



**UNIVERSITÀ
DEGLI STUDI
DI PADOVA**



DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA BIOMEDICA

**IMPLEMENTAZIONE DI UN ALGORITMO PER LA RILEVAZIONE
DEI GUASTI NEI SISTEMI DI INFUSIONE DI INSULINA PER I
PAZIENTI DIABETICI**

Relatore: Prof. Simone Del Favero

**Laureanda: Pertegato Laura
N. matricola: 1167560**

Correlatrice: Dott.ssa Elena Idi

ANNO ACCADEMICO 2021 – 2022

Data di laurea: 16 novembre 2022

Sommario

Il presente elaborato ha come tematica centrale l'implementazione dell'algoritmo sviluppato da Howsmon et al per la rilevazione dei guasti nei set di infusione della pompa insulinica.

Nel primo capitolo, d'introduzione all'argomento, viene illustrato il Diabete di Tipo 1, ovvero la patologia per cui può essere utilizzato il pancreas artificiale come strumento di cura. Successivamente, vengono elencati i vari test che si possono eseguire per ottenere un'accurata diagnosi e viene presentata una panoramica generale sulla diffusione della patologia nel mondo. Nella conclusione della prima parte, è descritto nello specifico il sistema di pancreas artificiale articolato nelle sue varie componenti: sensore di monitoraggio continuo del glucosio, algoritmo di controllo e pompa insulinica.

Nel secondo capitolo, è delineato l'algoritmo *Losses in Infusion Set Actuation* (LISAs) spiegandone le metriche sviluppate per la rilevazione dei guasti nella pompa insulinica. Le metriche si basano sulla elaborazione dei dati di glucosio e insulina di ogni paziente e si dividono in *Glucose Fault Metric* (GFM), *Insulin Fault Metric* (IFM) e *Glucose Slope* (GS). Il capitolo si conclude con la spiegazione delle performance dell'algoritmo ottenute da Howsmon et al.

Nel terzo capitolo, infine, viene eseguita l'implementazione dell'algoritmo LISAs illustrando i dati utilizzati, la struttura del programma e l'analisi dei valori ottenuti dalle metriche sopracitate. L'elaborato si conclude con un confronto fra le performance ottenute dall'algoritmo sviluppato da Howsmon et al e quello da me implementato.

Indice

1. Introduzione.....	7
1.1 Il diabete di Tipo 1.....	7
1.2 Metodi di diagnosi.....	8
1.3 Il diabete nel mondo.....	10
1.4 Il pancreas artificiale.....	11
1.4.1 Componenti del pancreas artificiale.....	11
2. Algoritmo di rilevazione dei guasti nei set d'infusione del Pancreas Artificiale (LISAs-Losses in Infusion Set Actuation)...	14
2.1 Metriche per il riconoscimento dei LISAs.....	14
2.1.1 <i>Glucose Fault Metric (GFM)</i>	15
2.1.2 <i>Insulin Fault Metric (IFM)</i>	15
2.1.3 <i>Glucose Slope (GS)</i>	18
2.2 Performance dell'algoritmo.....	18
3. Implementazione dell'algoritmo per il riconoscimento dei LISAs.....	20
3.1 Dati utilizzati.....	20

3.2 Struttura del programma.....	21
3.3 Analisi dei parametri.....	22
3.3.1 Sviluppo <i>Glucose Fault Metric</i>	22
3.3.2 Sviluppo <i>Glucose Slope</i>	24
3.3.3 Sviluppo <i>Insuline Fault Metric</i>	24
3.3.4 Grafici.....	25
3.4 Generazione di allarmi e performance dell'algoritmo.....	26
Conclusioni.....	28
Riferimenti.....	29
Appendice A.....	30

Capitolo 1

1. Introduzione

Il diabete è una patologia metabolica che si presenta quando il pancreas non è più in grado di produrre insulina o quando il corpo non riesce ad utilizzare l'insulina in maniera adeguata [1].

L'insulina è un ormone fondamentale per la regolazione dei livelli di glucosio nel sangue, essa infatti permette il trasporto del glucosio del cibo ingerito dal sangue alle cellule dove lo trasformano in fonte d'energia.

Se l'organismo si trova impossibilitato a produrre o ad utilizzare insulina la concentrazione di glucosio nel sangue subisce un aumento notevole, generando iperglicemia che, prolungata nel tempo, può portare a importanti complicazioni all'organismo come cecità, cardiopatia, insufficienza renale e, in condizioni critiche, alla morte [1]-[2].

1.1 Diabete di tipo 1

Il diabete di tipo 1 è il risultato dell'azione di eliminazione delle cellule Beta presenti nelle isole pancreatiche, anche dette isole di Langer-Hans, da parte del sistema immunitario.

In sostanza, le cellule T e B del sistema immunitario si inseriscono nelle isole di Langer-Hans e rilasciano proteine infiammatorie, come citochine e chemochine, che vanno a interagire con le cellule Beta fino a distruggerle [3].

In questo modo il pancreas perde le sue capacità di produzione di insulina e di regolazione della glicemia portando il soggetto affetto all' iperglicemia.

Per controllare la glicemia, quindi, il paziente avrà bisogno di costanti iniezioni di insulina che verranno fatte da un sistema esterno. La maggior parte dei soggetti affetti segue una terapia insulinica intensiva convenzionale basata sulla somministrazione manuale di insulina attraverso utilizzo di penne insuliniche; negli

ultimi anni, però, stanno prendendo piede sistemi più automatizzati, come il Pancreas Artificiale (AP), descritto più dettagliatamente nella sezione 1.4.

Generalmente questo tipo di diabete sopraggiunge più frequentemente nell'età dell'infanzia e dell'adolescenza e, sebbene le cause esatte non siano note, si attribuisce gran parte della responsabilità a fattori ambientali e genetici [3].

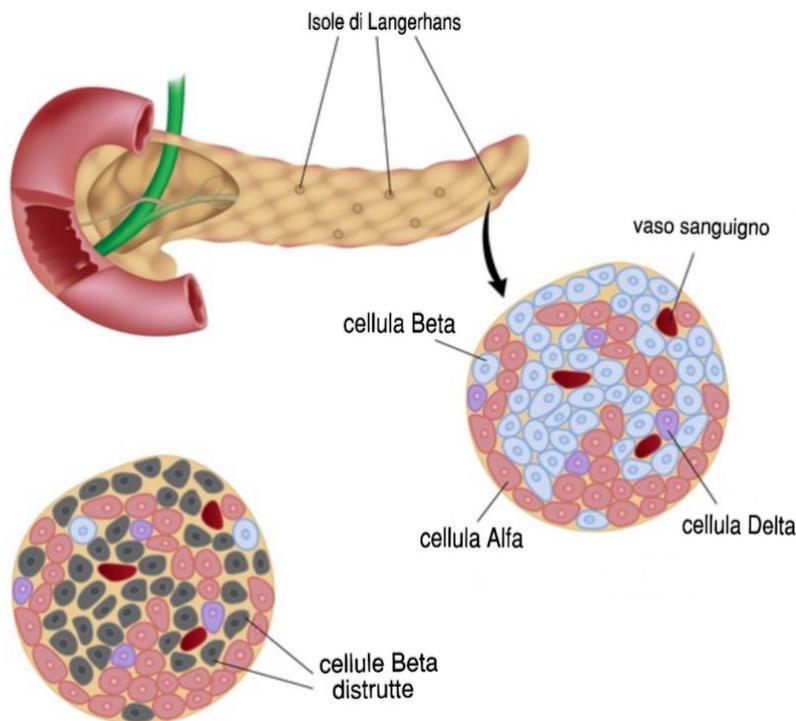


Figura 1 Confronto tra pancreas di un soggetto con cellule Beta sane e pancreas di un soggetto affetto da diabete di Tipo 1

1.2 Metodi di diagnosi

La ricerca del diabete in un individuo viene spesso suggerita in caso si presentino sintomi come: eccessiva sete, perdita di peso, stanchezza, infezioni ricorrenti e pre-coma.

L' International Diabetes Federation (IFD) assieme all' Organizzazione Mondiale della Sanità ha proposto dei criteri diagnostici standard per la rilevazione del *prediabete*¹ e del diabete [4]-[5].

Questi criteri sono:

- Test di tolleranza orale al glucosio (OGTT) da 75 grammi con misurazione del glucosio a digiuno e a due ore per rilevare IGT e IFT;
- Concentrazione di glucosio plasmatico venoso;
- *HbA1c*².

Nella tabella seguente presa da [4] vengono descritti brevemente i valori sopra i quali i vari test rilevano una condizione di diabete o di prediabete.

Test	Diabetes Should be diagnosed if ONE OR MORE of the following criteria are met	Impaired Glucose Tolerance (IGT) Should be diagnosed if BOTH of the following criteria are met	Impaired Fasting Glucose (IFG) Should be diagnosed if THE FIRST OR BOTH of the following are met
 Fasting plasma glucose	≥7.0 mmol/L (126 mg/dL)	<7.0 mmol/L (126 mg/dL)	6.1 – 6.9 mmol/L (110 – 125 mg/dL)
or			
 Two-hour plasma glucose after 75g oral glucose load (oral glucose tolerance test (OGTT))	≥11.1 mmol/L (200 mg/dL)	≥7.8 and <11.1 mmol/L (140–200 mg/dL)	<7.8 mmol/L (140 mg/dL)
or			
 HbA1c	≥48 mmol/mol (equivalent to 6.5%)		
or			
 Random plasma glucose in the presence of symptoms of hyperglycaemia	≥11.1 mmol/L (200 mg/dL)		

Tabella 1 Descrizione dei valori sopra i quali i test rilevano una condizione di diabete o prediabete tratto da [4].

¹ *Prediabete* è una condizione dove la concentrazione di glucosio a digiuno (IFG) e/o la tolleranza al glucosio (IGT) del soggetto sono alterate ma non così tanto da essere considerato diabete.

² *HbA1c* è un test sulla concentrazione di emoglobina glicata A1c nel sangue. L'emoglobina glicata è un tipo di emoglobina che si lega al glucosio, per cui stimando la sua concentrazione riusciamo a stimare anche la concentrazione di glucosio nel sangue.

1.3 Diabete nel mondo

L' *International Diabetes Federation* (IDF), nel 2021, ha stimato che più di 537,8 milioni di persone nel mondo siano affette da diabete di cui 1,2 milioni siano bambini e adolescenti. Questo numero si presuppone sia destinato ad aumentare, soprattutto nei paesi a basso-medio reddito fino ad arrivare a 786 milioni di diabetici entro il 2045.

Il diabete, essendo una delle maggiori malattie riscontrate al mondo, comporta notevoli costi economici ai Paesi, ai sistemi sanitari e ai pazienti stessi. Infatti, l'Atlante dell'IFD ha constatato che dal 2007 al 2021 la spesa sanitaria globale per assistere pazienti diabetici, indipendentemente che sia stata spesa pubblica o privata, è aumentata del 316% passando da 232 miliardi di dollari a 966 miliardi di dollari e, inoltre, prevede che continuerà ad aumentare fino a raggiungere i 1,05 trilioni di dollari nel 2045.

Sebbene i metodi di assistenza siano in continuo progresso e la vita dei pazienti sia molto migliorata, il diabete è ancora una delle principali cause di morte nel mondo. Uno dei principali paesi con il numero maggiore di morti annuali per diabete è la Cina con 1,4 milioni, a seguire gli Stati Uniti con 0.7 milioni, India (0.6 milioni), Pakistan (0.4 milioni) e Giappone (0.2 milioni) [4].

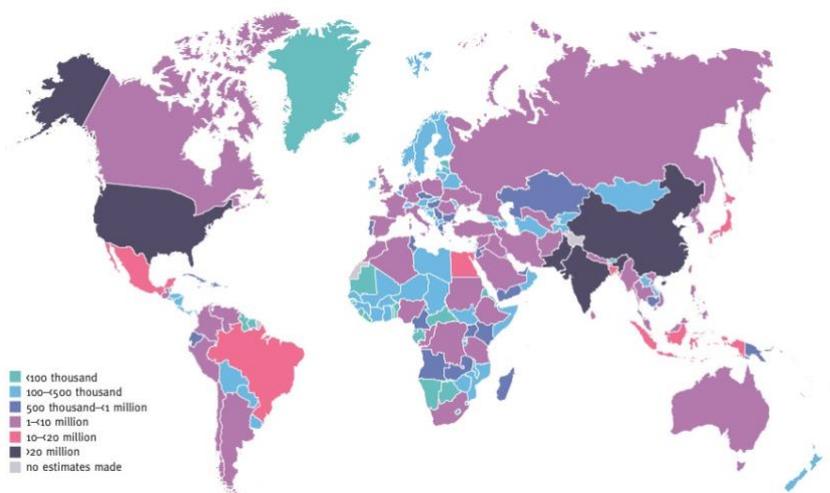


Figura 2 Distribuzione mondiale dei pazienti diabetici con età compresa tra i 20 e i 79 anni [4].

1.4 Pancreas artificiale (AP)

Il diabete di tipo 1 impedisce al pancreas di produrre insulina utile per la regolazione dei livelli di glucosio nel sangue. Per cui un soggetto affetto da tale forma di diabete avrà bisogno di un sistema esterno che inietti insulina nel paziente e tenga monitorato i livelli di glucosio nel sangue per evitare iper o ipoglicemia.

Una possibile alternativa alle terapie convenzionali che utilizzano penne o siringhe, è il sistema di pancreas artificiale.

Negli ultimi dieci anni questo tipo di sistema è stato testato in più di 50 studi condotti su pazienti affetti da diabete di tipo 1 e, una volta testata la loro efficacia in ambienti ospedalizzati, sono stati testati per poter essere utilizzati nella vita di tutti i giorni [6].

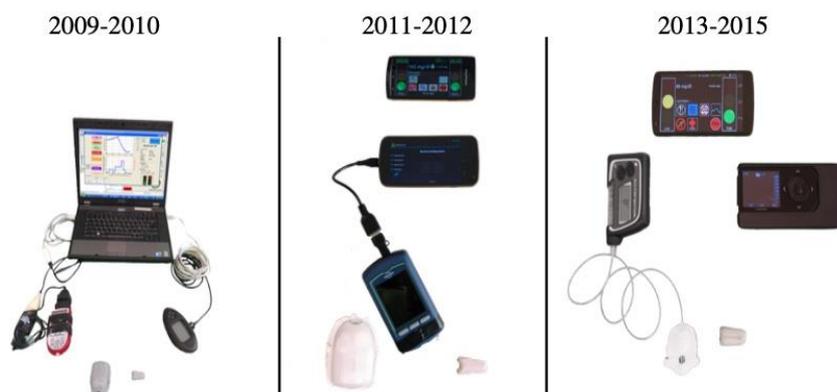


Figura 3 Evoluzione del hardware di un sistema di AP dal 2009 al 2015. [6]

1.4.1 Componenti del pancreas artificiale

Il pancreas artificiale è formato da tre elementi fondamentali:

- Pompa insulinica;
- Sensore di monitoraggio continuo del glucosio, detto *Continuous Glucose Monitoring* (CGM);
- Algoritmo di controllo.

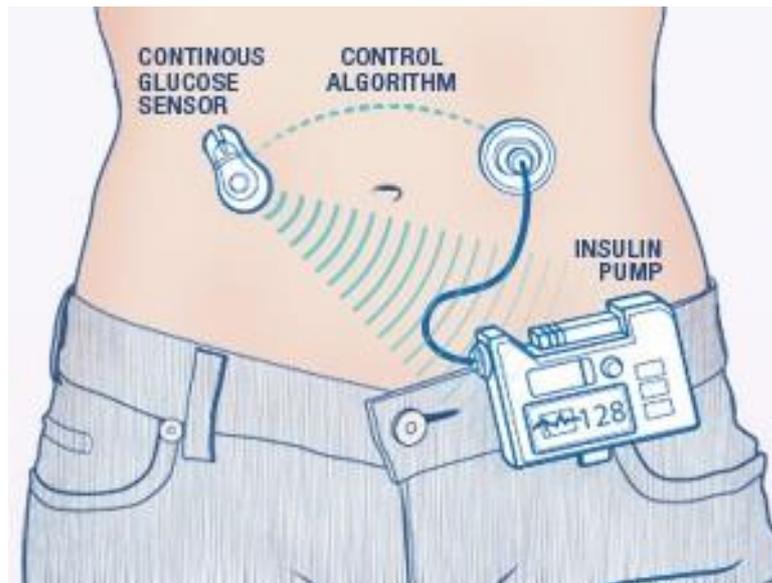


Figura 4 Rappresentazione sintetica di un sistema di pancreas artificiale

La pompa insulinica è un dispositivo che imita l'azione del pancreas rilasciando l'insulina necessaria all'organismo per il trasporto del glucosio dal sangue alle cellule.

Ogni pompa è composta da cinque elementi principali: un micro-computer programmabile, un pistone, un motore, un serbatoio e da un set d'infusione.

In sostanza, il motore attraverso movimenti impercettibili fa muovere il pistone che spinge l'insulina presente nel serbatoio all'interno del catetere del set d'infusione. Nella parte terminale del catetere è presente una cannula inserita nello strato sottocutaneo del paziente e attraverso di essa si infonderà la quantità di insulina programmata.[7]

Non esiste un solo tipo di infusione, infatti: durante la notte o lontano dai pasti si effettua un'infusione di insulina basale per mantenere nella norma i livelli glicemici del paziente; mentre nel momento dei pasti o di spuntini c'è la necessità di aumentare la quantità di insulina da iniettare per gestire i grammi di carboidrati presenti negli alimenti assunti. Una tale infusione di insulina è definita bolo.

Perciò la quantità di insulina iniettata non è sempre la stessa ma dipende dalla glicemia del momento e per conoscere tale concentrazione ci si avvale del sensore di monitoraggio continuo del glucosio il quale ne misura costantemente la

concentrazione nel fluido interstiziale attraverso l'utilizzo di un microsensore impiantato sotto la cute.

La pompa insulinica e il sensore di monitoraggio continuo del glucosio comunicano fra di loro grazie all'algoritmo di controllo. Tale algoritmo costituisce il fulcro del pancreas artificiale perché determina la quantità di insulina da somministrare a seconda dei valori registrati dal CGM e comunica tale quantità alla pompa insulinica. Per maggiori dettagli sui vari tipi di algoritmi sviluppati si veda [6].

Capitolo 2

2. L'algoritmo di rilevazione dei guasti nei set d'infusione del pancreas artificiale (LISAs - *Losses in Infusion Set Actuation*)

Uno dei maggiori problemi che può presentarsi nel sistema di pancreas artificiale è un guasto nel set d'infusione della pompa insulinica.

La rilevazione e la segnalazione al paziente di tali guasti è molto importante perché un guasto nel set d'infusione altera la quantità di insulina somministrata la paziente provocando iperglicemia, ovvero i livelli di glucosio nel sangue del soggetto sono molto elevati e se tale condizione persiste per troppo tempo possono apparire importanti complicazioni, come la chetoacidosi diabetica o il coma iperosmolare.

In questo capitolo analizzeremo la strategia per la rilevazione dei *Losses in Infusion Set Actuation* (LISAs) sviluppata da Howsmon et al [8].

2.1 Metriche per il riconoscimento dei LISAs

L'algoritmo sviluppato da Howsmon et al. prevede la raccolta di dati relativi a glucosio e insulina e il loro utilizzo per calcolare tre metriche di guasto rilevate su due finestre temporali diverse. Dopo aver definito le metriche ne hanno imposto un valore di soglia per ognuna nel breve periodo e, quando tutte e tre le metriche superano simultaneamente le soglie, l'algoritmo genererà un segnale d'allarme.

Le metriche designate per questo algoritmo sono: il *Glucose Fault Metric*, l'*Insulin Fault Metric* e *Glucose Slope*.

Questo algoritmo vuole essere robusto sia nella rilevazione in pazienti diversi sia nella rilevazione nello stesso paziente ma in momenti diversi, perché i livelli glicemici oltre ad essere differenti da paziente a paziente differiscono anche da giorno a giorno nello stesso paziente anche se assume gli stessi alimenti tutti i giorni.

2.1.1 Glucose Fault Metric (GFM)

Dal momento che in condizione di guasto i livelli glicemici sono elevati ho bisogno di calcolare due livelli di controllo glicemico: il livello di controllo basale e il livello di controllo attuale.

Il livello di controllo basale del paziente è la media dei valori di glucosio rilevati nel CGM in una finestra temporale di larghezza LW pari a un giorno ($\overline{CGM}_{k|k-LW}$).

Mentre, il livello di controllo attuale del paziente è pari alla media dei valori di glucosio più recenti rilevati nel CGM in una finestra temporale di larghezza SW equivalente a un'ora.

Questi due valori vengono combinati insieme per calcolare il Glucose Fault Metric, o, in breve, GFM.

Per trovare il GFM bisogna per prima cosa calcolare l'area presente tra il livello attuale e basale allo stesso tempo k, definita con l'abbreviazione AUC_k . Nel caso in cui il livello attuale è maggiore di quello basale allora AUC_k sommata al GFM al tempo k-1 rappresenterà il GFM al tempo k, altrimenti il GFM al tempo k sarà pari a zero.

Formalmente, considerando un tempo di campionamento $\Delta t_k = 5$ minuti, il calcolo del parametro viene descritto con le seguenti formule:

$$AUC_k = (\overline{CGM}_{k|k-SW} - \overline{CGM}_{k|k-LW}) \cdot \Delta t_k$$
$$GFM_k = \begin{cases} GFM_{k-1} + AUC_k & \text{se } \frac{\overline{CGM}_{k|k-SW}}{\overline{CGM}_{k|k-LW}} > 1 \\ 0 & \text{altrimenti} \end{cases}$$

2.1.2 Insulin Fault Metric (IFM)

Come abbiamo già detto nel capitolo 2.2, non sempre alti livelli di glucosio equivalgono a guasti nei set d'infusione.

Spesso questi aumenti improvvisi sono dovuti dal fatto che il paziente sta facendo un pasto o uno spuntino, per cui, per evitare la formazione di falsi positivi dobbiamo considerare un parametro complementare al GFM. Tale parametro è l'*Insulin Fault Metric* (IFM).

Per trovare l'*Insulin Fault Metric*, per prima cosa, dobbiamo stimare la quantità di insulina attiva nel sangue, cioè l'insulina plasmatica che qui indicheremo con l'abbreviazione PIE.

L'insulina realmente attiva nel sangue non è pari alla quantità di insulina iniettata, questo perché l'insulina somministrata prima di arrivare al sangue e influire sui livelli di glucosio del paziente deve essere assorbita dalla cute. Questo passaggio intermedio crea un ritardo tra il momento di infusione dell'insulina e il momento in cui svolge la sua funzione.

Possiamo descrivere il percorso fatto dall'insulina con il seguente schema compartimentale

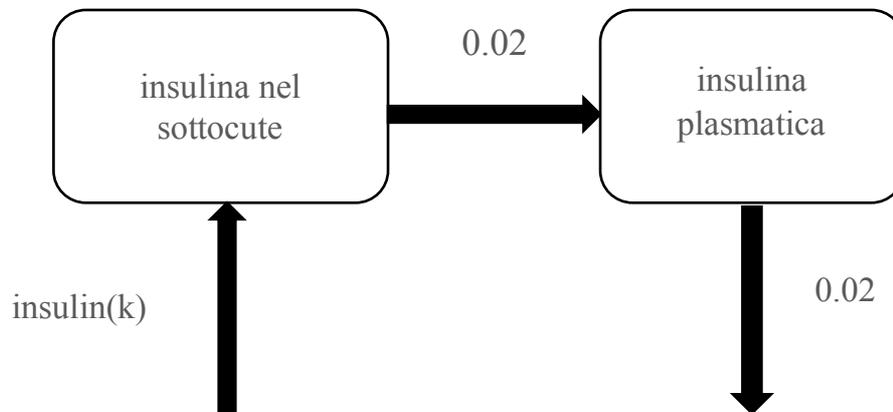


Figura 5 Schema compartimentale dello spostamento dell'insulina nel sangue

L'insulina inietta va ad aggiungersi all'insulina già presente nel sottocute e, della quantità di insulina già presente nel sottocute, il 2% andrà a defluire nel sangue aggiungendosi all'insulina già presente, generando così la reale quantità di insulina attiva che corrisponde alla PIE.

Matematicamente, si descrive tale processo con il seguente sistema lineare equivalente a un filtro del secondo ordine:

$$\begin{bmatrix} x_{k+1}^1 \\ x_{k+1}^2 \end{bmatrix} = \begin{bmatrix} 0.98 & 0.02 \\ 0 & 0.98 \end{bmatrix} \begin{bmatrix} x_k^1 \\ x_k^2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \text{Insulin}_k$$

$$\text{PIE}_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_k^1 \\ x_k^2 \end{bmatrix}$$

Dove:

- x_{k+1}^1 equivale alla concentrazione di insulina nel sangue al tempo k+1
- x_k^1 equivale alla concentrazione di insulina nel sangue al tempo k
- Insulin_k pari alla somma tra il bolo di insulina iniettato al tempo k e la velocità basale corrente in unità/minuto
- x_{k+1}^2 equivale alla concentrazione di insulina nel sottocute al tempo k+1
- x_k^2 equivale alla concentrazione di insulina nel sottocute al tempo k

Una cosa importante da ricordare è il tempo di campionamento utilizzato che è pari a un minuto; questo perché le variazioni di insulina e i boli possono presentarsi in tempi più brevi rispetto alle variazioni che possono verificarsi nel CGM [7].

Dopo aver calcolato la PIE per trovare il parametro IFM si esegue un confronto tra l'insulina plasmatica media nel breve periodo, $\text{PIE}_{k|k-SW}$, e l'insulina plasmatica media nel lungo periodo, $\text{PIE}_{k|k-LW}$.

In formula

$$\text{IFM}_k = \frac{\overline{\text{PIE}}_{k|k-SW} - \overline{\text{PIE}}_{k|k-LW}}{\overline{\text{PIE}}_{k|k-LW}} = \frac{\overline{\text{PIE}}_{k|k-SW}}{\overline{\text{PIE}}_{k|k-LW}} - 1$$

Si può notare che il confronto viene normalizzato rispetto a $\overline{\text{PIE}}_{k|k-LW}$ perché il fabbisogno individuale di insulina cambia da paziente a paziente.

2.1.3 Glucose Slope (GS)

In presenza di un guasto la pendenza dei livelli di glucosio varia bruscamente, per cui un ulteriore parametro utile per la rilevazione dei guasti è la derivata del segnale di glucosio o anche detta *Glucose Slope* (GS).

Considerando un tempo di campionamento di cinque minuti la *Glucose Slope* viene calcolata nel seguente modo:

$$GS_k = \frac{CGM_k - CGM_{k-1}}{\Delta t_k}$$

2.2 Performance dell'algoritmo

Per valutare le performance dell'algoritmo sono stati utilizzati set di dati clinici di sessantadue inserimenti di set d'infusione su venti pazienti.

Tutti i set di dati utilizzati per valutare l'algoritmo hanno rilevato clinicamente guasti nei set d'infusione; tuttavia, il momento esatto in cui sono iniziati non è noto, questo perché non sono stati indotti. Perciò, un *team* di medici e ingegneri l'ha determinato a priori facendolo coincidere con il punto di minimo più recente nei dati del CGM.

Questa incertezza nell'inizio del guasto assieme al fatto che i pazienti rispondono a un evento di allarme anziché ai singoli dati, hanno portato a valutare l'algoritmo su eventi di allarme piuttosto che sui singoli dati. Inoltre, il fatto che i test vengano eseguiti su tre set di dati indipendenti permette di generalizzare l'algoritmo anche a dati di nuovi pazienti e non solo a dati noti.

La sensibilità dell'algoritmo assieme al numero di falsi positivi al giorno sono due ulteriori caratteristiche da prendere in considerazione per valutare le performance dell'algoritmo. Infatti, definiamo l'algoritmo efficace se a un'alta sensibilità corrisponde un numero piccolo di falsi positivi.

Per ottenere una tale condizione sono state analizzate tutte le combinazioni possibili dei valori dei dati del set di training. Delle varie combinazioni si è poi messo in relazione la sensibilità con i falsi positivi generando una curva caratteristica operativa pseudo-ricevente (pROC). In tale curva si è poi selezionato il punto con una sensibilità del 71,8% e il numero di falsi positivi pari a 0.28 e infine si è andati a visualizzare nel set di training a che combinazione di valori dei parametri corrispondesse.

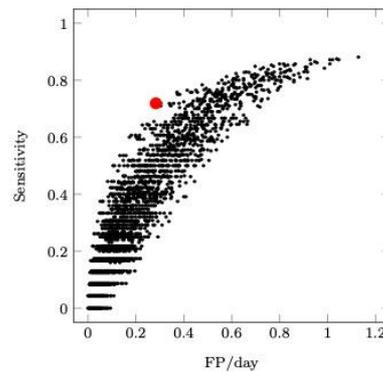


Figura 6 Curva pROC

La Figura 6 rappresenta la curva pROC, in cui il punto rosso equivale al valore di sensibilità pari a 71,8%, mentre i falsi positivi per giorno identificati sono pari a 0.28.

Parametri	Valori selezionati	Unità di misura
LW	24	ore
SW	1	ore
Soglia GFM	100	(mg/dL) •minuto
Soglia IFM	0.4	Numero puro
Soglia GS	0.3	(mg/dL) •minuto

Tabella 2 Parametri e valori corrispondenti al punto rosso della curva pROC

Capitolo 3

3. Implementazione dell'algoritmo per il riconoscimento dei LISAs

In questo ultimo capitolo vengono valutate le performance dell'algoritmo di Howsmon et al. L'algoritmo viene ricreato nel linguaggio MATLAB utilizzando un set di dati diversi da quelli già in precedenza impiegati per il *training* e la convalida. Lo spazio di lavoro utilizzato è MATLAB nella versione R2020b.

I dati che vengono impiegati per la valutazione delle prestazioni provengono da un database di cinque pazienti *in silicio* generati dal simulatore Uva/Padova.

Il codice completo è visibile nell'appendice A.

3.1 Dati utilizzati

Come già accennato nell'introduzione al paragrafo, i dati utilizzati riguardano cinque pazienti facenti parte di un database *in silicio* del simulatore UVA/Padova [9].

Il database raccoglie per ogni soggetto una simulazione di trenta giorni della concentrazione di glucosio e d'insulina con tempo di campionamento di cinque minuti, come previsto per i convenzionali sistemi di pancreas artificiale.

L'insulina basale viene somministrata continuamente e la sua quantità è regolata dall'algoritmo di controllo PID che permette di mantenere i livelli glicemici in un range di valori quasi normali.

La somministrazione dei boli, invece, viene sempre eseguita durante i pasti e in quantità proporzionale alla quantità di carboidrati stimata che varia da pasto a pasto e di giorno in giorno. Sebbene i valori relativi ai carboidrati assunti nei pasti ci vengano già forniti dal database per mantenere fede all'algoritmo sviluppato da Howsmon et al. non li considereremo.

I guasti della pompa, inoltre, avvengono vicino la mezzanotte e al mezzogiorno e durano esattamente sei ore semplificando l'elaborazione e la lettura del codice senza intaccare la qualità dei risultati raggiunti.

Il file, da cui estraiamo i dati utili, è un file CSV le cui colonne dividono i dati di ogni paziente per tipologia (orario, CGM, pasto, insulina, bolo, etc); mentre le righe indicano dei precisi orari di rilevazione. In seguito, raggiunta la visualizzazione di tutti i dati mensili di un singolo paziente si passa all'elenco degli stessi dati per il secondo, del terzo e così via fino all'ultimo (trentesimo) paziente posto in analisi.

Per quanto riguarda il nostro elaborato, non verranno considerati tutti i dati elencati nel database sopracitato, ma solamente quelli di interesse al fine del calcolo delle metriche di guasto e per l'ulteriore segnalazione dei guasti.

3.2 Struttura del programma

In primis, è necessario considerare che per poter rendere più flessibile il programma la maggior parte delle variabili è personalizzabile a seconda delle proprietà dei dati di cui si è in possesso.

A questo fine, quindi, sono modificabili i seguenti parametri:

- La lunghezza della finestra breve SW e di quella lunga LW;
- Il numero di giorni e ore di monitoraggio;
- La quantità di dati visualizzabili;
- Le soglie delle metriche.

Il programma estrae dal database del simulatore Uva/Padova i valori di interesse per il calcolo delle metriche e per la rappresentazione grafica.

Successivamente, dopo essersi calcolato il numero di pazienti monitorati, divide i valori estratti in precedenza dal database in matrici. Le matrici vengono elaborate facendo corrispondere ogni colonna a un diverso paziente e ogni riga a uno specifico campione rilevato da ogni paziente.

Prima di passare al calcolo dei parametri, il programma interpola i valori dell'insulina passando da un tempo di campionamento di cinque minuti a un tempo

di campionamento di un minuto. Questo processo viene realizzato perché, come scritto nell'articolo di Howsonm et al, le variazioni di insulina sono più rapide rispetto a quelle dei livelli glicemici.

Uno dei limiti principali del programma è il fatto che nelle prime ventiquattro ore i valori medi di glucosio, nel breve e lungo periodo, non sono affidabili per cui la rilevazione dei guasti non viene eseguita per nessuna metrica. Nel caso in cui, quindi, si verificassero dei guasti nelle prime ventiquattro ore non verrebbero rilevati dal programma.

I valori di *default* imposti nel programma equivalgono a quelli descritti da Howsonm et al, ovvero:

- Ampiezza finestra breve SW = 1 h (pari a 12 campioni per il calcolo del livello attuale del CGM e a 60 campioni per il calcolo della media a breve termine della insulina plasmatica);
- Ampiezza finestra lunga LW = 24 h (pari a 288 campioni per il calcolo del livello basale del CGM e a 1440 campioni per il calcolo della media a lungo termine della insulina plasmatica);
- Tempo di campionamento del CGM 5 minuti;
- Tempo di campionamento dell'insulina 1 minuto necessario per il calcolo dei valori di insulina;
- Soglie GFM = 100 mg/dL *minuto, IFM = 0.4 numero puro e GS = 0.3 mg/dL *minuto.

3.3 Analisi dei parametri

3.3.1 Sviluppo *Glucose Fault Metric*

Per il calcolo del GFM il programma estrae i valori medi del CGM attraverso una funzione detta '*mediasufinestra*', sia nel breve che nel lungo periodo. La funzione richiede in ingresso i valori del CGM e la larghezza della finestra su cui calcolare la media. In questo caso, la larghezza della finestra del lungo periodo è pari a LW =

288 campioni e, la larghezza della finestra nel breve periodo corrisponde a $SW=12$ campioni.

In uscita tale funzione rilascia il valore medio calcolato su tutti i campioni, e successivamente, da tali campioni estrae solo i valori dopo le prime ventiquattro ore. I dati vengono salvati poi in una matrice a parte per poter essere visualizzati successivamente in maniera grafica.

Sfruttando la funzione *'glucosefaultmetric'*, che richiede in ingresso il livello attuale, il livello basale e il tempo di campionamento, si ricavano i valori di GFM per ogni paziente.

Di seguito possiamo vedere una rappresentazione grafica dei valori ottenuti per il paziente 2 per gli ultimi venti giorni di monitoraggio.

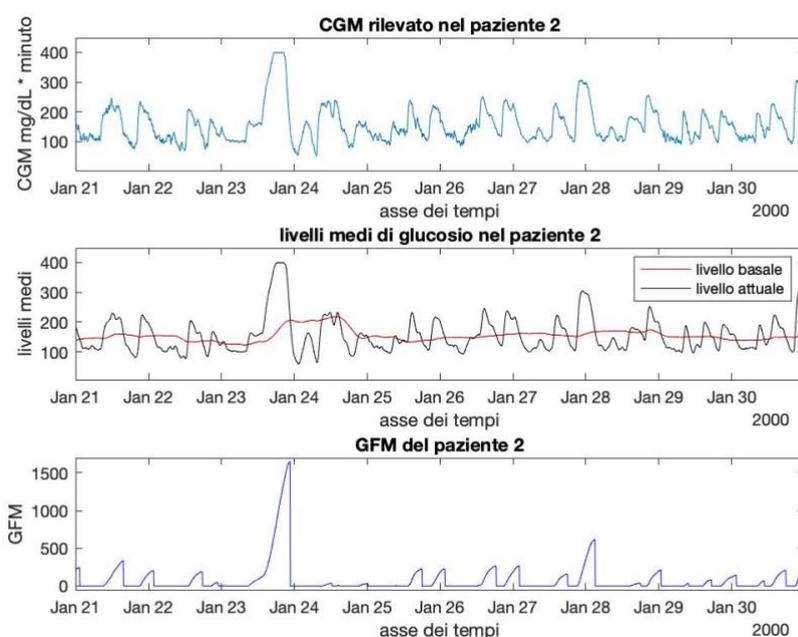


Figura 7 Esempio grafico dei valori di GFM rilevati nel paziente 2

3.3.2 Sviluppo *Glucose Slope*

Per il calcolo della derivata ci limitiamo a eseguire un rapporto tra la differenza di valori di CGM consecutivi e il tempo di campionamento espresso in minuti e sempre pari a 5.

Anche in questo caso i valori sono stati inseriti all'interno di una matrice utile sia per trovare i superamenti di soglia sia per la rappresentazione grafica.

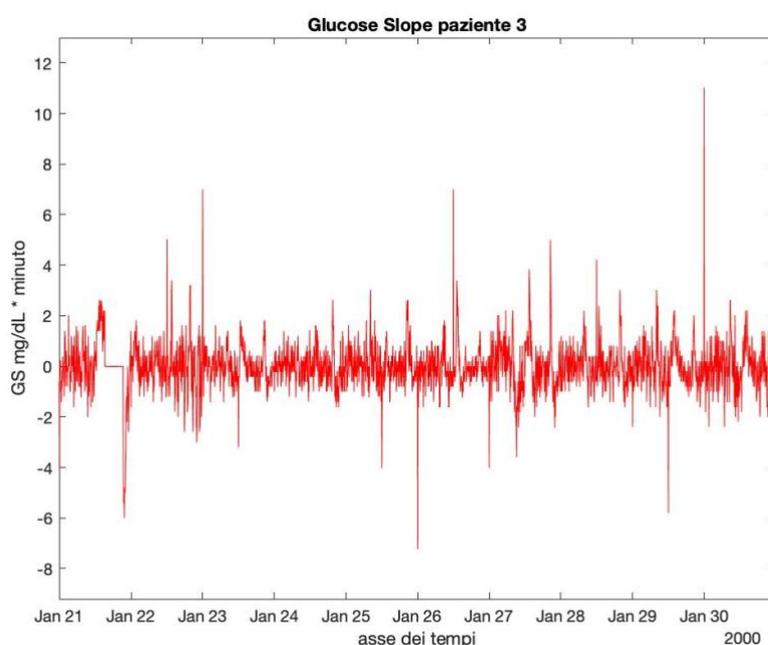


Figura 8 *Glucose Slope* del paziente 3 negli ultimi 20 giorni

3.3.3 Sviluppo *Insuline Fault Metric*

Prima di eseguire il calcolo del parametro si è calcolata la PIE sviluppando il filtro del secondo ordine.

Successivamente si è dato in ingresso alla funzione '*mediasufinestra*', già usata per il calcolo del livello basale e attuale, la PIE e la larghezza delle finestre su cui calcolare il valore medio nel breve e lungo periodo.

In questo caso, la larghezza della finestra del lungo periodo è pari a $LW = 1440$ campioni e, la larghezza della finestra nel breve periodo corrisponde a $SW = 60$ campioni perché il numero di campioni di insulina è più elevato rispetto a quello del CGM.

Dopo aver trovato i valori medi calcolo IFM attraverso una funzione, detta *'insulinfaultmetric'*, alla quale dando ingresso le PIE medie mi rilascia in uscita metrica IFM complementare al GFM.

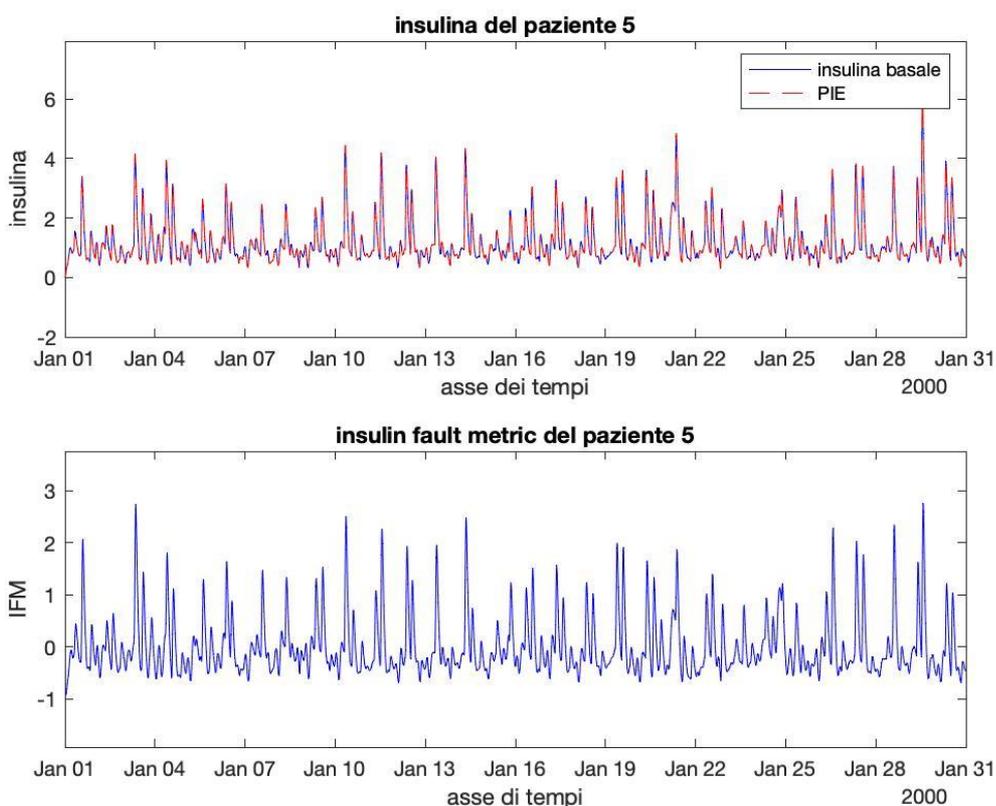


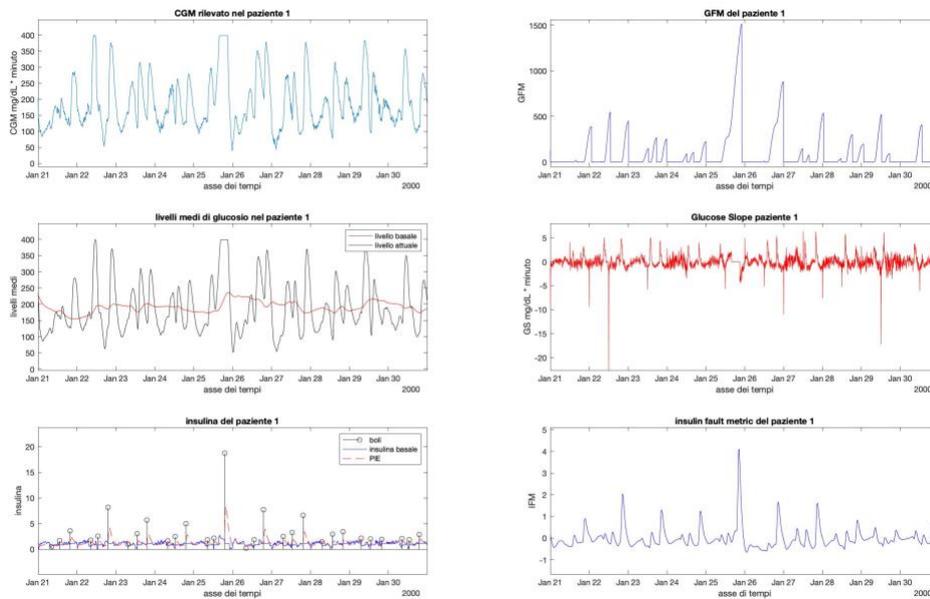
Figura 9 Grafico dei valori del sangue rilevati del paziente 3. Metrica dei seguenti valori.

3.3.4 Grafici

Al termine del calcolo dei parametri utili per la generazione degli allarmi il programma ne permette la visualizzazione grafica per ogni paziente, permettendo così di poter effettuare un confronto visivo e vedere come i vari valori glicemici ed

insulinici possono cambiare notevolmente da soggetto a soggetto che di giorno in giorno nello stesso soggetto.

L'immagine seguente rappresenta i valori rilevati nel paziente negli ultimi venti giorni di monitoraggio.



3.4 Generazioni di allarmi e performance dell'algorithm

Dopo aver calcolato le metriche di guasto il programma controlla il superamento delle soglie creando dei nuovi array di lunghezza pari alla quantità di campioni presenti per metrica di guasto.

All'interno di questi array il programma posiziona il valore 0 se la metrica alla stessa posizione non ha superato il valore di soglia e 1 se, invece, la soglia è stata superata. Prima di eseguire un confronto tra gli array delle varie metriche per generare l'allarme, vero o falso che sia, il programma riduce il numero di campioni dell'array creato per i superamenti di soglia dell'IFM allo stesso numero di campioni degli array creati per GFM e GS e, infine, verifica quando tutti e tre le metriche superano la soglia nello stesso momento salvando il tutto in una nuova matrice.

I valori presenti in tale matrice vengono visualizzati paziente per paziente in intervalli di 6 h per poter valutare le prestazioni dell'algoritmo nel rilevare correttamente i guasti.

Per fare ciò si devono conteggiare i seguenti valori:

- *True Positive (TP)*: numero di volte che è stato creato un allarme ed effettivamente c'era un guasto nella pompa.
- *False Positive (FP)*: numero di volte che è stato creato un allarme per superamento delle soglie ma non era presente un guasto della pompa, solitamente sono dovuti dai pasti che in questo programma non vengono considerati.
- *True Negative (TN)*: numero di volte che non è stato creato un allarme perché non c'è stato ne un superamento di soglia e ne guasto
- *False Negative (FN)*: numero di volte che il programma non ha segnalato la presenza di un guasto perché non c'è stato superamento delle soglie.
- Per capire poi se l'algoritmo ha delle buone performance è necessario calcolare il numero di *FP per giorno* e la sensibilità.
- Per trovare il numero di *FP per giorno* è necessario normalizzare la somma totale dei FP trovati per il prodotto tra il numero di giorni di rilevazione e il numero di pazienti monitorati. La sensibilità, invece, è semplicemente il rapporto tra il numero di TP trovati e il numero di guasti totali.
- In questo caso, con i valori di soglia convenzionali il programma ha presentato una sensibilità del 30 %, rilevando 3 guasti su 10, e il numero di FP per giorno è 0.33.

Confrontando i valori ottenuti nella simulazione con quelli trovati da Howsmon e dai suoi colleghi, rispettivamente sensibilità del 78% e FP/giorno pari a 0.28, la simulazione presenta una sensibilità peggiore e un numero di falsi positivi maggiore. Questo si verifica perché la metrica *Insulin Fault Metric* rimane sotto il valore di soglia prevista per tutte e sei le ore del guasto. Una soluzione applicabile per migliorare la sensibilità è quella di abbassare la soglia della metrica IFM. Infatti, già riducendo la soglia a 0.3 l'algoritmo rileva già due guasti in più.

Conclusioni

L'implementazione dell'algoritmo di Howsmon et al realizzata nel presente elaborato riporta risultati meno performanti in quanto, come precisato già nell'ultimo capitolo, sussiste un incremento sia di sensibilità che del numero di falsi positivi riscontrati. Alla luce dei risultati ottenuti si può affermare che l'algoritmo risulta sicuramente uno strumento promettente per i sistemi di pancreas artificiale utilizzabili nei pazienti diabetici nella vita di tutti i giorni. E', però, tutt'oggi soggetto di ricerche e di studi al fine di rilevare l'implementazione più efficiente che massimizzi la performance e minimizzi il margine d'errore.

Riferimenti

1. C. Cobelli, C. Dalla Man, G. Sparacino, L. Magni, G. De Nicolao and B. P. Kovatchev, “*Diabetes: Models, Signals, and Control*”, IEEE reviews in biomedical engineering, vol. 2, 2009, p. 54.
2. Online: www.idf.org
3. Online: <https://dri.hsr.it/news/le-cellule-beta-pancreatiche-rispondono-allinfiammazione-implicazioni-nello-sviluppo-del-diabete-tipo-1/>
4. D.J. Magliano et al. *IDF Diabetes Atlas 2021*, 10th ed. 2021, pp. 12-15, pp. 55-61.
5. J. C. Pickup and G. Williams, *Textbook of Diabetes 2*. Oxford, U.K.: Blackwell, 1991, pp. 28-29.
6. S. Del Favero, C. Toffanin, L. Magni and C. Cobelli, ‘*Deployment of Modular MPC for Type 1 Diabetes Control: the Italian experience 2008-2016*’, Pavia and Padua, Italy, 2018, pp. 2-5.
7. Online: <https://www.diabete.com/6-domande-sul-pancreas-artificiale/>
8. D. P. Howsmon, F. Cameron, N. Baysal, T. T. Ly, G. P. Forlenza, D. M. Maahs, B. A. Buckingham, J. Hahn and B.W. Bequette, ‘*Continuous Glucose Monitoring Enables the Detection of Losses in Infusion Set Actuation (LISAs)*’. Published 2017
9. L. Meneghetti, M. Terzi, S. Del Favero, G. A. Susto, Member, IEEE, C. Cobelli, Fellow, IEEE, ‘*Data-Driven Anomaly Recognition for Unsupervised Model-Free Fault Detection in Artificial Pancreas*’, pp. 39-40.

Appendice A

```
%importiamo di dati del simulatore
dati = importdata('simulator_dataset_2018_rng1.csv');
%estraiamo i dati del CGM, insulina, pasto, boli, guasti e ID dei pazienti
CGM = dati.data(:,1);
insulinabasale = dati.data(:,3);
boli = dati.data(:,4);%unità//ora
ID = dati.data(:,13);
guasti = dati.data(:,10);
%parametri modificabili
giornidimonitoraggio = 30;
oremonitoraggio = 24 * giornidimonitoraggio;
tempo_campionamento_CGM = 5; %minuti
tempo_campionamento_h_CGM = tempo_campionamento_CGM/60; %ore
tCGM=[0:tempo_campionamento_h_CGM:oremonitoraggio -...
tempo_campionamento_h_CGM];
tempo_campionamento_insulina = 1; %minuto
tempo_campionamento_in_h_insulina = tempo_campionamento_insulina/60;
tinsulina=[0:tempo_campionamento_in_h_insulina:oremonitoraggio-
tempo_campionamento_in_h_insulina];
LW_GFM = (24*(60/5)); %ampiezza dell'intervallo lungo per GFM= numero di
campioni da considerare per il calcolo della media
SW_GFM = (1*(60/5)); %ampiezza dell'intervallo corto per GFM
LW_IFM = 24*60; %ampiezza dell'intervallo lungo per IFM
SW_IFM = 1*60; %ampiezza dell'intervallo corto per IFM
%soglie
soglia_GFM = 100;
soglia_IFM = 0.4;
soglia_GS = 0.3;
guasto_pompa = -100;
%calcolo il numero di pazienti monitorati
campioni = length(ID);
```

```

indice = 1;
numeropazienti = 1;
while indice < campioni
    if ID(indice,1) == ID(indice+1,1)
        indice = indice +1;
    else
        numeropazienti = numeropazienti + 1;
        indice = indice +1;
    end
end

%estraggo i valori del CGM,dei pasti, dell'insulina,dei boli e dei guasti per ogni
paziente,inserisco i dati dentro a delle matrici il cui numero di righe equivale al numero di
campioni estratti per paziente e il numero di colonne il numero dei pazienti
monitorati,ogni colonna conterrà i valori rilevati nei 30 giorni per ogni paziente
numerocampioniperpaziente = campioni/numeropazienti;
for indice = 1:numeropazienti
    indici_paziente = find(dati.data(:,13) == indice);
    %CGM
    matrice_CGM(:,indice) = CGM(indici_paziente(1):indici_paziente(end)) ;
    %insulina basale
    matrice_insulina_basale_TC5(:,indice)=
    insulinabasale(indici_paziente(1):indici_paziente(end));
    %boli
    matrice_boli_TC5(:,indice)=boli(indici_paziente(1):indici_paziente(end));
    %guasti
    matrice_guasti(:,indice) = guasti(indici_paziente(1):indici_paziente(end));
end

%ATTENZIONE: i valori di insulina sono rappresentati con un tempo di campionamento
di 5 minuti dovremmo interpolare i dati per trasformarlo in tempo di campionamento di
un minuto

%ciclo per interpolazione dei valori di insulina
for indice = 1:numeropazienti
    %interpolazione insulina = insulina basale + bolo utile poi per il

```

%calcolo della PIE

```
insulina_basale_da_interpolare = matrice_insulina_basale_TC5(:,indice);  
insulina_basale_TC1 = interp1(tCGM,insulina_basale_da_interpolare,tinsulina);  
matrice_insulina_basale_TC1(:,indice) = insulina_basale_TC1;
```

```
bolo_da_interpolare = matrice_boli_TC5(:,indice);  
bolo_TC1 = interp1(tCGM,bolo_da_interpolare,tinsulina);
```

```
matrice_insulina_TC1(:,indice) = insulina_basale_TC1+bolo_TC1 ;
```

%unisco direttamente l'insulina con il bolo perché i dati del bolo sono già in unità/ora
%interpolazione singola di insulina basale mi servirà poi per la rappresentazione grafica dell'insulina basale divisa dal bolo

end

%calcolo dei parametri per rilevare i guasti

%GFM

%sfrutto la funzione media su finestra per calcolare il valore medio nel breve e lungo periodo

for indice = 1:numero_pazienti

```
CGMpaziente = matrice_CGM(:,indice);  
livello_basale_temp(:,indice) = media_sufinestra(CGMpaziente,LW_GFM);  
livello_attuale_temp(:,indice) = media_sufinestra(CGMpaziente,SW_GFM);  
%considero solo i valori dopo le prime 24 h  
livello_basale_paziente = livello_basale_temp([LW_GFM+1:end],indice);  
livello_attuale_paziente = livello_attuale_temp([LW_GFM+1:end],indice);  
matrice_livello_basale(:,indice) = livello_basale_paziente;  
matrice_livello_attuale(:,indice) = livello_attuale_paziente;
```

end

%calcolo del GFM

for indice = 1 : numero_pazienti

```
livello_basale = matrice_livello_basale(:,indice);  
livello_attuale = matrice_livello_attuale(:,indice);  
GFMpaziente = glucose_fault_metric(livello_basale, livello_attuale, tempocamp_CGM);  
GFM(:,indice) = GFMpaziente;
```

```

end
% calcolo dell'IFM
n = size(matrice_insulina_TC1,1);
insulinasangue =zeros(n,numeropazienti);
insulinasottocute = zeros(n,numeropazienti);
%calcolo la PIE
for indice1 = 1:numeropazienti
    for indice2 = 1:n
        insulinansangue(indice2+1,indice1)=...
            (0.98*insulinasangue(indice2,indice1))+(0.02*insulinasottocute(indice2,indice1));
        insulin(indice2,indice1) = matrice_insulina_TC1(indice2,indice1)/60;
        insulinasottocute(indice2+1,indice1) =...
            (0.98*insulinasottocute(indice2,indice1)) + insulin(indice2,indice1);
        PIE(indice2,indice1) = insulinasangue(indice2,indice1);
    end
end
% il numero di campioni da considerare per la PIE  $\sqrt{N}$  diverso da quello per il calcolo del
GFM per cui si utilizzerà una finestra con numero di campioni più elevata
%calcolo PIE medie
for indice = 1:numeropazienti
    PIEmediaLWtemp = mediasufinestra(PIE(:,indice),LW_IFM);
    PIEmediaSWtemp = mediasufinestra(PIE(:,indice),SW_IFM);
    PIEmediaLWpaziente = PIEmediaLWtemp(LW_IFM+1:end);
    PIEmediaSWpaziente = PIEmediaSWtemp(LW_IFM+1:end);
    IFMpaziente = insulinfaultmetric(PIEmediaLWpaziente,PIEmediaSWpaziente);
    IFM(:,indice) = IFMpaziente;
end
%calcolo della Glucose Slope
for indice1 = 1:numeropazienti
    ratedicrescita(:,indice1)=diff(matrice_CGM(:,indice1))/tempo_campionamento_CGM;
end
%generazione allarmi
%Le prime 24 del GFM e del IFM non sono considerate
%non considereremo neanche le prime 24h della GS per il calcolo dei guasti

```

```

lunghezza_finestra_da_non_considerare_GS = LW_GFM;

% IFM_per_guasti = IFM(lunghezza_finestra_da_non_considerare_IFM+1:end,:);
GS_per_guasti = ratedicrescita(lunghezza_finestra_da_non_considerare_GS+1:end,:);
guasti_pompa = matrice_guasti(lunghezza_finestra_da_non_considerare_GS+1:end,:);
%verifico i superamenti di soglia di GFM e GS
for i = 1 : numeropazienti
    GFMpaziente = GFM(:,i);%matrice contenente tutti i valori del GFM per i 5 pazienti
    GSpaziente = GS_per_guasti(:,i);%matrice contenete tutti i valori del GS per i 5
    pazienti dopo le prime 24h
    for t = 1 :length(GSpaziente)
        if GFMpaziente(t) > soglia_GFM
            allertaGFM(t,i) = 1;
        else
            allertaGFM(t,i) = 0;
        end
        if GSpaziente(t) > soglia_GS
            allertaGS(t,i) = 1;
        else
            allertaGS(t,i) = 0;
        end
    end
end
%faccio la stessa cosa per IFM
for i = 1 : numeropazienti
    IFMpazienteguasti = IFM(:,i);
    for g = 1:length(IFMpazienteguasti)
        if IFMpazienteguasti(g) > soglia_IFM
            superamentosoglia(g,i) = 1;
        else
            superamentosoglia(g,i) = 0;
        end
    end
end
end

```

```

% riduco il numero di campioni per allarme per IFM alla stessa quantità del GFM e del
GS, ovvero indico che c'è un superamento quando in 5 minuti almeno 3 valori del IFM
sono superiori alla soglia
for i = 1:numeropazienti
    superamento_soglia_paziente = superamentosoglia(:,i);
    k1 =1;
    for b =1:5:length(superamento_soglia_paziente)-4
        numerosuperamentosoglia=length(find(superamento_soglia_paziente(b:b+4)==1))
        if numerosuperamentosoglia > 3
            allertaIFM(k1,i) = 1;
        else
            allertaIFM(k1,i) = 0;
        end
        k1 = k1+1;
    end
end
% verifico quando tutti e 3 i parametri superano insieme la soglia
for i = 1:numeropazienti
    for k3 = 1: length(allertaGFM)
        if allertaGFM(k3,i) == 1 && allertaGS(k3,i) ==1 && allertaIFM(k3,i)==1
            allarme(k3,i) = 1;
        else
            allarme(k3,i) = 0;
        end
    end
end
%confronto con i valori della pompa per trovare quando l'algoritmo:
%segnala correttamente un guasto = True Positive
%segnala un superamento delle soglie ma non c'è un guasto = False Positive
%non segnala la presenza di un guasto ma in realtà il guasto c'è= False Negative
%non segnala la presenza di un guasto ed effettivamente il guasto non c'è= True Negative
n = length(allertaGFM);
oredirilevazione = 6; %ore
numerodicampioniperora=12;

```

```

campionidaconsiderare=oredirilevazione*numerodicampioniperora;
indiceiniziale = 1;
indicefinale =campionidaconsiderare;
for i = 1:numeropazienti
    indiceiniziale = 1;
    indicefinale =campionidaconsiderare;
    TP = 0;%true positive
    FP = 0;%false positive
    TN = 0;%true negative
    FN = 0;%false negative
    f = 1;
    while indicefinale <= size(allarme,1)
        allarme_6_ore = allarme(indiceiniziale:indicefinale,i);
        guasti_pompa_6_ore = guasti_pompa(indiceiniziale:indicefinale,i);
        indici_allarmi_metriche = find(allarme_6_ore==1);
        indici_guasti_pompa = find(guasti_pompa_6_ore==-100);
        if isempty(indici_allarmi_metriche)== 0 && isempty(indici_guasti_pompa)==0
            minutiprimadellarilevazione(f,i) = 5*(indici_allarmi_metriche(1)-...
            indici_guasti_pompa(1));
            f= f+1;
            TP = TP+1;
        elseif isempty(indici_allarmi_metriche) == 0 &&...
            isempty(indici_guasti_pompa)==1
            FP = FP+1;
        elseif isempty(indici_allarmi_metriche)== 1 &&
            isempty(indici_guasti_pompa)==0
            FN = FN +1;
        else
            TN = TN+1;
        end
        indiceiniziale = indicefinale +1;
        indicefinale = indicefinale+campionidaconsiderare;
    end
end

```

```

    valoripaziente = [TP FP FN TN]';
    matriceconteggio(:,i) = valoripaziente;
end
% calcolo i falsi positivi per giorno dei vari pazienti
Falsi_positivi_per_giorno = matriceconteggio(2,:)/(giornidimonitoraggio);
falsipositivitalipergiorno =
sum(matriceconteggio(2,:))/((giornidimonitoraggio)*numeropazienti);

%Rappresentazioni grafiche
%estraggo l'asse dei tempi da text data essendo una stringa li converto in data e ora
usando datetime
asse_tempi_CGM = datetime(dati.textdata(2:numerocampioniperpaziente+1,1));
asse_tempi_insulina=interp1(tCGM,asse_tempi_CGM,tinsulina);
%rappresentazione grafica
%per la rappresentazione grafica dei boli e insulina basale trovo gli indici in cui i boli
sono diversi da 0
for i = 1 :numeropazienti
    indicebolo = find(matrice_boli_TC5(:,i)>0);
    %trovo i valori all'interno della matrice boli e li inserisco in un vettore per
    rappresentarli con stem
    vettoretuttiiboli = matrice_boli_TC5(:,i);
    n = length(indicebolo);
    for k = 1:n
        vettorebolo(k) = vettoretuttiiboli(indicebolo(k));
        posizionebolosuasse(k) = asse_tempi_CGM(indicebolo(k));
    end
    matrice_dei_boli_maggiori_di_0(:,i) = vettorebolo;
    matrice_posizione_bolo(:,i) = posizionebolosuasse;
end
for indice= 1:numeropazienti
    %trovo i valori da rappresentare per ogni paziente dentro le varie matrici
    % CGM
    CGMpaziente = matrice_CGM(:,indice);

```

%livello basale e attuale

livellobasalepaziente = matricelivellobasale(:,indice);

livelloattualepaziente = matricelivelloattuale(:,indice);

%boli: scalo il valore per rappresentazione

vettoreboli = matrice_dei_boli_maggiori_di_0(:,indice)/12;

posizionebolosuasse = matrice_posizione_bolo(:,indice);

%insulina basale

insulinabasalepaziente = matrice_insulina_basale_TC1(:,indice);

%PIE

PIEpaziente = PIE(:,indice);

%IFM

IFMpaziente = IFM(:,indice);

%GFM

GFMpaziente = GFM(:,indice);

%derivata

ratepaziente = ratedicrescita(:,indice);

figure(indice)

subplot(3,2,1)

plot(asse_tempi_CGM,CGMpaziente)

xlim([asse_tempi_CGM(12*24*20) asse_tempi_CGM(end)])

ylim([min(CGMpaziente)-50 max(CGMpaziente)+50])

title(['CGM rilevato nel paziente ' num2str(indice)])

xlabel('asse dei tempi')

ylabel('CGM mg/dL * minuto')

subplot(3,2,2)

plot(asse_tempi_CGM(LW_GFM+1:end),GFMpaziente, 'b')

title(['GFM del paziente ' num2str(indice)])

xlim([asse_tempi_CGM(20*24*12) asse_tempi_CGM(end)])

ylim([min(GFMpaziente)-50 max(GFMpaziente)+50])

xlabel('asse dei tempi')

ylabel('GFM')

subplot(3,2,3)

```

plot(asse_tempi_CGM(LW_GFM+1:end),livellobasalepaziente,'r')
hold on
plot(asse_tempi_CGM(LW_GFM+1:end),livelloattualepaziente,'k')
title(['livelli medi di glucosio nel paziente ' num2str(indice)])
xlim([asse_tempi_CGM(12*24*20) asse_tempi_CGM(end)])
ylim([min(livelloattualepaziente)-50 max(livelloattualepaziente)+50])
xlabel('asse dei tempi')
ylabel('livelli medi')
legend('livello basale', 'livello attuale')
subplot(3,2,4)
plot(asse_tempi_CGM(2:end),ratepaziente,'r')
title(['Glucose Slope paziente ' num2str(indice)])
xlabel('asse dei tempi')
ylabel('GS mg/dL * minuto')
xlim([asse_tempi_CGM(12*24*20) asse_tempi_CGM(end)])
ylim([min(ratepaziente) max(ratepaziente)])
subplot(3,2,5)
stem(posizionebolosuasse,vettoreboli,'k')
hold on
plot(asse_tempi_insulina,insulinabasalepaziente,'-b')
hold on
plot(asse_tempi_insulina,PIEpaziente,'--r')
xlabel('asse dei tempi')
ylabel('insulina')
xlim([asse_tempi_CGM(12*24*20) asse_tempi_CGM(end)])
ylim([min(vettoreboli)-5 max(vettoreboli)+5])
legend('boli', 'insulina basale', 'PIE')
title(['insulina del paziente ' num2str(indice)])
subplot(3,2,6)
plot(asse_tempi_insulina(LW_IFM+1:end),IFMpaziente,'b')
xlabel('asse di tempi')
ylabel('IFM')
xlim([asse_tempi_CGM(12*24*20) asse_tempi_CGM(end)])

```

```

ylim([min(IFMpaziente)-1 max(IFMpaziente)+1])
title(['insulin fault metric del paziente ' num2str(indice)])
pause(0.1)
end

%Funzioni sviluppate per il calcolo delle medie e delle metriche
function valorimedi = mediasufinestra(x,finestra)
%funzione per il calcolo della concentrazione media di glucosio all'interno di un
%intervallo di tempo la cui larghezza = numero di campioni da considerare per calcolare
la media
n = length(x); %numero di campioni del segnale
for k = n:-1:finestra+1
    valore = x(k-finestra:k); % valori del CGM su cui calcolare la media
    valoremedio(k) = mean(valore);
end
valorimedi = valoremedio;
end

function GFM =glucosefaultmetric(livellobasale,livelloattuale,tempodicampionamento)
%funzione per il calcolo della metrica di guasti del glucosio dando in
%ingresso il livello basale ed attuale di glucosio del paziente
n = length(livellobasale);
%il primo valore di GFM va trovato fuori dal ciclo
AUC(1) = (livelloattuale(1) - livellobasale(1))*tempodicampionamento;
if AUC(1) > 0
    GFM(1) = AUC(1);
else
    GFM(1) = 0;
end
for indice = 2:n
    %calcolo l'area presente tra i livelli di glucosio
    AUC(indice) =(livelloattuale(indice)-livellobasale(indice))*tempodicampionamento;
    if AUC(indice) > 0
        GFM(indice) = GFM(indice-1) + AUC(indice);
    end
end

```

```
    else
        GFM(indice) = 0;
    end
end
end
end
```

```
function IFM = insulinfaultmetric(PIEmediaLW,PIEmediaSW)
```

```
%calcolo del parametro complementare al GFM per rilevare guasti nella pompa insulinica
```

```
n = length(PIEmediaLW);
for indice = 1:n
    IFM(indice) = (PIEmediaSW(indice)/PIEmediaLW(indice))-1;
end
end
```