Progetto Industry Lab

A. Risaro (825113), D. Mingolla (866167)

2021/2022

Indice

1 Introduzione **3**
1.1 Introduzione al problema 3
1.2 Contestualizzazione nell'ambito dell'Industria 4.0 3
1.3 Assunzioni e riferimenti scientifici 4
1.4 Struttura del report 6
1.5 Struttura del codice 6

2 Preparazione ed esplorazione del dataset **9**
2.1 Raccolta dati 9
2.2 Pulizia del Dataset e Analisi descrittive 10

3 Training dei modelli **15**
3.1 Algoritmi 15
3.1.1 Clustering 15
3.1.2 Anomaly Detection 18
3.2 Descrizione della fase di training 20
3.3 Informazioni acquisite 29

4 Testing **31**

5 Deploy **34**

6 Risultati **35**

7 Conclusioni **40**
7.1 Consigli per l'azienda 40
7.2 Sviluppi futuri 41

8 Bibliografia **42**

1 Introduzione

1.1 Introduzione al problema

Il problema sottopostoci dal *Centro Ricerche Fiat - Fiat Chrysler Automobiles (FCA)* riguarda l'identificazione dei punti di anomalia all'interno delle curve generate dai loro processi di saldatura. Tali saldature sono state svolte utilizzando la metodologia a punti su parti di telaio di automobili che attraversano le stazioni di saldatura 005 situate all'interno di uno degli stabilimenti dell'azienda.

L'obiettivo di questo lavoro è quello di migliorare l'efficienza produttiva e diminuire il numero di anomalie durante il processo di saldatura. Per raggiungere tale obiettivo sono stati utilizzati degli algoritmi di Machine Learning non supervisionati e degli strumenti di analisi statistica descrittiva e inferenziale.

1.2 Contestualizzazione nell'ambito dell'Industria 4.0

Con il termine *Industria 4.0* si definisce la trasformazione digitale in atto oggi nel settore industriale. Con questa locuzione si intende descrivere la capacità che diversi strumenti digitali hanno di poter comunicare tra di loro all'interno di un ambiente comune, con il fine di portare a compimento una serie di obiettivi prestabiliti. In tal modo la produzione tradizionale viene rinnovata utilizzando le moderne tecnologie *ICT* e i sistemi *"smart"* a oggi disponibili, automatizzando il processo produttivo e traendo da esso la massima efficacia ed efficienza. Lo scambio di informazioni tra operatori, macchinari e strumenti rende tali tecnologie "intelligenti" adatte a diversi ambiti e settori. Alla base della *Industria 4.0* vi è poi l'utilizzo dei sensori, ovvero dispositivi in grado di raccogliere dati in tempo reale relativi al macchinario industriale sul quale sono installati. In seguito questi dati possono essere analizzati, monitorati e condivisi con altri macchinari e/o operatori.

I dati fornitici da *FCA* sono un esempio di quanto detto precedentemente in quanto raccolti tramite sensoristica. Tali dati sono stati infatti estrapolati tramite un sistema chiamato *Weld Quality System (WQS)*, mentre un ulteriore sistema denominato *Weld Management System (WMS)* si è occupato di generare le curve di saldatura caratteristiche del processo.

La velocità computazionale in aumento all'interno dei sensori assieme alla loro capacità di analisi, permette alle aziende di agire in tempo reale sul processo produttivo migliorandolo.

La possibilità di registrare dati durante il processo produttivo risulta essere un grande vantaggio competitivo per un'azienda, in quanto permette sia di avere uno storico degli andamenti produttivi su cui fare analisi ex-post attraverso algoritmi di intelligenza artificiale, sia di poter sviluppare algoritmi per analisi in tempo reale.

È infatti possibile svolgere operazioni di manutenzione predittiva sui macchinari che includono la rilevazione di anomalie o difetti nella produzione e di ottimizzazione dei processi.

In tale contesto è possibile utilizzare i modelli sviluppati in questo lavoro installandoli sui sensori presenti all'interno del processo industriale che si occupa di monitorare lo stato operativo delle saldature. Tali modelli possono essere eseguiti anche su hardware non particolarmente potenti e permetterebbero all'azienda di monitorare la qualità dei propri telai.

All'interno dell'articolo [5] è sottolineata l'importanza dell'evitare che l'intensità di corrente durante la saldatura subisca dei picchi improvvisi onde evitare sbavature nella stessa. È facile intuire come sia possibile sfruttare l'immediatezza nell'agire dei sensori per fermare la saldatura al momento ritenuto più opportuno, per poi riprenderla una volta terminate le dovute operazioni di ripristino della qualità. Questo consentirebbe di allungare la vita dell'elettrodo utilizzato per saldare, in quanto si eviterebbe che il processo continui nonostante siano stati già identificati degli andamenti anomali.

Gli strumenti sviluppati durante lo svolgimento di tale lavoro possono essere utilizzati come base per lo sviluppo di applicativi dall'interfaccia intuitiva. Tali applicazioni consentirebbero agli operatori sul campo di visualizzare graficamente e in tempo reale lo svolgimento della saldatura ed eventualmente bloccarla manualmente, tutto questo senza la necessità di avere particolari conoscenze tecniche.

1.3 Assunzioni e riferimenti scientifici

I materiali utilizzati per le costruzione delle carrozzerie delle automobili sono diversi in quanto ognuno di essi offre dei vantaggi e delle caratteristiche qualitative specifiche. Pertanto, si trovano spesso componenti, strutture o carrozzerie in cui vengono combinati elementi di diversa natura. Di norma, la mescolanza di diverse tipologie di materiale ha come fine la riduzione

del peso del veicolo e una sua maggiore resistenza e sicurezza grazie all'uso di materiali più leggeri, ma più durevoli. Secondo l'articolo [1], l'acciaio è presente nella maggior parte delle auto, specialmente in quelle di classe medio/bassa. In base alla percentuale di carbonio presente nel ferro, si vanno a costituire delle leghe specifiche con particolari proprietà meccaniche. La tipologia di materiale su cui avviene la saldatura, assieme alla tipologia della stessa e allo spessore delle lamiere, rappresentano delle informazioni fondamentali per l'identificazione delle anomalie. Questo è vero in quanto in base alle caratteristiche precedentemente elencate, l'amperaggio raggiunto durante la saldatura può essere eccessivo in alcuni casi, provocando dunque delle sbavature (anomalie), e normale in altri, come descritto accuratamente nell'articolo di Zhang et al..

Da tale articolo ne consegue che valori di intensità di corrente di molto superiori a 8500A o di molto al di sotto degli 8000A, causano delle sbavature come quelle mostrate in Figura 1.

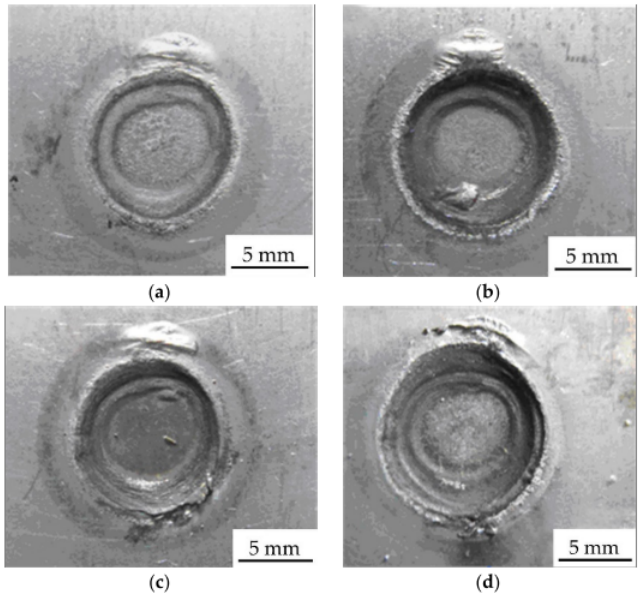

Figure 3. Resistance spot weld appearance gained with different welding currents: **(a)** 8.5 kA; **(b)** 10.0 kA; **(c)** 11.0 kA; **(d)** 12.0 kA.

Figura 1: Immagine tratta dall'articolo di Zhang et al.

Abbiamo dunque assunto che i dati forniti siano relativi a delle saldature effettuate su telai in *acciaio*. Inoltre, come descritto nell'articolo [2], la *salda-*

tura elettrica a resistenza a punti è tipicamente utilizzata per saldare lamiere e piastre sottili (in genere di spessore inferiore a 10 mm) e l'acciaio rientra tra i materiali saldabili. Tale saldatura a punti può essere effettuata in modo automatico tramite delle attrezzature sulla estremità di robot antropomorfi che permettono un'alta flessibilità di movimento, nonché la ripetibilità del processo di saldatura. Sulla base di questa affermazione, abbiamo ritenuto coerente assumere che la tipologia di saldatura a cui i nostri dati fanno riferimento è la saldatura elettrica a resistenza a punti.

1.4 Struttura del report

Il report è stato suddiviso in 7 sezioni. Nella prima sezione di introduzione viene introdotto il problema affrontato in questo lavoro e contestualizzato nell'ambito della Industry 4.0. All'interno di tale sezione si è pensato inoltre di inserire una panoramica sulle assunzioni fatte e sui riferimenti scientifici utilizzati per comprendere al meglio il contesto operativo. Nella seconda sezione è descritto il processo di costruzione e preparazione del dataset, il tutto corredato da una descrizione della fase esplorativa svolta sul dataset basata su svariate analisi statistiche descrittive da cui sono state tratte informazioni utili ai fini risolutivi del problema. Nella terza sezione si sono evidenziati gli algoritmi utilizzati nel nostro lavoro assieme ai pregi e ai difetti di ognuno di essi. In tale sezione viene inoltre descritto il processo di allenamento dei modelli e le informazioni acquisite in seguito alla loro applicazione. Nella quarta sezione vi è una dimostrazione dell'efficacia dei nostri modelli nel differenziare le curve di saldatura in base al loro andamento, seppur tali curve rappresentano dei nuovi dati per i modelli stessi.

Nella quinta sezione è presente una breve descrizione della fase di deploy, consistita principalmente nella simulazione del comportamento di un API a cui, se passato in input un file JSON contenente una curva di saldatura e il rispettivo spotname, restituisce il numero di anomalie al suo interno e il livello di concordanza tra i modelli.

Infine nella sesta e settima sezione sono stati descritti i risultati ottenuti, i possibili sviluppi futuri del nostro progetto e alcuni consigli per l'azienda.

1.5 Struttura del codice

In questa sezione è descritta l'impostazione organizzativa dell'intero codice scritto durante lo svolgimento del progetto. La cartella principale del pro-

getto è denominata *industry_project* e principalmente contiene al suo interno i seguenti file e le seguenti cartelle:

- **Attempts**: la seguente cartella contiene un notebook con all'interno il codice relativo ad alcuni tentativi di analisi dei dati che sono stati svolti e poi scartati.
- **Data**: la seguente cartella contiene tutti i file necessari per la corretta esecuzione del codice. Contiene al suo interno le seguenti sottocartelle:
  - *Original*: la seguente cartella contiene al suo interno i file originali inviatici dall'azienda in formato JSON.
  - *Prepared*: la seguente cartella contiene al suo interno i dati risultanti da un processo che a partire da dati originali, ha apportato delle modifiche specifiche a quest'ultimi in modo da poter essere utilizzati nelle fasi di training, testing, deploy e di analisi descrittive delle anomalie. Tutti i file al suo interno sono stati memorizzati in formato *pickle*.
  - *Inference*: la seguente cartella contiene al suo interno un campione stratificato acquisito dai dati originali e memorizzato in formato pickle. Sebbene il problema trattato in questo lavoro è di tipo non supervisionato, abbiamo ritenuto utile riservarci un insieme di dati non utilizzato in fase di training. Tali dati sono stati utilizzati per testare l'efficacia dei nostri modelli evitando di incrementare il bias che sarebbe potuto esserci se i modelli fossero stati testati su un sottoinsieme dei dati di training.
- **Models**: la seguente cartella contiene al suo interno i modelli utilizzati per l'identificazione delle anomalie memorizzati in formato pickle. Oltre a quest'ultimi, al suo interno sono presenti:
  - il pickle degli scaler utilizzati per standardizzare le curve di ogni spotname prima che vi venisse applicato il DBSCAN;
  - un dataframe Pandas contenente gli intervalli di confidenza per ogni punto delle curve rappresentanti l'andamento medio globale di ogni spotname.
- **Notebooks**: la seguente cartella contiene al suo interno i notebook Jupyter numerati in ordine crescente secondo l'ordine di esecuzione da seguire.

- **Documentation**: la seguente cartella contiene al suo due documenti in formato pdf: uno contenente una guida su come riprodurre l'ambiente di sviluppo sul proprio PC, l'altro contenente la presentazione del problema dataci da FCA.

- **poetry.lock** e **pyproject.toml**: sono i due file principali utilizzati da *Poetry* per l'installazione dell'ambiente virtuale e la gestione delle librerie e delle rispettive dipendenze. Grazie a questi file *Poetry* è in grado di avvertire anticipatamente lo sviluppatore della presenza di eventuali conflitti tra la libreria che sta cercando di installare e quelle già presenti nell'ambiente di sviluppo.

- **requirements.txt**: file contenente le librerie necessarie per la corretta esecuzione del progetto permettendone la riproducibilità su ambienti diversi da *Poetry*, come *Virtualenv* ad esempio.

2 Preparazione ed esplorazione del dataset

In questa sezione verrà spiegato il processo di raccolta e aggregazione dei dati forniti da FCA, assieme all'esplorazione di quest'ultimi attraverso molteplici analisi descrittive.

2.1 Raccolta dati

I dati sono stati forniti da FCA in più file JSON, in seguito da noi aggregati e inseriti in un dataframe Pandas. Sono stati scartati tutti i campi al di fuori dei 4 ritenuti utili per lo svolgimento del problema. Tali campi sono:

- Timestamp (str): data e ora di inizio della saldatura.
- SpotName (str): punto in cui è avvenuta la saldatura.
- CurrentCurve (Array): lista di valori di corrente (A) per ogni istante di saldatura.
- VoltageCurve (Array): lista di valori di tensione (V) per ogni istante di saldatura.

Si è poi proceduto a una pulizia preliminare: la colonna Timestamp è stata trasformata in formato datetime ed alcune osservazioni con SpotName mancante sono state arbitrariamente imputate con il valore '9999_9_99'.

	TimeStamp	spotName	CurrentCurve	VoltageCurve
75849	2019-09-17 17:33:48	60061_0_00	[312, 2067, 3862, 5267, 6398, 7217, 7959, 8700...	[134, 756, 1292, 1623, 1812, 1876, 1946, 2010,...
75857	2019-09-17 17:33:48	60095_0_00	[309, 2010, 3618, 4763, 5660, 6247, 6619, 6990...	[139, 754, 1294, 1655, 1885, 2008, 2032, 2090,...
75856	2019-09-17 17:33:48	60093_0_00	[312, 2034, 3630, 4788, 5571, 6165, 6666, 6979...	[138, 766, 1329, 1664, 1867, 1973, 2055, 2063,...
75855	2019-09-17 17:33:48	60091_0_00	[308, 1976, 3613, 4786, 5620, 6207, 6639, 7010...	[133, 768, 1328, 1695, 1929, 2029, 2088, 2129,...
75850	2019-09-17 17:33:48	60077_0_00	[293, 2055, 3850, 5220, 6231, 6982, 7406, 7667...	[127, 759, 1284, 1597, 1739, 1809, 1774, 1717,...
...	...	...	...	...
222857	2020-05-28 15:00:04	60017_0_00	None	None
222863	2020-05-28 15:00:17	60025_0_00	[344, 2225, 4011, 5234, 6017, 6613, 7051, 7365...	[134, 763, 1284, 1602, 1779, 1863, 1946, 1972,...
222864	2020-05-28 15:00:17	60027_0_00	[303, 2034, 3644, 4767, 5526, 6103, 6437, 6680...	[136, 744, 1264, 1600, 1784, 1904, 1944, 1960,...
222862	2020-05-28 15:00:17	60023_0_00	[317, 2065, 3781, 4926, 5688, 6229, 6801, 7118...	[139, 769, 1303, 1652, 1853, 1976, 2107, 2142,...
222865	2020-05-28 15:00:44	9999_9_99	[339, 2926, 5598, 7380, 8567, 9585, 10349, 105...	[125, 906, 1533, 1798, 1857, 1879, 1828, 1644,...

222866 rows × 4 columns

Figura 2: Come si presentano i dati

2.2 Pulizia del Dataset e Analisi descrittive

Una volta creato il dataset si è proceduto alla fase di esplorazione. La prima cosa su cui si ci è concentrati è stata la verifica dell'eventuale presenza di valori nulli nei dati che avevamo a disposizione, assieme a uno studio del loro impatto.

Si è notato che i valori nulli di CurrentCurve e VoltageCurve non solo erano sotto forma di None, ma anche di valori [-1] e [0]. Si è quindi proceduto all'imputazione di questi ultimi con valori None. Di seguito viene rappresentata la percentuale di valori nulli per ogni attributo del nostro dataset.

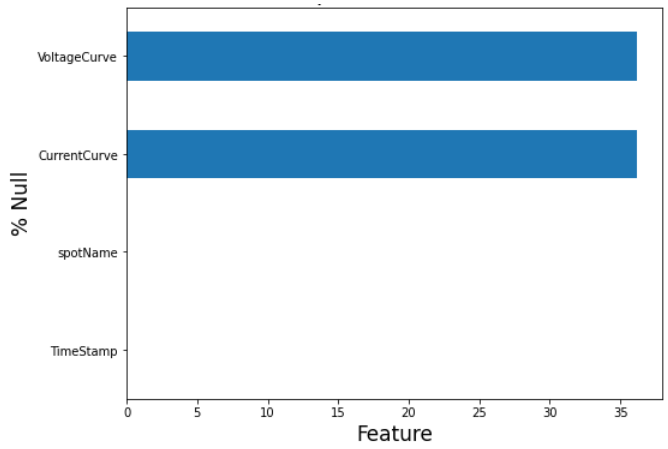

Figura 3: Percentuale di valori nulli per ogni attributo

Si è inoltre constatato che i valori all'interno delle colonne CurrentCurve e VoltageCurve o erano presenti contemporaneamente, o mancavano entrambi. Si è quindi indagata la distribuzione dei valori nulli rispetto a ogni singolo SpotName.

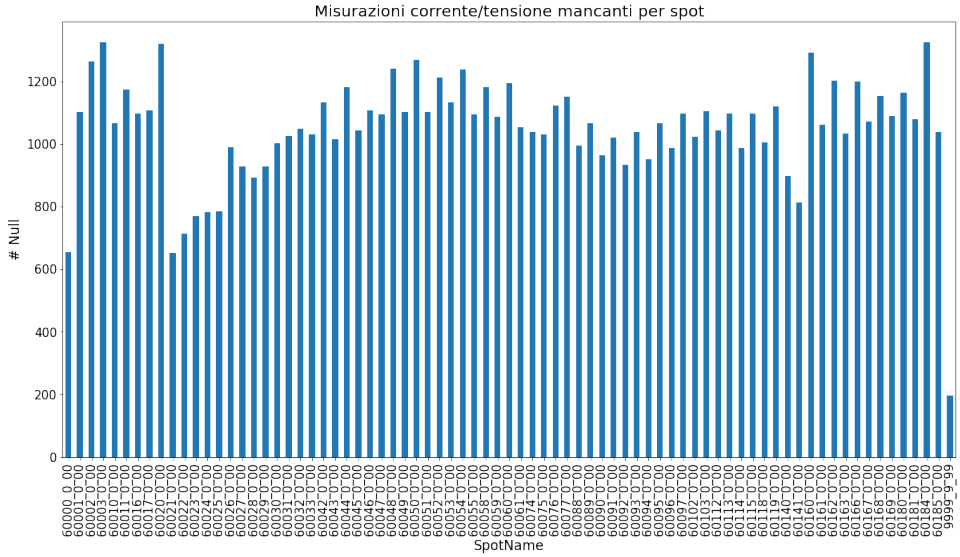

Figura 4: Numero di valori nulli per ogni SpotName

Da questo grafico si è potuto evincere che tutti gli SpotName hanno un numero elevato di valori mancanti.

Si è indagata inoltre la presenza di eventuali valori nulli (sotto forma di 0, -1, None) nelle colonne VoltageCurve e CurrentCurve considerando le sole liste non nulle. Quello che è emerso è l'assenza di valori nulli all'interno delle liste rappresentanti le curve di saldatura.

Si è voluto inoltre capire se i dati mancanti fossero legati ad un solo arco temporale, oppure fossero dispersi nel tempo. Si sono quindi andate a creare delle nuove variabili giorno, mese ed anno per poi andare a contare il numero di giorni mancanti per ogni mese.

Si è osservato che tale distribuzione dei valori mancanti è molto simile tra i diversi SpotName.

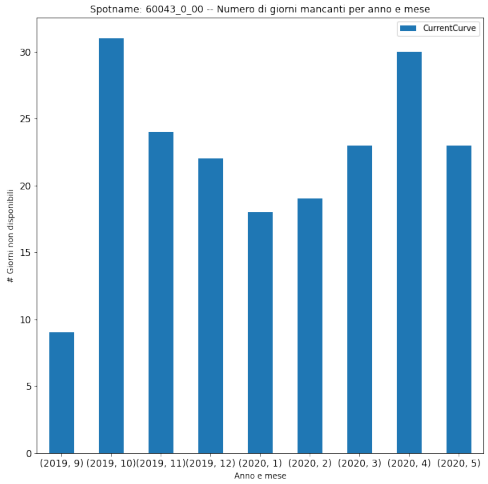

Figura 5: Numero di giorni mancanti per uno specifico SpotName per ogni mese

Da questa immagine possiamo evincere che vi sono alcuni alcuni spotname in cui vi è una completa assenza di dati in specifici mesi o, più in generale, vi è un elevato numero di valori mancanti durante tutto l'arco temporale di rilevazione dei dati fornitici. Queste constatazioni hanno portato alla scelta di non cercare di imputare i valori mancanti, ma di procedere alla loro rimozione.

Successivamente dato che i valori di campionamento di CurrentCurve e VoltageCurve durante il processo di saldatura vengono registrati ogni millisecondo (ms), si è indagato quale fase la loro durata media per ogni spotname.

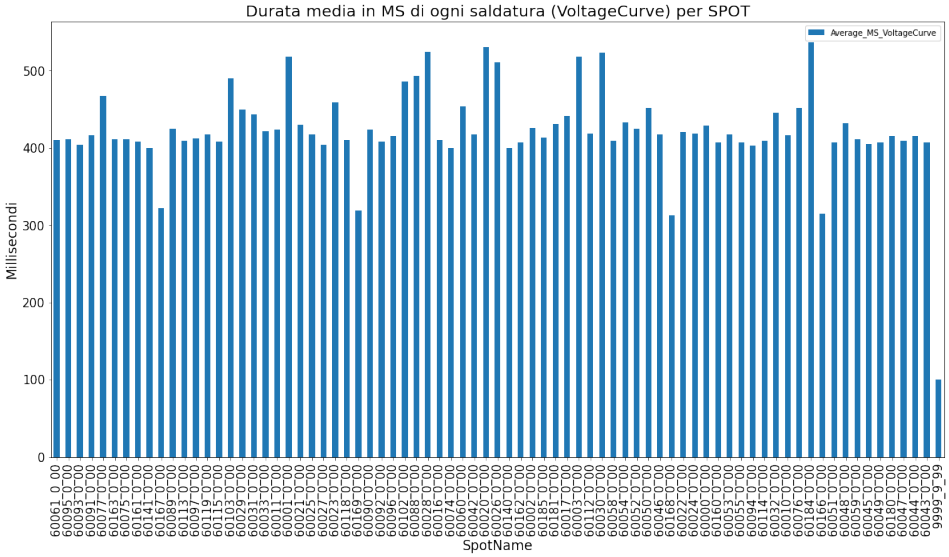

Figura 6: Durata media in millisecondi di ogni curva di saldatura (VoltageCurve) per ogni spotname

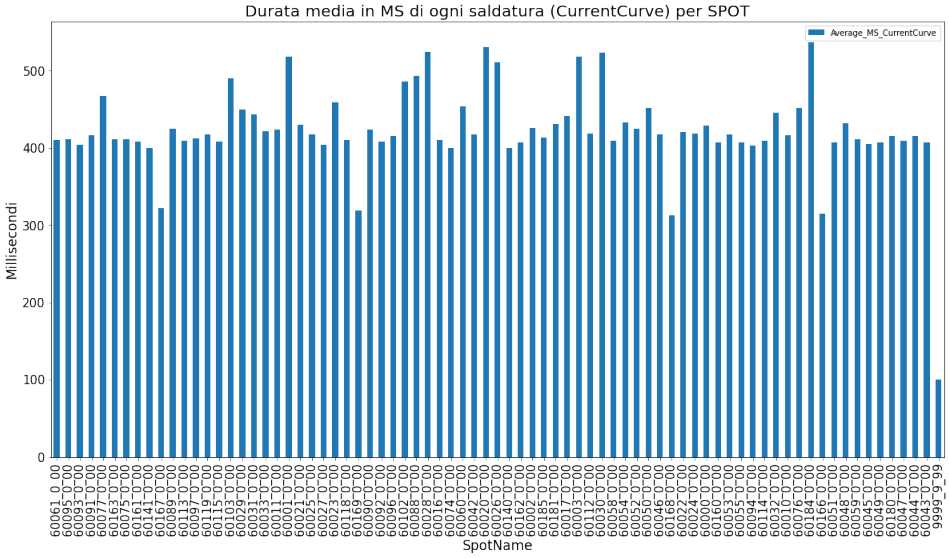

Figura 7: Durata media in millisecondi di ogni saldatura (CurrentCurve) per ogni spotname

Da questi risultati si è potuto constatare che la durata media del processo di saldatura è variabile rispetto a ogni specifico SpotName. Tale constatazione è ragionevole se consideriamo quanto detta nella sezione 1.3, ovvero che le caratteristiche di una saldatura dipendono fortemente dal tipo di materiale e dal punto su cui avviene.

Si è inoltre constatato che, per ogni osservazione, la lunghezza della curva nella colonna CurrentCurve è pari a quella della curva presente nella colonna VoltageCurve. A nostro parere questo implica che il sensore in questione funzioni correttamente in quanto, oltre ad acquisire i dati sull'andamento della saldatura nel momento stesso in cui quest'ultima avviene senza generare alcun valore nullo, è in grado di acquisire (o calcolare) due valori contemporaneamente (corrente e tensione).

Infine attraverso l'utilizzo delle *cross-correlations* si è indagato se le variabili CurrentCurve e VoltageCurve fossero correlate su diversi lag temporali, e se fosse stato quindi possibile considerare solo una delle due in modo da diminuire lo spazio necessario alla memorizzazione del dataset. In seguito a tale operazione è emerso che le due colonne fossero in realtà legate da un rapporto di dipendenza lineare spiegata dalla *Legge di Ohm*, e che dunque bastasse considerarne solo una delle due in quanto l'altra non avrebbe contribuito in alcun modo al miglioramento dei modelli. Alla luce di questa constatazione si è deciso di lavorare unicamente sulla variabile CurrentCurve.

3 Training dei modelli

Nella seguente sezione è descritta la procedura seguita ai fini dell'allenamento dei modelli di *anomaly detection* e di *clustering*.

3.1 Algoritmi

Nella seguente sottosezione verrà data una panoramica generale degli algoritmi di *clustering* e di *anomaly detection* utilizzati.

3.1.1 Clustering

Nella seguente sottosezione verrà descritto brevemente il funzionamento degli algoritmi di *clustering* che abbiamo deciso di confrontare durante lo svolgimento del progetto. Per ogni algoritmo è stata inoltre stilata una lista contenente i vantaggi e gli svantaggi che ci hanno indotto a considerarne l'utilizzo o meno.

3.1.1.1 DBSCAN

Il *DBSCAN* è un algoritmo di *clustering* basato sulla ricerca di zone dense, ovvero zone caratterizzate dalla presenza di molte osservazioni tra loro vicine. Le osservazioni appartenenti a zone non dense, come picchi positivi o negativi nel caso delle serie storiche, saranno invece classificate come anomale. Per l'identificazione di tali zone dense sono necessari due iperparametri:

1. **Eps**: corrisponde alla distanza minima tra due osservazioni per poterle considerare vicine.

2. **Min_samples**: corrisponde al minimo numero di punti necessari per formare una regione densa.

La determinazione di questi parametri non segue delle regole precise e, nel nostro caso specifico, abbiamo deciso di testare 4 diverse combinazioni di quest'ultimi tenendo in considerazione soltanto la coppia di parametri a cui al rispettivo modello corrispondeva il valore del *silhouette score* più alto. In linea generale si ha che:

- Se il valore di *eps* è troppo basso, molte osservazioni non rientrerebbero all'interno di un *cluster* e verrebbero considerate anomale. Al contrario,

se il valore di *eps* dovesse risultare troppo elevato, la maggior parte dei punti rientrerebbe in un unico *cluster* e non ci sarebbero osservazioni anomale.

- Similmente a quanto accade con il valore di *eps*, un valore troppo elevato associato a *min_samples* non consentirebbe la corretta determinazione delle osservazioni anomale.

È doveroso sottolineare che la metrica utilizzata dal nostro modello *DBSCAN* è quella *euclidea*, è stato dunque necessario normalizzare le osservazioni che altrimenti avrebbero potuto differire di molto in termini di grandezza, rendendo l'algoritmo stesso meno efficace.

- Vantaggi:
  - Non richiede una conoscenza a priori del numero di cluster.
  - Definisce i cluster indipendentemente dalla loro forma.
  - È comprovata la sua robustezza ed efficacia nell'ambito dell'anomaly detection (Dang).
- Svantaggi:
  - Nel caso in cui venga applicato su un insieme di dati la cui densità è variegata, i risultati potrebbero non essere quelli desiderati.
  - I risultati dipendono fortemente dagli iperparametri utilizzati che sono difficili da selezionare empiricamente.
  - Non funziona particolarmente bene se applicato a dati caratterizzati da un'elevata dimensionalità, in quanto ha un peso computazionale particolarmente oneroso.

3.1.1.2 Hierarchical Clustering

Gli algoritmi di *Hierarchical clustering* sono anch'essi basati sull'idea che le osservazioni tra loro vicine condividono una qualche relazione, al contrario di quelle tra loro distanti. Si è scelto in particolare un algoritmo di clustering gerarchico che segua un approccio *bottom-up* o *agglomerativo*. Ogni osservazione viene posizionata all'interno del proprio cluster, dopodiché i cluster vengono tra loro uniti in insiemi più grandi sulla base di una particolare condizione (distanza, incremento di varianza nel cluster, ecc.). Il risultato di tale procedura è rappresentato da un apposito grafico chiamato *dendrogramma*.

- Vantaggi:
  - Non richiede una conoscenza a priori del numero di cluster.
  - Generalmente produce dei cluster meno sensibili alla presenza di valori anomali.
  - Può essere usato come un buon punto di partenza per determinare il numero di cluster da poi applicare ad algoritmi non gerarchici come il *K-means*.
- Svantaggi:
  - Ha una complessità temporale pari a $O(n^3)$ che rende il suo utilizzo poco adatto se applicato a grandi quantità di dati (Dang).
  - Ha difficoltà nello gestire cluster di dimensione e forma tra loro di molto differenti.

3.1.1.3 Time Series K-Means

L'algoritmo di clustering denominato K-Means si basa sull'utilizzo dei centroidi, ovvero dati sintetici non effettivamente osservati, assegnati a ogni cluster. I centroidi vengono utilizzati dall'algoritmo per cercare di minimizzare la distanza tra essi e i punti all'interno del rispettivo cluster. Il numero di cluster (o di centroidi) viene definito dall'utente. Riassumendo, l'algoritmo svolge in ordine i seguenti passi:

1. Vengono selezionati randomicamente k osservazioni da utilizzare come centroidi, dove k rappresenta il numero di cluster.
2. Ogni osservazione viene assegnata al cluster con il centroide più vicino.
3. I centroidi di ogni cluster vengono ricalcolati sulla base delle nuove osservazioni al suo interno. Rispetto allo step numero 1, in questa fase i nuovi centroidi non faranno parte dei dati osservati.
4. Lo step 2 e 3 vengono ripetuti fino a quando la condizione di stop non è soddisfatta. Il criterio di stop è tipicamente scelto tra i seguenti:
   - I centroidi calcolati al punto 3 non variano rispetto ai precedenti.
   - Il numero massimo di iterazioni è stato raggiunto.

    - A seguito dello step 2, i punti sono rimasti nei cluster assegnati precedentemente.

- Vantaggi:
    - È possibile analizzare il centroide di ogni cluster per poterne comprendere i tratti che lo differenziano dai centroidi degli altri cluster. Abbiamo sfruttato tale caratteristica per poter comprendere al meglio l'andamento e la forma delle curve di saldatura sulla quale stavamo lavorando, per poi allenare sui centroidi il modello Isolation Forest.

- Svantaggi (Seither):
    - Necessita della conoscenza a priori del numero di cluster k. Non esiste un metodo specifico per far fronte a ciò e ad oggi il metodo più utilizzato è quello denominato *elbow method*.
    - Il corretto funzionamento dell'algoritmo si basa sull'assunzione che tutti i cluster abbiano la stessa dimensione e varianza. Nella realtà questa assunzione non è quasi mai verificata e questo potrebbe portare l'algoritmo a generare dei cluster errati.

3.1.2 Anomaly Detection

In questa sezione verranno presentati gli algoritmi di anomaly detection utilizzati in questo lavoro e in particolare ne verranno presentati i vantaggi e gli svantaggi.

3.1.2.1 Isolation Forest

L'Isolation Forest è un algoritmo di anomaly detection la cui logica di funzionamento si basa sul classificare come anomali quei punti che, trovandosi molto distanti dagli altri, sono più facili da isolare. Il funzionamento dell'algoritmo è riassumibile nei seguenti step:

1. Partizionamento ricorsivo dello spazio delle feature rappresentato da diversi alberi binari. A ogni step l'algoritmo sceglie randomicamente un attributo e un valore di split da utilizzare come condizione da applicare alle osservazioni.

2. Partizionare lo spazio delle feature fino a quando ogni osservazione è stata isolata.

3. Ritornare al punto 1 fino a quando si sarà sviluppata una foresta di alberi composta da un numero di alberi passato come iperparametro all'algoritmo.

È importante sottolineare come il numero di partizioni necessarie per isolare un punto sia pari all'altezza dell'albero binario descritto precedentemente. Minore è la distanza tra la radice dell'albero e un punto, con maggiore facilità tale punto è stato isolato dagli altri e maggiori sono le probabilità che quest'ultimo sia un'anomalia. Da sottolineare che l'Isolation Forest utilizza un iperparametro chiamato *contamination* come soglia per la classificazione della anomalie. Tale parametro rappresenta la proporzione tra valori anomali e non anomali che ci si aspetta di trovare nel dataset. Nella fase di test del modello, i nuovi punti vengono fatti passare attraverso tutti gli alberi binari precedentemente allenati e viene assegnato loro un punteggio di anomalia che dipende fortemente dalla distanza media di ogni punto dalla radice di tutti gli alberi. Minore è tale distanza media, maggiore sarà il punteggio di anomalia assegnato a tale punto. Il punteggio di anomalia sarà dato dalla seguente formula:

$$s(x, m) = 2^{\frac{-E(h(x))}{c(m)}}$$

dove $E(h(x))$ rappresenta la media delle altezze del punto x calcolata sull'intera foresta di alberi binari, mentre $c(m)$ rappresenta l'altezza media di una foglia dato un albero contenente m osservazioni.

- Vantaggi:
  - L'algoritmo ha complessità lineare e richiede poco spazio in memoria, funziona dunque molto bene se applicato a grandi quantità di dati.
  - Il suo funzionamento è facilmente spiegabile anche a persone prive di un background tecnico.
- Svantaggi:
  - Richiede la conoscenza a priori della proporzione tra valori anomali e non anomali in modo da calcolare la soglia oltre la quale considerare un punto anomalo.

- Qualora i punti anomali dovessero essere molto vicini a quelli non anomali, l'algoritmo potrebbe avere delle difficoltà a identificarli.
- Qualora ci fossero molti punti anomali, quest'ultimi potrebbero addensarsi in uno specifico spazio rendendo la loro identificazione più difficile.

3.1.2.2 Confidence Interval

L'idea sulla quale tale approccio si basa è quella di considerare come anomali i punti al di fuori dell'intervallo di confidenza calcolato utilizzando le diverse osservazioni relative allo stesso spotname. Nello specifico, per ogni spotname è stata calcolata una curva rappresentante il suo andamento medio globale ottenuta calcolando la media di tutte le osservazioni relative allo spotname in ogni istante di tempo. Per ogni punto di tale curva è stato poi calcolato un intervallo di confidenza con un livello di confidenza pari al 99%. Per il calcolo di tale intervallo abbiamo utilizzato il metodo *bootstrap* che ci ha permesso approssimare la distribuzione dei valori in uno specifico istante di tempo, campionando ripetutamente dall'insieme di tutti i valori per ogni istante di cui disponevamo per ogni spotname. Nella Figura 8 è possibile notare come vengano considerati anomali i soli punti al di fuori dell'intervallo di confidenza colorato in verde.

- Vantaggi:
  - All'aumentare della quantità di dati a disposizione, l'intervallo di confidenza si avvicinerà sempre più al vero valore assunto dalla curva in quel punto.
- Svantaggi:
  - Il metodo *bootstrap* richiede molto tempo e risorse computazionali per poter essere eseguito.
  - I valori considerati anomali sono i soli valori all'esterno dell'intervallo di confidenza, nulla vieta che un valore pur trovandosi all'interno di quest'ultimo, non sia in realtà anomalo.

3.2 Descrizione della fase di training

Successivamente alla pulizia del dataset e all'analisi descrittiva dei dati a nostra disposizione, si è proceduto all'applicazione dei modelli di clustering

e di anomaly detection. A causa della grande quantità di dati mancanti per ogni spotname, si è scartata la possibilità di imputare le curve mancanti per non incrementare di molto il bias dei modelli. Si è considerato dunque lo sviluppo di un metodo che identificasse le anomalie in ogni curva attraverso il confronto tra quest'ultima e l'andamento medio globale del rispettivo spotname. In tal modo si sono sfruttate tutte le informazioni a nostra disposizione senza aggiungerne di artificiali. Nella prima fase è stato calcolato per ogni spotname il suo andamento medio globale, ovvero sono state considerate tutte le sue curvature a nostra disposizione e ne è stata fatta una media. Basandoci sulle stesse singole saldature per spotname, per ogni istante di tempo è stata poi calcolata la distribuzione dei valori e da qui abbiamo derivato un intervallo di confidenza con un livello di confidenza pari al 99%. Una dimostrazione di quanto descritto precedentemente è raffigurata nella Figura 8 in cui è possibile osservare l'andamento medio globale dello spotname '60003_0_00' (linea blu) e l'intervallo di confidenza per ogni suo punto (colorato in verde). La media globale e il relativo intervallo di confidenza sono stati poi confrontati con un singolo andamento giornaliero dello stesso spotname e si sono identificati svariati punti anomali (in rosso).

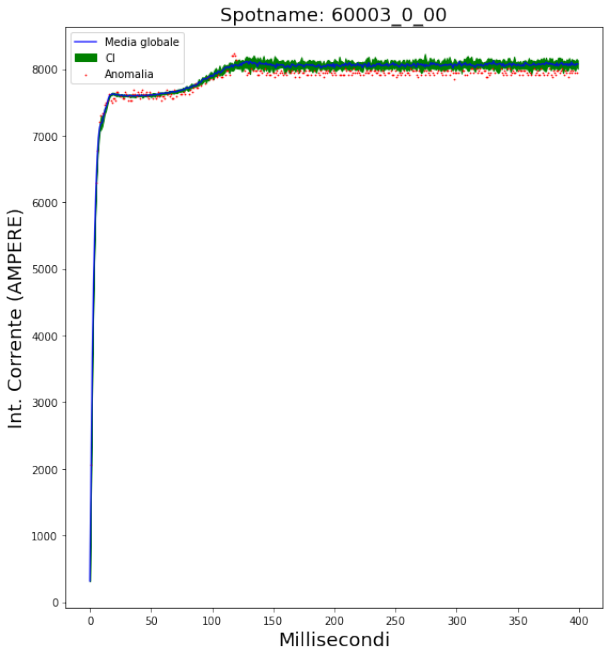

Figura 8: Andamento medio globale di uno spotname con annesso intervallo di confidenza

Non avendo alcun dato certo relativo alla tipologia di saldatura e sul materiale su cui essa è stata svolta, si è pensato di confrontare diversi algoritmi di clustering che ci permettessero di verificare se l'andamento globale medio relativo a ogni spotname fosse simile a quello di un altro spotname ed, eventualmente, capirne il perché. Ancora prima di mostrare i risultati ottenuti in seguito al clustering e la procedura che ci ha portati a essi, è importante sottolineare come nel caso dell'applicazione dell'algoritmo *DBSCAN* e di un algoritmo di *hierarchical clustering*, si è assunto che tutte le curve partissero dallo stesso istante di tempo, ovvero 0. In tal modo è stato possibile utilizzare la distanza euclidea per la formazione dei cluster. Precedentemente all'applicazione dell'algoritmo DBSCAN, i dati sono stati standardizzati. In seguito è stata svolta una ricerca dei parametri in misura limitata alle nostre capacità computazionali. Sono state create due liste: una contenente i valori del parametro *eps* da testare, l'altra contenente i valori del parametro *min_samples*. Per ogni coppia di valori si è salvato il *silhouette score* ottenuto, l'identificativo dello spotname sul quale tale valore è stato ottenuto e i rispettivi parametri. Tali informazioni ci serviranno in seguito per poter

applicare il DBSCAN sui nuovi dati. Nella Figura 9 sono mostrati i risultati ottenuti dal DBSCAN. Da precisare che per un questione di facilità di sviluppo e generalizzazione del codice, il cluster 0 mostrato in figura corrisponde in realtà l valore -1 assegnato dal DBSCAN, ovvero in tale cluster sono presenti gli spotname i cui andamenti medi globali sono stati etichettati come anomali.

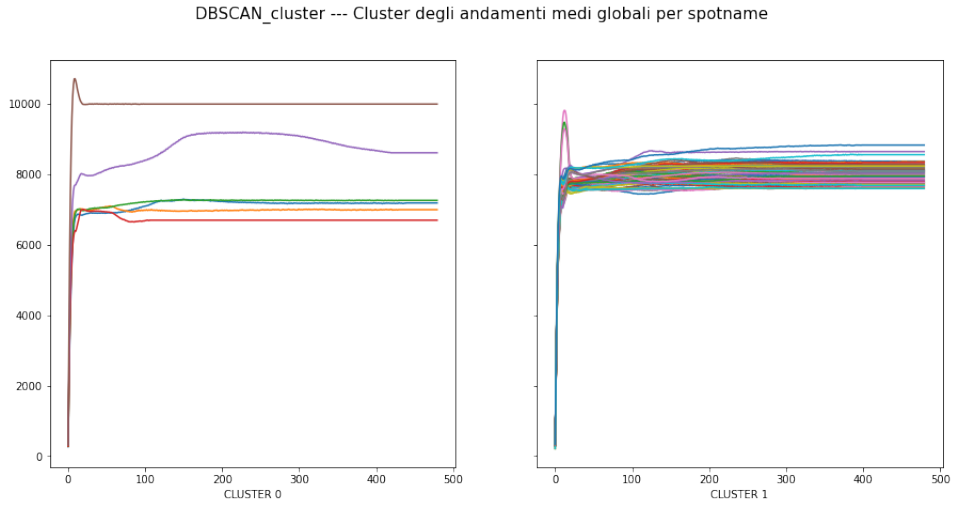

Figura 9: Risultati del DBSCAN applicato sugli andamenti medi globali per spotname

Nella Figura 11 sono invece raffigurati i risultati ottenuti in seguito all'applicazione dell'algoritmo di hierarchical clustering della libreria *scipy*. In seguito a vari test, si è deciso in maniera empirica di impostare una determinata soglia (linea rossa nella Figura 10) che ci consentisse di ottenere 4 cluster. Questo numero è risultato essere il numero di cluster più adatto al nostro problema in quanto, come vedremo in seguito, ci ha consentito di comprendere le 4 caratteristiche che principalmente differenziano gli andamenti delle saldature dei diversi spotname. Il risultato ottenuto dall'applicazione dell'algoritmo di hierarchical clustering è mostrato nel dendrogramma nella Figura 10.

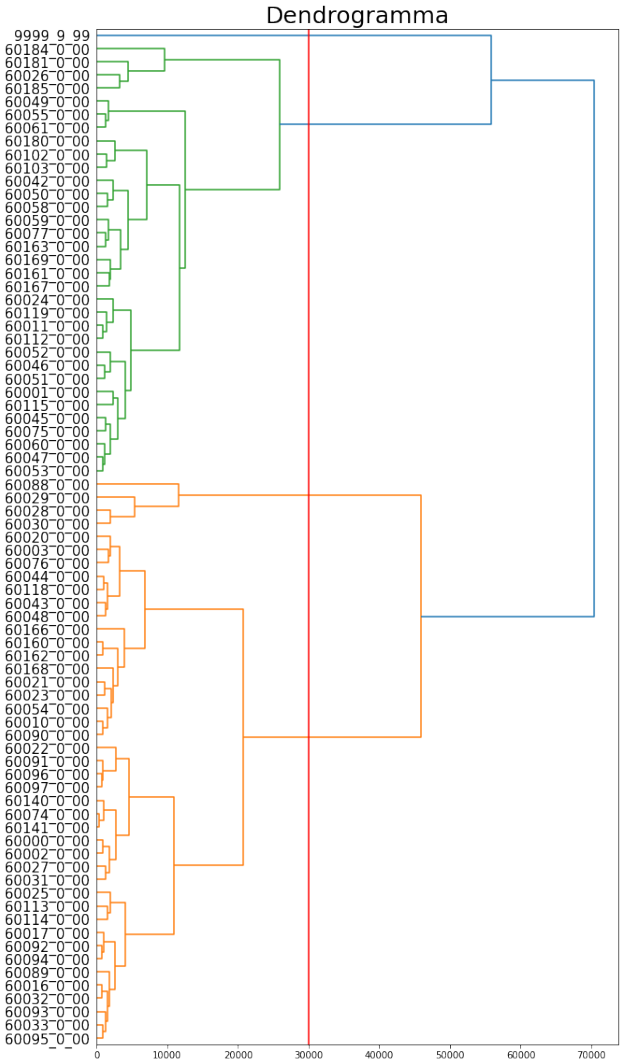

Figura 10: Dendrogramma rappresentante i cluster degli andamenti globali ottenuti

Lo stesso risultato è stato riassunto nella Figura 11 dove ricordiamo che all'interno di ogni cluster, il colore differenzia gli andamenti medi globali per spotname.

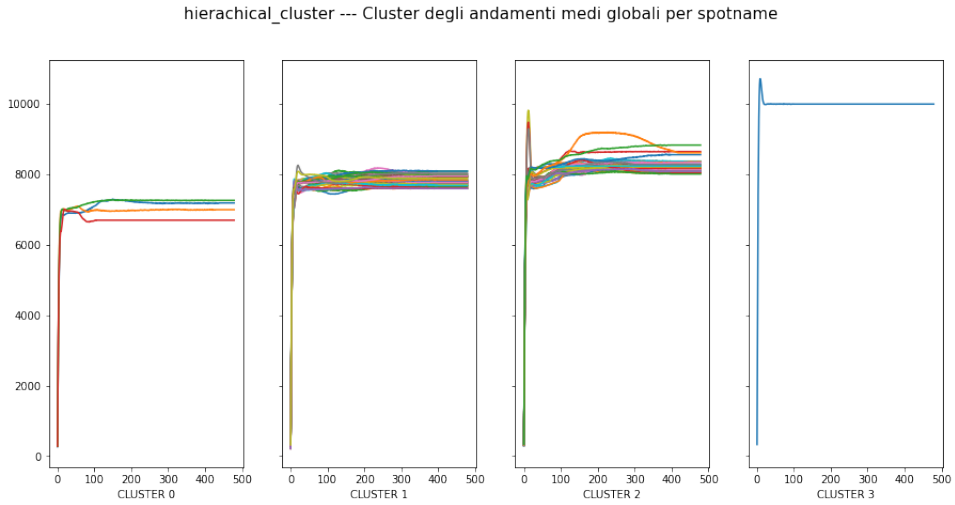

Figura 11: Risultati dell'algoritmo di Hierarchical clustering applicato sugli andamenti medi globali per spotname

Per l'applicazione del DBSCAN e dell'algoritmo di hierarchical clustering è stata utilizzata la distanza euclidea come metrica per il calcolo della vicinanza tra le osservazioni. Nel caso specifico dell'hierarchical clustering, è stato inoltre utilizzato il metodo *ward* per la definizione di un metodo di aggregazione. Quest'ultimo metodo unisce due cluster se e solo se tale unione porta a un incremento della varianza interna del cluster finale inferiore a quello che si otterrebbe unendo uno dei due cluster iniziali con qualsiasi altro cluster. Specificare la metrica e il metodo di aggregazione utilizzati è importante considerando che i risultati ottenuti dagli algoritmi di clustering dipendono fortemente da entrambi. La distanza euclidea non è tipicamente consigliata nell'ambito delle time series, in quanto queste ultime spesso hanno punti iniziali differenti. Due time series identiche ma una registrata con un qualche secondo di anticipo o ritardo rispetto all'altra, potrebbero essere considerate molto distanti (dissimili) se si utilizza la distanza euclidea. Nonostante questo abbiamo proceduto con le nostre analisi e per poter verificare l'eventuale correttezza o meno della nostra assunzione, si è deciso di applicare l'algoritmo K-Means che a sua volta utilizzava l'algoritmo denominato *Dynamic Time Warping (DTW)* per il confronto di due time series (o curve nel nostro caso) raccolte a velocità diversa. Nella Figura 12 notiamo come i risultati ottenuti dall'applicazione dell'algoritmo K-Means assieme all'algoritmo DTW, non si discostino di molto da quelli ottenuti utilizzando

approcci più classici come il DBSCAN e l'algoritmo di hierarchical clustering con metrica euclidea.

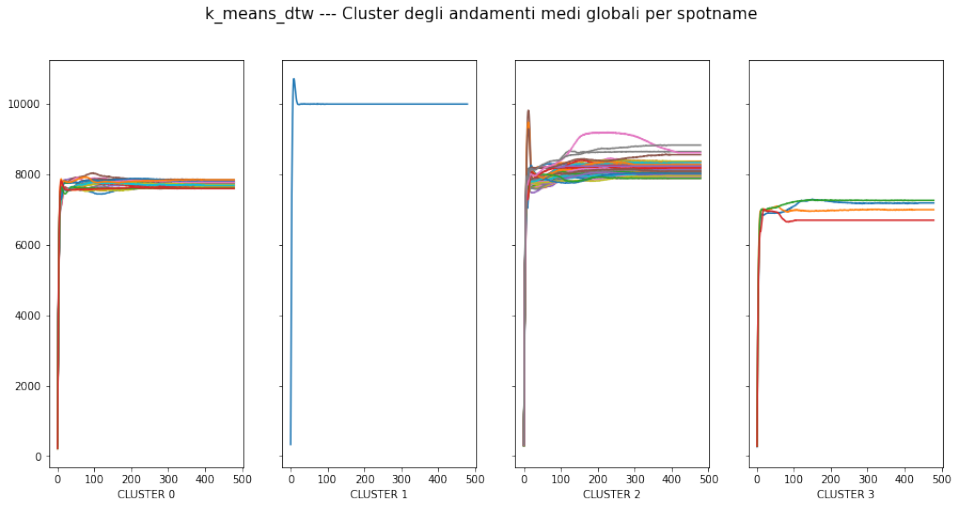

Figura 12: Risultati dell'algoritmo K-Means (DTW) applicato sugli andamenti medi globali per spotname

L'applicazione di quest'ultimo algoritmo è stata per noi particolarmente importante in quanto ci ha permesso di estrarre i centroidi di ogni cluster, ottenendo dunque delle rappresentazioni riassuntive dei fattori caratterizzanti di ognuno di essi. Nella Figura 13 sono raffigurati i centroidi di ogni cluster ed è importante soffermarsi su come essi differiscano l'uno dell'altro, in modo da comprendere il perché si è in seguito deciso di allenare il modello di anomaly detection denominato *Isolation Forest* su di essi.

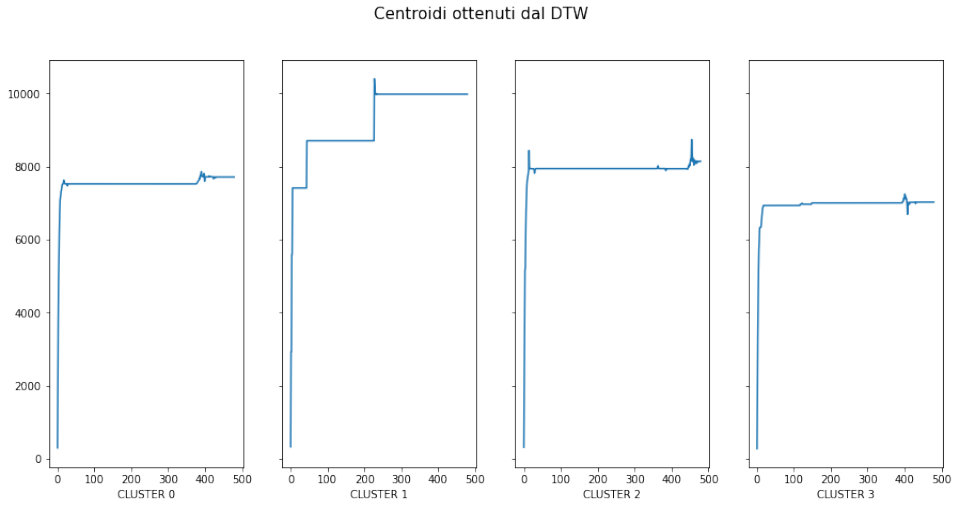

Figura 13: Centroidi ottenuti dall'applicazione dell'algoritmo K-Means (DTW)

La parte finale della fase di training è consistita nell'allenare su ogni singolo centroide di ogni cluster un diverso modello di Isolation Forest. In tal modo si è ottenuto per ogni cluster un modello di Isolation Forest in grado comprendere le caratteristiche delle curve di saldatura al suo interno, e dunque in grado di identificare eventuali curve anomale non appartenenti a uno specifico cluster. Tali modelli sono stati salvati in un dizionario le cui chiavi corrispondono all'identificatore dello spotname e i cui valori corrispondono al rispettivo modello. Tale dizionario sarà utile nelle fasi successive, particolarmente nella fase di deploy in cui si è simulato lo sviluppo di un API di alto livello che dato l'identificatore di uno spotname e una curva di saldatura, consente di identificarne le eventuali anomalie al suo interno. Il valore dell'iperparametro *contamination* è stato settato per tutti i modelli a 0.02, in quanto empiricamente si è visto che ipotizzando un numero di anomalie in media pari al 2% delle osservazioni totali l'algoritmo era in grado di identificare gli andamenti particolari, ovvero quegli andamenti nella curva che si comportavano in modo differente dal resto. Si ricorda che tale parametro è facilmente modificabile e che dunque può essere adattato in base alle singole esigenze. Se si necessita un modello più sensibile alle anomalie, si potrebbe pensare di alzare leggermente tale valore considerando però che potrebbero in conseguenza aumentare il numero dei falsi positivi. Nella Figura 14 sono rappresentate le anomalie ottenute da un modello di Isolation Forest allenato

sul centroide del cluster 3 e applicato a una misurazione giornaliera di uno spotname facente parte dello stesso cluster. Notiamo come il modello sia stato in grado di identificare un incremento sospetto dell'amperaggio a partire dall'80esimo millisecondo fino al 130esimo circa.

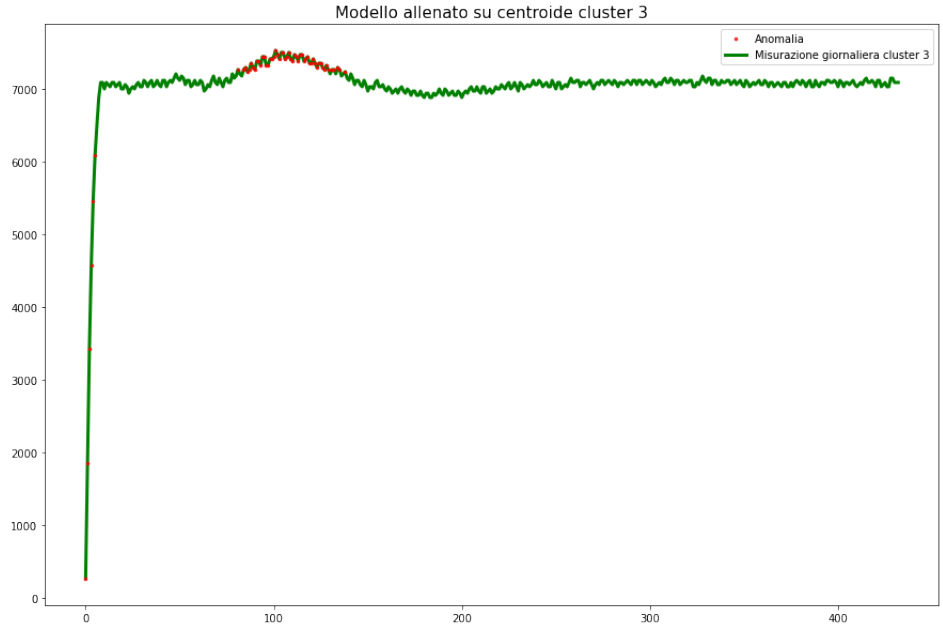

Figura 14: Risultato dell'applicazione dell'I.F. su una curva appartenente allo stesso cluster

Nella Figura 15 è invece rappresentata la situazione opposta a quella precedente. Mantenendo la stessa misurazione giornaliera presa da uno spotname appartenente al cluster numero 3, su di essa è stato testato un modello allenato sul centroide del cluster numero 2. Come possiamo notare, il modello non ha notato anomalie a partire dall'80esimo millisecondo fino 130esimo, come accadeva in Figura 14. Ciò significa che tale andamento non è sospetto per una misurazione giornaliera di uno spotname appartenente al cluster 2.

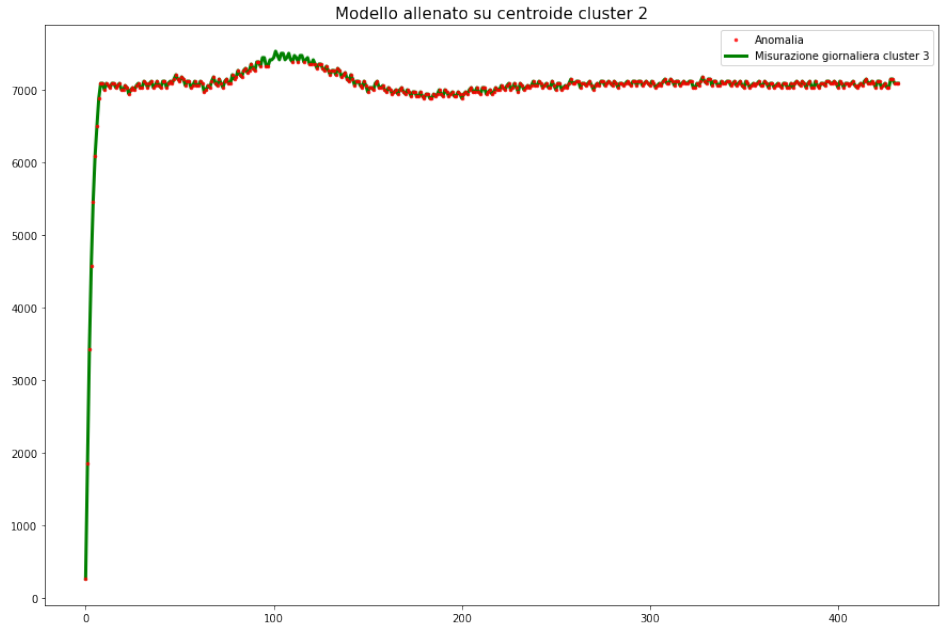

Figura 15: Risultato dell'applicazione dell'I.F. su una curva appartenente a un cluster diverso

3.3 Informazioni acquisite

Come è possibile osservare in Figura 12 e in Figura 13, vi sono delle differenze importanti negli andamenti di ogni spotname che ne caratterizzano l'appartenenza o meno a un determinato cluster. Considerando la Figura 12, notiamo:

- CLUSTER 0: le curve medie globali degli spotname appartenenti a tale cluster sono caratterizzate dall'assumere durante tutta la durata della saldatura, un amperaggio mai superiore agli 8000A. Notiamo come nella fase che va dall'inizio della saldatura fino ai primi 30 millisecondi circa, fase che si presuppone essere di riscaldamento della testina per la saldatura, l'amperaggio rimane sempre al disotto degli 8000A, situazione che si ripete soltanto nel cluster numero 3.

- CLUSTER 1: le curve medie globali degli spotname appartenenti a tale cluster sono caratterizzate dall'assumere durante tutta la durata della saldatura valori molto elevati, attorno ai 10000A circa. Le curve

appartenenti a questo cluster sono dello spotname 9999_9_99, uno spotname creato artificialmente sostituendo gli identificatori degli spotname mancanti con il medesimo codice alfanumerico.

- CLUSTER 2: le curve medie globali degli spotname appartenenti a tale cluster sono caratterizzate dall'avere valori elevati nella fase di riscaldamento della testina (i primi 30 millisecondi circa) e da un'ulteriore innalzamento dell'amperaggio a partire dal millisecondo 100esimo fino al millisecondo 130esimo circa.

- CLUSTER 3: le curve medie globali degli spotname appartenenti a tale cluster sono caratterizzate dall'avere dei valori sempre ben al disotto degli 8000A e da una decrescita dell'amperaggio subito dopo la fase di riscaldamento della testina, a cui tipicamente segue una stabilizzazione dell'amperaggio stesso.

4 Testing

Nonostante il nostro problema fosse di tipo non supervisionato, si è deciso lo stesso di inserire una fase di testing all'interno della quale applicare i modelli su nuovi dati. L'obiettivo di tale fase è stato quello di mostrare la capacità dei modelli di essere in grado di comprendere gli andamenti caratteristici di una curva e, di conseguenza, di classificarla come anomala o meno. I nuovi dati sono stati ottenuti a partire da un campione stratificato estratto nella fase di preparazione del dataset.

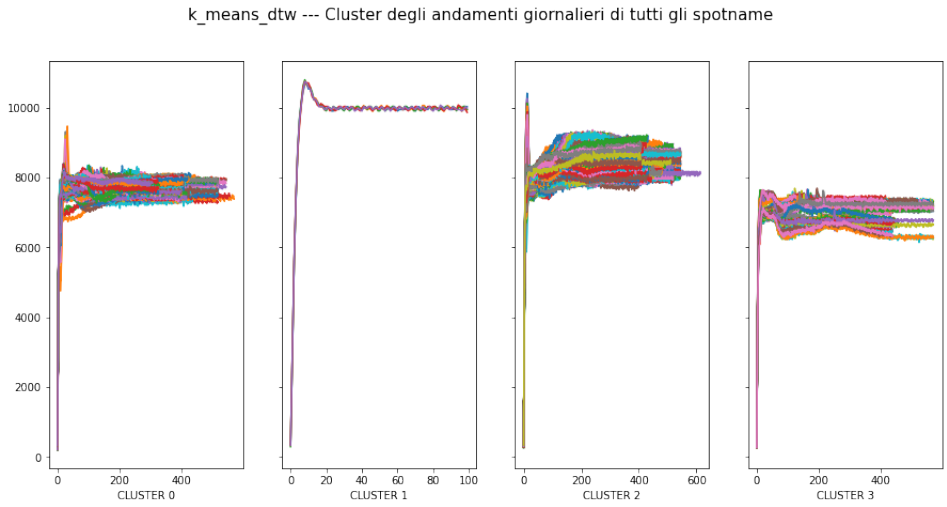

Figura 16: Risultati K-Means sui dati di test

Figura 17: Risultati DBSCAN sui dati di test

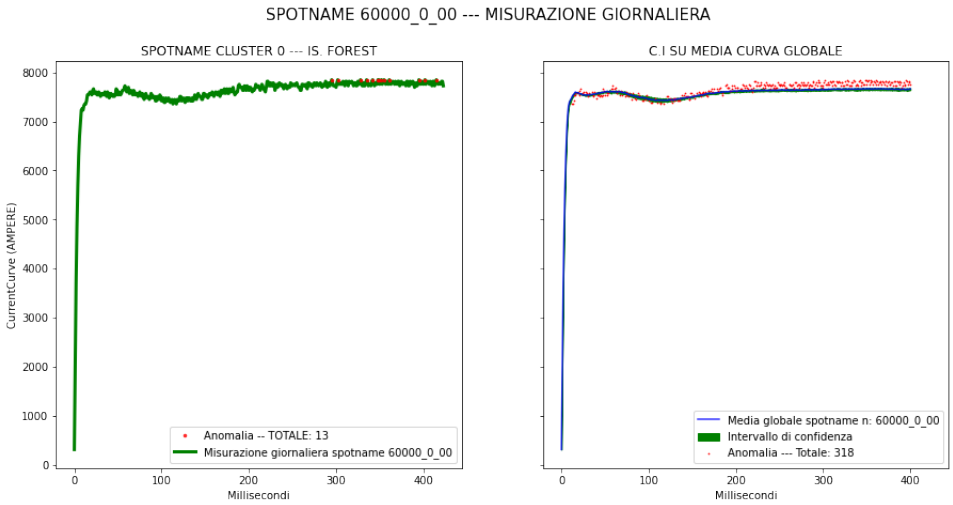

Figura 18: Risultati I.F. e C.I. sui dati di test (considerando uno spotname)

Nella Figura 16 si mostra come il modello K-Means sia stato capace di suddividere ogni spotname nel rispettivo cluster basandosi soprattutto sull'andamento dal 150esimo al 250esimo millisecondo circa, delle curve di saldatura. Nella Figura 17 è mostrato un risultato ottenuto dal modello DBSCAN

i cui iperparametri sono stati identificati durante la fase di training. Notiamo come le curve ritenute anomale (grafico a sinistra) abbiano un andamento divergente rispetto a quelle non ritenute tali (grafico a destra). Nella Figura 18 sono rappresentati i risultati ottenuti dai modelli Isolation Forest e di Confidence Interval applicati su un solo spotname. Per la visione dei grafici relativi a ogni curva di saldatura nel campione, si rimanda alla visione del rispettivo notebook. Al termine di questa fase, per ogni coppia di modelli è stata calcolata la percentuale di volte in cui le previsioni di entrambi concordavano. Si sono ottenuti i seguenti risultati:

- DBSCAN - CONFIDENCE INTERVAL: 91.37%
- DBSCAN - ISOLATION FOREST: 34.50%
- CONFIDENCE INTERVAL - ISOLATION FOREST: 37.24%

5 Deploy

Per la fase di deploy, ovvero la fase relativa all'applicazione dei modelli sviluppati su nuovi dati, si è simulata una chiamata a un *API*. A quest'ultima viene passato in input un file *JSON* e in seguito a varie rielaborazione, l'API restituisce come output un grafico su cui sono segnati i punti della curva identificati come anomali e il numero di modelli (su 3) che concordano nel dire che tale curva è anomala. All'utente è data la scelta di decidere se si vogliono mostrare o meno graficamente le eventuali anomalie trovate, e nel primo caso otterrebbe un grafico simile a quello nella Figura 19.

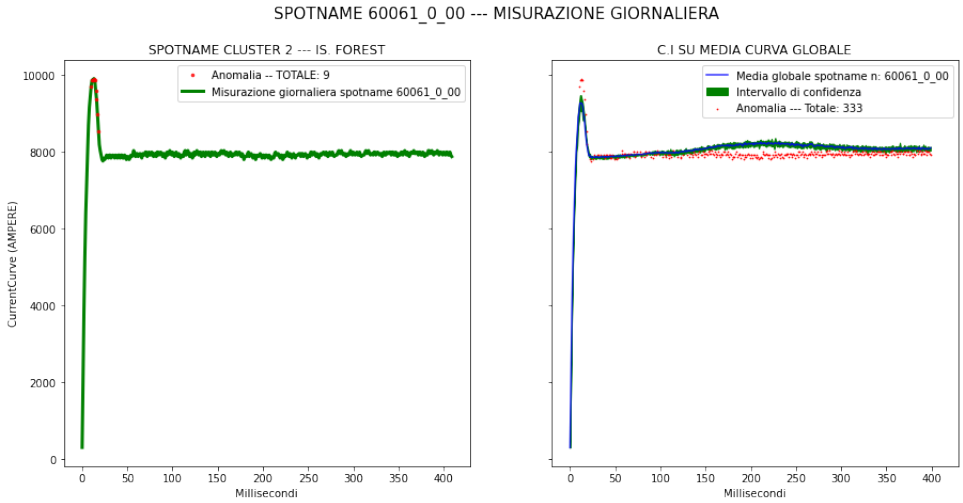

Figura 19: Grafico in output dato dall'API

Come è possibile notare dalla figura sopra riportata, a ogni curva è associato il relativo numero di anomalie identificate dai rispettivi modelli. Tale numero riveste un ruolo fondamentale in quanto, se superiore a una specifica soglia, determina la classificazione della rispettiva curva come anomala o meno. Una nota importante relativa a questa fase è che il codice è stato scritto in modo tale da poter essere facilmente esposto come API al pubblico attraverso una sua implementazione tramite framework web come *Flask* o *Django*.

6 Risultati

I modelli sviluppati nelle fasi precedentemente descritte sono stati applicati sulla totalità dei dati relativi alle singole curve di saldatura per ogni spotname, e in seguito si è calcolato il numero di punti anomali che ogni singolo modello identificava. Per ogni curva giornaliera si sono dunque ottenuti due valori indicanti il numero di anomalie ricavati dai modelli Isolation Forest e di Confidence Interval, più un terzo valore (-1 o 0) relativo alla previsione del DBSCAN. In base a delle soglie manualmente impostate, si è verificato che i primi due valori non fossero eccessivamente elevati e qualora lo fossero, si è assegnato il valore -1 indicante che tale curva è per noi anomala. La scelta di restituire un risultato binario (-1 o 0) per ogni modello anziché un valore di anomaly score, è data dall'idea di ridurre il numero di fattori da considerare nel momento in cui si è in presenza di un'anomalia. Questo consente all'impiegato addetto al controllo qualità di incrementare la velocità decisionale ed evitare che il pezzo sui cui la saldatura sta avvenendo venga ulteriormente danneggiato.

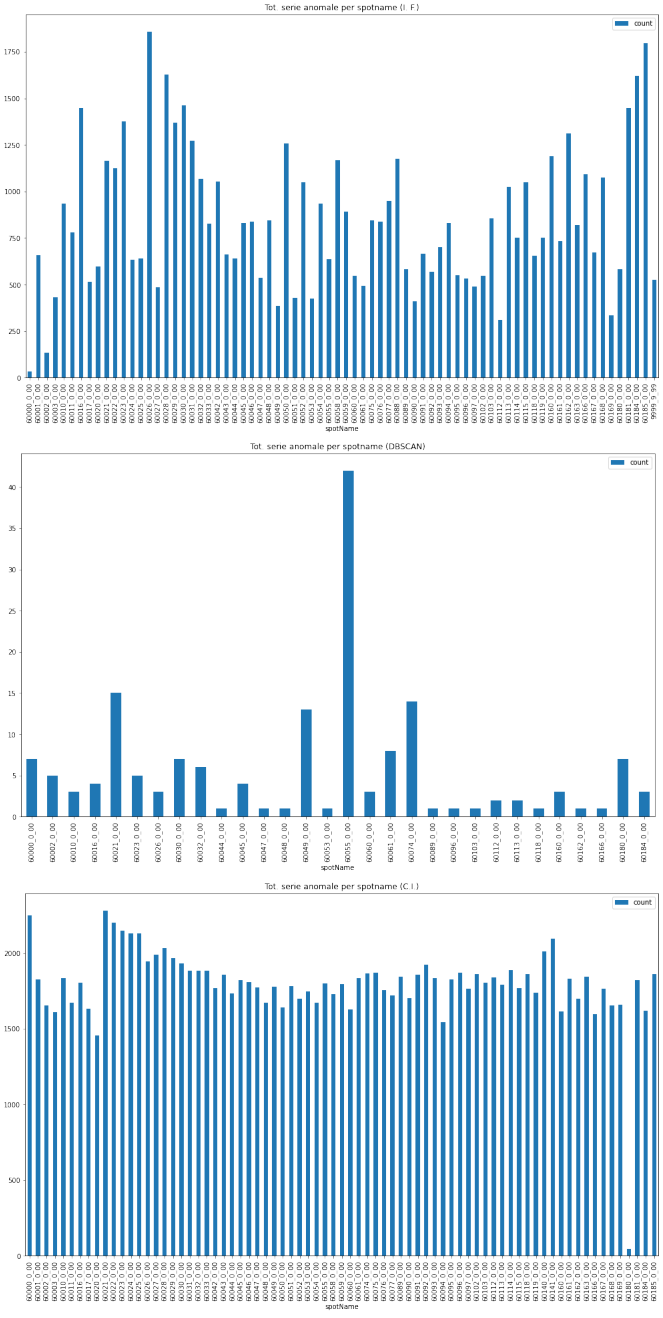

Figura 20: Numero di anomalie per ogni spotname

Nella Figura 20 sono rappresentati tre grafici ognuno contenente il numero di curve classificate come anomale per ogni spotname ognuno relativo ad uno specifico modello. Premettendo che tali risultati dipendono fortemente dalle soglie di anomalie impostate manualmente, notiamo come il modello basato sugli intervalli di confidenza rilevi un numero di curve anomale per spotname maggiore rispetto agli altri due modelli, essendo quest'ultimo intrinsecamente più restrittivo rispetto agli altri. Sempre all'interno della stessa figura, è osservabile come il modello DBSCAN non abbia rilevato nessuna curva anomala in molti degli spotname, mentre il modello Isolation Forest si è dimostrato più moderato rispetto agli altri due. Supponiamo che tale incapacità da parte del modello DBSCAN di non essere stato in grado di rilevare alcuna anomalia in alcuni spotname, sia dovuta alla non adeguata ricerca dei parametri più adatti per il modello. Tale problema è stato ulteriormente discusso nella sezione 7.2.

Nelle immagini che seguiranno sono mostrati i risultati ottenuti.

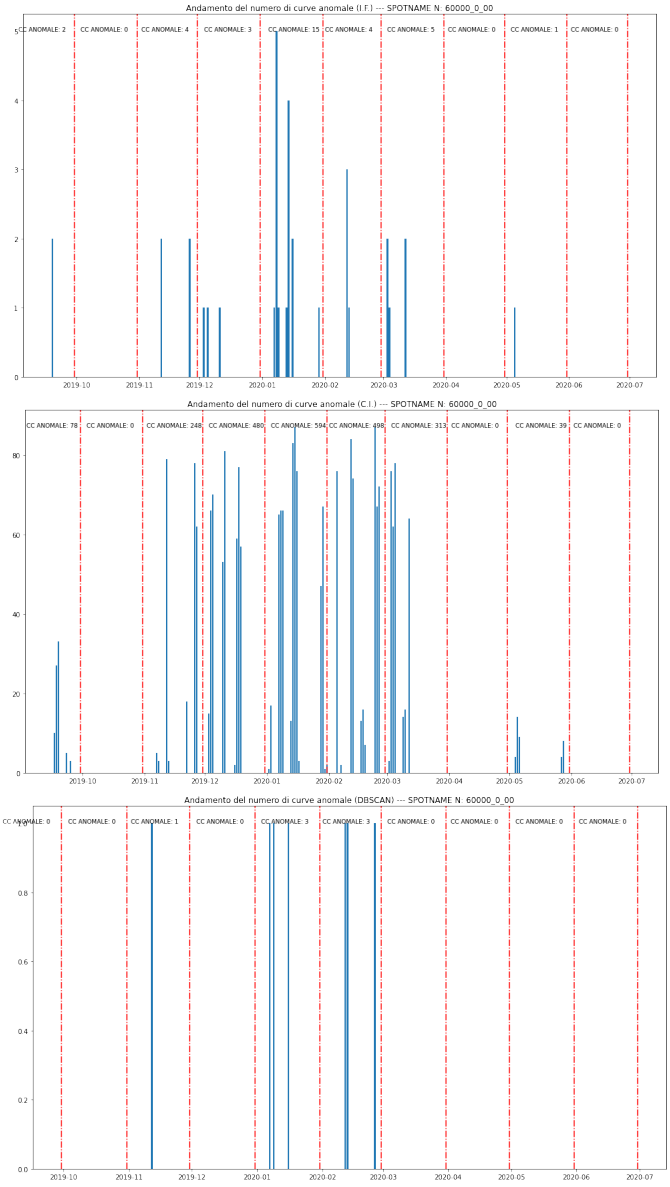

Figura 21: Andamento del numero di anomalie per lo spotname 60000_0_00

In Figura 21 ogni grafico rappresenta l'andamento del numero di curve anomale nel tempo per un modello specifico e per uno specifico spotname (n. 60000_0_00). All'interno del notebook è presente un grafico di questo

tipo per ogni spotname, mentre nel report si è deciso di inserirne solo uno rappresentativo anche degli altri onde evitare di eccedere con il numero di immagini. Notiamo come da Dicembre 2019 a Marzo 2020 tutti i modelli concordano nel dire che c'è stato un incremento nel numero di anomalie. A detta di alcuni operatori nell'ambito automotive, la manutenzione degli strumenti per saldare avviene tipicamente ogni inizio anno dunque quest'ultimi manterrebbero delle performance elevate durante i mesi iniziali e intermedi dell'anno (subito dopo la manutenzione), per poi peggiorare man mano nel tempo a causa del loro utilizzo, determinando un incremento nel numero di anomalie nei mesi finali dell'anno. Si raccomanda dunque l'azienda di porre una maggiore attenzione alla qualità dei telai soprattutto verso fine anno.

7 Conclusioni

Nella seguente sezione sono riportate alcune osservazioni che si ritiene essere di fondamentale importanza per l'azienda per ottenere un miglioramento dei modelli predittivi sviluppati, ottimizzando dunque il proprio processo produttivo.

7.1 Consigli per l'azienda

Sulla base del lavoro di analisi svolto, consigliamo all'azienda di fornire maggiori informazioni riguardo il processo di produzione. Si suppone che modelli maggiormente accurati possano essere sviluppati se si dispone di ulteriori informazioni quali:

- *materiale soggetto a saldatura*;
- *spessore delle lamiere*: consente di calcolare in maniera precisa gli intervalli di corrente elettrica che non causano sbavature;
- *diametro dell'elettrodo*: è alla base del mantenimento di una forma costante e compatta dell'arco di saldatura;
- *numero di saldature per elettrodo*: a un maggior utilizzo di un elettrodo corrisponde una maggiore probabilità di incorrere in problemi e difetti di saldatura (instabilità dell'arco, difficoltà di innesco);
- *storico manutenzione dei bracci robotici (o altro)*: si potrebbe impostare una soglia di tollerabilità nel numero di anomalie nel tempo e, in base allo storico della manutenzione del dispositivo atto alla saldatura, avvisare l'operatore se è necessario o meno svolgere la manutenzione se non si vuole oltrepassare tale soglia.

In aggiunta a quanto elencato sopra, consigliamo all'azienda di raccogliere dati che abbiano una continuità nel tempo evitando di avere lunghi intervalli temporali (giorni, settimane, mesi) su cui non si hanno informazioni. Tale sporadicità nei dati relativi alle curve di saldatura per ogni spotname, ha reso per noi impossibile l'applicazione di alcuni modelli molto performanti di analisi predittiva (LSTM, ecc.) se non attraverso la creazione di dati sintetici. Tale approccio è stato però da noi scartato immediatamente in quanto avrebbe causato un incremento nei bias dei modelli. Ricordiamo

infine che il codice è stato appositamente sviluppato per essere reso flessibile e adattabile alle molteplici esigenze aziendali. Attraverso la modifica dei diversi parametri presenti al suo interno, è possibile rendere i modelli di rilevamento delle anomalie più o meno sensibili. Ad esempio, è possibile pensare a modi più performanti e flessibili di calcolare le soglie che determinano il classificare o meno come anomala una curva di saldatura.

7.2 Sviluppi futuri

Riteniamo che a partire dal lavoro svolto il progetto possa essere ulteriormente migliorato e ampliato attraverso l'uso di hardware più potente e lo sviluppo di un'interfaccia grafica. Nel primo caso diventerebbe possibile testare un maggior numero di iperparametri, come nel caso degli algoritmi DBSCAN e K-Means, con il fine di sviluppare modelli maggiormente performanti che tengano conto delle singole caratteristiche di ogni spotname. Relativamente all'interfaccia grafica si potrebbe pensare allo sviluppo di un applicativo da installare su dispositivi tablet android/iOS o di una semplice pagina web. Tale interfaccia verrebbe utilizzata dagli operatori atti al controllo della qualità consentendo loro una più semplice e immediata verifica delle curve di saldatura e delle rispettive statistiche descrittive, grazie all'utilizzo di grafici aggiornati in tempo reale.

8 Bibliografia

[1] Quali materiali sono utilizzati nella fabbricazione di carrozzerie?, May 2020. URL https://avtotachki.com/it/tehnicheskaya-stat-ya-kakie-materialy-ispol-zuyutsya-v-proizvodstve-kuzovov/.

[2] Saldatura a resistenza, Oct 2021. URL https://www.techmec.it/saldatura-a-resistenza/.

[3] Shilpa Dang. Performance evaluation of clustering algorithm using different datasets. *IJARCSMS*, 3:167–173, Jan 2015.

[4] Julian Seither. Anomaly detection: (dis-)advantages of k-means clustering, Jul 2017. URL https://www.inovex.de/de/blog/disadvantages-of-k-means-clustering/.

[5] Xinge Zhang, Fubin Yao, Zhenan Ren, and Haiyan Yu. Effect of welding current on weld formation, microstructure, and mechanical properties in resistance spot welding of cr590t/340y galvanized dual phase steel. *Materials (Basel, Switzerland)*, 11(11):E2310, Nov 2018. ISSN 1996-1944. doi: 10.3390/ma11112310.