre il rumore;
- infine viene creato l'istogramma delle parole SFA per valutare la similarità.

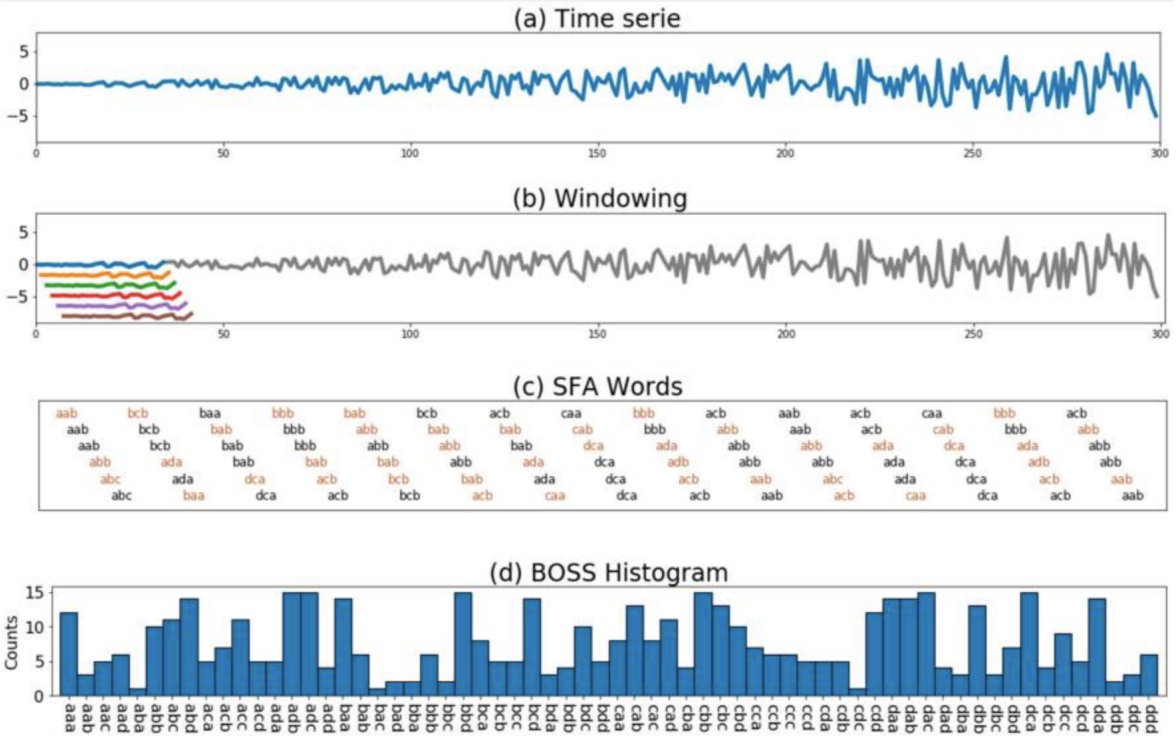


Figura 2.9: *Esempio di costruzione del modello BOSS*

Funzione di Windowing

La funzione di *windowing* permette di dividere una serie temporale $T = (t_1, ..., t_n)$ di lunghezza n in finestre $S_i; w = (t_i, ..., t_{i+w-1})$ di lunghezza fissa w.
Due finestre consecutive in posizione i e $i + 1$ si sovrappongono in $w - 1$ posizioni:

$$windows(T, w) = \left\{ \underbrace{S_{1;w}}_{(t_1,\ldots,t_w)} , \underbrace{S_{2;w},}_{(t_2,\ldots,t_{w+1})} \ldots, S_{n-w+1;w} \right\}$$

Symbolic Fourier Approximation (SFA)

La SFA permette di rappresentare una serie temporale con una sequenza di simboli, denominati parole SFA, utilizzando un alfabeto finito di simboli.

La trasformazione SFA applica:

- un filtro passa basso per rimuovere il rumore associato a rapidi cambiamenti all'interno del segnale. La lunghezza delle parole SFA determina il numero di coefficienti di Fourier utilizzati e di conseguenza la banda del filtro ;
- una rappresentazione tramite stringhe, grazie ad un meccanismo di quantizzazione che permette l'applicazione di un algoritmo di corrispondenza tramite stringhe. La grandezza dell'alfabeto determina il grado di quantizzazione.

L'SFA è composta da due operazioni, l'approssimazione e la quantizzazione, come mostrato in figura 2.10.

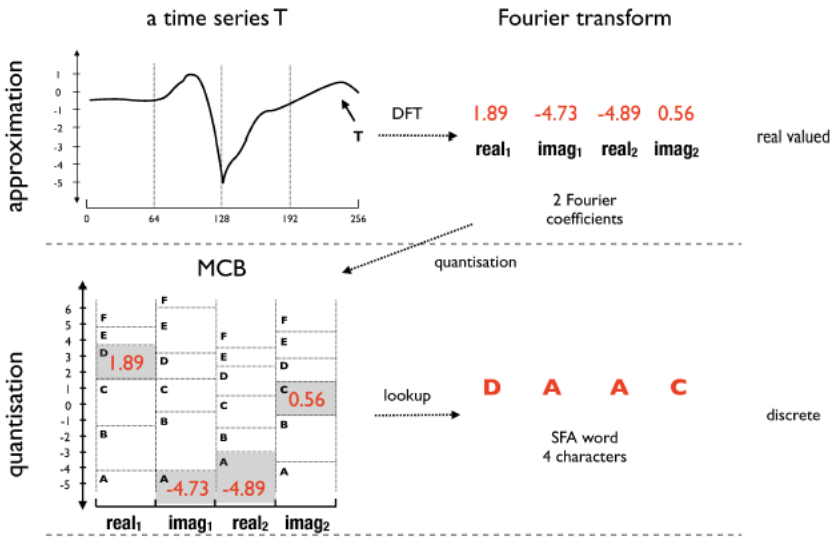


Figura 2.10: *Approssimazione serie temporale tramite DFT e quantizzazione tramite MCB*

L'approssimazione è effettuata calcolando i coefficienti di Fourier, tramite la Discrete Fourier Transform (DFT), che decompone un segnale T di lunghezza n nella somma di funzioni ortogonali usando onde sinusoidali. Ogni onda è rappresentata da un numero complesso $X_u = (real_u, imag_u)$ con $u = 0, 1, ..., n-1$, chiamato coefficiente di Fourier:

$$DFT(T) = X_0 ... X_{n-1} = (real_0, imag_0, ..., real_{N-1}, imag_{n-1})$$

con

$$X_u = \frac{1}{n} \sum_{x=0}^{n-1} T(x) \cdot e^{-j2\pi ux/n}, \ con\ u \in [0, n), \ j = \sqrt{-1}$$

L'approssimazione consente di rappresentare il segnale di lunghezza n con un segnale trasformato di lunghezza l. Il segnale è filtrato passa-basso utilizzando i primi l $\ll$ n coefficienti di Fourier, cioè quelli correlati alle gamme di frequenza più basse e al lento cambiamento del segnale. I coefficienti di ordine più alto sono correlati a una frequenza più alta relativa a rapidi cambiamenti del segnale, spesso associati a rumore, e vengono quindi eliminati.

Il processo di quantizzazione consiste nel generare una tabella di ricerca nelle serie temporali, i cui intervalli definiscono le lettere che compongono la parola generata dopo il processo di discretizzazione. Questa tabella di ricerca è definita Multiple Coefficient Binning (MCB). Lo scopo è ridurre al minimo la perdita di informazioni dovuta al processo di quantizzazione.
Gli intervalli MCB sono calcolati partendo da una matrice A, composta dalle trasformate di Fourier di N serie temporali di addestramento, usando solo i primi $\frac{l}{2}$ coefficienti, dove l è la lunghezza della parola SFA:

$$A = \begin{pmatrix} DFT(T_1) \\ . \\ DFT(T_i) \\ . \\ DFT(T_N) \end{pmatrix} = \begin{pmatrix} real_{11} & imag_{11} & \dots & real_{1\frac{l}{2}} & imag_{1\frac{l}{2}} \\ . & . & \dots & . & . \\ real_{i1} & imag_{i1} & \dots & real_{i\frac{l}{2}} & imag_{i\frac{l}{2}} \\ . & . & \dots & . & . \\ real_{N1} & imag_{N1} & \dots & real_{N\frac{l}{2}} & imag_{N\frac{l}{2}} \end{pmatrix} = \begin{pmatrix} C_1 & \dots & C_j & \dots & C_l \end{pmatrix}$$

La j-esima colonna C_j corrisponde ai valori reali o immaginari di tutti gli N segnali di allenamento. Date le colonne C_j ordinate, con $j = 1, .., l$ e un alfabeto Σ di

dimensione c, MCB determina c+1 breakpoints $B_j(0) < ... < B_j(c)$ per ogni colonna C_j, usando intervalli equi-distanti. Otteniamo cosi l set da c+1 intervalli, come mostrato in figura 2.10.
Alla fine etichettiamo ogni intervallo MCB con l'a-esimo simbolo dell'alfabeto Σ. Per ogni coppia (j, a) con $j = 1, ..., l$ e $a = 1, ..., c$:

$$[\beta_j(a-1), \beta_j(a)) \stackrel{\Delta}{=} symbols_a \in \Sigma$$

La trasformazione di una qualsiasi serie temporale in un insieme di parole SFA si ottiene utilizzando gli intervalli MCB precedentemente calcolati.
La rappresentazione simbolica $SFA(t) = s_1, ..., s_l$ di una time series T con $DFT(T) = real_0, imag_0, ..., real_{\frac{l}{2}-1}, imag_{\frac{l}{2}-1} = t'_1, ..., t'_l$ si ottiene mappando i valori reali e immaginari con un simbolo dell'alfabeto Σ.
Il j-esimo valore t'_j è mappato nell' a-esimo simbolo, se cade nel suo intervallo:

$$(\beta_j(a-1) \leq t'_j < \beta_j(a)) \Rightarrow s_j \equiv symbol_a \in \Sigma$$

Modello BOSS

L'algoritmo BOSS utilizza le funzioni sopra descritte per trasformare ogni serie temporale in un insieme non ordinato di parole SFA. Dopo aver diviso la serie temporale in finestre di dimensione fissa w, ogni finestra viene normalizzata per avere deviazione standard pari a 1 e successivamente si applica una trasformazione SFA per ottenere l'insieme delle parole.
L'utilizzo di un insieme non ordinato fornisce l'invarianza dell'allineamento orizzontale di ciascuna sottostruttura all'interno della serie temporale. In sezioni stabili del segnale, è probabile che la SFA di due finestre vicine crei parole identiche. Per evitare di dare peso maggiore a sezioni stabili viene eseguita una riduzione della numerosità, andando a ignorare le serie di duplicati consecutivi.
Ad esempio se $S =$ **abb** *abb abb abb abb* **bbc** *bbc* **abb bcb** *bcb bcb*, applicando la riduzione si ottiene $S' =$ **abb bbc abb bcb**.
Contando le occorrenze per ogni parola si costruisce l'istogramma. Nell'esempio precedente si ottiene $B: \ abb = 2, \ bbc = 1, \ bcb = 1$.

Due serie temporali sono simili se hanno lo stesso insieme di parole SFA.
La distanza tra due serie temporali T_1 e T_2, dati i relativi istogrammi delle occorrenze B_1 e B_2, è definita come:

$$D(T_1, T_2) = dist(B_1, B_2) = \sum_{a \in B_1; B_1(a) > 0} [B_1(a) - B_2(a)]^2$$

La condizione $B_1(a) > 0$ permette di omettere i conteggi di parole SFA pari a 0 nel confronto tra le serie temporali, in quanto l'assenza di parole SFA ha due ragioni: il rumore distorce le sottostrutture o una sottostruttura non è contenuta in un altro segnale.
Questa condizione porta però ad avere misure di distanza non simmetriche, cioè la distanza tra T_1 e T_2 risulta diversa da quella tra T_2 e T_1.
Se si vuole mantenere la simmetria tra le distanze ottenute questo vincolo può essere omesso, ed è possibile calcolare la distanza tra gli istogrammi delle occorrenze utilizzando la classica distanza Euclidea, come fatto all'interno nel progetto.

2.4 Confronto

Applicando l'algoritmo BOSS al dataset originale si ottiene la matrice delle distanze tra ogni serie temporale. La matrice ottenuta avrà quindi la diagonale pari a 0, in quanto la distanza di ogni serie temporale da se stessa è nulla, e sarà simmetrica, cioè la distanza tra T_1 e T_2 risulterà uguale a quella tra T_2 e T_1.
Una volta ottenuta questa matrice, è possibile effettuare un confronto grafico tra il dataset contente le features estratte e il dataset originale contenente le serie temporali, tramite l'applicazione degli algoritmi di clustering. Per il primo dataset vengono utilizzati il DBSCAN e il KMedoids classici, basati sulla distanza euclidea, mentre nel secondo vengono utilizzati gli stessi algoritmi di clustering ma con delle piccole modifiche, in quanto il calcolo dei parametri sarà basato sulla matrice delle distanze calcolata tramite BOSS.
Oltre al confronto visivo, è possibile confrontare la bontà degli algoritmi di clustering in entrambi i dataset tramite 3 indici già descritti in precedenza: *Average Silhouette Index*, *Global Silhouette Index* e *Dunn Index*. Anche in questo caso, per il da-

taset contenente le serie temporali saranno necessarie delle piccole variazioni per poter calcolare questi indici basandosi sulla matrice delle distanze.

2.4.1 K-MEDOIDS

Il K-Medoids [21] è un algoritmo di clustering partizionale che, dato un insieme di n elementi e un numero K di cluster definito a priori, suddivide l'insieme dei dati in K cluster differenti, ognuno rappresentato da un *medoid*.
Un *medoid* può essere definito come un elemento di un cluster la cui dissimilarità media rispetto a tutti gli oggetti nel cluster è minima, in questo modo esso sarà il punto più centrale di un dato insieme di punti.
L'obiettivo dell'algoritmo è minimizzare l'errore quadratico medio della distanza tra punti di un cluster e il punto designato per esserne il centro. L'idea principale è di aver come *medoids* punti del dataset stesso, in modo da renderli interpretabili.

L'algoritmo procede in questo modo:

- si selezionano casualmente K oggetti come *medoids* da un insieme di n punti (con K < n) ;
- assegna ogni punto dell'insieme al *medoid* più simile, dove il concetto di similarità è basato sulla funzione di costo che è definita in termini di distanza;
- seleziona un elemento non *medoid* casuale *O'*;
- calcola il costo S_i che è la somma dei costi dei singoli elementi dal corrispondente *medoid* iniziale e il costo totale S_f usando come *medoid O'* e se ne calcola la differenza $S = S_f - S_i$;
- se $S < 0$ si scambia il *medoid* iniziale con il nuovo;

si ripete il procedimento sino a quando si hanno cambiamenti nell'insieme dei medoid.

Scelta automatica del numero di cluster

L'algoritmo K-Medoids necessita di sapere a priori il numero di cluster K in cui suddividere l'insieme di dati.
È stata introdotta una metodologia per il calcolo di K in modo automatico.
Dato un range di possibili valori di K (nel caso in esame da 1 a 10) viene applicato ripetutamente l'algoritmo K-Medoids, settando il numero di cluster pari a K, e calcolando di volta in volta l'SEE come la somma delle distanze dei campioni dal loro *medoid* più vicino.
Come descritto in precedenza per il DBSCAN, plottando queste distanze graficamente si ottiene una curva, e l'obiettivo è trovare il punto di gomito della curva stessa, a cui corrisponderà uno specifico valore di K.
Per una maggior accuratezza, la scelta finale del numero di cluster viene fatta valutando la miglior combinazione di tre indici di bontà del clustering , l'*Average Silhouette Index*, il *Global Silhouette Index* e il *Dunn Index* (descritti nella sezione successiva) calcolati applicando tre volte l'algoritmo K-Medoids utilizzando un numero di cluster pari a K-1, K e K+1.

2.4.2 Visualizzazione grafica

La rappresentazione grafica della suddivisione in cluster eseguita tramite DBSCAN o K-Medoids è mostrata tramite l'utilizzo di scatter-plot tridimensionali, facilmente interpretabili dall'utente.
Viste le grandi dimensioni dei dati in esame, una rappresentazione 3D necessita di tecniche di riduzione dimensionale che preservino comunque il più possibile la struttura dei dati iniziale.
Nel framework proposto sono state utilizzate due tecniche: t-SNE e PCA.

t-SNE

IL t-distributed stochastic neighbor embedding (t-SNE) [22] è un algoritmo di riduzione della dimensionalità non lineare che si presta particolarmente per ridurre dataset ad alta dimensionalità in uno spazio a due o tre dimensioni, rappresentabili

tramite un grafico a dispersione. L'algoritmo modella i punti in modo che oggetti vicini nello spazio originale risultino vicini nello spazio a dimensionalità ridotta, e oggetti lontani risultino lontani, cercando di preservare la struttura locale.
Inizialmente l'algoritmo calcola la probabilità di somiglianza tra due punti x_i e x_j nello spazio ad alta dimensione, calcolando la probabilità condizionata $p_{j|i}$.
In seguito, calcola la probabilità di somiglianza dei relativi punti y_i e y_j nello spazio a bassa dimensione corrispondente, calcolando la probabilità condizionata $q_{j|i}$.
Si cerca quindi di minimizzare la differenza tra queste probabilità condizionali nello spazio dimensionale superiore e in quello inferiore per una perfetta rappresentazione dei punti nello spazio inferiore. Per fare ciò, t-SNE cerca di minimizzare la somma della divergenza di *Kullback-Leibler (KL)* della distribuzione Q rispetto a P, usando un metodo di discesa a gradiente.
L'uso della divergenza di *Kullback-Leibler* consente di avere penalità elevate se punti vicini nello spazio originale vengono considerati lontani nello spazio a dimensionalità ridotta, mentre il viceversa ha un'influenza minore, tendendo quindi a preservare la struttura locale della distribuzione dei punti.
Se i dati in esame sono di dimensioni molto grandi è consigliabile eseguire prima una riduzione dimensionale tramite PCA.

PCA

La Principal Component Analysis (PCA) [17] è una tecnica non-supervisionata utilizzata nel'ambito della statistica multivariata per la semplificazione dei dati d'origine.
Lo scopo primario è la riduzione della dimensione del dataset analizzato tramite una trasformazione lineare delle variabili originarie proiettate in un nuovo sistema cartesiano nel quale le componenti vengono ordinate in base alla varianza in ordine decrescente: pertanto, la variabile con maggiore varianza viene proiettata sul primo asse, la seconda sul secondo asse e così via. Ogni componente segue la direzione lungo la quale i dati hanno maggior varianza, minimizza la distanza quadratica media tra i punti e le loro proiezioni ed è ortogonale alla precedente.
La riduzione della complessità avviene limitandosi ad analizzare le principali com-

ponenti tra le nuove variabili.
Matematicamente, PCA è implementato nel seguendo modo:

- Standardizzare i dati: tutti gli attributi del nostro dataset avranno media pari a 0 e varianza pari a 1;
- Calcolare la matrice di covarianza, cioè una matrice simmetrica che rappresenta la variazione di ogni variabile rispetto alle altre;
- Calcolare gli autovalori e i rispettivi autovettori della matrice. Gli autovettori altro non sono che le nuove componenti cercate e più è grande un autovalore, più importanza avrà il corrispettivo autovettore;
- Ordinare gli autovettori in ordine decrescente di importanza e vedere quanti autovettori servono per garantire una varianza spiegata sufficiente;

Ricerca delle features importanti e rappresentazione tramite Boxplot e Radar Chart

Data la grande quantità di features estratte per ogni split, viene fatta una rappresentazione grafica delle 5 features più importanti tramite *Boxplot* e *Radar Chart*. La ricerca delle top 5 features viene eseguita tramite l'algoritmo Random Forest.

Random Forest
La Random Forest [17] è un algoritmo di apprendimento automatico supervisionato che si basa sul concetto di Bagging, cioè sull'apprendimento di più modelli previsionali per formare un unico modello di previsione più potente.
È una foresta casuale che combina molti alberi decisionali in un unico modello, decorrelando tra di loro gli alberi utilizzando nel decidere gli split all'interno di ciascun albero, solamente un campione casuale di m predittori, dove m solitamente è uguale alla radice quadrata del numero totale di predittori. Poiché ad ogni split consideriamo una selezione nuova di m predittori, la probabilità di sovrapposizione o di correlazione tra gli alberi diminuisce. Questo dovrebbe portare ad una maggiore riduzione della varianza del modello.
È possibile calcolare la frequenza relativa di ciascuna caratteristica nella fase di ad-

destramento del modello, ridimensionando l'importanza di ogni caratteristica in modo che la somma di tutti i punteggi sia 1. Questo punteggio permette di capire le caratteristiche più importanti per la costruzione del modello stesso.

Boxplot

Il diagramma a scatola e baffi [23], altrimenti detto Boxplot, è una visualizzazione usata in statistica per rappresentare la distribuzione di una serie di eventi relativi ad un campione esaminato, che si utilizza per variabili quantitative.
In particolare, il boxplot permette di rappresentare sullo stesso grafico cinque tra le misure di posizione più utilizzate in statistica. La scatola è delimitata dal primo e dal terzo quartile ed è divisa in due parti dalla mediana. I baffi rappresentano la differenza fra i quartili e i valori minimo e massimo. La lunghezza dei baffi mostra la normalità (baffi corti) o eccezionalità (baffi lunghi) dei fenomeni, e i valori anomali che si collocano oltre i baffi stessi.

Radar Chart

Il Radar Chart [24] è un metodo grafico per mostrare dati su variabili multiple in forma di un grafico bidimensionale rappresentate su assi con la stessa origine. La posizione relativa e l'angolo degli assi sono privi di importanza. Questi assi hanno scale diverse in base al valore minimo e al valore massimo dell'attributo rappresentato su di esso e i punti sui diversi assi sono uniti con segmenti, in modo che il grafico abbia la forma di una stella o di una ragnatela, come in figura 2.11. Questo tipo di grafico viene spesso utilizzato per l'immediatezza con cui si possono confrontare n-uple di valori relativi ad osservazioni diverse.

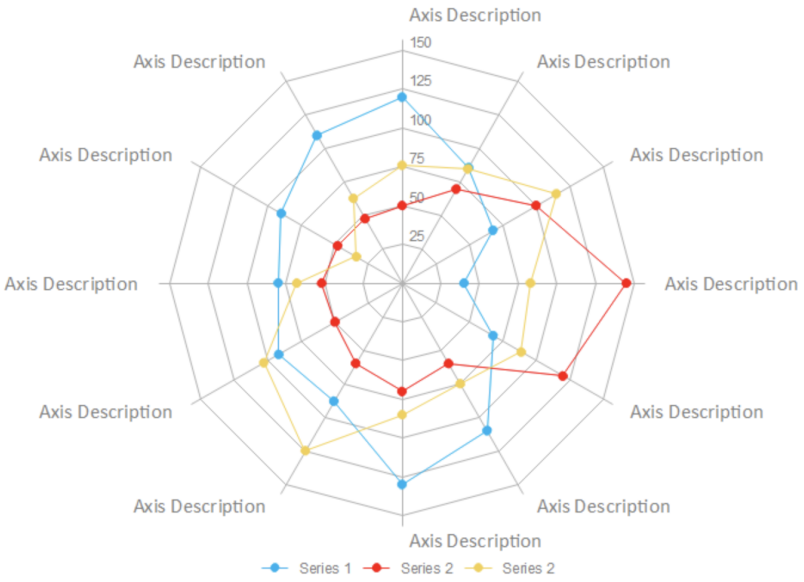


Figura 2.11: *Radar Chart*

Capitolo 3

Architettura e Workflow

In questo capitolo descriverò l'estensione del framework ADESCA, creato e descritto da Paolo Bethaz all'interno del suo progetto di tesi "Automated Data Exploration To Discover Structures And Models Hidden In The Data" per l'analisi dei dati automatizzata.
Il framework proposto permette di automatizzare il processo di analisi di uno specifico dataset, minimizzando il contributo dell'esperto di dati. Questo processo è suddiviso in identificazione del tipo di dato, caratterizzazione dei dati, rimozione degli outliers, clustering, classificazione, trasformazione e visualizzazione dei dati.
Lo scopo dell'estensione da me realizzata è adattare ADESCA al contesto dell'Industria 4.0, permettendo l'analisi automatica di dataset contenenti delle serie temporali, applicando la metodologia descritta in precedenza.

3.1 Selezione del dataset

All'apertura, il programma si presenta con una pagina iniziale in cui l'utente può selezionare il dataset che vuole esaminare, scegliendolo da una qualsiasi directory presente nel proprio computer, sia in formato csv. sia in formato xls.

Figura 3.1: *Upload del dataset*

3.2 Visualizzazione del dataset

ADESCA è in grado di svolgere un'analisi automatizzata ed esaustiva del dataset fornito dall'utente. Lo scopo dell'estensione da me realizzata è permettere l'analisi automatizzata dei dati raccolti nell'Industria 4.0.
Di conseguenza, prima di visualizzare il dataset selezionato in formato tabulare, il framework è in grado di capire se esso contiene delle serie temporali. Questo riconoscimento è eseguito tramite la ricerca di un attributo temporale (timestamp) e la ricerca di un attributo contenente un vettore numerico di grosse dimensioni (time-series).
Se il dataset non contiene delle serie temporali, il framework proporrà all'utente un'analisi dei dati standard, come da progetto iniziale.
Se invece il dataset contiene delle serie temporali, il framework mostrerà all'utente i dati in formato tabulare, informandolo della presenza di timeseries e proponendo

due metodologie d'analisi: *Feature based analysis* e *Time-series based analysis*.

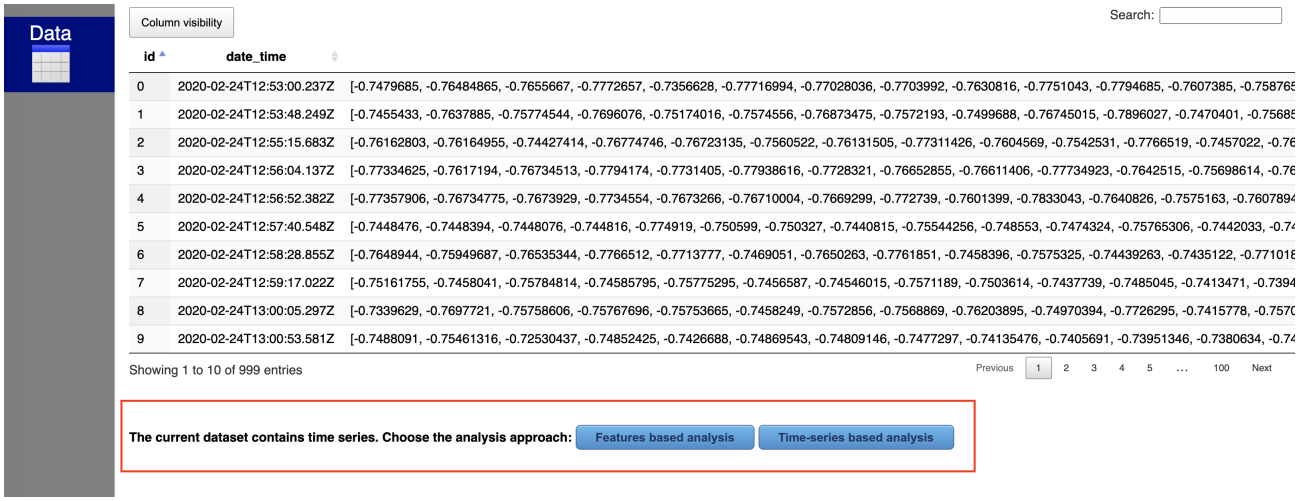


Figura 3.2: *Visualizzazione del dataset - Rilevata la presenza di serie temporali*

3.3 Features based Analysis

Se l'utente decide di procedere con la *Features based Analysis* viene attuata la metodologia per l'estrazione delle features principali.

3.3.1 Scelta del numero di split

Il primo passo è la scelta degli split. Il programma mostrerà all'utente il grafico contenente l'andamento nel tempo della prima time series presente nel dataset, supponendo che anche le altre serie temporali abbiano, in linea generale, la stessa rappresentazione.
Il framework propone all'utente una doppia scelta: automatico o manuale.
All'apertura di questa nuova pagina, la modalità predefinita per la scelta degli split è automatica. Di conseguenza sul grafico vengono mostrati degli assi verticali rossi che permettono di visualizzare la suddivisione del segnale, in base al calcolo automatico del numero di split effettuato, come spiegato in precedenza. Essendo in modalità automatica, non è possibile modificare la dimensione di questi blocchi. Nell'esempio in figura 3.3 è mostrato l'andamento di una serie temporale e la divi-

sione automatica in 5 split della stessa dimensione.
Inoltre nella parte inferiore sono mostrati: a sinistra una tabella teorica con una breve descrizione degli indici presenti, con il range dei possibili valori e se lo scopo è massimizzare o minimizzare l'indice considerato. In questo modo l'utente può farsi un'idea sulla bontà degli indici visualizzati; a destra sono presenti i 5 indici ottenuti tramite la scelta degli split automatica.

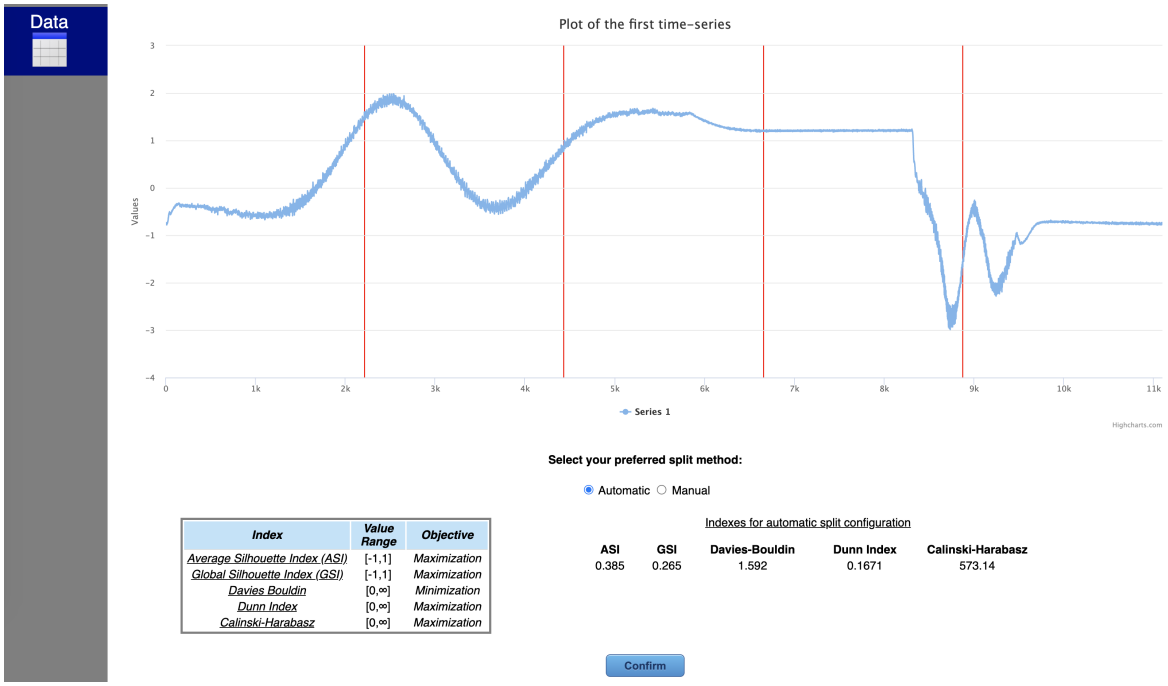


Figura 3.3: *Visualizzazione della prima serie temporale con divisione in split automatica*

Per gli utenti più esperti è disponibile una modalità manuale. Osservando l'andamento del segnale, l'utente può decidere di cambiare il numero di split per suddividere il segnale nella maniera che ritiene più corretta per la propria analisi. Cliccando sulla modalità 'manual', tramite un'apposita tendina a scorrimento è possibile scegliere il numeri di split in un intervallo da 1 a 24. Cambiando questo valore il grafico mostrerà tante linee verticali rosse equidistanziate in base al numero di blocchi scelti. É inoltre possibile spostare manualmente ogni singolo asse verticale rosso a destra e a sinistra lungo l'asse orizzontale, in modo da dividere il segnale nelle porzioni desiderate. In figura 3.4 è stato scelto un numero di split pari a 8 e inoltre gli assi sono stati spostasti nelle posizioni scelte dall'utente.

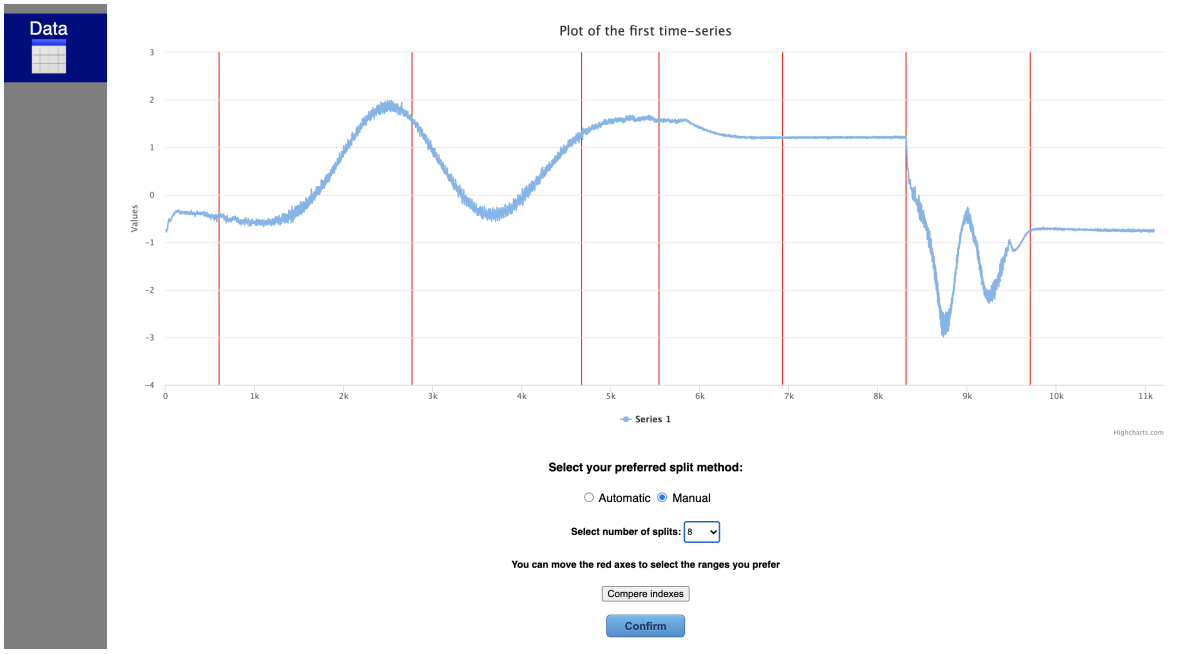


Figura 3.4: *Configurazione manuale per la scelta degli split*

Cliccando *Compare indexes* verrà mostrato un confronto tra gli indici di bontà del clustering ottenuti dalla suddivisione in split calcolati in modalità automatica e la configurazione attuale decisa dall'utente in modalità manuale. La sezione si presenta all'utente come in figura 3.5. Analizzando questi valori l'utente può capire le differenze tra le due soluzioni e decidere se tornare alla configurazione automatica, provare manualmente un'altra configurazione o mantenere la scelta fatta.

Index	Value Range	Objective
Average Silhouette Index (ASI)	[-1,1]	Maximization
Global Silhouette Index (GSI)	[-1,1]	Maximization
Davies Bouldin	[0,∞]	Minimization
Dunn Index	[0,∞]	Maximization
Calinski-Harabasz	[0,∞]	Maximization

Indexes for automatic split configuration

ASI	GSI	Davies-Bouldin	Dunn Index	Calinski-Harabasz
0.385	0.265	1.592	0.1671	573.14

Indexes for manual split configuration

ASI	GSI	Davies-Bouldin	Dunn Index	Calinski-Harabasz
0.222	0.187	2.279	0.2242	154.06

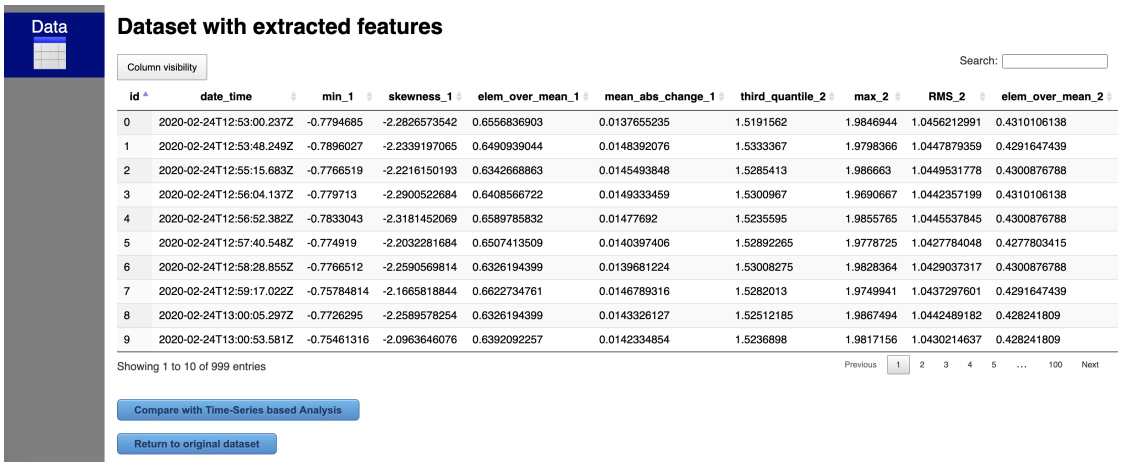
Confirm

Figura 3.5: *Confronto tra configurazione automatica e manuale*

Eseguita la scelta e il posizionamento degli assi, l'utente può premere il bottone di conferma per consentire al framework di eseguire il calcolo degli Smart Data.

3.3.2 Calcolo e visualizzazione degli Smart Data

Dopo aver ricevuto la posizione degli assi rossi al momento della conferma, il framework suddivide ogni serie temporale in base agli split scelti e calcola per ognuno di esse le 14 features descritte in precedenza, ottenendo un nuovo insieme di dati.
Per ridurre ulteriormente le dimensioni del dataset ottenuto, viene svolta una estrazione delle features più significative, calcolando la matrice di correlazione tra le varie features e mantenendo solo quelle colonne in cui la media dei valori assoluti risultano inferiori ad una certa soglia. Il valore della soglia è variabile, in quanto viene ridotta finché il dataset ottenuto non abbia un numero di colonne inferiori a 100, numero ritenuto ragionevole per avere un insieme di dati che contenga ancora molte informazioni, ma non di dimensioni esagerate. Se inizialmente il dataset ottenuto conteneva meno di 100 colonne allora l'estrazione delle features più significative non viene eseguita.
Si ottiene così il dataset trasformato contenente gli Smart Data, più facile da comprendere in modo visivo e più maneggevole, che viene mostrato all'utente in formato tabulare, come in figura 3.6.



id	date_time	min_1	skewness_1	elem_over_mean_1	mean_abs_change_1	third_quantile_2	max_2	RMS_2	elem_over_mean_2
0	2020-02-24T12:53:00.237Z	-0.7794685	-2.2826573542	0.6556836903	0.0137655235	1.5191562	1.9846944	1.0456212991	0.4310106138
1	2020-02-24T12:53:48.249Z	-0.7896027	-2.2339197065	0.6490939044	0.0148392076	1.5333367	1.9798366	1.0447879359	0.4291647439
2	2020-02-24T12:55:15.683Z	-0.7766519	-2.2216150193	0.6342668863	0.0145493848	1.5285413	1.986663	1.0449531778	0.4300876788
3	2020-02-24T12:56:04.137Z	-0.779713	-2.2900522684	0.6408566722	0.0149333459	1.5300967	1.9690667	1.0442357199	0.4310106138
4	2020-02-24T12:56:52.382Z	-0.7833043	-2.3181452069	0.6589785832	0.01477692	1.5235595	1.9855765	1.0445537845	0.4300876788
5	2020-02-24T12:57:40.548Z	-0.774919	-2.2032281684	0.6507413509	0.0140397406	1.52892265	1.9778725	1.0427784048	0.4277803415
6	2020-02-24T12:58:28.855Z	-0.7766512	-2.2590569814	0.6326194399	0.0139681224	1.53008275	1.9828364	1.0429037317	0.4300876788
7	2020-02-24T12:59:17.022Z	-0.75784814	-2.1665818844	0.6622734761	0.0146789316	1.5282013	1.9749941	1.0437297601	0.4291647439
8	2020-02-24T13:00:05.297Z	-0.7726295	-2.2589578254	0.6326194399	0.0143326127	1.52512185	1.9867494	1.0442489182	0.428241809
9	2020-02-24T13:00:53.581Z	-0.75461316	-2.0963646076	0.6392092257	0.0142334854	1.5236898	1.9817156	1.0430214637	0.428241809

Figura 3.6: *Visualizzazione del dataset con gli Smart Data*

Una volta ottenuto il dataset trasformato contenente le feautures, il framework propone sia tornare alla configurazione precedente, ritornando al formato sotto forma di serie temporali, sia eseguire un confronto tra le due metodologie proposte.

3.4 Time-series based Analysis

Se la scelta iniziale ricade su *Time-series based analysis* il framework calcola la matrice delle distanze tra tutte le serie temporali presenti nel dataset, tramite l'applicazione dell'algoritmo BOSS. Viene mostrata all'utente una nuova pagina, come in figura 3.7, che presenta in alto la tabella contenente una piccola descrizione degli indici valutati mentre successivamente è divisa in due colonne: a sinistra sono mostrati i risultati di clustering ottenuti tramite DBSCAN e a destra quelli ottenuti tramite K-Medoids, entrambi basati sulla matrice delle distanze. In basso è presente un bottone che permette il confronto tra i risultati ottenuti con *Time-series based Analysis* e quelli ottenuti con *Features based Analysis*, in questo caso con tutti i parametri scelti automaticamente.

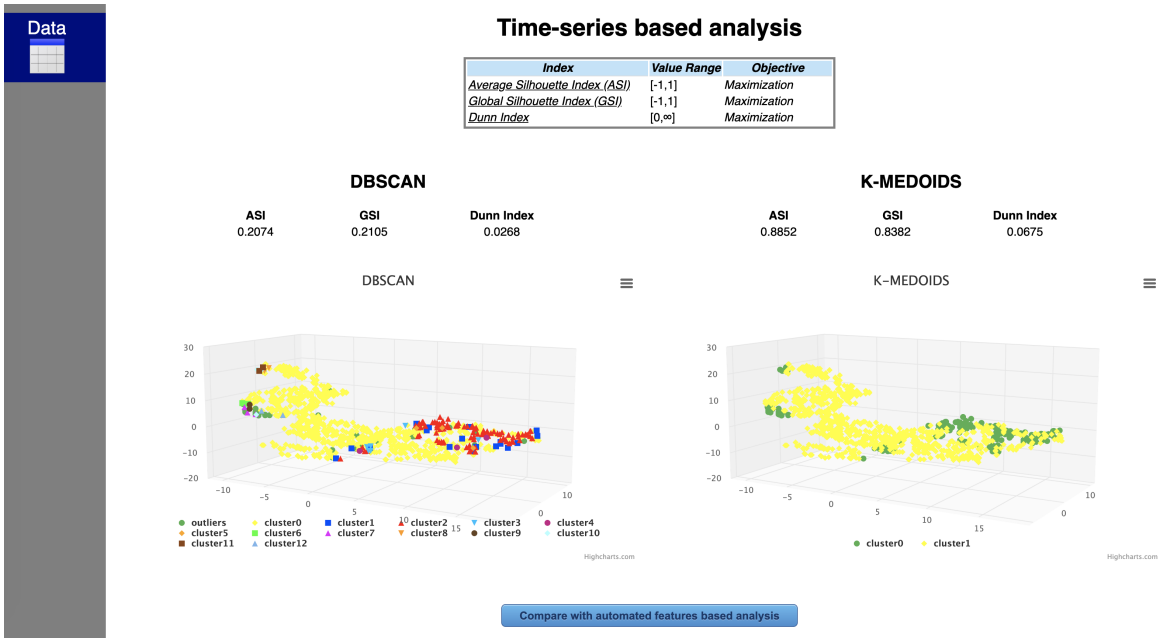


Figura 3.7: *Clustering ottenuto con l'approccio Time-series based*

3.5 Confronto tra dataset

Qualunque sia la scelta iniziale, è sempre possibile eseguire un confronto con l'altro approccio. Se è stato scelto l'approccio *Features based analysis*, il confronto viene fatto tra i risultati di clustering ottenuti dal dataset contenente le features in base alla

scelta degli split fatta dall'utente e i risultati di clustering ottenuti tramite la matrice delle distanze calcolate tramite BOSS. Se, invece, la scelta iniziale è stata *Time-series based analysis* il confronto è lo stesso, ma per la creazione del dataset con le features viene utilizzata la configurazione automatica, visto che in questo caso l'utente non può prendere nessun tipo di decisione. La pagina di confronto mostra all'utente una schermata costituita da due colonne strutturate allo stesso modo per un confronto visivo tra le due metodologie di facile interpretazione. A sinistra è presentato l'approccio *Features based analysis* e a destra l'approccio *Time-series based analysis*, come in figura 3.8.

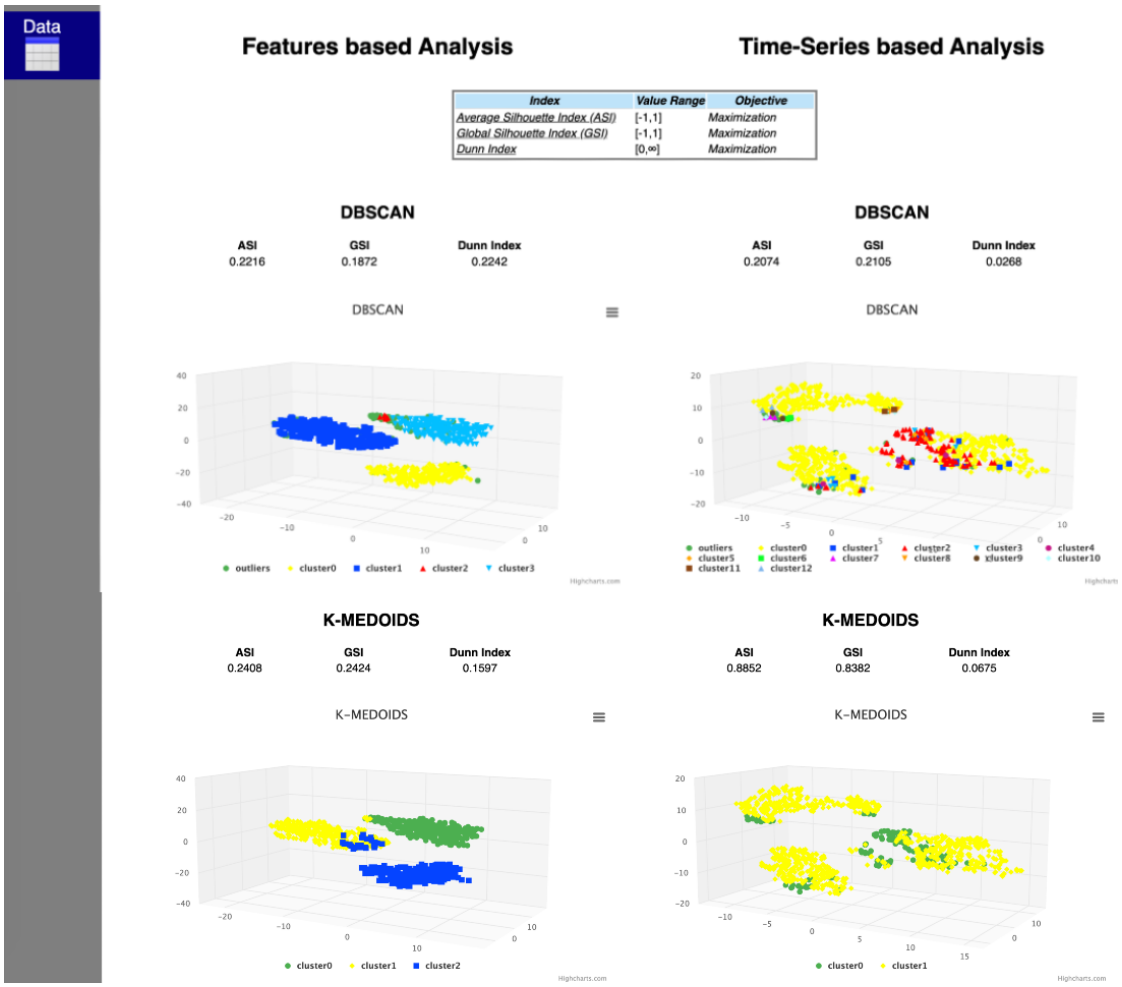


Figura 3.8: *Confronto tra le due metodologie di analisi di serie temporali*

In entrambi i casi è possibile visualizzare i risultati dell'applicazione dei due algoritmi di clustering, il DBSCAN e il K-Medoids, tramite il calcolo di tre indici di

bontà, l'*Average Silhouette Index*, il *Global Silhouette Index* e il *Dunn Index*, e una rappresentazione grafica dei cluster tramite scatter plot tridimensionali, ottenuti dalla combinazione di PCA e TSNE sui dati.
Gli indici di bontà del clustering permettono la valutazione del clustering ottenuto nei vari casi, mentre la rappresentazione grafica permette di capire il numero di cluster creati e la loro distribuzione.
Nell'approccio *Features based analysis*, cliccando su un qualsiasi cluster presente nei grafici, si apre una finestra pop-up che mostra all'utente delle informazioni aggiuntive sul clustering ottenuto. In particolare, viene raffigurato in alto un grafico contenente i boxplot delle 5 features più importanti, valutate secondo l'algoritmo Random Forest, come in figura 3.9. Il grafico è suddiviso in split a seconda della scelta iniziale fatta dall'utente e i boxplot sono posizionati nello split di riferimento della features che rappresentano. Questo permette di capire da quale divisione del segnale iniziale è stata estratta ognuna delle features ritenute più importanti.

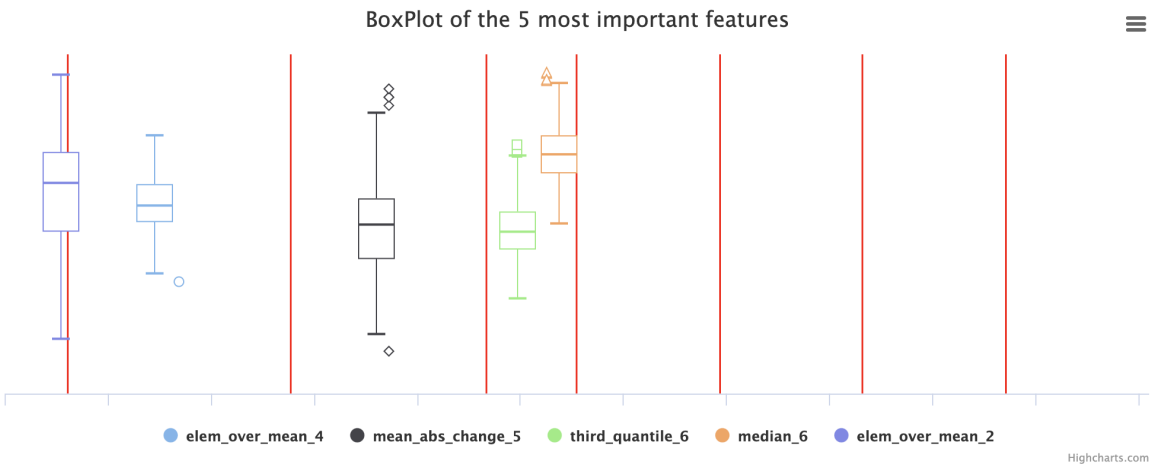


Figura 3.9: *Boxplot delle 5 features più importanti*

Nella parte sottostante viene fatta una caratterizzazione del cluster cliccato dall'utente, come in figura 3.10. Viene quindi rappresentata la stessa tipologia di grafico appena descritta, ma i boxplot presenti fanno riferimento solo ai dati appartenenti al cluster selezionato.

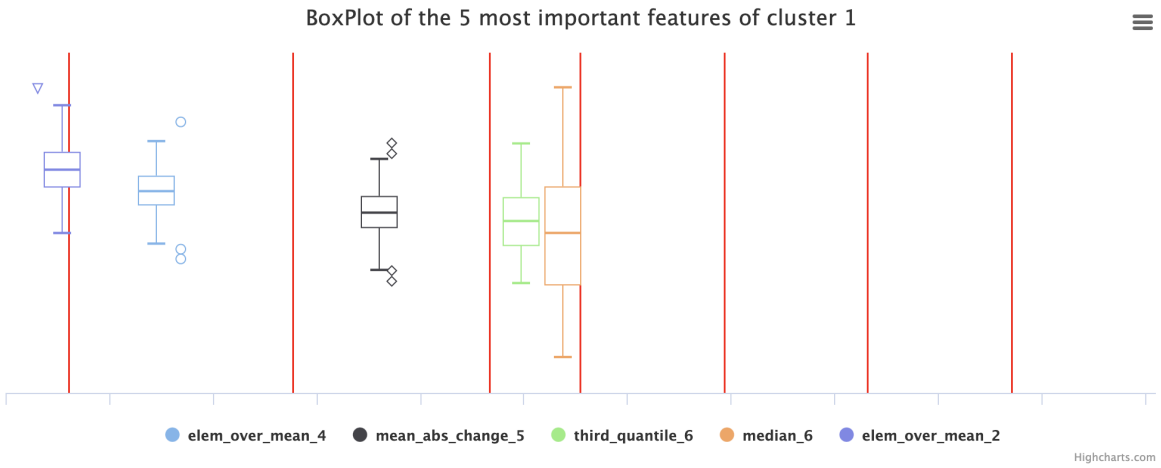


Figura 3.10: *Caratterizzazione cluster cliccato dall'utente*

Per evidenziare le differenze tra i vari cluster viene inoltre utilizzato un Radar Chart che raffigura i valori delle top 5 features più importanti. Il cluster selezionato è di colore blu, mentre gli altri cluster sono disegnati in rosso.

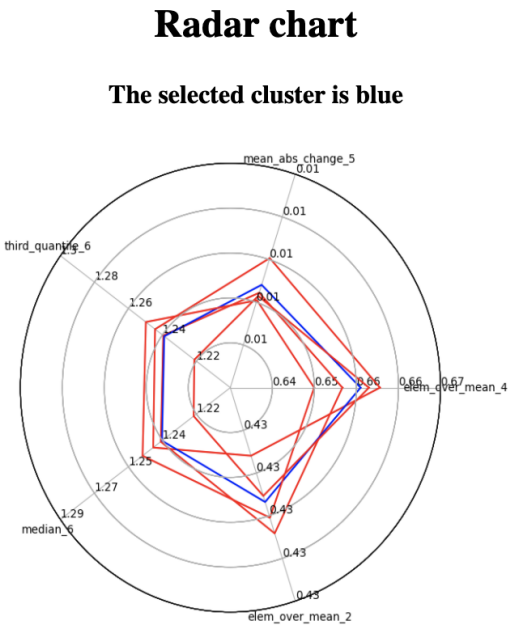


Figura 3.11: *Radar chart delle top 5 features*

Anche nell'approccio *Time-series based analysis*, cliccando su un qualsiasi cluster raffigurato tramite scatterplot, si apre una finestra pop-up che contiene due grafici, come in figura 3.12 e 3.13. Il primo grafico mostra il diverso andamento dei centroi-

di dei cluster presenti, calcolati come la media di tutti i valori delle serie temporali appartenenti ai cluster di riferimento. Il secondo grafico invece permette di osservare l'andamento nel tempo del centroide del cluster scelto dall'utente e una fascia colorata data dal range di valori ottenuta sommando e sottraendo da ogni valore medio la deviazione standard.

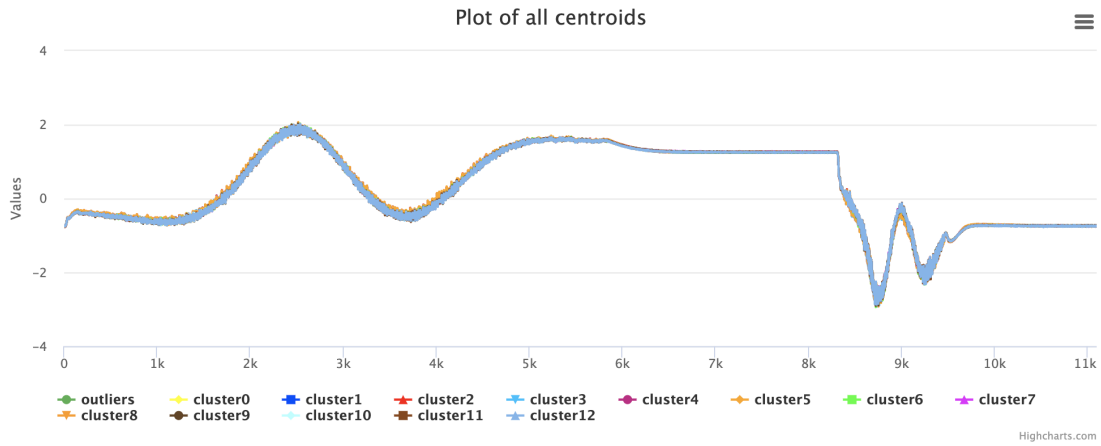


Figura 3.12: *Centroidi cluster ottenuti con approccio Time-series based analysis*

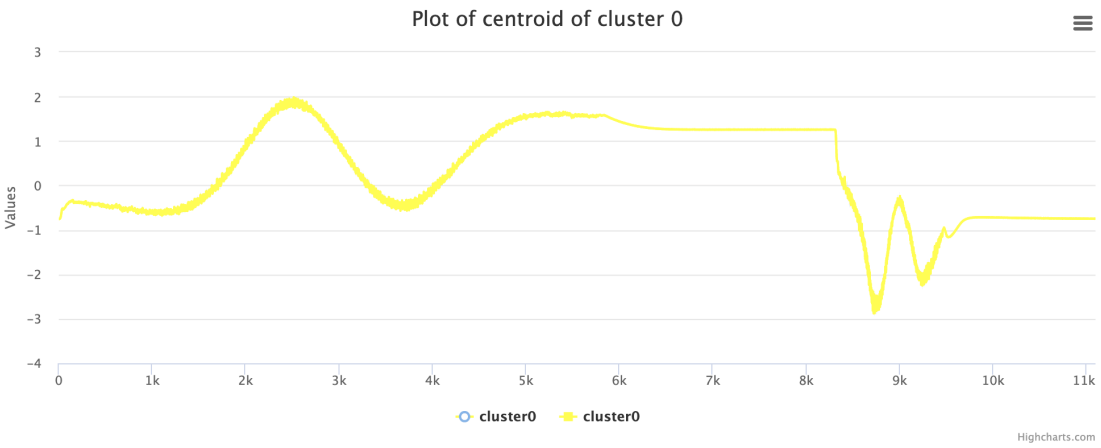


Figura 3.13: *Centroide cluster selezionato con approccio Time-series based analysis*

Capitolo 4

Casi d'uso e risultati

In questo capitolo mostrerò l'applicazione dell'estensione del framework realizzata con 3 casi d'uso proposti nel progetto di ricerca europeo SERENA [25] nell'ambito industriale 4.0 e verranno rappresentati i diversi risultati ottenuti.
I 3 dataset esaminati contengono delle serie temporali univariate, cioè relative ad un unico sensore e quindi ad un'unica misurazione.

4.1 1° dataset

Il primo insieme di dati in esame è fornito da un'azienda incentrata sulla produzione di bracci di traino per rimorchi. Il macchinario interessato in questo processo è un laminatoio che presenta dei rivestimenti resistenti all'usura. Più prodotti vengono lavorati dalla macchina del laminatoio, maggiore è l'usura del rivestimento.
Un secondo macchinario è utilizzato per misurare le dimensioni di ciascun prodotto uscito dal laminatoio e valutarne la loro precisione in base all'usura del macchinario. Le dimensioni del macchinario sono rappresentate come due serie temporali x e y che, combinate tra di loro, formano un insieme di coordinate puntuali su di un piano. Siccome le serie temporali x sono tutte uguali (le misurazioni sono prese sempre negli stessi valori di ascissa), abbiamo utilizzato le serie temporali y.
Il dataset è composto da 5232 righe, ognuna delle quali è rappresentata da uno specifico timestamp, ed è suddiviso in tre attributi: *ID*, *TS_mes* e *y_mes*.
L'attributo *TS_mes* contiene i timestamp, composti dalla data e dall'orario in cui so-

no state rilevate le misure. L'attributo *y_mes* contiene le serie temporali relative ad ogni timestamp, di lunghezza minima di 1271 elementi.
Inizialmente il framework mostra l'insieme di dati caricato, come in figura 4.1.

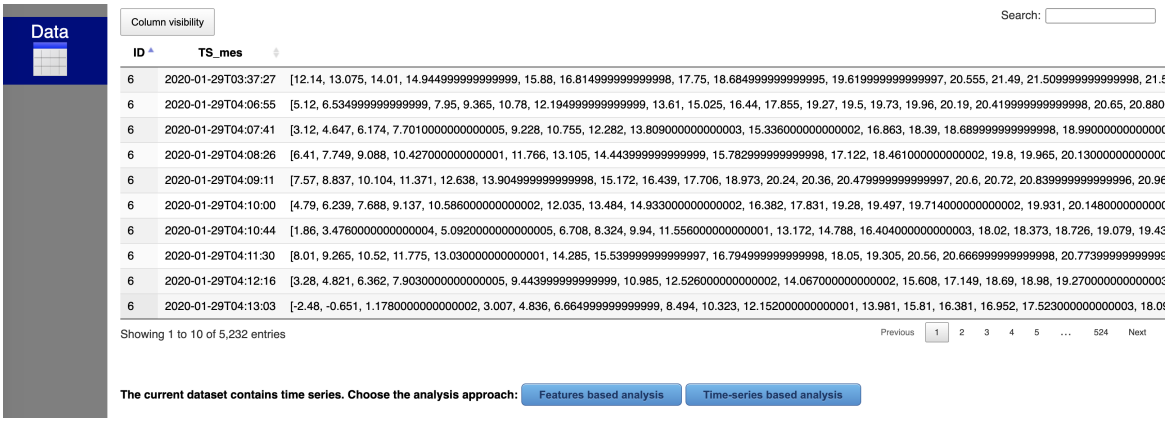


Figura 4.1: *Apertura del dataset contenente le serie temporali*

Il programma rileva la presenza di serie temporali e propone all'utente i due approcci per l'analisi. Cliccando su *Time-Series based analysis* viene mostrata la schermata seguente:

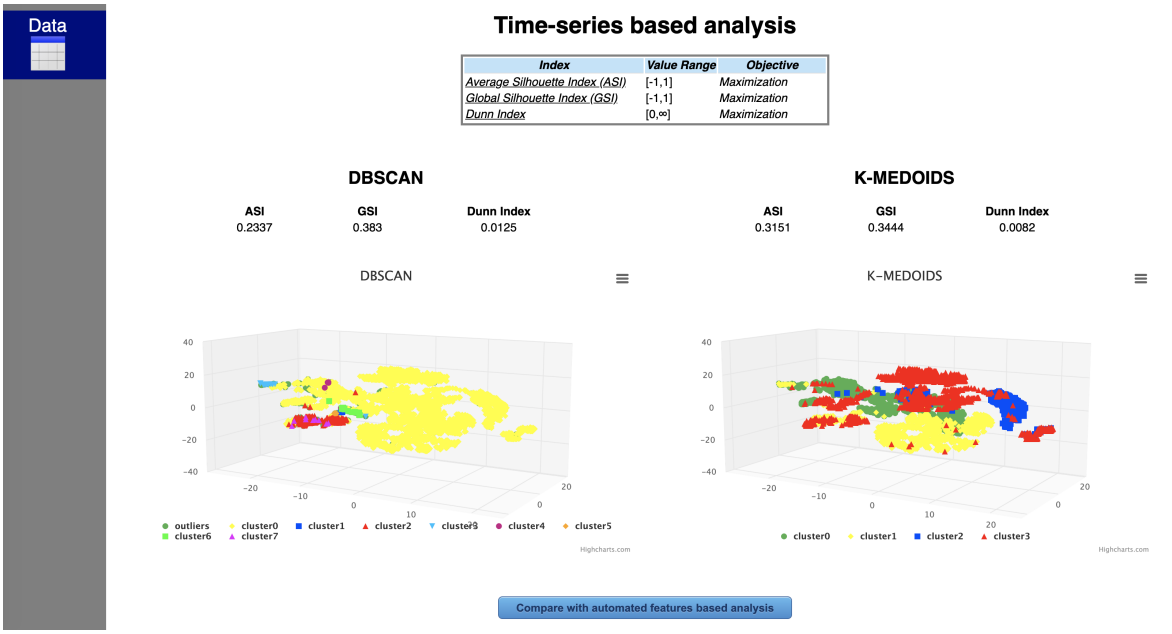


Figura 4.2: *Risultati approccio 'Time-series based'*

Scegliendo invece l'altro approccio, cioè *Features based analysis*, viene invece mostrata la pagina per la scelta degli split:

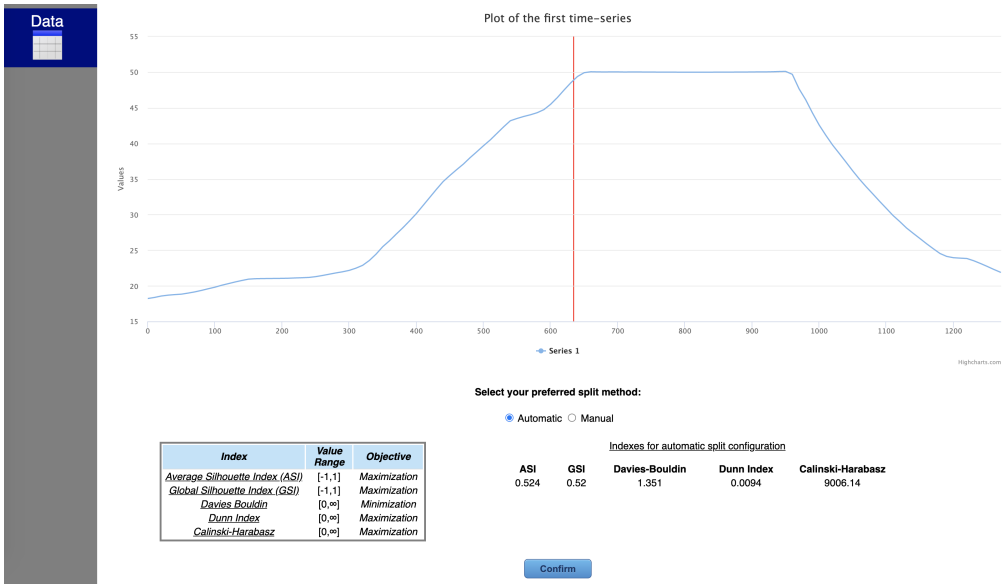


Figura 4.3: *Scelta split: modalità automatica*

Inizialmente viene proposta la soluzione con il numero di split pari a 2, calcolati automaticamente. Un esempio di scelta manuale viene mostrata in figura 4.4:

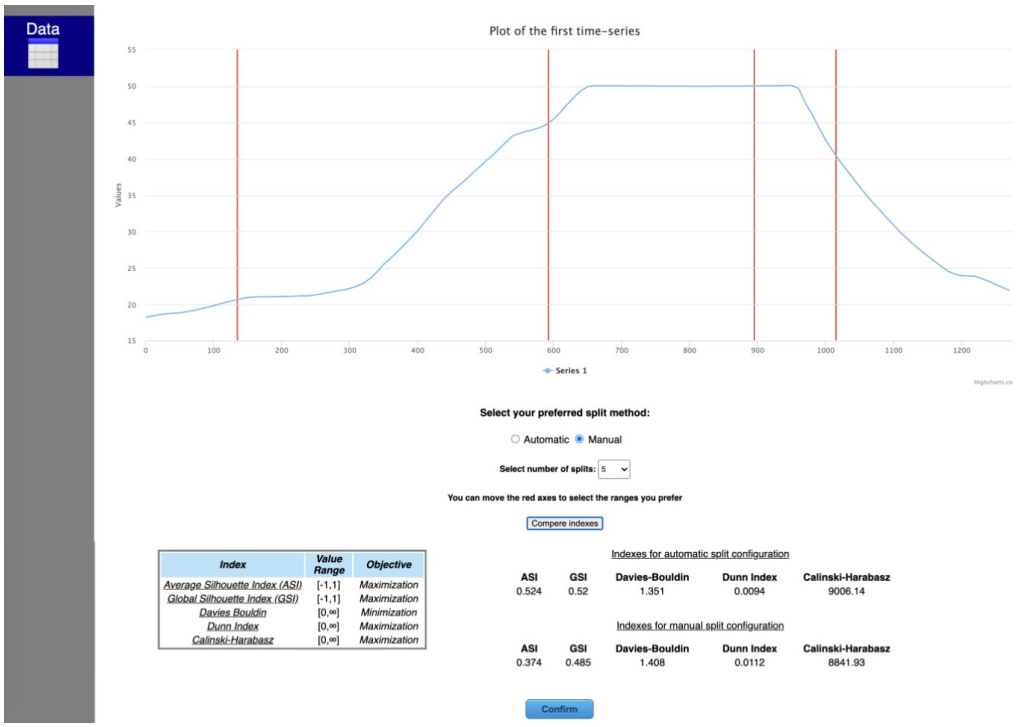
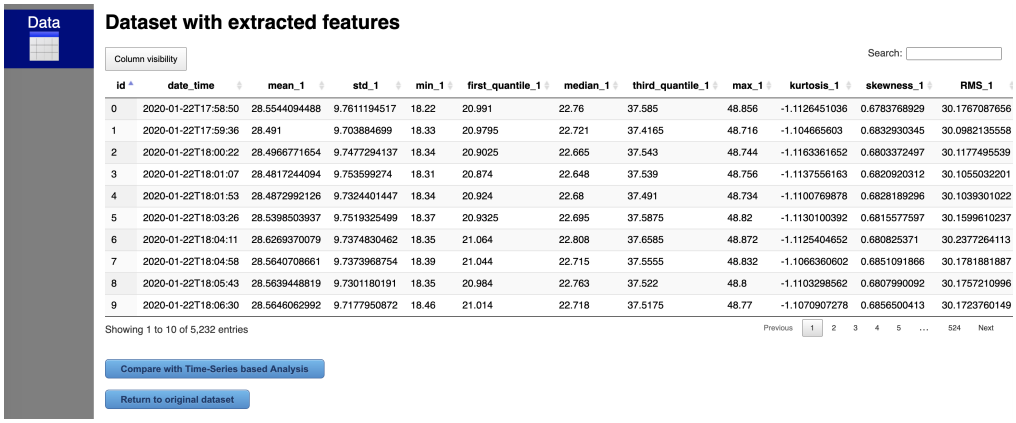


Figura 4.4: *Scelta split: modalità manuale*

Visto che 4 indici su 5 sono peggiori rispetto ai precedenti, la scelta è ricaduta sulla modalità automatica.

Cliccando su *Confirm* viene creato e visualizzato il dataset contente le features estratte. Il dataset presenta l'attributo *date_time*, contente gli stessi timestamp iniziali, seguito dalle 28 features estratte dal primo e dal secondo split.



id	date_time	mean_1	std_1	min_1	first_quantile_1	median_1	third_quantile_1	max_1	kurtosis_1	skewness_1	RMS_1
0	2020-01-22T17:58:50	28.5544094488	9.7611194517	18.22	20.991	22.76	37.585	48.856	-1.1126451036	0.6783768929	30.1767087656
1	2020-01-22T17:59:36	28.491	9.703884699	18.33	20.9795	22.721	37.4165	48.716	-1.104665603	0.6832930345	30.0982135558
2	2020-01-22T18:00:22	28.4966771654	9.7477294137	18.34	20.9025	22.665	37.543	48.744	-1.1163361652	0.6803372497	30.1177495539
3	2020-01-22T18:01:07	28.4817244094	9.753599274	18.31	20.874	22.648	37.539	48.756	-1.1137556163	0.6820920312	30.1055032201
4	2020-01-22T18:01:53	28.4872992126	9.7324401447	18.34	20.924	22.68	37.491	48.734	-1.1100769878	0.6828189296	30.1039301022
5	2020-01-22T18:03:26	28.5398503937	9.7519325499	18.37	20.9325	22.695	37.5875	48.82	-1.1130100392	0.6815577597	30.1599610237
6	2020-01-22T18:04:11	28.6269370079	9.7374830462	18.35	21.064	22.808	37.6585	48.872	-1.1125404652	0.680825371	30.2377264113
7	2020-01-22T18:04:58	28.5640708661	9.7373968754	18.39	21.044	22.715	37.5555	48.832	-1.1066360602	0.6851091866	30.1781881887
8	2020-01-22T18:05:43	28.5639448819	9.7301180191	18.35	20.984	22.763	37.522	48.8	-1.1103298562	0.6807990092	30.1757210996
9	2020-01-22T18:06:30	28.5646062992	9.7177950872	18.46	21.014	22.718	37.5175	48.77	-1.1070907278	0.6856500413	30.1723760149

Figura 4.5: *Dataset contenente gli Smart Data estratti*

In figura 4.6 è mostrato il confronto tra i due approcci.

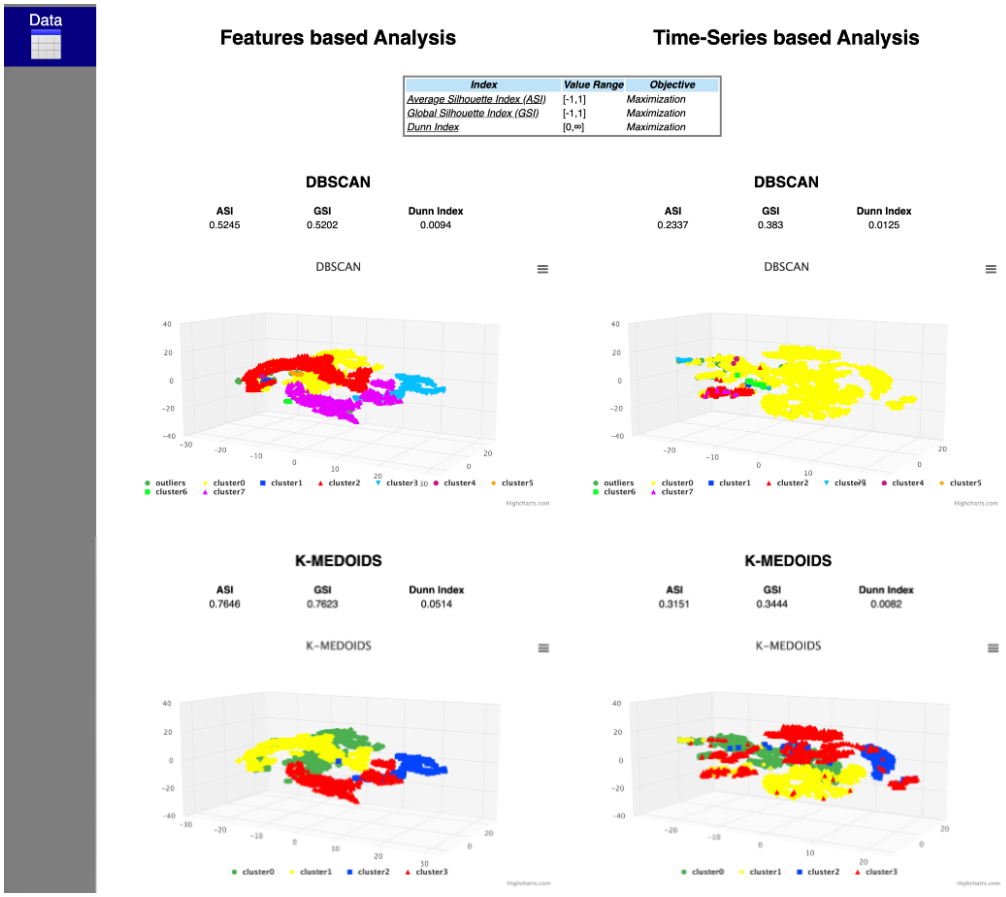


Figura 4.6: *Confronto tra approccio 'Features based' e 'Time-series based'*

Cliccando su un qualsiasi cluster presente nei grafici basati sul dataset contenenti le features, ad esempio il cluster 2, viene aperta una finestra con i seguenti grafici:

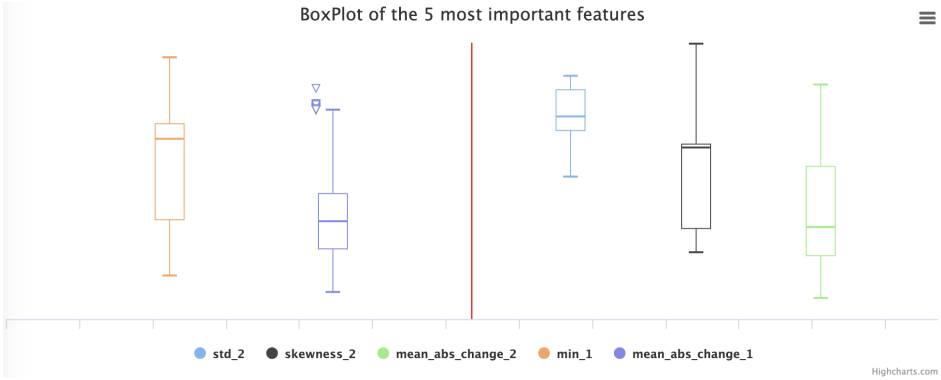


Figura 4.7: *Boxplot delle top 5 features riferiti a tutto il dataset*

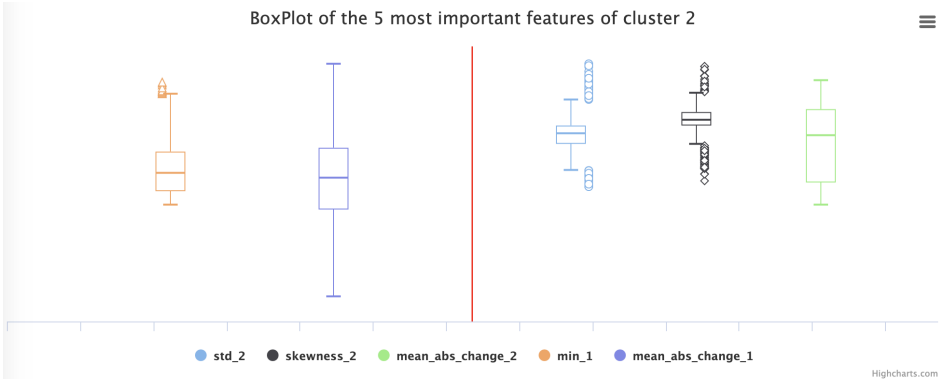


Figura 4.8: *Boxplot delle top 5 features riferiti al cluster 2*

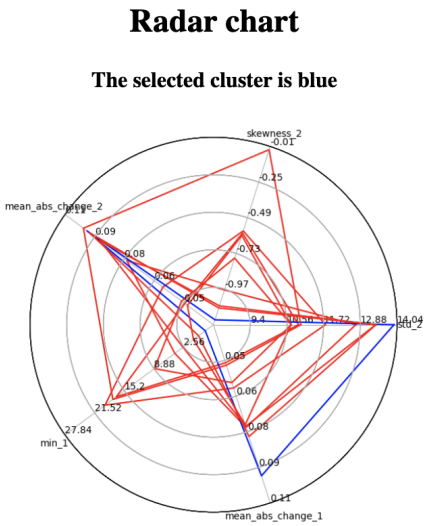


Figura 4.9: *Radar chart delle top 5 features: in blu quelle riferite al cluster 2*

Cliccando invece su un qualsiasi cluster presente nei grafici basati sull'approccio *Time-series based*, ad esempio il cluster 0, viene aperta una finestra con i seguenti grafici:

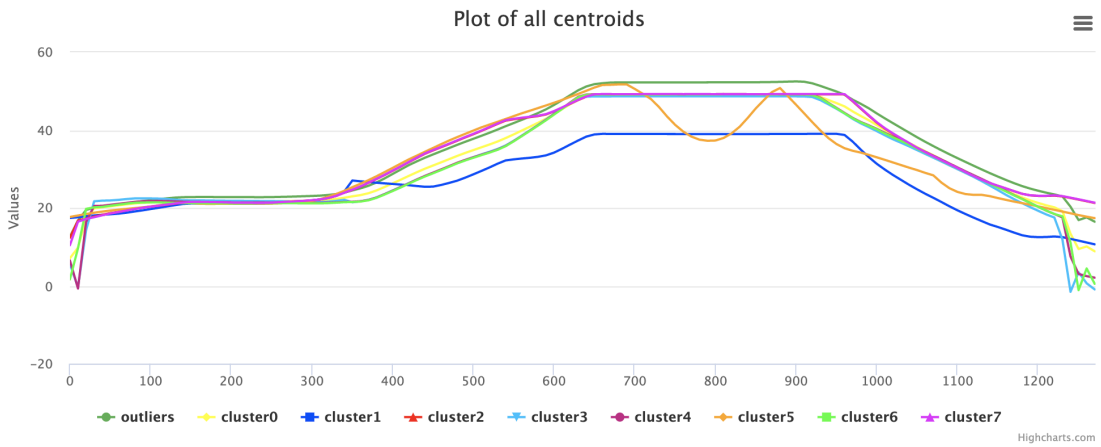


Figura 4.10: *Rappresentazione dell'andamento di tutti i centroidi*

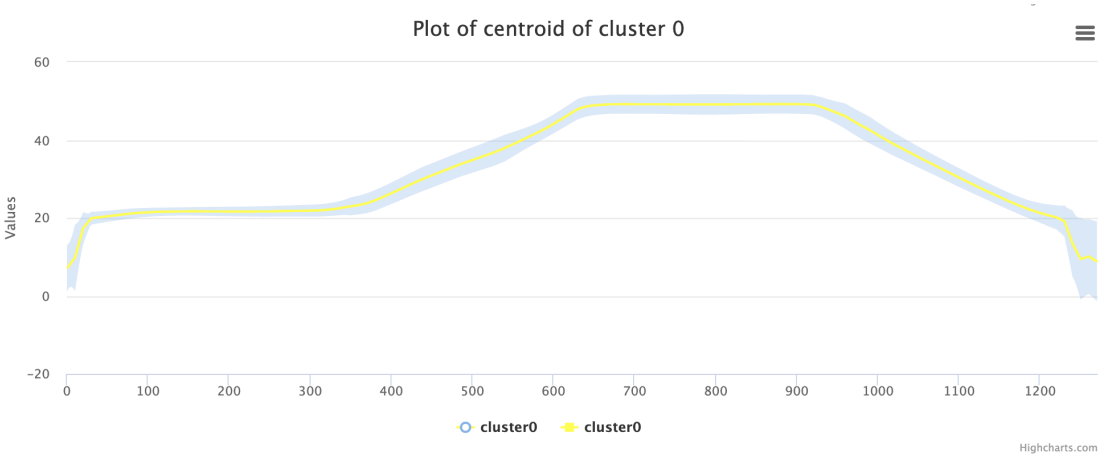


Figura 4.11: *Rappresentazione dell'andamento del centroide del cluster 0*

4.2 2° dataset

Il secondo caso di studio riguarda il monitoraggio di un robot di un'importante azienda di fama internazionale e leader mondiale nel campo dell'automazione. I dati raccolti dai sensori riguardano valori di corrente consumata dal robot durante numerosi cicli di lavorazione. Sulla base di questi valori di corrente ciascun ciclo produttivo è etichettato con una certa label.
Il dataset contiene 1000 serie temporali di lunghezza minima pari a 11096 elementi. È composto da tre attributi: *date_time*, *data_value* e *class*. L'attributo *date_time* contiene i timestamp, composti dalla data e dall'orario in cui sono state rilevate le misure. L'attributo *data_value* contiene le serie temporali relative ad ogni timestamp, che in questo caso si riferiscono a misurazioni di corrente, in Ampere. Infine abbiamo l'attributo *class* che permette la suddivisione in tre classi numeriche: 0, 10, 15.
Nella sequenza di immagini successive sono mostrate le diverse schermate mostrate all'utente, seguendo la stessa procedura descritta nel caso precedente:

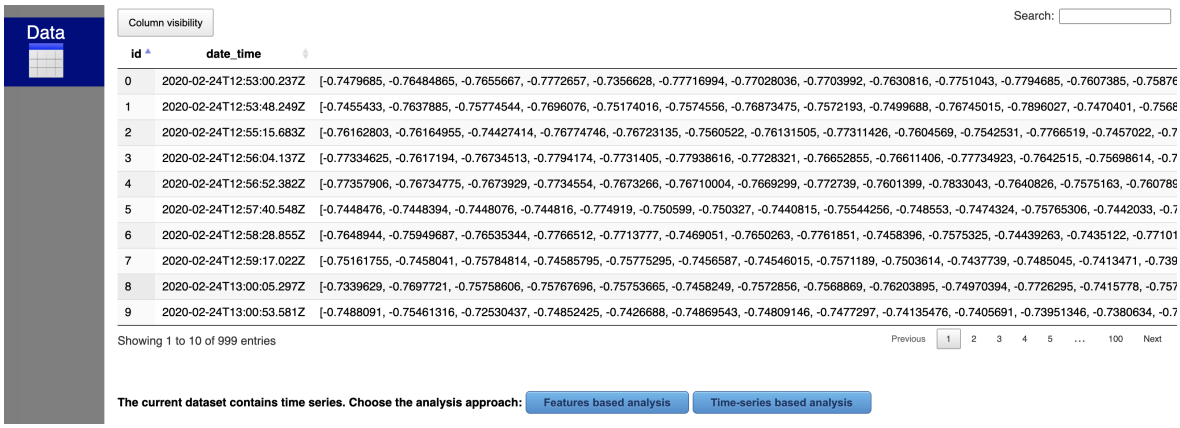


Figura 4.12: *Apertura del dataset contenente le serie temporali*

Approccio *Time-series based*:

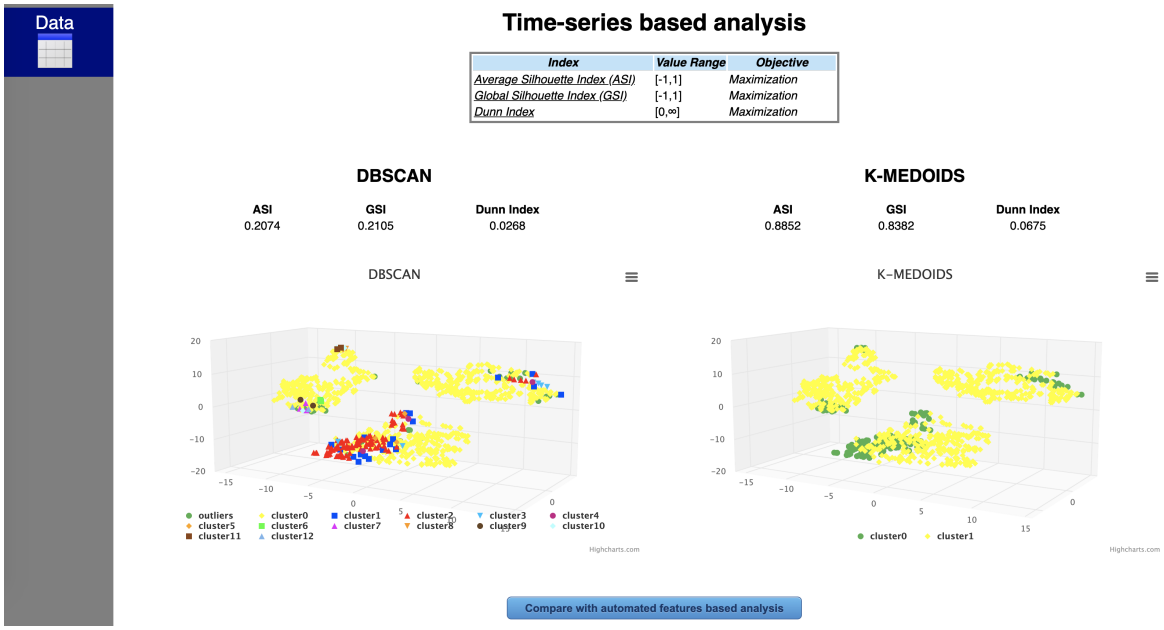


Figura 4.13: *Risultati approccio 'Time-series based'*

Approccio *Features based*:

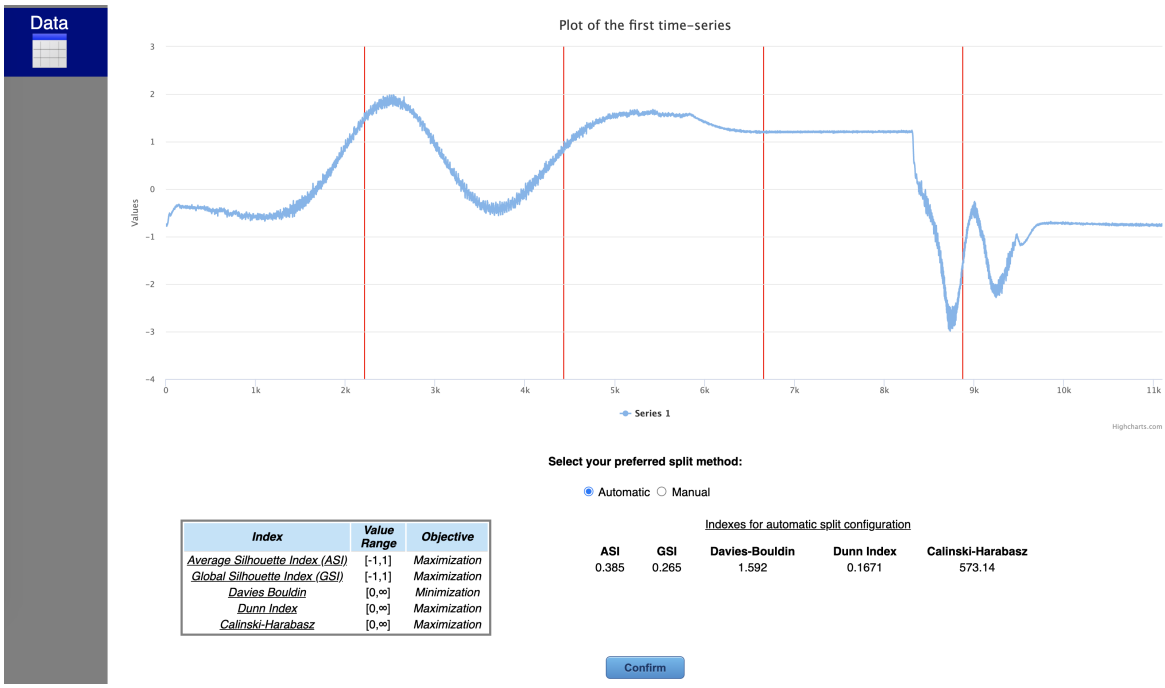
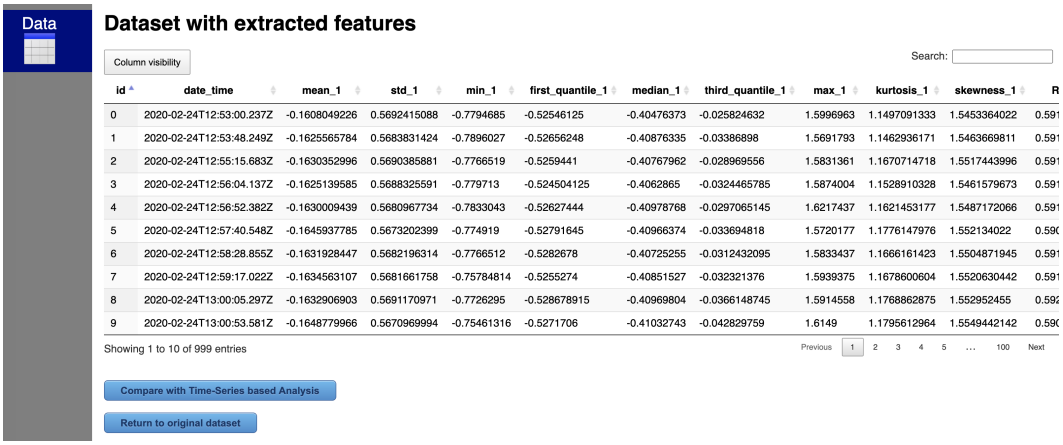


Figura 4.14: *Scelta split: modalità automatica*



id	date_time	mean_1	std_1	min_1	first_quantile_1	median_1	third_quantile_1	max_1	kurtosis_1	skewness_1	R
0	2020-02-24T12:53:00.237Z	-0.1608049226	0.5692415088	-0.7794685	-0.52546125	-0.40476373	-0.025824632	1.5996963	1.1497091333	1.5453364022	0.591
1	2020-02-24T12:53:48.249Z	-0.1625565784	0.5683831424	-0.7896027	-0.52656248	-0.40876335	-0.03386898	1.5691793	1.1462936171	1.5463669811	0.591
2	2020-02-24T12:55:15.683Z	-0.1630352996	0.5690385881	-0.7766519	-0.5259441	-0.40767962	-0.028969556	1.5831361	1.1670714718	1.5517443996	0.591
3	2020-02-24T12:56:04.137Z	-0.1625139585	0.5688325591	-0.779713	-0.524504125	-0.4062865	-0.0324465785	1.5874004	1.1528910328	1.5461579673	0.591
4	2020-02-24T12:56:52.382Z	-0.1630009439	0.5680967734	-0.7833043	-0.52627444	-0.40978768	-0.0297065145	1.6217437	1.1621453177	1.5487172066	0.591
5	2020-02-24T12:57:40.548Z	-0.1645937785	0.5673202399	-0.774919	-0.52791645	-0.40966374	-0.033694818	1.5720177	1.1776147976	1.552134022	0.590
6	2020-02-24T12:58:28.855Z	-0.1631928447	0.5682196314	-0.7766512	-0.5282678	-0.40725255	-0.0312432095	1.5833437	1.1666161423	1.5504871945	0.591
7	2020-02-24T12:59:17.022Z	-0.1634563107	0.5681661758	-0.75784814	-0.5255274	-0.40851527	-0.032321376	1.5939375	1.1678600604	1.5520630442	0.591
8	2020-02-24T13:00:05.297Z	-0.1632906903	0.5691170971	-0.7726295	-0.528678915	-0.40969804	-0.0366148745	1.5914558	1.1768862875	1.552952455	0.592
9	2020-02-24T13:00:53.581Z	-0.1648779966	0.5670969994	-0.75461316	-0.5271706	-0.41032743	-0.042829759	1.6149	1.1795612964	1.5549442142	0.590

Figura 4.15: *Dataset contenente gli Smart Data estratti*

Confronto tra le due metodologie:

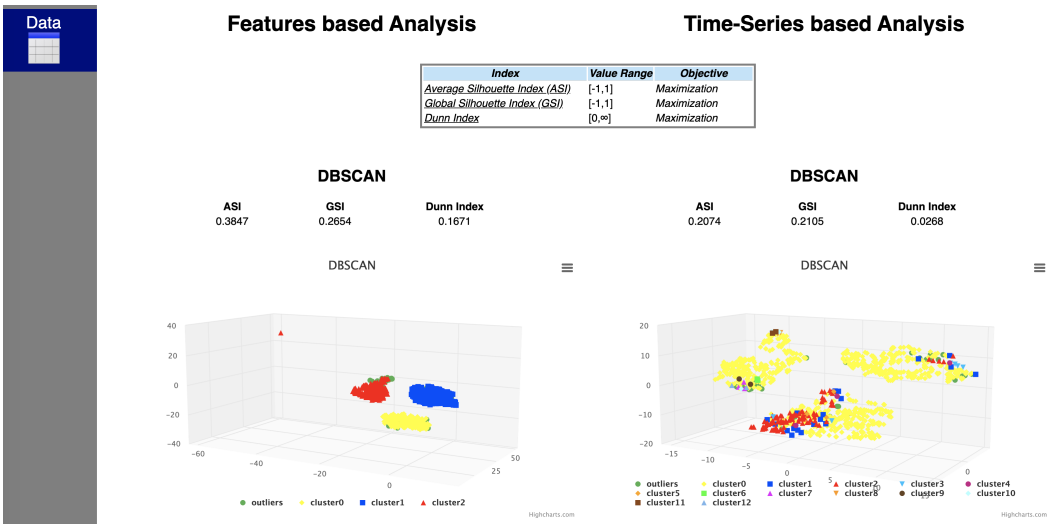


Figura 4.16: *Confronto tra approccio 'Features based' e 'Time-series based': DBSCAN*

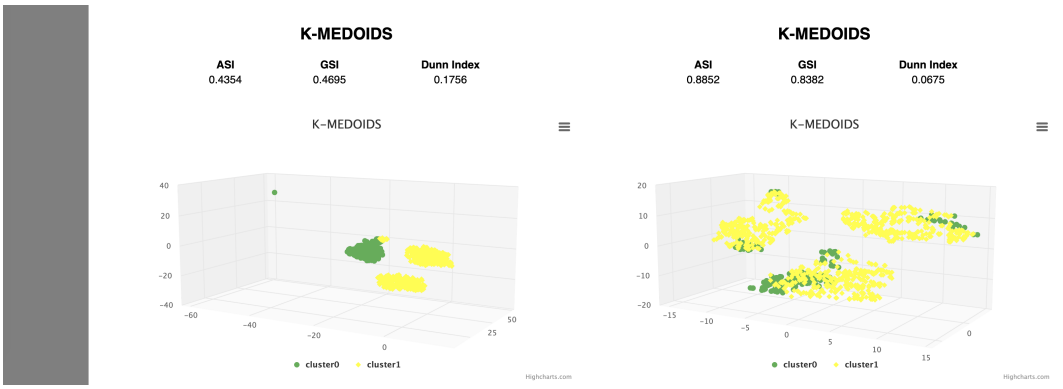


Figura 4.17: *Confronto tra approccio 'Features based' e 'Time-series based': K-MEDOIDS*

Caratterizzazione dei cluster nelle due metodologie:

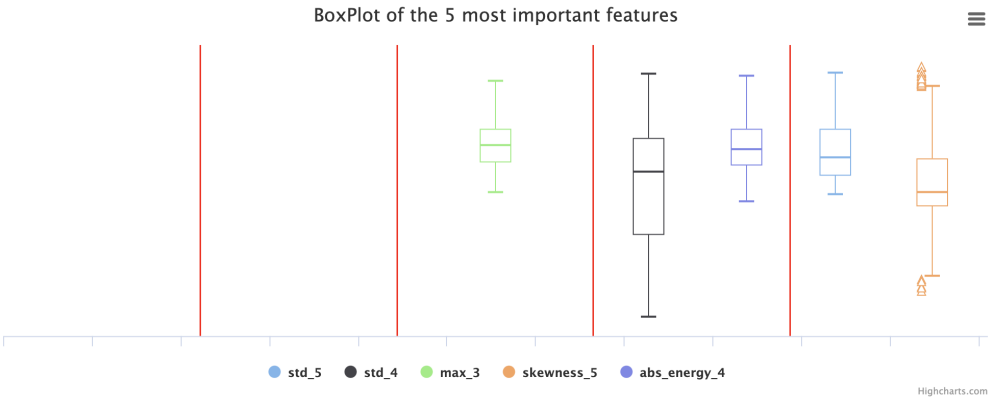


Figura 4.18: *Boxplot delle top 5 features riferiti a tutto il dataset*

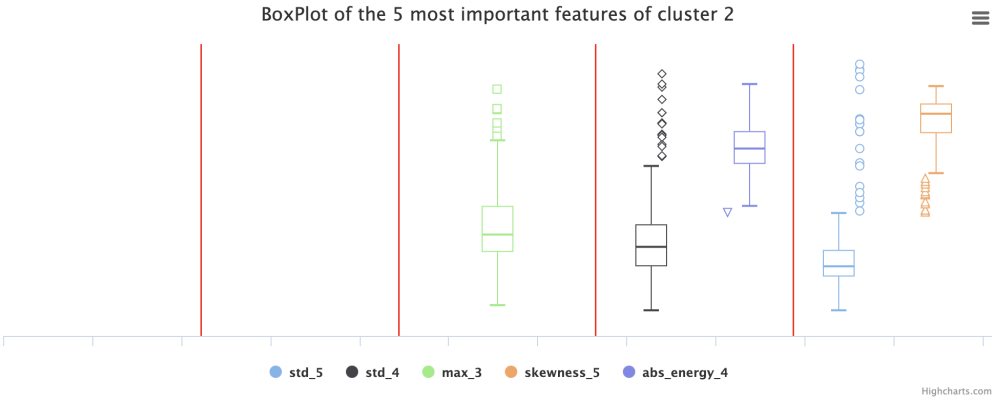


Figura 4.19: *Boxplot delle top 5 features riferiti al cluster 2*

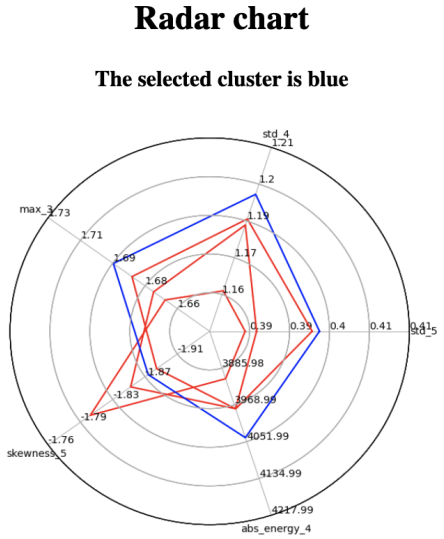


Figura 4.20: *Radar chart delle top 5 features: in blu quelle riferite al cluster 2*

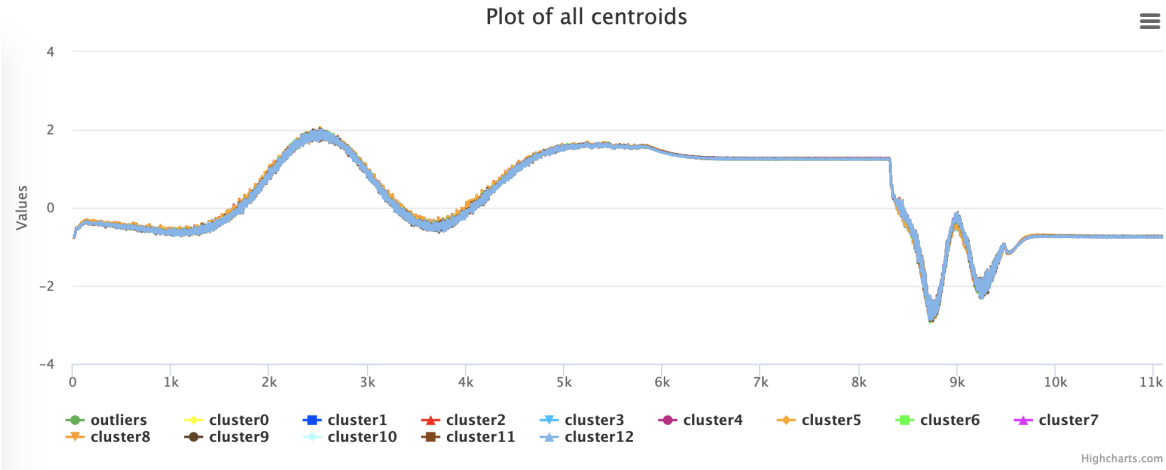


Figura 4.21: *Rappresentazione dell'andamento di tutti i centroidi*

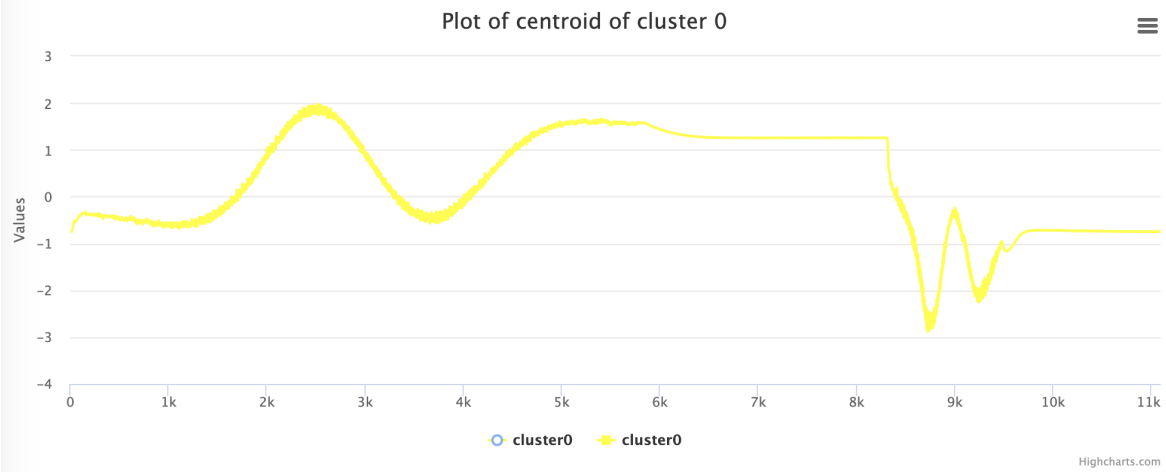


Figura 4.22: *Rappresentazione dell'andamento del centroide del cluster 0*

4.3 3° dataset

L'ultimo caso d'uso riguarda una nota azienda multinazionale produttrice di elettrodomestici. In fase di produzione, un ugello che inietta una combinazione di due sostanze chimiche reagenti è stato sensorizzato per monitorare l'intero processo. In questo caso, la variabile monitorata rappresenta i valori di pressione (Bars) delle sostanze chimiche in fase di iniezione.
Il dataset esaminato è composto da 29503 serie temporali e sono presenti tre attributi: *id*, *date_time* e *data_values*.
L'attributo *date_time* contiene i timestamp relativi al tempo di raccolta del dato, mentre l'attributo *data_values* contiene le serie temporali, in cui la time-series minima ha 1000 elementi.
Nella sequenza di immagini successive sono mostrate le diverse schermate mostrate all'utente, seguendo la stessa procedura descritta precedentemente:

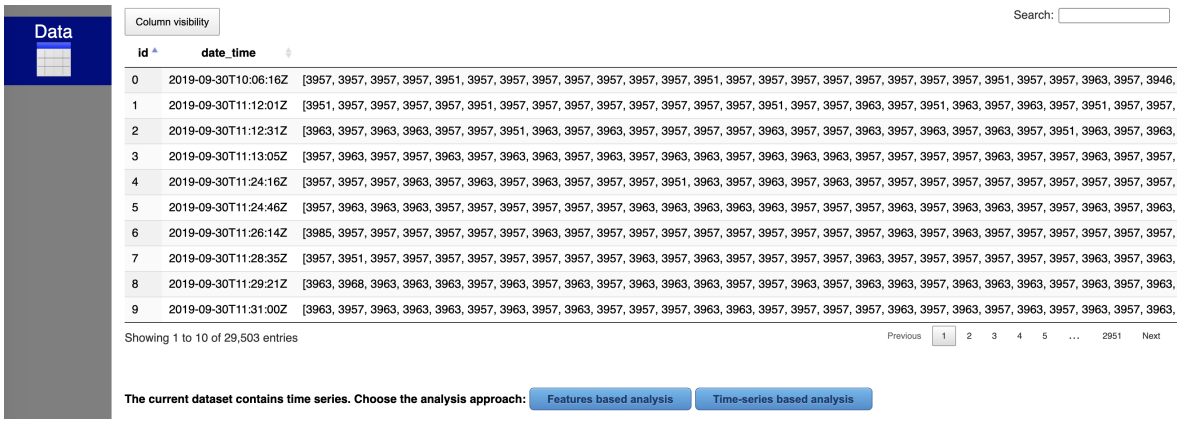


Figura 4.23: *Apertura del dataset contenente le serie temporali*

Approccio *Time-series based*:

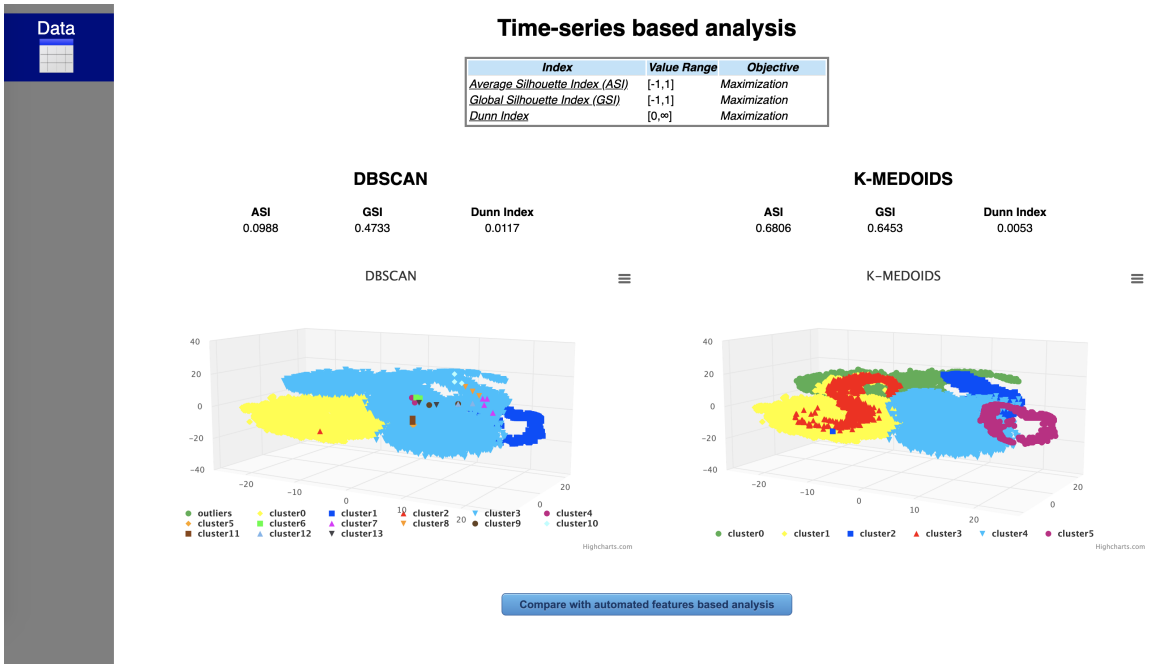


Figura 4.24: *Risultati approccio 'Time-series based'*

Approccio *Features based*:

Figura 4.25: *Scelta split: modalità automatica*

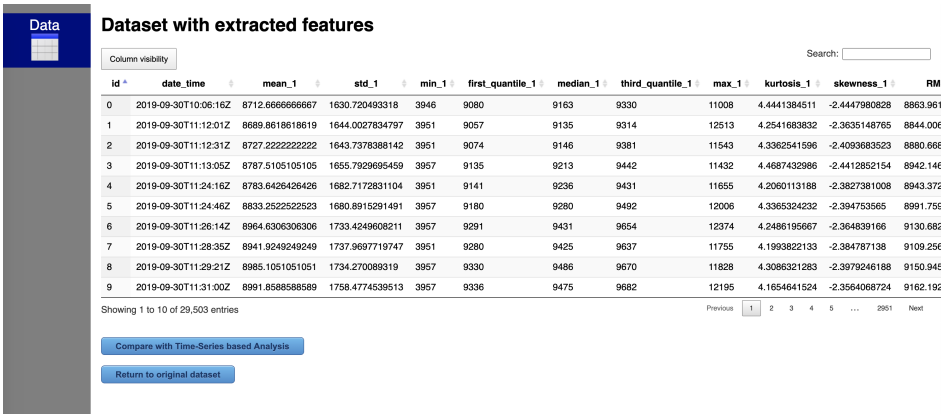


Figura 4.26: *Dataset contenente gli Smart Data estratti*

Confronto tra le due metodologie:

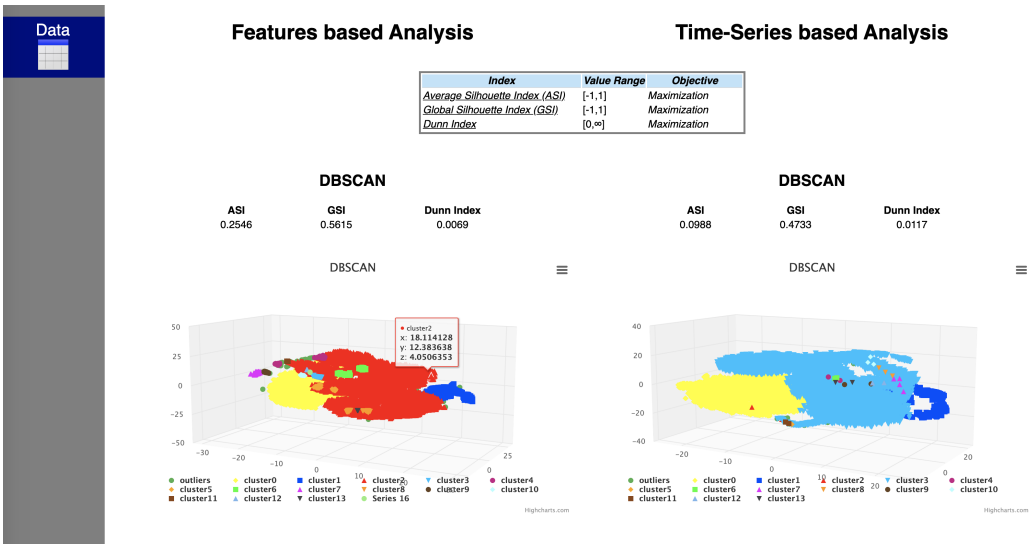


Figura 4.27: *Confronto tra approccio 'Features based' e 'Time-series based': DBSCAN*

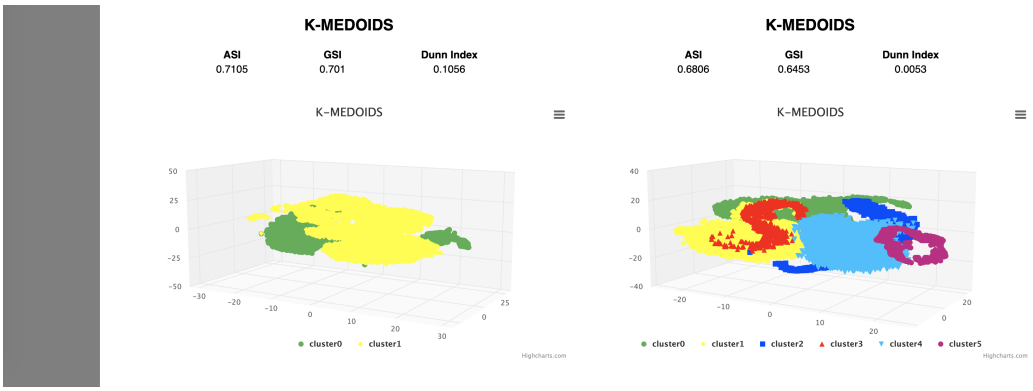


Figura 4.28: *Confronto tra approccio 'Features based' e 'Time-series based': K-MEDOIDS*

Caratterizzazione dei cluster nelle due metodologie:

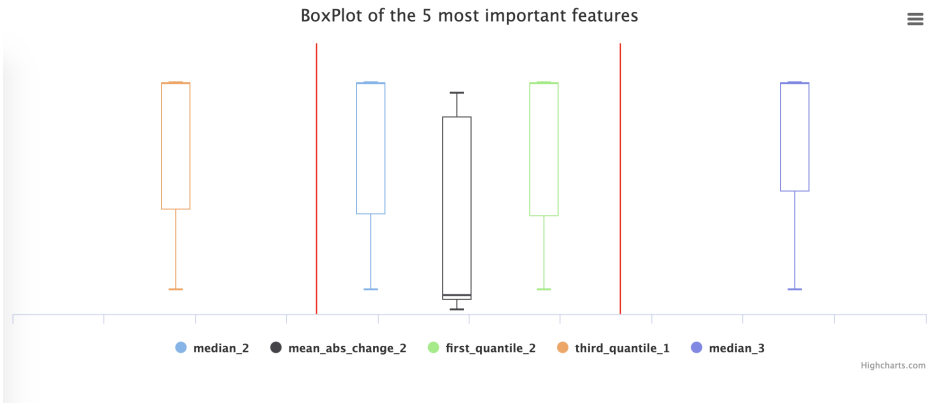


Figura 4.29: *Boxplot delle top 5 features riferiti a tutto il dataset*

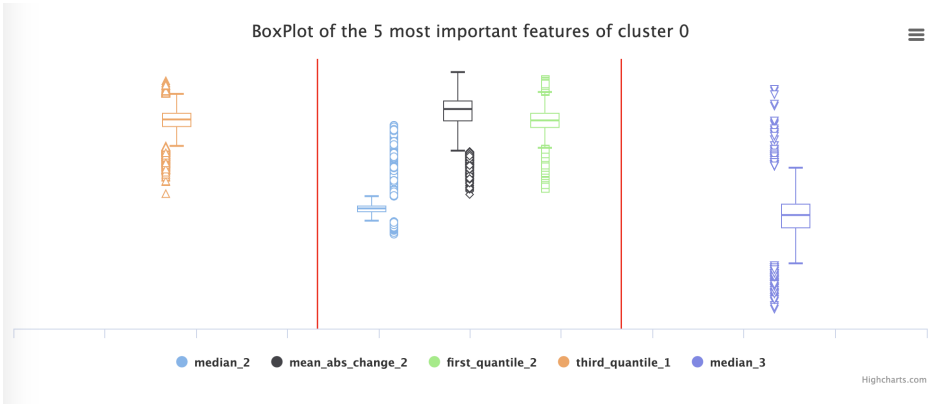


Figura 4.30: *Boxplot delle top 5 features riferiti al cluster 0*

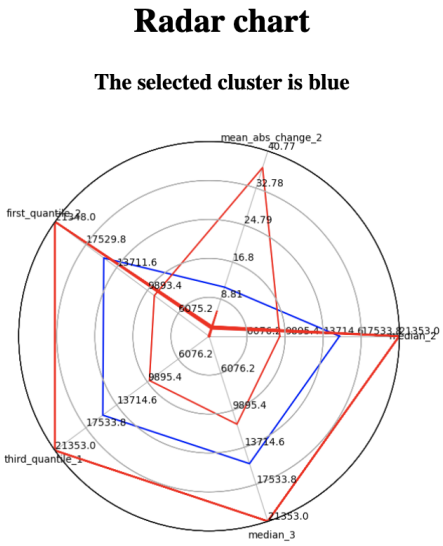


Figura 4.31: *Radar chart delle top 5 features: in blu quelle riferite al cluster 0*

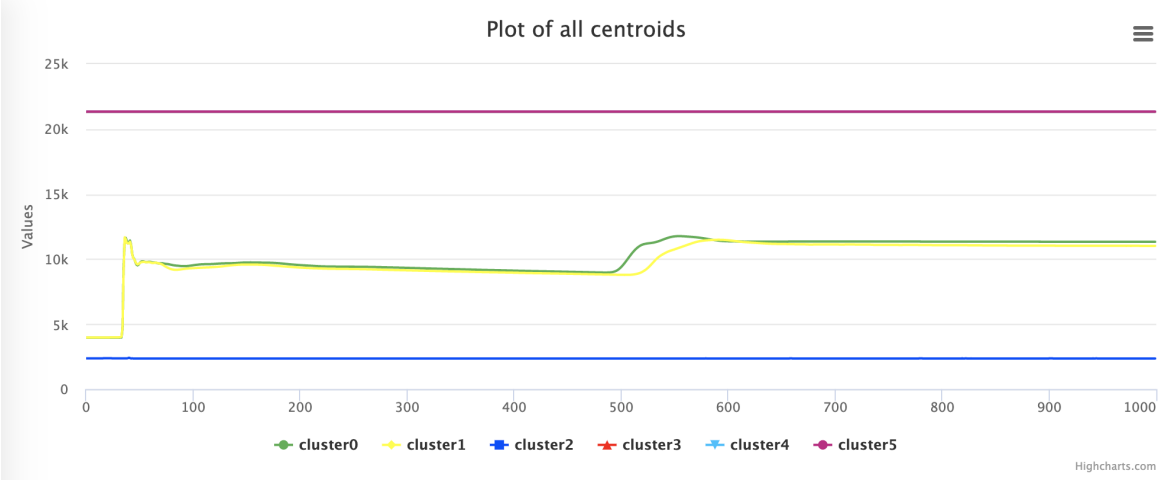


Figura 4.32: *Rappresentazione dell'andamento di tutti i centroidi*

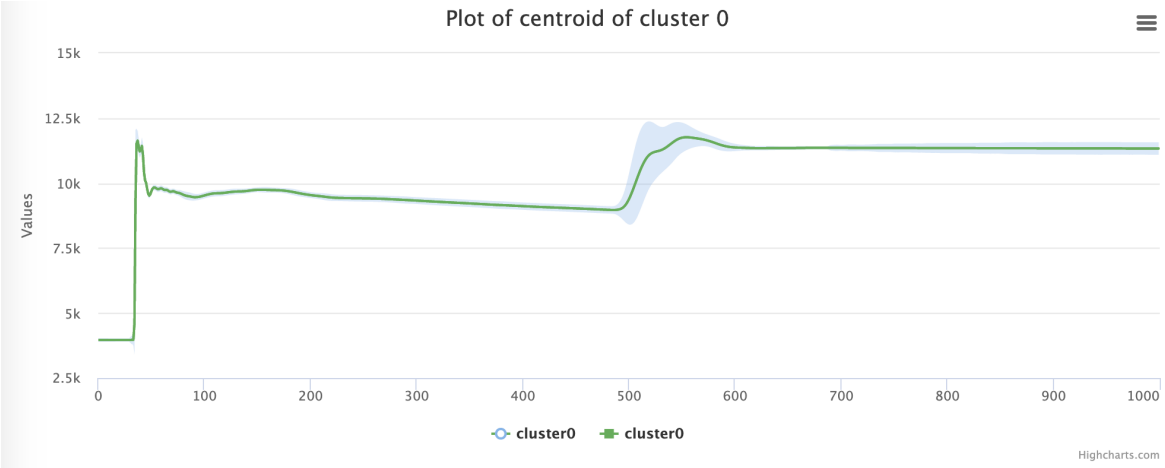


Figura 4.33: *Rappresentazione dell'andamento del centroide del cluster 0*

4.4 Analisi dei risultati

I risultati ottenuti nei 3 casi analizzati mostrano come l'approccio *Features based* fornisca, la maggior parte delle volte, indici di bontà del clustering migliori. Solo in un caso, nell'analisi del secondo dataset considerato, l'applicazione dell'algoritmo K-Medoids porta a indici di bontà migliori nell'approccio *Time-series based*.
Anche la rappresentazione grafica tramite scatter plot mostra risultati visivi migliori nell'approccio *Features based*, in quanto è possibile notare una suddivisione più netta dei cluster rappresentati.
Si può quindi affermare che tramite l'approccio *Features based*, pensato e sviluppato all'interno del progetto di tesi, si ottiene alla fine un'analisi buona laddove l'obiettivo è valutare il trend delle serie temporali e non si è interessati alla variabilità del dato. Va ricordato infatti che tramite l'estrazione delle features viene perso il riferimento temporale, riducendo la variabilità del dato iniziale.
Ulteriori vantaggi dell'approccio *Features based* sono i costi computazionali ridotti rispetto al calcolo della matrice delle distanze tramite BOSS, soprattutto all'aumentare del numero di time-series presenti nell'insieme di dati di partenza, e una maggior facilità di comprensione e di utilizzo della metodologia da parte dell'utente finale.

Capitolo 5

Conclusione

Con questo progetto di tesi è stata introdotta un'estensione del framework ADESCA per l'esplorazione automatizzata di insieme di dati contenenti serie temporali.
Il lavoro svolto ha permesso lo studio e la realizzazione di una metodologia d'analisi automatica, in modo che tutti gli utenti, anche i meno esperti, possano trarre vantaggio dall'utilizzo del framework e possano sfruttare i risultati mostrati, cercando di rendere il flusso di esecuzione il più semplice possibile.
Non essendoci strategie d'analisi esaustive allo stato dell'arte che possano fornire risultati approfonditi sull'analisi di serie temporali, l'introduzione di una nuova metodologia, basata sull'estrazione di features che caratterizzano ogni segmento del segnale, non è stata di facile sviluppo.
Il lavoro svolto è solo un punto di partenza per la gestione delle serie temporali, in quanto la loro analisi risulta un argomento complesso, soprattutto se lo scopo è rendere il tutto automatico, minimizzando il contributo umano. Occorre trovare algoritmi e strategie che possano essere generalizzati il più possibile e applicabili a diversi insiemi di dati.
Si può affermare che i risultati prodotti dall'estrazione di features sono buoni per analizzare il trend di un insieme di dati contenenti serie temporali, nonostante siano possibili diversi miglioramenti.

In seguito sono mostrate alcuni proposte che potrebbero essere sviluppate in futuro all'interno del framework per rendere l'analisi di serie temporali più interessante e più completa.

Rimozione degli outliers

Studiare una metodologia per il rilevamento e la rimozione delle serie temporali che possono essere considerate degli outliers all'interno dell'insieme di dati.

Implementazione di uno *StoryTelling* automatico

Realizzare uno *StoryTelling* automatico per qualsiasi insieme di dati, anche quelli contenenti serie temporali, per fornire una panoramica completa del dataset esaminato.

Raccolta feedback dell'utente

Attività di sperimentazione del tool ADESCA a utenti esperti di Data Science e non esperti con somministrazione di un questionario per raccogliere feedback per migliorare le caratteristiche della versione attuale. I feedback degli utenti saranno anche analizzati per definire strategie opportune per la definizione semantica dello *StoryTelling*.

Bibliografia e Sitografia

[1] Luciana Maci. «Che cos'è l'Industria 4.0 e perché è importante saperla affrontare». In: *EconomyUp* (2019). URL: https://www.economyup.it/innovazione/cos-e-l-industria-40-e-perche-e-importante-saperla-affrontare/.

[2] Giuseppe Maneschi. *Il potere dei dati nell'Industria 4.0*. URL: https://www.cosmanitalia.it/it/blog/il-potere-dei-dati-nell-industria-4-0/.

[3] Extrared. *L'evoluzione dell'industria 4.0*. URL: https://www.extrasys.it/it/red/industria-4-0.

[4] Elvis Hozdić. «Smart factory for industry 4.0: A review». In: *International Journal of Modern Manufacturing Technologies* 7.1 (2015), pp. 28–35.

[5] Li Da Xu, Eric L. Xu e Ling Li. «Industry 4.0: state of the art and future trends». In: *International Journal of Production Research* 56.8 (2018), pp. 2941–2962. DOI: 10.1080/00207543.2018.1444806. eprint: https://doi.org/10.1080/00207543.2018.1444806. URL: https://doi.org/10.1080/00207543.2018.1444806.

[6] Francisco Almada-Lobo. «The Industry 4.0 revolution and the future of Manufacturing Execution Systems (MES)». In: *Journal of innovation management* 3.4 (2015), pp. 16–21.

[7] Andreja Rojko. «Industry 4.0 Concept: Background and Overview». In: *International Journal of Interactive Mobile Technologies (iJIM)* 11.5 (2017), pp. 77–90. ISSN: 1865-7923. URL: https://onlinejour.journals.publicknowledgeproject.org/index.php/i-jim/article/view/7072.

[8] Baotong Chen et al. «Smart factory of industry 4.0: Key technologies, application case, and challenges». In: *IEEE Access* 6 (2017), pp. 6505–6519.

[9] V. Alcácer e V. Cruz-Machado. «Scanning the Industry 4.0: A Literature Review on Technologies for Manufacturing Systems». In: *Engineering Science and Technology, an International Journal* 22.3 (2019), pp. 899–919. ISSN: 2215-0986. DOI: https://doi.org/10.1016/j.jestch.2019.01.006. URL: http://www.sciencedirect.com/science/article/pii/S2215098618317750.

[10] David Romero et al. «Towards an Operator 4.0 Typology: A Human-Centric Perspective on the Fourth Industrial Revolution Technologies». In: ott. 2016.

[11] M. Bellini. «Blockchain: cos'è, come funziona e gli ambiti applicativi in Italia». In: (2018). URL: https://www.blockchain4innovation.it/esperti/blockchain-perche-e-cosi-importante/.

[12] Mauro Bellini. «IoT (Internet of Things): cos'è, come funziona ed esempi». In: (2020). URL: https://www.internet4things.it/iot-library/internet-of-things-gli-ambiti-applicativi-in-italia/.

[13] Marco Giannini. «Industria 4.0, evoluzione della logistica e applicazione del Knowledge Triangle: la centralità delle competenze professionali e il progetto Framelog». In: *Methodology* 37.3 (2019).

[14] Deloitte. «Italia 4.0: siamo pronti?Il percepito degli executive in merito agli impatti economici, tecnologici es ociali delle nuove tecnologie». In: (2018). URL: https://www2.deloitte.com/content/dam/Deloitte/it/Documents/process-and-operations/Report%20Italia%204.0%20siamo%20pronti_Deloitte%20Italy.pdf.

[15] Adeline Bailly. «Time Series Classification Algorithms with Applications in Remote Sensing». Tesi di dott. 2018.

[16] Tim Hall. *The Role of Data in Industry 4.0*. 2020. URL: https://industrytoday.com/the-role-of-data-in-industry-4-0/.

[17] Gareth James et al. *An introduction to statistical learning*. Vol. 112. Springer, 2013.

[18] Pang-Ning Tan, Michael Steinbach e Vipin Kumar. «Data mining cluster analysis: basic concepts and algorithms». In: *Introduction to data mining* (2013), pp. 487–533.

[19] Enrico Pegoraro. *Statistica per Data Science con R - V. 03*. 2019. URL: http://www.r-project.it/_book/introduzione-allalgoritmo-dbscan.html.

[20] Patrick Schäfer. «The BOSS is concerned with time series classification in the presence of noise». In: *Data Mining and Knowledge Discovery* 29 (nov. 2015). DOI: 10.1007/s10618-014-0377-7.

[21] Preeti Arora, Shipra Varshney et al. «Analysis of k-means and k-medoids algorithm for big data». In: *Procedia Computer Science* 78 (2016), pp. 507–512.

[22] Laurens van der Maaten e Geoffrey Hinton. «Visualizing data using t-SNE». In: *Journal of machine learning research* 9.Nov (2008), pp. 2579–2605.

[23] Umberto Santucci. *Diagramma a scatola e baffi*. URL: http://www.umbertosantucci.it/atlante/diagramma-a-scatola-e-baffi/.

[24] edraw. *When to Use a Spider Chart*. URL: https://www.edrawsoft.com/chart/when-to-use-spider-chart.html.

[25] URL: https://serena-project.eu/.