

Digital Twin e le Opportunità del Cloud

A. Gravino, D. Trinchese, C. Napolitano

Dicembre 2022 - Gennaio 2023

Abstract

Un Digital Twin è la rappresentazione digitale di un sistema fisico, che simula il suo stato e i suoi processi in corso. Il nascita del concetto di Digital Twin è attribuita nel corso del progetto "Apollo" della NASA, quando fu ideato un "twin" per fare mirroring delle condizioni di volo di una navicella realmente lanciata nello spazio. L'uso di Digital Twin ha subito conquistato diversi settori ed aree di ricerca, da quello industriale a quello dell'healthcare. L'utilizzo di tecnologie DT-based potrebbe rivoluzionare la medicina che noi oggi conosciamo; diversi sono stati i tentativi della rappresentazione di un "twin" digitale per la rappresentazione di diversi scenari biomedici (ospedali, strutture sanitarie, medici, pazienti).

La realizzazione di un Patient-Twin è la nuova frontiera dell'healthcare; diversi sono stati i tentativi di realizzazione di gemelli digitali per modellare gli aspetti clinicamente rilevati di un paziente. Il seguente lavoro pone attenzione su una proposta sperimentale di "Patient Digital Twin" (*Graph Representation Forecasting of Patient's Medical Conditions: Toward a Digital Twin - Pietro Barbiero et al.* [1]). Partendo dalla proposta applicativa sperimentale degli autori e dallo studio ed'analisi di tale soluzione, per essa è stata realizzata un'infrastruttura che permette, contemporaneamente, di godere integralmente sia degli strumenti dell'applicazione proposta da *Pietro Barbiero et al.* sia dei vantaggi derivanti dall'uso di tecnologie di Cloud-Computing.

La motivazione alla base della realizzazione del lavoro qui descritto trova risposta nei vantaggi che l'unione interdisciplinare tra Healthcare e Machine-Learning può ottenere dall'integrazione di strumenti di cloud-computing a supporto delle operazioni. Diversi aspetti cruciali sono stati affrontati durante la realizzazione del lavoro, dallo studio del dominio applicativo e tecnico, nonchè gli aspetti di Medicina e Machine-Learning che concernono la proposta di *Pietro Barbiero et al.*, allo studio degli attuali providers di Cloud Services, strumenti strategie risolutive da essi messi a disposizione.

1 Introduzione

Digital Twin: cos'è?

Un Digital Twin può essere definito come un *modello digitale di un sistema fisico e dei suoi processi in corso* attraverso una connessione dati che consente di convertire il sistema fisico in un sistema virtuale mantenendo un alto livello di sincronizzazione tra di loro. In termini più semplici, un Digital Twin (**DT**) è un modello virtuale altamente complesso, che è *l'esatta replica* del suo corrispettivo fisico. Questo può essere qualsiasi cosa: *da un'auto, a un macchinario industriale, a un aereo, un ponte, un edificio, una persona e così via.*

Il DT viene aggiornato in tempo reale dai dati raccolti dai **sensori** collegati all'asset fisico e usa programmi di simulazione, l'apprendimento automatico e il ragionamento per fornire informazioni utili sull'asset e per elaborare modelli predittivi delle prestazioni future e delle reazioni dell'oggetto a determinate condizioni.

Nei DTs, *il flusso dei dati è automatico e integrato in entrambe le direzioni* tra sistemi fisici e digitali per sincronizzare l'oggetto digitale con lo stato attuale del fisico e anche inviare informazioni di controllo ad esso.

In realtà, il concetto di Digital Twin *non è niente di nuovo*, come spiega Micheal Grieves [2], ricercatore e professore presso l'Università del Michigan, a cui viene attribuita l'invenzione del gemello digitale; infatti, seguendo le sue parole, "lo spazio virtuale che gli umani hanno usato da sempre per modellare il gemello digitale di un sistema è stato la **mente**, soltanto grazie alle nuove tecnologie questi spazi virtuali hanno trovato posto negli **spazi digitali dei computer**".

Benefici dei Digital Twins

Implementare DTs ha numerosi vantaggi. Alcuni di questi benefici possono essere:

- **Monitoraggio in tempo reale, controllo e acquisizione dei dati:** tramite il gemello digitale gli aggiornamenti in tempo reale vengono scambiati tra i sistemi fisici e digitali consentendo il monitoraggio e l'acquisizione di tutti i dati necessari per le decisioni aziendali e consentendo il controllo sul sistema fisico quando necessario.
- **Efficienza aumentata:** facilitando il processo di test di vari scenari e casi prima della loro implementazione in sistemi fisici; i gemelli digitali forniscono una piattaforma per sperimentare soluzioni in anticipo e implementare le migliori per aumentare l'efficienza generale del sistema.
- **Supporto alle decisioni:** avendo tutti i dati in tempo reale raccolti in una piattaforma e facilmente accessibili per alimentare strumenti di analisi dei dati, si possono prendere decisioni aziendali più veloci, meglio informate ed efficienti.
- **Risk assessment migliorato:** un gemello digitale consente il test virtuale di varie soluzioni per eseguire un'analisi "what-if" per valutare tali soluzioni senza influire sul sistema fisico.

Possibili applicazioni dei Digital Twins

Grazie ai numerosi vantaggi derivanti dal loro uso e dalle loro potenzialità di simulazione di sistemi fisici reali, i Digital Twins ha trovato applicazione in diversi domini. Alcuni di questi domini sono:

- **Settore industriale e manifatturiero:** i DTs per sistemi industriali e manifatturieri possono fornire copie digitali delle loro fabbriche e linee di produzione su cui possono essere testati, migliorati e ottimizzati tutti i processi, senza influire sulla linea produttiva, fornendo monitoraggio, controllo e ottimizzazione in tempo reale.
- **Smart cities:** creare il Digital Twin di una città consente la visualizzazione di tutte le risorse nella città e l'interazione delle persone e dei veicoli all'interno; consente il monitoraggio delle infrastrutture, delle utilità e delle attività commerciali, nonché la pianificazione di sviluppi futuri.
- **Istruzione:** Sono state già proposte diverse soluzioni DT-based per l'istruzione, come gli ambienti di apprendimento intelligenti con i framework di apprendimento adattativo personalizzati, con strumenti di data mining, integrando tecnologie IoT.
- **Medicina:** Il modello dei Digital Twin ha anche la potenzialità di rivoluzionare l'assistenza sanitaria e permettere terapie personalizzate alle caratteristiche e ai bisogni di ciascun paziente. Creare una replica digitale di un paziente, con tutte le sue informazioni – dna, patologie e particolari predisposizioni, storia familiare ecc. – permetterà in futuro agli specialisti sanitari di scegliere la migliore terapia per il singolo. Creare un gemello digitale di una persona, inoltre, permetterebbe al personale sanitario di avere un quadro completo del paziente, indipendentemente dalla struttura e dal personale a cui si rivolge.

Esempi concreti di Digital Twin

In letteratura è possibile osservare come applicazioni di Digital Twins siano già state realizzate, come esse siano di grande aiuto ai processi aziendali e di come essi siano estremamente preziosi nel contesto applicativo di appartenenza. In questa breve sezione elencheremo alcuni esempi concreti di soluzioni DT-based.

- **Fincantieri**, azienda italiana operante nel settore della cantieristica navale, si è classificata tra le aziende vincitrici del secondo bando del Competence Center Smact [3] proprio con un progetto finalizzato alla creazione di un gemello digitale per ottimizzare e ridurre i costi del processo di costruzione di una nave. Il progetto mira al miglioramento delle operazioni di monitoraggio, di assistenza agli operatori, della pianificazione della supply chain e del processo produttivo in ambito industriale e navale con l'applicazione di tecnologie Industry 4.0: Digital Twin e IoT. Il Digital Twin di una officina del cantiere navale di Fincantieri, fornisce la possibilità di simulare le attività quotidiane e fornire informazioni critiche sul funzionamento dei processi in esecuzione e tenere sotto controllo lo svolgimento di tutte le attività produttive, la loro sincronizzazione ed il loro stato di avanzamento. Il Digital Twin dello stabilimento produttivo sarà invece applicato ad un macchinario per la produzione e il controllo di pacchi statorici o rotorici di motori elettrici. [4]
- **Patient's Digital Twin**: in (Pietro Barbiero et al.) [1] è possibile osservare un esempio di Digital Twin in ambito healthcare. Gli autori rimarcano l'obiettivo della medicina moderna, nonché quello di traslare essa dall'essere una disciplina di "cura reattiva" ad una scienza interdisciplinare reattiva. A tal fine, propongono un "gemello digitale" dei pazienti che modella il corpo umano come un tutto e fornisce una visione panoramica sulla condizione individuale. Testano il modello DT su due casi clinici simulati, combinando informazioni a livello di organo, tessuto e cellulare, fornendo una visione d'insieme sia sulla condizione attuale che su quella futura del paziente monitorando e prevedendo dei punti finali clinicamente rilevanti che rappresentano l'evoluzione dei parametri vitali del paziente.

In letteratura è possibile osservare come in realtà siano numerosi i tentativi di applicazioni di tecnologie DT in ambito biomedico, questo perché l'idea di realizzare un Patient Digital Twin così come proposto da *Pietro Barbiero et al.* è affascinante nel campo della ricerca tanto quanto sembra. Qui seguono altri esempi di applicazioni DT-based in ambito biomedico, dall'uso di DT come proposte di gemelli digitali per pazienti a proposte di applicazioni DT per la gestione di strutture sanitarie

- **Human Digital Twin for Personalized Healthcare: Vision, Architecture and Future Directions** (*Samuel D. Okegbile; Jun Cai; Changyan Yi; Dusit Niyato*) [5]: consiste nella proposta di un HDT (Human Digital Twin) che può facilitare lo sviluppo di nuovi farmaci e vaccini senza sottoporre gli umani ad alcuna forma di rischio. Può predire lo stato futuro di salute di un individuo per consentire l'attivazione di misure preventive efficienti. L'HDT aggiungerà valore all'industria sanitaria attraverso diagnosi personalizzate e selezioni terapeutiche.
- **Digital Twins in Healthcare: an architectural proposal and its application in a social distancing case study** (*Alessandra De Benedictis, Nicola Mazzocca, Alessandra Somma, Carmine Strigaro*) [6]: propongono una classificazione delle applicazioni DT in quattro categorie; propongono un'architettura Digital Twin generalizzata che può essere utilizzata come riferimento per identificare i principali componenti funzionali di un sistema Digital Twin; presentano CanTwin, un caso di studio industriale reale sviluppato da Hitachi e rappresentante il Digital Twin di un servizio di mensa che serve 1.100 lavoratori, impostato per monitorare il distanziamento sociale, ispezionare le code, contare le persone e tracciarle, supervisionare l'occupazione dei tavoli.
- **A Healthcare Digital Twin for Diagnosis of Stroke** (*Iqram Hussain, Md. Azam Hossain, Se-Jin Park*) [6]: L'obiettivo di questo studio è proporre un concetto di "digital twin" sanitario e utilizzare i dati dell'EEG (Elettroencefalogramma) e i modelli machine-learning per costruire un digital twin per pazienti con ictus. È stato sviluppato un digital twin salutare per le diagnosi di ictus.

- **The Digital Twin Revolution in Healthcare** (*Tolga Erol and Arif Furkan Mendi and Dilara Dogan*) [7]
- **A user interface design for a patient oriented digital patient** (*Nikolaos Th. Ersotelos and Xia Zhao and Youbing Zhao and Hui Wei and Enjie Liu and Gordon J. Clapworthy and Feng Dong*) [8]

Digital Patient (Pietro Barbiero et al.)

L'applicazione oggetto di studio di questo lavoro è disponibile presso la repository GitHub [9]. Il "gemello digitale" realizzato (Pietro Barbiero et al. [1]) consiste in un sistema assistito da IA modulare che può essere utilizzato per *modellare il corpo umano* nel suo insieme e per *prevedere l'evoluzione delle condizioni patofisiologiche*. Il primo modulo si basa su una rete neurale a grafo (**GNN**) che prevede endpoint clinicamente rilevanti (come la pressione sanguigna), mentre il secondo è rappresentato da una rete generativa avversaria (**GAN**) che fornisce una proof-of-concept di integrabilità multi-omica.

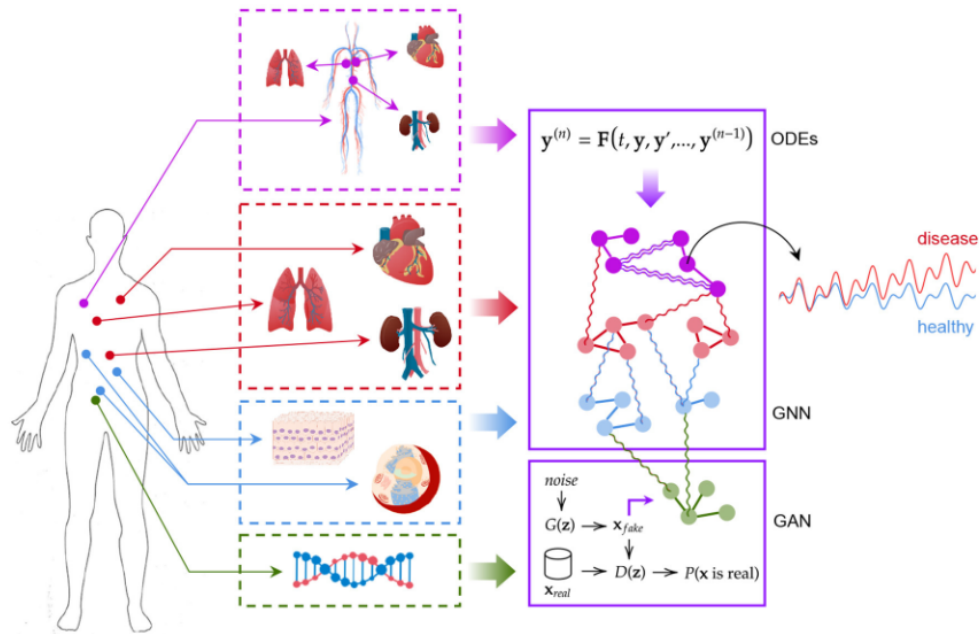


Figure 1: Modello del Digital Twin [1].

I modelli di Deep-Learning sono difficili da interpretare, quindi c'è un crescente interesse nell'interpretabile AI e nell'interpretabile Deep-Learning. I GNN hanno cominciato ad attirare l'attenzione per la loro struttura a grafo facilmente comprensibile e le loro caratteristiche come non linearità, interpretabilità, geometria non euclidea, modularità, cross-modalità, generatività, multiscale e densità spettrale. Queste proprietà rendono i GNN adatti per l'**analisi dei dati medici**. Il modello digitale del paziente è composto da quattro strati biologici: transcriptomic layer, cellular layer, organ layer, exposomic layer.

Obiettivi di progetto: opportunità sul Cloud

Il nostro obiettivo è duplice. Il primo obiettivo è sicuramente quello di comprendere a fondo il dominio applicativo (medico) e tecnico (machine-learning) di ciò che è stato sviluppato da Pietro Barbiero et al. così da poter usufruire degli strumenti messi a disposizione dall'applicazione ([9]). Il secondo obiettivo, per poterlo comprendere a pieno, è da posizionarsi su un livello di sfida superiore.

La nostra idea è quella di presentare una proof-of-concept per un sistema che, oltre a fornire strumenti di machine-learning che operano nel campo della medicina (come ad esempio il Digital Twin proposto nella sottosezione precedente), integra al suo interno elementi di tecnologie cloud. Questo obiettivo fonda le sue motivazioni nei vantaggi intrinseci che deriverebbero dall'uso di Cloud-Computing technologies in ambito Machine-Learning. Le tecnologie cloud offrono la capacità di scalare rapidamente e in modo efficace le applicazioni di machine learning che richiedono maggiore capacità di elaborazione. Ciò significa che le aziende possono implementare rapidamente ed usare efficientemente applicazioni di machine learning che richiedono anche grandi quantità di risorse di elaborazione.

Prevediamo che i progressi nella medicina e nell'informatica possano, in un futuro molto prossimo, richiedere notevoli quantità di potenza computazionale così come grandi necessità di scalabilità, affidabilità e sicurezza sicuramente desiderabili in un campo critico tanto quanto quello generato dall'interazione e dalle sinergie tra medicina ed informatica. Pertanto il nostro obiettivo è quello di presidiare un'infrastruttura pronta ad ospitare Digital-Twin providers e rendere per essi disponibili tutti i vantaggi delle tecnologie basate su cloud.

2 Background

In questa sezione si discuterà degli aspetti salienti delle tecnologie utilizzate, a partire dalla GNN che rappresenta in tutto e per tutto il digital twin virtuale, passando per Docker che ci ha permesso di eseguire il modello in un ambiente isolato e minimale, per arrivare al cloud con Amazon Web Services (AWS) e i relativi servizi come Elastic Container Registry (ECR), Elastic Container Service (ECS), Fargate ed S3.

Grafi e reti neurali: Graph Neural Network

I grafi sono strutture matematiche usate per modellare una insieme di nodi, cioè oggetti, e gli archi, cioè le loro relazioni. I grafi vengono impiegati in molti ambiti di ricerca siccome forniscono una struttura dati flessibile per la modellazione di sistemi del mondo reale. Le GNN sono un modello di deep-learning basato sul dominio dei grafi. La teoria dei grafi combinata con gli elementi delle reti neurali hanno reso le GNN uno degli strumenti più promettente per l'analisi di sistemi complessi in tale dominio. Il vantaggio principale delle GNN rispetto agli altri modelli di ML consiste nella loro architettura estremamente flessibile ed interpretabile.

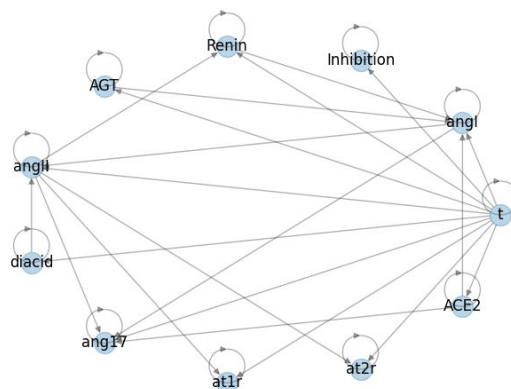


Figure 2: Grafo sistema renina-angiotensina

Il framework utilizzato, proposto da (Battaglia et al.)[10] e basato sulle GNN, si poggia su moduli chiamati graph network block (GN blocks) che rappresentano le unità di calcolo fondamentali della GNN.

Blocchi GN multipli possono essere composti o combinati con altre reti neurali per generare architetture complesse. Una GNN può essere definita come una tupla $G = (\mathbf{u}, H, E)$. $H = \{\mathbf{h}_i\}_{i=1:N^v}$ è l'insieme di nodi dove le caratteristiche di ogni nodo sono indicate da \mathbf{h}_i . $E = \{(\mathbf{e}_k, r_k, s_k)\}$ è l'insieme di archi in cui ogni nodo è rappresentato dalle proprie caratteristiche \mathbf{e}_k , il nodo che riceve \mathbf{r}_k ed il nodo che invia \mathbf{s}_k . \mathbf{u} denota un insieme di attributi globali che rappresentano lo stato del sistema sottostante. Ogni blocco GN consiste di tre funzioni di aggiornamento ϕ ,

$$\begin{aligned} \mathbf{e}'_k &= \phi^e(\mathbf{e}_k, \mathbf{h}_{r_k}, \mathbf{h}_{s_k}, \mathbf{u}) \\ \mathbf{h}'_i &= \phi^h(\bar{\mathbf{e}}_k, \mathbf{h}_i, \mathbf{u}) \\ \mathbf{u}' &= \phi^u(\mathbf{e}', \mathbf{h}', \mathbf{u}) \end{aligned} \quad (1)$$

e tre funzioni di aggregazione ρ ,

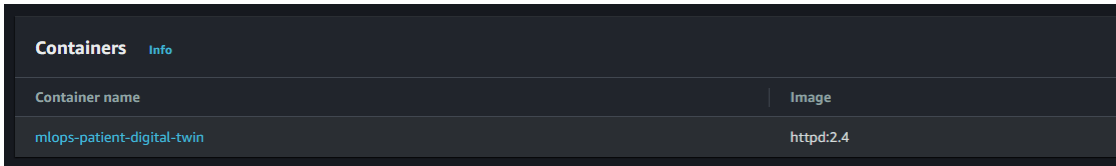
$$\begin{aligned} \bar{\mathbf{e}}'_i &= \rho^{e \rightarrow h}(E'_i) \\ \bar{\mathbf{e}}' &= \rho^{e \rightarrow u}(E') \\ \bar{\mathbf{h}}' &= \rho^{h \rightarrow u}(H') \end{aligned} \quad (2)$$

dove $E'_i = \{(\mathbf{e}'_k, r_k, s_k)\}$, $E' = \bigcup_i E'_i = \{(\mathbf{e}'_k, r_k, s_k)\}_{k=1:N^e}$, $H' = \{(\mathbf{h}'_i)\}_{i=1:N^v}$. Per effettuare l'addestramento completo di un blocco GN, sono necessari 6 steps, alternando l'aggiornamento con l'aggregazione. Per ogni arco, E'_i è calcolato attraverso la funzione di aggiornamento ϕ^e . Il risultato è poi aggregato tramite la funzione $\rho^{e \rightarrow v}$. L'output $\bar{\mathbf{e}}'_i$ corrisponde all'aggiornamento dell'arco ed è utilizzato per aggiornare la rappresentazione del nodo \mathbf{h}'_i tramite ϕ^h . $\rho^{e \rightarrow u}$ e $\rho^{h \rightarrow u}$ eseguono l'aggregazione generando $\bar{\mathbf{e}}'$ e $\bar{\mathbf{h}}'$ rispettivamente dall'aggiornamento di archi e nodi. Gli attributi globali rappresentati da \mathbf{u}' sono calcolati sfruttando le informazioni da $\bar{\mathbf{e}}'$, $\bar{\mathbf{h}}'$, e \mathbf{u} attraverso la funzione ϕ^u . Il processo di apprendimento di ogni blocco GN potrebbe dipendere o meno dagli altri blocchi. Agli archi, come al flusso delle informazioni o agli attributi globali potrebbero essere applicati dei vincoli, ma in questo lavoro di Barbiero et al. [1] non è stato applicato alcun vincolo.

Docker

Docker è una piattaforma di containerizzazione che consente di distribuire facilmente software in modo portatile. Un container Docker è un'immagine di un'applicazione che include tutte le dipendenze necessarie, come librerie e file di configurazione, in modo da poter essere eseguita in qualsiasi ambiente che supporti Docker.

Docker rende facile distribuire e gestire le applicazioni in modo consistente, indipendentemente dall'ambiente in cui vengono eseguite. Ad esempio, è possibile utilizzare Docker per creare un'immagine di un'applicazione e quindi eseguirla in modo identico sia in un ambiente di sviluppo che in produzione, senza doversi preoccupare della configurazione dell'ambiente o della gestione delle dipendenze.



Containers Info	
Container name	Image
mlops-patient-digital-twin	httpd:2.4

Figure 3: Container Docker deployato su Cloud.

Docker offre anche un'ampia gamma di strumenti per la creazione, il push e il pull di immagini di container su un registry, come ad esempio Docker Hub, e per la gestione dei container in esecuzione, come ad esempio Docker Compose per la definizione di workflow multi-container.

Elastic Container Service & Elastic Container Registry

Amazon Elastic Container Registry è un servizio di container registry di Amazon Web Services che consente di archiviare, gestire e distribuire immagini di container. ECR supporta i principali formati di

container, come Docker e OCI, e offre integrazioni con altri servizi AWS, come ad esempio Amazon ECS.

Amazon Elastic Container Service è un servizio di container orchestration di AWS che consente di eseguire, gestire e scalare facilmente container su una varietà di infrastrutture, come Amazon EC2 o Amazon Fargate. ECS supporta una varietà di modelli di distribuzione, come l'orchestrazione di task o di servizi, e offre funzionalità di integrazione con altri servizi AWS, come ad esempio Amazon ECR.

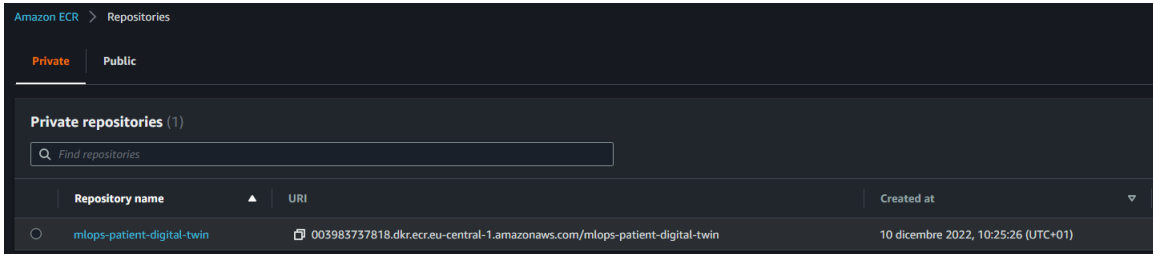


Figure 4: Elastic Container Repository del progetto digital-patient-twin.

In pratica, ECR può essere utilizzato per archiviare e gestire immagini di container, mentre ECS può essere utilizzato per eseguire e gestire container basati su queste immagini su una varietà di infrastrutture. Ad esempio, si potrebbe utilizzare ECR per archiviare un'immagine di un modello di machine learning e quindi utilizzare ECS per eseguire il modello in produzione, gestendo automaticamente il rollout di nuove versioni e la scalabilità delle risorse di calcolo.

Fargate

AWS Fargate è un servizio di Amazon Web Services (AWS) che consente di eseguire container Docker senza dover gestire i cluster di container o i nodi di calcolo. Con Fargate, è possibile distribuire facilmente le applicazioni containerizzate su ECS o Amazon Elastic Kubernetes Service (EKS) senza dover preoccuparsi della configurazione dell'infrastruttura sottostante. Fargate offre un modo semplice e scalabile per eseguire i container in modo nativo su AWS, eliminando la necessità di gestire manualmente i cluster di container.

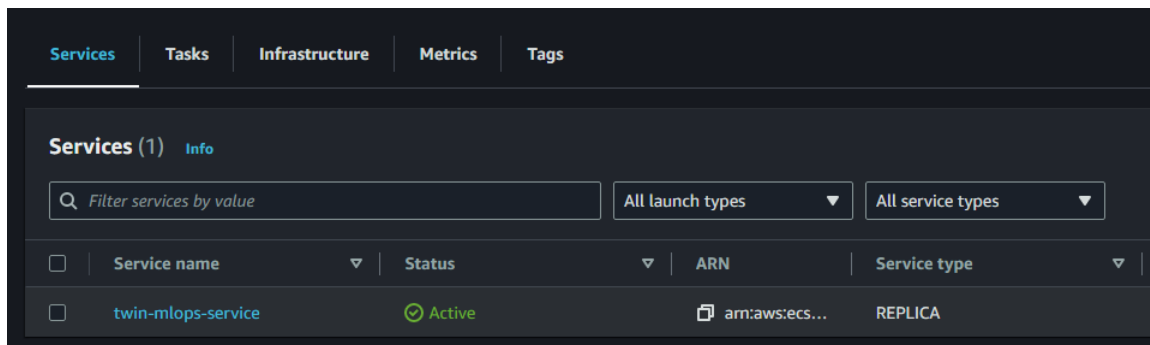


Figure 5: Cluster Fargate del digital-patient-twin.

S3: Storage as a Service

Amazon S3 (Simple Storage Service) è un servizio di archiviazione su cloud di Amazon Web Services (AWS). È progettato per offrire una scalabilità senza precedenti, alta affidabilità e bassi costi di gestione.

Gli utenti possono usare S3 per archiviare e recuperare dati da qualsiasi luogo su Internet, in modo facile e a basso costo. S3 offre diversi tipi di archiviazione, tra cui archiviazione standard, archiviazione a freddo (per dati che vengono utilizzati meno frequentemente) e archiviazione ad alte prestazioni (per dati che richiedono tempi di accesso rapidi).

S3 è utile per molti tipi di organizzazioni e può essere utilizzato per archiviare qualsiasi tipo di dato, come file di testo, file multimediali, backup di dati, immagini e molto altro ancora. È particolarmente adatto per le aziende che hanno bisogno di archiviare grandi quantità di dati, ma non vogliono dover gestire e mantenere un sistema di archiviazione locale. Inoltre, S3 offre funzionalità di sicurezza avanzate, come la crittografia dei dati in transito e in riposo, per garantire la protezione dei dati degli utenti.

CI/CD e GitHub Actions

La CI/CD è una pratica di sviluppo software che mira ad automatizzare il processo di integrazione e distribuzione del codice sorgente di un'applicazione. In pratica, con la CI/CD, ogni volta che uno sviluppatore effettua un commit del proprio codice, questo viene automaticamente compilato, testato e distribuito, senza che sia necessario intervenire manualmente.

Il vantaggio principale della CI/CD è che consente di ridurre i tempi di rilascio di un'applicazione, poiché il codice viene testato e distribuito in modo automatico, senza che sia necessario attendere che gli sviluppatori eseguano manualmente queste operazioni. Inoltre, la CI/CD aiuta a garantire la qualità del codice, poiché il processo di integrazione e distribuzione viene eseguito in modo automatico e ripetuto frequentemente, il che permette di individuare e risolvere i problemi in modo tempestivo.

La CI/CD è spesso utilizzata in concomitanza con il versionamento del codice, come ad esempio Git, e con una piattaforma di gestione dei progetti, come ad esempio Jira. Inoltre, sono disponibili diverse piattaforme di automazione del workflow, come ad esempio Jenkins, Travis CI o GitHub Actions, che possono essere utilizzate per automatizzare il processo di CI/CD.

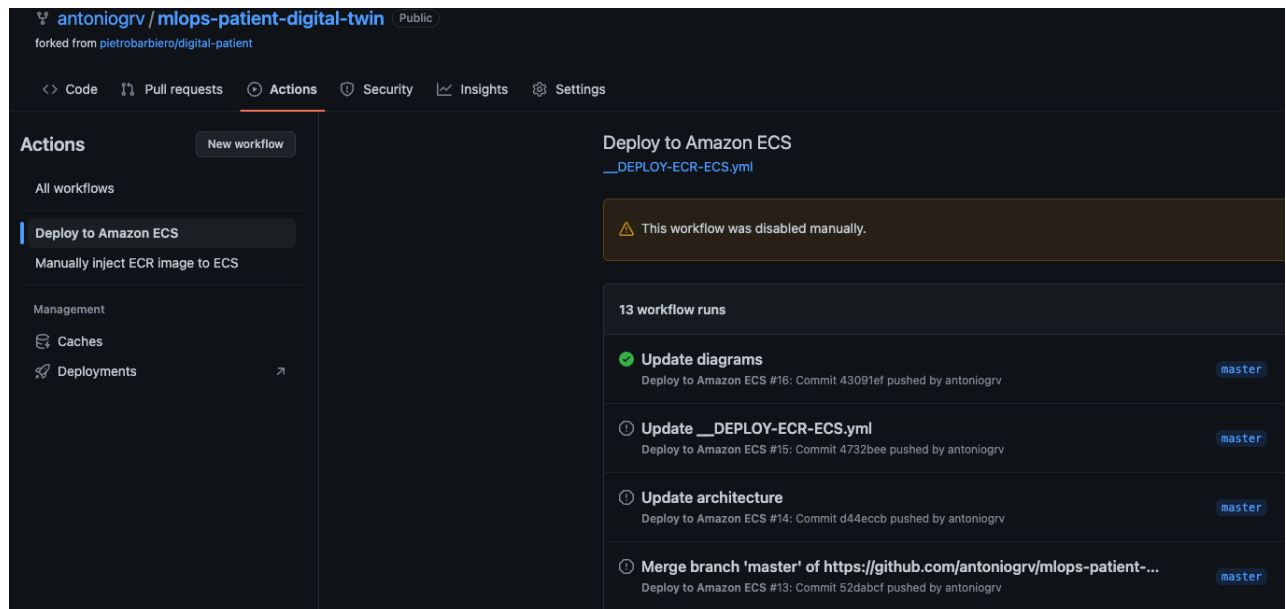


Figure 6: Automatizzazione del processo di costruzione dell'immagine Docker e del suo deploy su ECS via CI/CD.

La CI/CD in GitHub Actions è dunque realizzata attraverso dei workflows scritti in YAML. Il seguente workflow, incluso nella repository del progetto, permette di adempiere alle task principali che s'intende automatizzare. Ulteriori workflows (come l'automatizzazione della creazione della task definition di Fargate, necessaria per la buona riuscita del deploy) coesistono nella medesima directory.

- name: Costruisci l'immagine Docker e deployala su ECR
id: build-image
env:


```

    ECR_REGISTRY: ${ steps.login-ecr.outputs.registry }
    IMAGE_TAG: ${ github.sha }
  run: |
    docker build -t $ECR_REGISTRY/$ECR_REPOSITORY:$IMAGE_TAG .
    docker push $ECR_REGISTRY/$ECR_REPOSITORY:$IMAGE_TAG
    echo "image=$ECR_REGISTRY/$ECR_REPOSITORY:$IMAGE_TAG" >> $GITHUB_OUTPUT

- name: Aggiorna la task definition su ECS
  id: task-def
  uses: aws-actions/amazon-ecs-render-task-definition@v1
  with:
    task-definition: ${ env.ECS_TASK_DEFINITION }
    container-name: ${ env.CONTAINER_NAME }
    image: ${ steps.build-image.outputs.image }

- name: Deploy il container Docker su ECS
  uses: aws-actions/amazon-ecs-deploy-task-definition@v1
  with:
    task-definition: ${ steps.task-def.outputs.task-definition }
    service: ${ env.ECS_SERVICE }
    cluster: ${ env.ECS_CLUSTER }
    wait-for-service-stability: true

```

3 Sperimentazione e relativa implementazione

In questa sezione dapprima vi sarà una descrizione dello scenario clinico relativa alla sperimentazione effettuata poi si affronteranno alcuni aspetti implementativi, che conducono ai risultati si quest'ultima. L'obiettivo è dunque quello di evidenziare lato codice come si arriva al risultato, e di mostrare piccole modifiche apportate ad un lavoro complesso e completo lato ML. La spiegazione del codice e delle scelte effettuate, inoltre, sarà accompagnata dalla motivazione medica laddove possibile.

Scenario clinico

Lo scenario clinico consiste in un paziente anziano che soffre di ipertensione, diabete ed ha contratto un'infezione da SARS-CoV2. Lo stile di vita del paziente è sedentario ed ha una dieta ricca di carboidrati. L'obiettivo è quello di mostrare come un modello di paziente digitale può essere utilizzato per simulare l'evoluzione nel tempo degli endpoint clinici.

Sistema renina-angiotensina e SARS-CoV2

Il sistema renina-angiotensina (RAS) è un sistema ormonale che regola la vasocostrizione e la risposta infiammatoria. Il principale regolatore del RAS è l'ormone peptidico Angiotensina II (ANG-II), generato dall'enzima di conversione dell'angiotensina (ACE). L'ANG-II esercita le sue funzioni attraverso due recettori di tipo proteina-G (AT1R e AT2R) e l'eptapeptide Angiotensina (1-7) (ANG(1-7)), che si lega e attiva il recettore di tipo proteina-G Mas (MAS). L'ANG(1-7) può essere generato sia dall'ACE2 a partire dall'ANG-II, sia dall'enzima endopeptidasi neutra (NEP) a partire dall'ANG-I. I tre recettori di tipo proteina-G (AT1R, AT2R e MAS) aiutano il corpo a regolare la pressione sanguigna durante il corso della giornata [11][12]. L'AT1R stimola la vasocostrizione, l'ipertensione e la risposta infiammatoria, mentre il MAS promuove la vasodilatazione, l'ipotensione e la vasoprotezione. Il ruolo dell'AT2R è controverso, ma in condizioni fisiologiche normali contrasta gli effetti dell'AT1R. Tuttavia, gli effetti vasodilatatori dell'AT2R non vengono associati a una significativa riduzione della pressione sanguigna [13]. I fattori esterni che influiscono sul RAS includono la concentrazione di glucosio, i trattamenti con inibitori dell'ACE e le infezioni virali che si legano all'ACE2, come avviene per il SARS-CoV-2. Un'elevata concentrazione

di glucosio può determinare condizioni di ipertensione cronica, per questo i trattamenti per l'ipertensione spesso includono farmaci inibitori dell'ACE per compensare la produzione eccessiva di ANG-II e AT1R [14]. Le infezioni virali come COVID-19 possono anche avere un impatto negativo sul RAS, poiché il virus si lega all'ACE2 per entrare nella cellula ospite, impedendo l'attività dell'enzima nella generazione dell'ANG(1-7) [15].

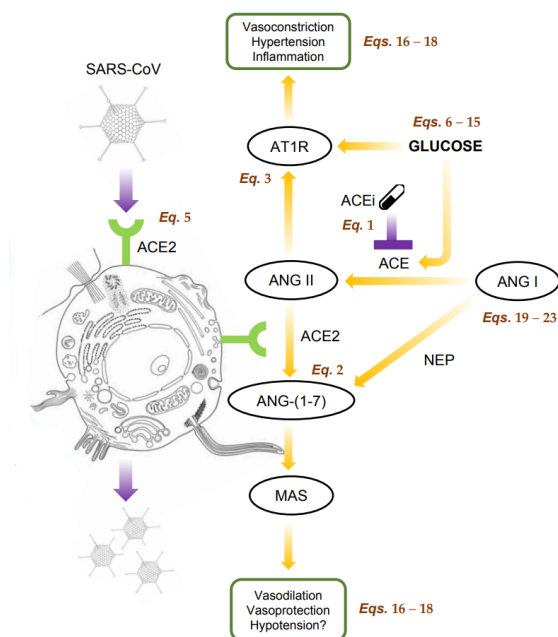


Figure 7: Relazione tra RAS e Covid

Dataset

I due dataset utilizzati sono dunque relativi al paziente anziano precedentemente descritto.

Dataset	Tipo Features	Numero Features	Numero Sample
CARDIO	Sistema Cardiaco	27	1809
RAS	Sistema renina-angiotensina	13	2501

Table 1: Caratteristiche Dataset

Questi due dataset sono strettamente correlati come è possibile dedurre dalla sottosezione relativa al RAS e COVID. Infatti come abbiamo appena visto, il RAS è una delle principali vie biologiche che regolano la pressione del sangue ed è strettamente collegata all'infezione da SARS-CoV2, siccome il virus si lega all'ACE2 per entrare nella cellula ospite. Ed il dataset cardio contiene valori (pressioni) la cui alterazione è dovuta proprio a fattori relativi al RAS, come ad esempio l'ACE2 che impedisce l'attività dell'ANG(1-7).

DigitalTwin: uno sguardo sull'implementazione

Un DT come anticipato nella sezione introduttiva è un modello virtuale progettato per riflettere accuratamente un oggetto fisico, quest'ultimo attraverso sensori produce dati che vengono trasmessi alla copia digitale. Quando la copia ottiene i dati, questi possono essere utilizzati per effettuare simulazioni, studi ed in linea generale ottenere informazioni preziose. Il DT prodotto da (Pietro Barbiero et al.)[1] ha come

obiettivo quello di fornire una panoramica sulle condizioni fisiopatologiche attuali e future del paziente esaminato.

Il DT in questione è realizzato mediante la DGL (DeepGraphLibrary), vengono dunque utilizzati i deep graph che è possibile inquadrare come l'applicazione di grafi alle reti neurali con più "hidden layer", cioè funzioni matematiche ciascuna progettata per ottenere un output specifico, spiegate nella sezione precedente (Equazione 1 e 2). Nel concreto i nodi e gli archi utilizzati per la costruzione del deep graph vengono fissati:

```
elist = [
    #colonne relative a x_ras lega ognuno con se stesso (crea un nodo)
    ('t', 't'), ('angI', 'angI'), ('Inhibition', 'Inhibition'),
    ('Renin', 'Renin'), ('AGT', 'AGT'), ('angII', 'angII'),
    ('diacid', 'diacid'), ('ang17', 'ang17'), ('at1r', 'at1r'),
    ('at2r', 'at2r'), ('ACE2', 'ACE2'),

    #colonne relative a x_ras lega il tempo con tutti
    ('t', 'angI'), ('t', 'Inhibition'), ('t', 'Renin'), ('t', 'AGT'),
    ('t', 'angII'), ('t', 'diacid'), ('t', 'ang17'), ('t', 'at1r'),
    ('t', 'at2r'), ('t', 'ACE2'),

    #colonne relative a x_ras legami angI
    ('AGT', 'angI'), ('Renin', 'angI'), ('angI', 'ang17'),
    ('angI', 'angII'),

    #colonne relative a x_ras legami angII
    ('diacid', 'angII'), ('angII', 'Renin'), ('angII', 'ang17'),
    ('angII', 'at1r'), ('angII', 'at2r'),

    #colonne relative a x_ras legami ACE2
    ('ACE2', 'ang17'), ('ACE2', 'angI'),

    #colonne relative a x_cardio lega tempo con tutti
    ('t2', 'Pra'), ('t2', 'Prv'), ('t2', 'Pla'), ('t2', 'Plv'),

    #colonne relative a x_cardio lega tutti 4 valori restanti fra loro
    ('Pra', 'Prv'), ('Pra', 'Pla'), ('Pra', 'Plv'),
    ('Prv', 'Pra'), ('Prv', 'Pla'), ('Pra', 'Plv'),
    ('Pla', 'Pra'), ('Pla', 'Prv'), ('Pla', 'Plv'),
    ('Plv', 'Pra'), ('Plv', 'Prv'), ('Plv', 'Pla'),
]
```

La modellazione del grafo viene dunque effettuata sfruttando conoscenze mediche. I nodi corrispondono alle variabili rappresentate dalle equazioni differenziali presenti nel lavoro (Barbiero e Lió)[16]

Organizzazioni dei dati

Come è possibile vedere, nella tabella 1, il dataset CARDIO ha 27 features, ma in realtà solo 5 di queste vengono utilizzate:

- **t2**: il tempo
- **Pra,Prv,Pla,Plv**: indicano rispettivamente la pressione dell'atrio e ventricolo destro e sinistro;

Anche per il dataset DKD non tutte le features vengono utilizzate, infatti delle 13, 2 vengono scartate:

- **t**: il tempo
- **angI,Inhibition,Renin,AGT,angII,diacid,ang17,at1r,at2r,ACE2**: indicano valori relativi al sistema renina-angiotensina

Si procede poi con la riduzione del numero di sample, passando da 1809 per CARDIO e 2501 per RAS ad un unico insieme di 300 samples. Questi nuovi valori sono ottenuti mediante una funzione di interpolazione applicata a 300 valori generati in un intervallo fissato. La funzione di interpolazione è invece ottenuta interpolando ogni feature di un dataset con il rispettivo tempo. L'insieme di 300 samples avrà le 16 features in precedenza elencate e sulle quali il deep graph viene creato, come viene mostrato nel codice nella sezione relativa al DT. I samples vengono poi concatenati 20 volte ottenendo un insieme finale con shape(6000,16).

Lo step conclusivo è quello di separare i samples dalle labels e viene effettuato nel seguente modo:

```
#parte da 0 arriva a 4001
for batch in range(x.shape[0] - 2 * window_size + 1):
    samples.append(x[batch:batch + window_size - 2])
    labels.append(x[batch + window_size - 1:batch + 2 * window_size - 3])

#ogni sample cosi come ogni lable sara'
#una lista di valori, cioe'

#il primo sample saranno i valori x[0:998]
#mentre la prima label saranno i valori x[999:1997]

#il secondo sample saranno i valori x[1:999]
#mentre la prima label saranno i valori x[1000:1998]

#l'ultimo sample saranno i valori x[4000:4998]
#mentre la prima label saranno i valori x[4999:5997]
```

Training e risultati

I dati per test e train vengono suddivisi andando ad utilizzare la funzione train_test_split, in particolare il 70% viene utilizzato per il training ed il 30% utilizzato per il test. Tale funzione automatizza un blocco di istruzioni in cui i dati venivano organizzati, andando inoltre a snellire il codice. Continuando con i dettagli relativi al training, il numero di epoche utilizzato è 20 ed il learning rate è 0.01.

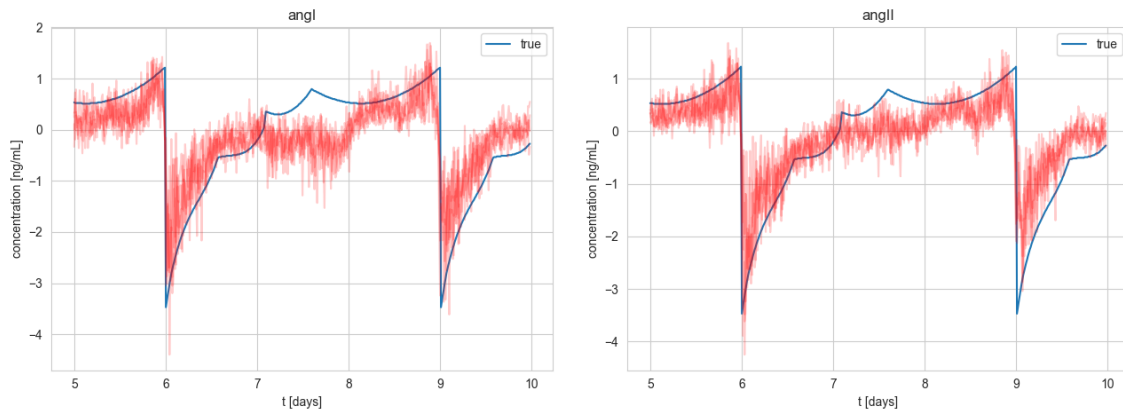


Figure 8: Rispettivamente Angiotensin I e Angiotensina2

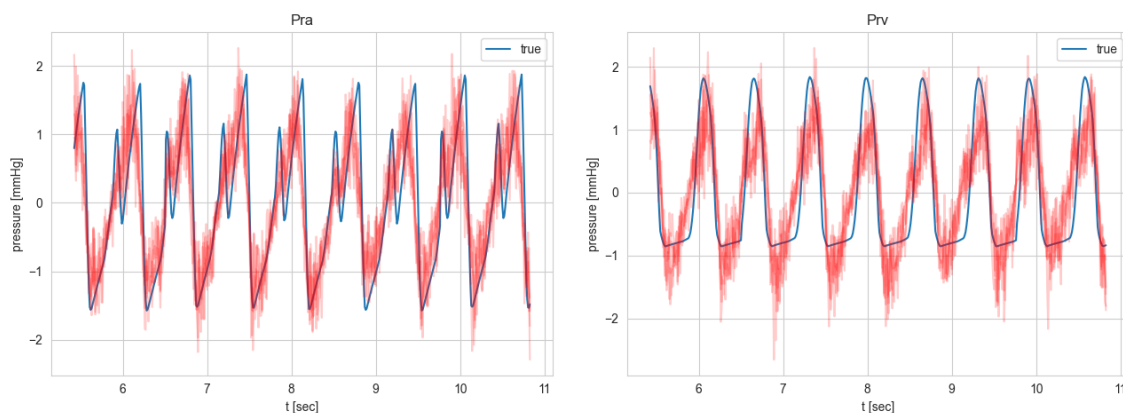


Figure 9: Rispettivamente pressione atrio e ventricolo destro

I grafici visibili nella figura 8 sono i risultati relativi all'angiotensina 1 e all'angiotensina 2, in particolare vi è una previsione di quella che sarà la concentrazione (ng/mL). La linea blu traccia i valori effettivi di concentrazione nel tempo (giorni), mentre la parte rossa indica la previsione del modello. Quelli visibili nella figura 9 sono invece i risultati relativi alla pressione dell'atrio e ventricolo destro. Vi è una previsione di quella che sarà la pressione in millimetri di mercurio (mmHg). Anche qui la linea blu traccia i valori effettivi di pressione nel tempo (secondi), mentre la parte rossa indica la previsione del modello.

4 Cloud & Machine Learning Operations

L'MLOps (anche noto come *Machine Learning Operations*, o anche *DevOps per il Machine Learning*) è una pratica che mira a rendere più efficiente il processo di sviluppo e implementazione di modelli di machine learning in ambienti di produzione. Ciò può includere l'automazione del processo di addestramento e distribuzione di modelli di machine learning, la gestione dei dati e delle risorse di calcolo, il monitoraggio dei modelli in produzione e il rollout continuo di nuove versioni.

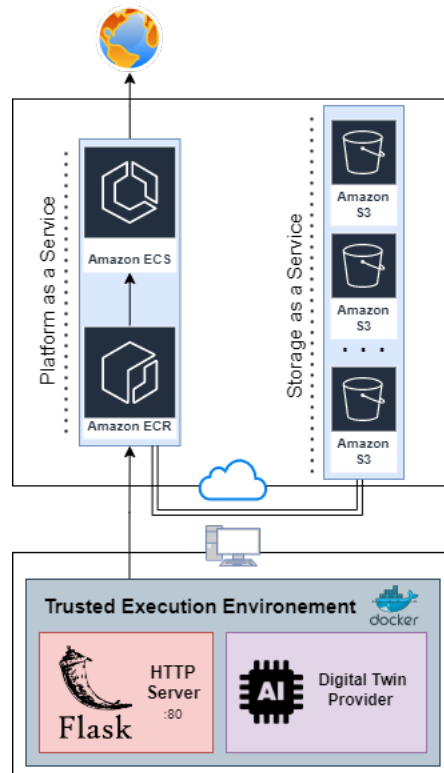
4.1 Caratteristiche e vantaggi del MLOps

L'obiettivo principale del MLOps è quello di rendere più veloce ed efficiente il ciclo di vita dei modelli di machine learning, permettendo alle organizzazioni di sfruttare al meglio le opportunità offerte dal machine learning e di rispondere rapidamente ai cambiamenti nel mercato. MLOps richiede l'adozione di best practice e strumenti specifici per il machine learning, come ad esempio framework di orchestrazione dei modelli, sistemi di tracciamento delle versioni e strumenti di monitoraggio dei modelli.

Ci sono diversi motivi per cui potrebbe essere utile deployare una rete neurale su cloud:

- **Scalabilità:** il cloud offre risorse di calcolo e storage a disposizione in modo elastico, permettendo di scalare facilmente il numero di risorse utilizzate in base alle esigenze. Ciò è particolarmente importante per le reti neurali, che possono richiedere molta potenza di calcolo durante l'addestramento o l'esecuzione.
- **Facilità d'uso:** molti fornitori di cloud offrono servizi di machine learning "fully managed", che consentono di eseguire il training e il deployment di modelli di machine learning senza doversi preoccupare della gestione dell'infrastruttura sottostante.
- **Accessibilità:** il cloud rende accessibili risorse di calcolo e storage da qualsiasi luogo, semplicemente utilizzando una connessione a Internet. Ciò è particolarmente utile per le organizzazioni che non hanno accesso a hardware di calcolo potente in loco o che non vogliono investire nell'acquisto di hardware.

- **Riduzione dei costi:** utilizzando il cloud, è possibile pagare solo per le risorse di calcolo e storage utilizzate, evitando di dover investire in hardware costoso e di doversi preoccupare della sua manutenzione.
- **Disponibilità:** i fornitori di cloud offrono livelli elevati di disponibilità e ridondanza, il che significa che è meno probabile che il servizio venga interrotto a causa di problemi hardware o di altro tipo. Ciò è particolarmente importante per i modelli di machine learning che devono essere disponibili 24/7.



L'architettura proposta prevede di realizzare un'immagine Docker contenente tutte le dipendenze necessarie per poter permettere agli script Python pre-esistenti, forniti dagli accademici autori della repository originale, di essere lanciati senza problemi, in maniera portabile e sicura. Fra le dipendenze pre-installate sull'immagine Docker si annoverano **pytorch**, **numpy** e **pandas**. L'immagine Docker viene generata ogni volta che un collaboratore alla repository effettua una push via GitHub. Il trigger della push permette di inizializzare una routine di **CI/CD** implementata via **GitHub Actions**.

Il frontend dell'applicazione ha lo scopo di permettere un minimo e limitato grado di interazione con il Digital Twin Provider ed è reso disponibile nel contesto di un container Docker portabile e isolato rispetto al resto del sistema operativo. La virtualizzazione, in questo caso, permette di installare pacchetti Python

critici per il funzionamento del backend, come flask e flask-cors, senza contaminare la macchina ospitante e incorrere in potenziali problemi di versioning.

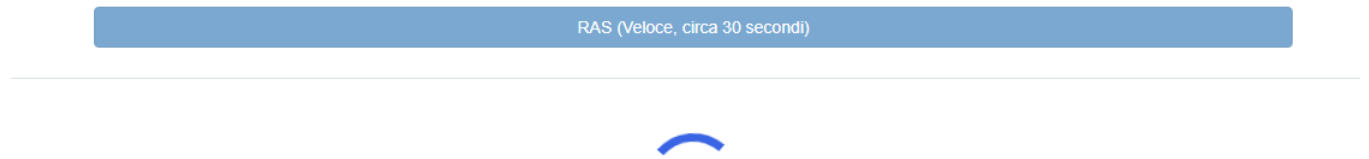


Figure 11: L'attivazione del meccanismo di polling restituisce uno spinner di loading fino al caricamento dei dati su S3.

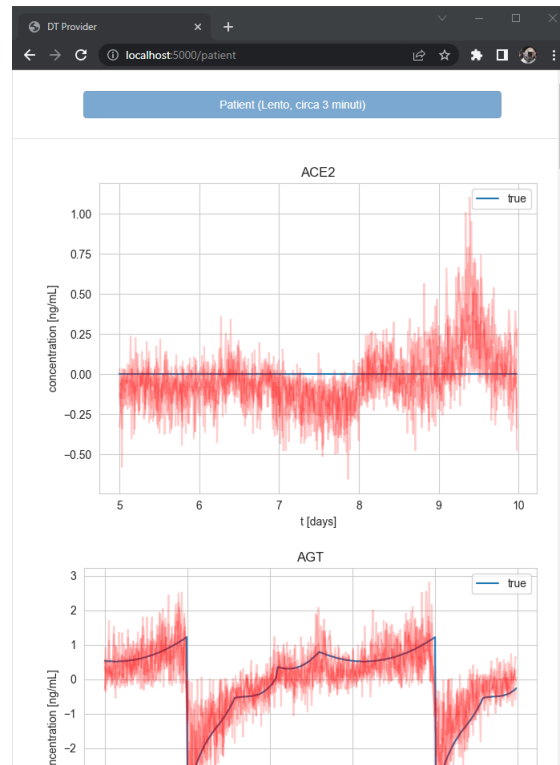
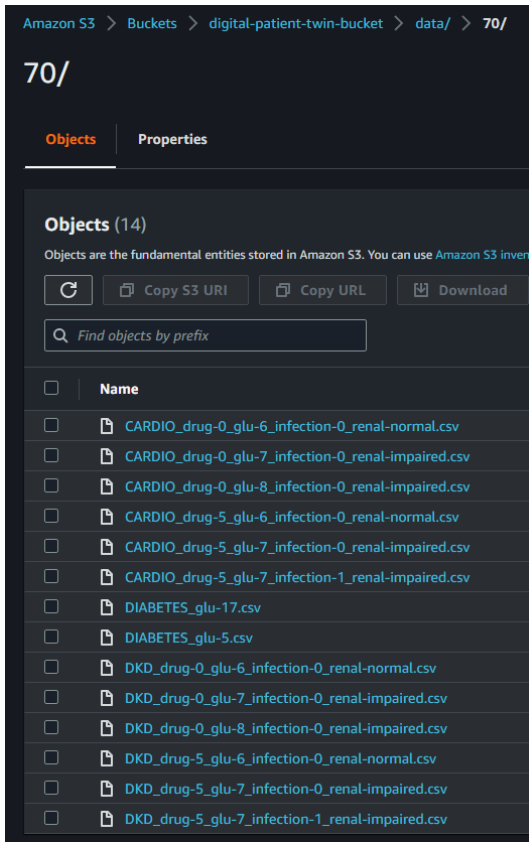
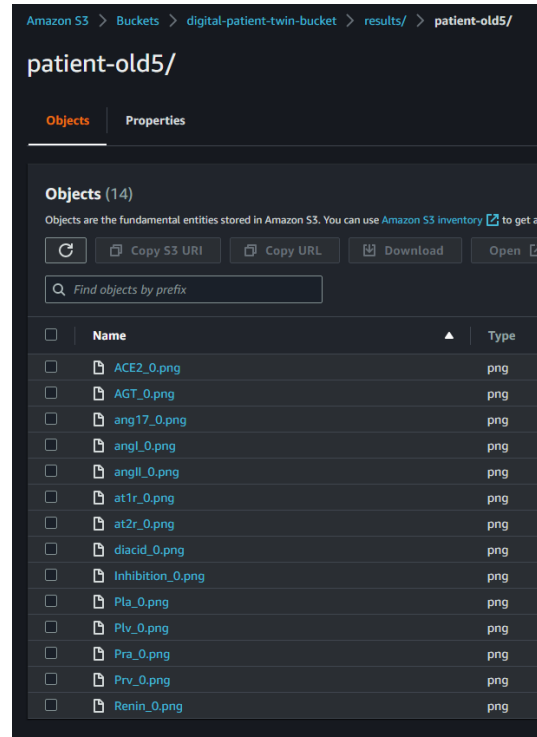


Figure 12: I risultati vengono mostrati in maniera sequenziale. L'applicativo, inoltre, è responsive.

In generale, la piattaforma - realizzata come proof of concept - permette di connettersi al provider tramite un canale bidirezionale realizzato tramite polling. Alla pressione di un certo bottone, verrà invocata una RESTful API (organicamente indicata fra le route della webapp Flask) che, a sua volta, girerà la richiesta al digital twin provider effettivo, trasmettendo infine i risultati su S3. Il polling, in questo caso, permette di determinare se il caricamento su S3 è avvenuto con successo o meno.



(a) Dataset caricato su S3



(b) Risultati dello script patient.py sul cloud

Figure 13: Sequenza di operazioni su Amazon S3

I dati caricati su S3 riguardano sia i dataset per il modello di machine learning, sia i risultati generati da quello stesso modello.

A basso livello, la webapp in Flask è in realtà un applicativo molto semplice e costituito da una manciata di endpoints. La route principale è la seguente, e permette di interfacciare l'ambiente HTTP al Digital Twin Provider.

```
@app.route('/run/<command>')
def run(command):
    dir = str(Path('__twin/examples/' + command + '.py').parent.absolute())
    path = str(dir) + '/' + command + '.py'
    print(path)
    out = os.popen('python3.7 ' + path).read()
```

Riferimenti

- [1] Pietro Barbiero, Ramon Viñas Torné, and Pietro Lió. “Graph Representation Forecasting of Patient’s Medical Conditions: Toward a Digital Twin”. In: *Frontiers in Genetics* 12 (Sept. 2021). DOI: 10.3389/fgene.2021.652907. URL: <https://doi.org/10.3389/fgene.2021.652907>.
- [2] John Vickers Michael Grieves. “Origins of the Digital Twin Concept”. In: (Aug. 2016). DOI: 10.13140/RG.2.2.26367.61609. URL: <https://doi.org/10.13140/RG.2.2.26367.61609>.

- [3] Nicoletta Pisanu. *Dal letto smart al digital twin navale, i 13 vincitori del bando del Competence center Smact*. <https://www.innovationpost.it/attualita/formazione-e-competenze/letto-smart-al-digital-twin-navale-competence-center-smact-vincitori-bando/>.
- [4] Fincantieri. *Digital Twin and Industrial Internet of Things for Manufacturing 4.0*. <https://www.fincantieri.com/it/innovazione/progetti-di-innovazione/aware/>.
- [5] Samuel D. Okegbile et al. “Human Digital Twin for Personalized Healthcare: Vision, Architecture and Future Directions”. In: *IEEE Network* (2022), pp. 1–7. DOI: 10.1109/mnet.118.2200071. URL: <https://doi.org/10.1109/mnet.118.2200071>.
- [6] Alessandra De Benedictis et al. “Digital Twins in Healthcare: an architectural proposal and its application in a social distancing case study”. In: *IEEE Journal of Biomedical and Health Informatics* (2022), pp. 1–12. DOI: 10.1109/jbhi.2022.3205506. URL: <https://doi.org/10.1109/jbhi.2022.3205506>.
- [7] Tolga Erol, Arif Furkan Mendi, and Dilara Dogan. “The Digital Twin Revolution in Healthcare”. In: *2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*. IEEE, Oct. 2020. DOI: 10.1109/ismsit50672.2020.9255249. URL: <https://doi.org/10.1109/ismsit50672.2020.9255249>.
- [8] Nikolaos Th. Ersotelos et al. “A user interface design for a patient oriented digital patient”. In: *13th IEEE International Conference on BioInformatics and BioEngineering*. IEEE, Nov. 2013. DOI: 10.1109/bibe.2013.6701588. URL: <https://doi.org/10.1109/bibe.2013.6701588>.
- [9] Pietro Barbiero. *digital-patient*. <https://github.com/pietrobarbiero/digital-patient>.
- [10] Peter W. Battaglia et al. *Relational inductive biases, deep learning, and graph networks*. 2018. DOI: 10.48550/ARXIV.1806.01261. URL: <https://arxiv.org/abs/1806.01261>.
- [11] Mariela M. Gironacci et al. “Angiotensin (1-7) Induces Mas Receptor Internalization”. In: *Hypertension* 58.2 (Aug. 2011), pp. 176–181. DOI: 10.1161/hypertensionaha.111.173344. URL: <https://doi.org/10.1161/hypertensionaha.111.173344>.
- [12] Luis C. Matavelli and Helmy M. Siragy. “AT2 Receptor Activities and Pathophysiological Implications”. In: *Journal of Cardiovascular Pharmacology* 65.3 (Mar. 2015), pp. 226–232. DOI: 10.1097/fjc.0000000000000208. URL: <https://doi.org/10.1097/fjc.0000000000000208>.
- [13] Sébastien Foulquier, U. Muscha Steckelings, and Thomas Unger. “Impact of the AT2 Receptor Agonist C21 on Blood Pressure and Beyond”. In: *Current Hypertension Reports* 14.5 (July 2012), pp. 403–409. DOI: 10.1007/s11906-012-0291-6. URL: <https://doi.org/10.1007/s11906-012-0291-6>.
- [14] Mohammad Amin Zaman, Suzanne Oparil, and David A. Calhoun. “Drugs targeting the renin–angiotensin–aldosterone system”. In: *Nature Reviews Drug Discovery* 1.8 (Aug. 2002), pp. 621–636. DOI: 10.1038/nrd873. URL: <https://doi.org/10.1038/nrd873>.
- [15] Andrew M. South et al. “Controversies of renin–angiotensin system inhibition during the COVID-19 pandemic”. In: *Nature Reviews Nephrology* 16.6 (Apr. 2020), pp. 305–307. DOI: 10.1038/s41581-020-0279-4. URL: <https://doi.org/10.1038/s41581-020-0279-4>.
- [16] Pietro Barbiero and Pietro Lió. *The Computational Patient has Diabetes and a COVID*. 2020. DOI: 10.48550/ARXIV.2006.06435. URL: <https://arxiv.org/abs/2006.06435>.