



UNIVERSITÀ DEGLI STUDI DI SALERNO

Dipartimento di Informatica

Corso di Laurea Triennale in Informatica

TESI DI LAUREA

MLOps: Una revisione sistematica della letteratura sulle pratiche e proprietà MLOps

RELATORE

Prof. Fabio Palomba

CANDIDATO

Damiana Buono

Matricola: 0521207221

CORRELATORI

Dott. Giammaria Giordano

Dott. Gilberto Recupito

Anno Accademico 2022-2023

“Non si deve mai avere paura di dire “Non lo so” o “Non capisco” o di fare domande “stupide”, poiché nessuna domanda è stupida. Bisogna continuare anche quando le cose sembrano impossibili, anche quando i cosiddetti esperti dicono che è impossibile; bisogna essere autonomi ed essere diversi; non aver paura di sbagliare o di ammettere i propri errori, perché solo chi fallisce tanto, può avere tanto successo.”

Margaret Heafield Hamilton

Tratto da un'intervista del 2016

Abstract

Lo sviluppo e l'implementazione di applicazioni di machine learning (ML) è un campo molto popolare che viene utilizzato in un numero sempre maggiore di progetti in tutto il mondo. Tuttavia, è molto impegnativo automatizzare e rendere operativi i prodotti ML e quindi non riescono a soddisfare le loro aspettative. Il paradigma di Machine Learning Operations (MLOps) risolve questo problema, il quale richiede l'uso di determinate pratiche e tool. MLOps include diversi aspetti come best practice. Il ciclo di vita dello sviluppo del software per un machine learning è davvero molto complesso e la sua natura sequenziale comporta diverse attività ripetitive per gli sviluppatori. L'automazione in questo aspetto ridurrebbe notevolmente il tempo e lo sforzo manuale richiesto. L'obiettivo di questo lavoro di tesi è di analizzare quali pratiche, tool e proprietà MLOps vengono utilizzate in letteratura dagli autori tramite una revisione sistematica della letteratura.

Indice	ii
Elenco delle figure	iv
1 Introduzione	1
1.1 Motivazioni e Obiettivi	1
1.2 Risultati	6
1.3 Struttura della tesi	6
2 Stato dell'arte	7
2.1 Lavoro correlato	7
3 Metodologia della ricerca	11
3.1 Obiettivi e Domande della ricerca	11
3.2 Query di ricerca	13
3.3 Sorgenti di ricerca	13
3.4 Criteri di esclusione e inclusione	14
3.5 Valutazione della qualità	14
3.6 Estrazione dei dati	15
3.7 Analisi dei dati	18
3.8 Esecuzione del processo di ricerca	20
4 Analisi dei risultati	22
4.1 RQ1. Le proprietà necessarie per supportare i sistemi ML tramite MLOps . .	22

4.2	RQ2. Le pratiche e tool di MLOps adoperate nei sistemi ML	25
4.3	RQ3. Le pratiche o tool che permettono di soddisfare le proprietà MLOps . .	27
5	Conclusioni e Sviluppi Futuri	29
5.1	Sviluppi Futuri	30
	Ringraziamenti	31

Elenco delle figure

1.1	MLOps.	2
1.2	Ciclo di vita DevOps.	4
1.3	Ciclo di vita MLOps.	5
3.1	Questa tabella riporta le informazioni estratte dei primi sette paper	16
3.2	Questa tabella riporta le informazioni estratte dei paper da P8 a P13	17
3.3	Questa tabella riporta le informazioni estratte dei paper da P14 a P18	18
4.1	Grafico delle proprietà MLOps.	23
4.2	Proprietà MLOps.	24
4.3	Grafico delle pratiche/tool MLOps.	25
4.4	Pratiche e Tool MLOps.	26
4.5	Pratiche/Tool che soddisfano le proprietà MLOps	27
4.6	Tabella che riporta per ogni proprietà le pratiche/tool riscontrate.	28

1.1 Motivazioni e Obiettivi

L'Intelligenza Artificiale (AI) è un campo di studio che è in ascesa per sviluppare applicazioni adatte a procurarsi informazioni e fornire modelli ML che siano incorporati in applicazioni che fungono da moduli di rilevamento facciale o riconoscimento vocale. Lo sviluppo e l'implementazione di applicazioni ML diventano più difficili e complesse rispetto alle applicazioni tradizionali [1]. Ad esempio, il ciclo di vita delle applicazioni ML differisce in modo significativo dalle applicazioni tradizionali nei cicli di feedback, il monitoraggio e la valutazione di un modello possono tornare indietro a qualsiasi fase precedente dello sviluppo, come la pre-elaborazione dei dati o l'addestramento del modello, che modificherà il comportamento del modello.

Le infrastrutture e le piattaforme per la gestione del ciclo di vita end-to-end delle applicazioni ML contengono più pipeline di lavoro a causa delle molteplici fasi e artefatti del processo [1]. Le fasi del ciclo di vita generalmente includono diverse parti, come l'elaborazione dei dati, la convalida dei dati, l'estrazione delle funzionalità, la progettazione del modello, l'addestramento del modello, la valutazione del modello, lo sviluppo dell'applicazione, i controlli di qualità, la distribuzione dell'applicazione e la manutenzione dell'applicazione. La natura complessa delle applicazioni ML necessitano di un supporto per le diverse attività che compongono le pipeline. Tra le attività più comuni è necessario un sistema di controllo della versione dei dati (DVC), il quale favorisce la tracciabilità del sistema andando a definire

versioni differenti per ogni insieme di dati utilizzato per il training di un modello. Quindi la soluzione alle sfide dei modelli di Machine Learning (ML) avviene tramite l'utilizzo delle pratiche e tools MLOps (**Figura 1.1**).

MLOps è DevOps per l'apprendimento automatico. Consente agli sviluppatori di collaborare e aumentare il ritmo con cui i modelli di intelligenza artificiale possono essere sviluppati, implementati, ridimensionati, monitorati e riaddestrati.

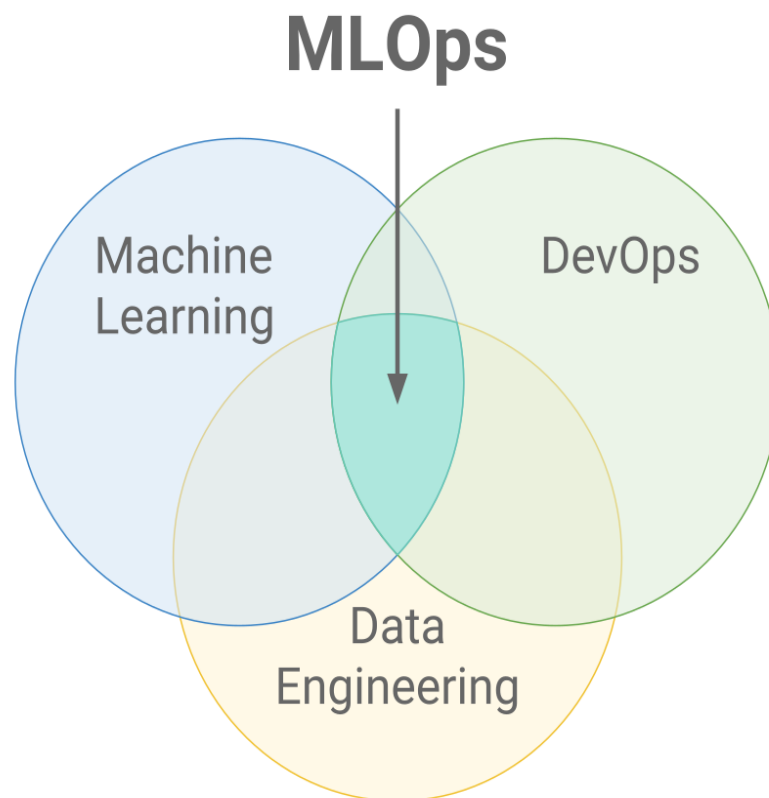


Figura 1.1: MLOps.

DevOps (**Figura 1.2**)¹ o "Development and Operations" è uno sforzo organizzativo collaborativo e multidisciplinare per automatizzare la consegna continua di nuovi aggiornamenti software garantendone la correttezza e l'affidabilità². Ci sono principalmente cinque concetti in DevOps [2]:

¹<https://blog.osservatori.net/it/t/devops-come-funziona-fondamentale-per-direzione-it>

²Link del documento: <https://dl.acm.org/doi/abs/10.1145/3359981>

- **Continuous monitoring:** consentono di automatizzare i check di device e servizi e per notificare i malfunzionamenti al reparto tecnico, permettendo agli admin di essere più efficienti. A differenza del metodo tradizionale, con il Continuous Monitoring:
 - Le attività di monitoraggio sono estremamente focalizzate su indicatori specifici;
 - Tutti i controlli e le transizioni vengono verificate in tempo reale e in maniera continua;
 - Anche i fallimenti dei controlli vengono identificati velocemente, assicurando di minimizzare i rischi che ne conseguono;
 - È possibile allineare le esigenze del business con le attività svolte, adattandole ai nuovi obiettivi e concentrandosi sui controlli più importanti.
- **Continuous Delivery e/o Continuous Deployment (CD):** La Continuous delivery (CD) si pone l'obiettivo di accelerare il rilascio del software. Consente infatti di costruire artefatti eseguibili e production-ready (build) a partire dalla base di codice comune, validata attraverso i processi di CI. Il Continuous deployment è l'estremizzazione e il passo finale del processo di distribuzione continua. Permette infatti di distribuire nell'ambiente di produzione ogni singola modifica al codice che abbia superato i test.
- **Continuous operation:** Il concetto di Continuous operations afferisce alla disponibilità delle applicazioni per l'utente finale e alla continuità di servizio. Qualsiasi modifica venga apportata al software o all'infrastruttura deve essere trasparente per l'utilizzatore. L'obiettivo è evitare un'interruzione dell'operatività.
- **Continuous Integration(CI):** Essa è la parte del processo dedicata principalmente agli sviluppatori del codice e si occupa di effettuare l'integrazione, il test unitario e l'unificazione del codice all'interno di una base di codice condivisa in modo automatico al fine di evidenziare o meno la corretta integrazione del nuovo codice nel repository.
- **Collaboration:** strumenti di gestione del codice che tengono traccia delle modifiche realizzate nel tempo e delle diverse versioni, richiamandole al bisogno e permettendo agli sviluppatori di lavorare in modo competitivo e parallelo.

L'adozione delle pratiche DevOps nello sviluppo del software è diventata un requisito nella maggior parte degli scenari al giorno d'oggi. DevOps si inserisce perfettamente, grazie alla sua natura dinamica basata sull'apprendimento e sui miglioramenti continui. Il ciclo infinito

DevOps mira a ritrattare le divisione dei ruoli dei teams di sviluppatori di applicazioni (Dev) e delle operazioni IT (Ops) in un singolo team [4].

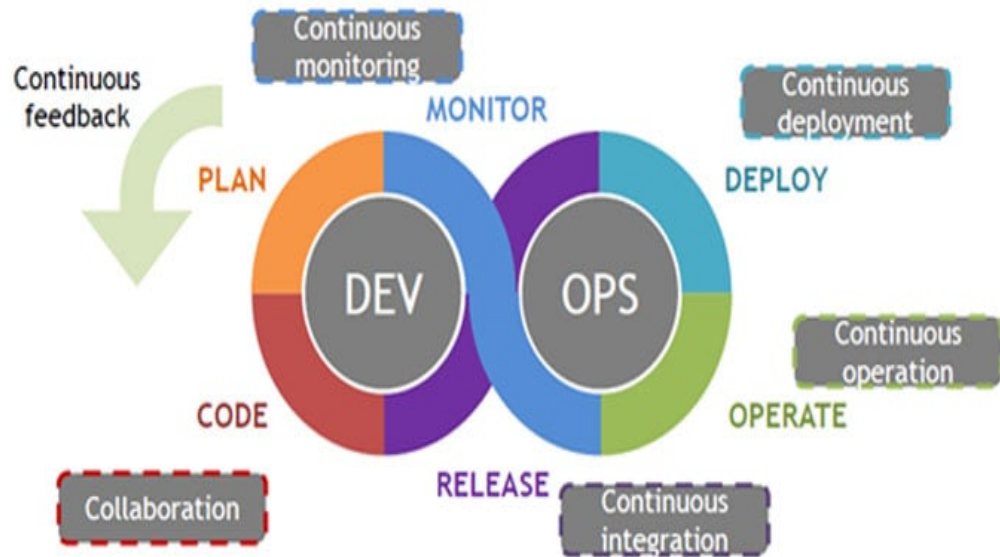


Figura 1.2: Ciclo di vita DevOps.

Quindi, MLOps (**Figura 1.3**) è un insieme di pratiche e mira a supportare tutte le fasi del ciclo di vita della modellazione ML, dall'acquisizione dei dati allo sviluppo e alla convalida di un modello ML, alla messa a disposizione di questi modelli. Diversi tool favoriscono l'impiego di MLOps all'interno dei sistemi ML mira a supportare tutte le fasi del ciclo di vita della modellazione ML, dall'acquisizione dei dati allo sviluppo e alla convalida di un modello ML, alla messa a disposizione di questi modelli. Inoltre, stabilire buone pratiche e tool MLOps aiuta un'organizzazione ad assicurarsi che il modello possa essere continuamente aggiornato e distribuito mentre i dati vengono aggiunti e può consentire la condivisione del modello tra i team assicurandosi che gli artefatti siano continuamente aggiornati e distribuiti[3].

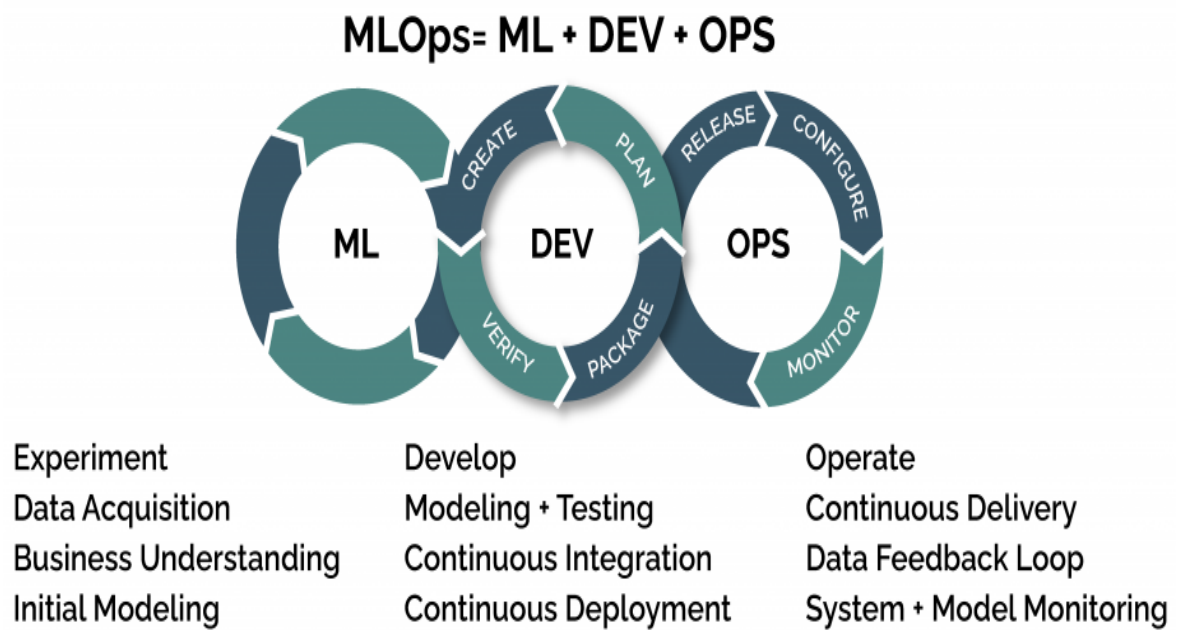


Figura 1.3: Ciclo di vita MLOps.

Esistono piattaforme MLOps come KubeFlow [8], ModelOps[18] e MLFlow[7] [8] ecc. oppure strumenti di orchestrazione come Kubernetes [8], Jenkins[6][8] e Apache AirFlow[8] utilizzano un approccio valido il quale va ad utilizzare i processi esistenti in alcune fasi del ciclo di vita. Inoltre, le piattaforme ML come TensorFlow[8], ModelOps[18] e Kubeflow[8] hanno fornito le soluzioni per la gestione del ciclo di vita end-to-end delle applicazioni ML orchestrandone le fasi in pipeline ML multistep.

Pertanto, in questa tesi, è stata condotta una Systematic Literature Review (SLR) sull'utilizzo di pratiche, tool e proprietà MLOps. Come linea guida è stato utilizzato il documento [5] per cercare sistematicamente in letteratura le pratiche, tool e proprietà MLOps utilizzate dagli autori. Prima, attraverso l'utilizzo di keywords specifiche del contesto è stata trovata la query di ricerca. Sono stati ritrovati inizialmente 1920 articoli. E poi, attraverso il processo di review, sono stati trovati 18 articoli. Infine, sono state utilizzate delle **parole chiavi**:

- **MLOps o MLOperations:** utilizzate nella query di ricerca, per ricercare paper che trattavano argomenti MLOps.
- **best practice:** utilizzate nella query di ricerca, per ricercare documenti che discutevano di pratiche migliori nell'abito MLOps .
- **AI o Artificial Intelligence:** utilizzate nella query di ricerca, per ricercare articoli che analizzano argomenti sull'intelligenza artificiale.

- **SLR o Systematic Literature Review:** utilizzate nella ricerca di articoli correlati a tale documento.

1.2 Risultati

In questa tesi di ricerca si è esteso il lavoro portato avanti dagli esperti di ML fornendo ulteriori informazioni sul MLOps, tramite l'utilizzo di una Systematic Literature Review(SLR). Esattamente sono state ricercate, studiate e riportate:

- Le proprietà necessarie che permettono di supportare i sistemi ML tramite MLOps;
- Le pratiche e tool di MLOps adoperate nei sistemi ML;
- Le pratiche e tool che permettono di soddisfare le proprietà MLOps;

1.3 Struttura della tesi

Le sezioni sono organizzate come segue:

Sezione 2. In questa sezione vengono riportati i lavori correlati a tale documento, nel quale si analizza il significato MLOps, oltre ad elaborare le revisioni della letteratura esistente sul MLOps e su come il nostro lavoro differisce da esse.

Sezione 3. In questa sezione riportiamo le domande di ricerca e la metodologia sistematica impiegate per ricercare e sintetizzare la letteratura.

Sezione 4. In questa sezione si discutono i risultati raggiunti.

Sezione 5. In questa sezione si conclude il documento e si delinea il nostro futuro programma di ricerca

Che cos'è MLOps? In parole povere, MLOps è DevOps per l'apprendimento automatico. Consente agli sviluppatori di collaborare e aumentare il ritmo con cui i modelli di intelligenza artificiale possono essere sviluppati, implementati, ridimensionati, monitorati e riaddestrati[2].

MLOps imita le pratiche DevOps e introduce azioni aggiuntive specifiche per ML. Ciò sembra naturale, poiché in molti casi le funzionalità ML sono incorporate in un sistema software più ampio che ospita, fornisce accesso e monitora le funzionalità ML. Infatti, spesso il modello, che può essere il cuore dell'applicazione, è solo una piccola parte dell'intero sistema software, quindi l'interazione tra il modello e il resto del software e del contesto è essenziale. Basandosi sul successo degli approcci di sviluppo continuo del software, in particolare DevOps, è diventato desiderabile distribuire anche componenti di machine learning (ML) in tempo reale[10]. A tal fine, MLOps fa riferimento alla promozione dell'automazione e del monitoraggio in tutte le fasi dello sviluppo e dell'implementazione del sistema ML.

2.1 Lavoro correlato

Questo studio evidenzia l'interesse emergente per MLOps e la crescente adozione di queste pratiche e tool nei sistemi ad alta intensità di software. Questo è un segnale positivo in quanto indica che sempre più aziende si stanno orientando all'utilizzo di MLOps per ottenere piattaforme completamente automatizzate. Le SLR sottolineano il fatto che i team

interfunzionali con competenze di ingegneria dei dati, scienza dei dati o operazioni possono facilitare MLOps. Sulla base degli approfondimenti presenti in letteratura [10], [15] e [16], si nota che il registro dei modelli e i cicli di retroazione possono abbreviare la transizione dei modelli dal prototipo alla fase di produzione. Di conseguenza, promuovono l'automazione, il controllo delle versioni, la spiegabilità e la tracciabilità. Quindi, gli sviluppi del sistema di apprendimento automatico sono caratterizzati da alcune sfide che devono essere affrontate, al fine di ottenere continui miglioramenti; una delle sfide più critiche è la raccolta dei dati, l'estrazione dei dati e la pulizia dei dati. Negli ultimi anni sono stati condotti degli studi sull'automatizzazioni del ciclo di vita delle pipeline ML tramite una revisione sistema della letteratura. La maggior parte di loro ([10], [15] e [16]) ha affrontato il problema di riuscire ad automatizzare il ciclo di vita delle pipeline ML utilizzando delle piattaforme ML, pratiche DevOps con l'automatizzazione delle pratiche CI/CD e la definizione delle proprietà MLOps.

Kreuzberger et al.[5] hanno condotto una ricerca con metodo misto, tra cui una revisione della letteratura, una revisione degli strumenti e interviste agli esperti. Come risultato di queste indagini, forniscono una panoramica aggregata delle proprietà, dei componenti e dei ruoli necessari, nonché dell'architettura e dei flussi di lavoro associati. Da tale documento sono state ricavate le proprietà MLOps. Tali proprietà sono una guida su come le cose dovrebbero essere realizzate in MLOps e sono strettamente correlate al termine "best practices". Discutono delle proprietà MLOps e di come vengono applicate nelle varie piattaforme ML. Le proprietà ricavate sono:

- **Automazione CI/CD.** L'automazione CI/CD fornisce integrazione continua, distribuzione continua e distribuzione continua. Esegue le fasi di compilazione, test, consegna e distribuzione. Fornisce un rapido feedback agli sviluppatori in merito al successo o al fallimento di determinati passaggi, aumentando così la produttività complessiva.
- **Orchestrazione del flusso di lavoro.** L'orchestrazione del flusso di lavoro coordina le attività di una pipeline del flusso di lavoro ML in base a grafici aciclici diretti (DAG). I DAG definiscono l'ordine di esecuzione delle attività considerando relazioni e dipendenze.
- **Riproducibilità.** La riproducibilità è la capacità di riprodurre un esperimento ML e ottenere esattamente gli stessi risultati
- **Versione.** Il versioning garantisce il versioning di dati, modello e codice per consentire non solo la riproducibilità, ma anche la tracciabilità.

- **Collaborazione.** La collaborazione garantisce la possibilità di lavorare in modo collaborativo su dati, modello e codice. Oltre all'aspetto tecnico, questo principio sottolinea una cultura del lavoro collaborativa e comunicativa che mira a ridurre i silos di dominio tra ruoli diversi.
- **Formazione e valutazione continue sul machine learning.** Formazione continua significa riqualificazione periodica del modello ML sulla base di nuovi dati sulle funzionalità. La formazione continua è abilitata attraverso il supporto di un componente di monitoraggio, un ciclo di feedback e una pipeline del flusso di lavoro ML automatizzata. La formazione continua include sempre una corsa di valutazione per valutare il cambiamento nella qualità del modello.
- **Tracciamento/registrazione dei metadati ML.** I metadati vengono tracciati e registrati per ogni attività del flusso di lavoro ML orchestrata. Il tracciamento e la registrazione dei metadati sono necessari per ogni iterazione del processo di addestramento (ad es. data e ora dell'addestramento, durata e così via), inclusi i metadati specifici del modello, ad es. i parametri utilizzati e le metriche delle prestazioni risultanti, derivazione del modello: dati e codice utilizzati, per garantire la piena tracciabilità delle esecuzioni sperimentali.
- **Monitoraggio continuo.** Il monitoraggio continuo implica la valutazione periodica dei dati, del modello, del codice, delle risorse dell'infrastruttura e delle prestazioni del servizio del modello.
- **Cicli di retroazione.** Cicli di feedback multipli sono necessari per integrare gli approfondimenti dalla fase di valutazione della qualità nel processo di sviluppo o ingegnerizzazione (ad esempio, un ciclo di feedback dalla fase di progettazione del modello sperimentale alla precedente fase di progettazione delle caratteristiche). Un altro ciclo di feedback è richiesto dal componente di monitoraggio (ad esempio, osservando le prestazioni del servizio del modello) allo scheduler per consentire il riaddestramento.

Quindi, rispetto al lavoro di Kreuzberger, Kühl e Hirschl [5], tale documento può essere visto come un'analisi sistematica complementare in cui ci concentriamo sulle proprietà MLOps andando a stimare il loro utilizzo in letteratura, così da fornire una percentuale che indica quando una determinata proprietà viene utilizzata in letteratura.

Hummer et al.[18] ha proposto un quadro e una piattaforma basati su cloud per lo sviluppo end-to-end e la gestione del ciclo di vita delle applicazioni di intelligenza artificiale

(AI). Discutono delle sfide della creazione di operazioni AI scalabili e dell'implementazione di un prototipo ModelOps attorno ai concetti di versioning dei metadati, astrazioni del dominio AI, modelli riutilizzabili, pipeline basate su eventi e perfetta integrazione delle funzionalità del ciclo di vita. Il collegamento che l'articolo [18] ha con tale documento è l'applicazione delle proprietà MLOps (Kreuzberger et al. [5]) nel prototipo ModelOps.

Karamitsos et al. [19] propongono, tramite una revisione sistematica della letteratura (SLR), le pratiche DevOps per l'applicazione di machine learning, integrando perfettamente sia l'ambiente di sviluppo che quello operativo e presentano come applicare principi, pratiche e strumenti di integrazione continua (CI) e consegna continua (CD) in modo da ridurre al minimo gli sprechi e supportare cicli di feedback rapidi. Karamitsos et al. rispondono alle loro domande di ricerca proponendo un nuovo approccio per introdurre i due principi DevOps nell'approccio MLOps. L'idea principale è progettare una pipeline automatizzata di apprendimento automatico utilizzando due principi DevOps: l'integrazione continua (CI) e la consegna continua (CD). Praticare MLOps significa che sosteniamo l'automazione e il monitoraggio in tutte le fasi della costruzione del sistema ML.

Ciò che differenzia questo documento con il nostro è dato dalla ricerca delle pratiche e tool in generale, cioè il nostro studio non si basa solo sul DevOps ma anche su altre pratiche e tool come il ModelOps, MLFlow, TensorFlow ecc.

Quindi, sulla base dell'analisi dello stato dell'arte, possiamo affermare che il nostro lavoro presenta un'indagine sistematica aggiornata che si focalizza su due aspetti: (i) ricerca delle pratiche e tool MLOps utilizzate in letteratura; (ii) ricerca delle proprietà MLOps adoperate in letteratura. Grazie a tali ricerche possiamo fornire una percentuale di utilizzo per ogni pratica e proprietà MLOps trovate in letteratura.

Metodologia della ricerca

L'obiettivo dello studio è esaminare la letteratura di ricerca che ha applicato metodi di intelligenza artificiale per rilevare le pratiche e tool utilizzate per la gestione del ciclo di vita di una pipeline ML, tramite l'utilizzo di MLOps, con lo scopo di fornire ai professionisti di ingegneria del software elementi e approfondimenti utilizzabili che possono sfruttare per automatizzare le fasi che coinvolgono lo sviluppo di applicazioni ML. Per raggiungere tale scopo, è stata condotta una Systematic Literature Review (SLR), che è un processo di sintesi attraverso il quale i documenti di ricerca esistenti su un argomento di interesse vengono sistematicamente identificati, selezionati e analizzati dalla critica per analizzare l'insieme dei lavori di ricerca che hanno contribuito alla definizione e all'evoluzione del campo d'interesse.

3.1 Obiettivi e Domande della ricerca

L'obiettivo specifico di ricerca della revisione sistematica della letteratura è riportato di seguito:

Obiettivo: Come è supportata la gestione delle pipeline ML tramite MLOps?

Tali obiettivi hanno guidato la definizione delle domande di ricerca:

RQ1. Quali sono le proprietà necessarie che permettono di supportare i sistemi ML tramite MLOps?

Questa domanda mira ad affrontare quali sono le proprietà utilizzate per la gestione di una pipeline ML. La domanda di ricerca è motivata dalla volontà di fornire nel contesto di MLOps una proprietà, che sarebbe una guida su come le cose dovrebbero essere realizzate in MLOps ed è strettamente correlato al termine “best practices”. Attraverso un’analisi approfondita dei documenti sono state trovate nove proprietà ML: Automatizzazione CI/CD, Formazione Continua/Valutazione Continua sul ML, Collaborazione, Monitoraggio Continuo, Cicli di retroazione, Tracciabilità/Registrazione dei metadati ML, Riproducibilità, Versione e Orchestrazione del flusso di lavoro.

RQ2. Quali sono le pratiche o tool di MLOps adoperate nei sistemi ML?

Questa domanda mira ad affrontare la ricerca delle pratiche e tool MLOps utilizzate nei sistemi ML. La domanda di ricerca è motivata dalla volontà di fornire le pratiche e tool MLOps adottate nei sistemi ML, come ad esempio l’utilizzo delle pratiche del DevOps (CD/CI). Dalla ricerca, ottenuta tramite l’analisi dei vari articoli, sono state trovate dodici pratiche e tool che sono: Continuous Integration (CI), Continuous Deployment (CD), Continuous Training (CT), TensorFlow, MLFlow, Integration Patterns, CD4ML, ModelOps, Kubernetes, Apache AirFlow, KubeFlow e Jenkins.

RQ3. Quali pratiche o tool permettono di soddisfare le proprietà MLOps?

Questa domanda mira ad affrontare le pratiche e tool che soddisfano le proprietà MLOps. La domanda di ricerca è motivata dalla volontà di fornire le pratiche citate nella RQ2 che soddisfano le proprietà MLOps (RQ1). In questo caso si è ricercato per ogni pratica o tool trovata quale delle nove proprietà permettevano di soddisfarle.

3.2 Query di ricerca

Uno dei passaggi metodologici chiave di una revisione sistematica della letteratura è l'identificazione di termini di ricerca appropriati che possono aiutare a recuperare un insieme completo di fonti. A questo proposito, è stata adottata la seguente strategia:

- Per ogni domanda di ricerca sono state estratte le parole chiave più rilevanti;
- Per tutti i termini rilevanti sono stati identificati possibili sinonimi o ortografia alternativa con *;
- Sono stati utilizzati operatori booleani (\wedge e \vee) per comporre il file domanda di ricerca.

Il risultato è stato la seguente query di ricerca:

**(MLOps \vee MLOperation*) \wedge (Artificial Intelligence \vee AI) \wedge (pipeline* \vee stage*) \wedge
(practice* \vee "best practice")**

Come mostrato, sono stati messi in OR (\vee) tutti i sinonimi dello stesso concetto, mentre più concetti sono stati combinati usando l'operatore AND (\wedge). Prima di continuare la revisione sistematica della letteratura, si è verificata la presenza dei termini della query di ricerca relativi ai problemi delle pratiche e tool di MLOps nelle revisioni sistematiche della letteratura. L'idea alla base di questo passaggio era verificare la coerenza e la completezza dei termini selezionati rispetto a documenti che riportavano un'indagine sistematica della letteratura. Pertanto, dovrebbero contenere una mappatura completa dei termini utilizzati in letteratura per indicare le pratiche e tool di MLOps. Questo passaggio ci ha permesso di includere parole alternative nella query di ricerca nel caso in cui inizialmente non fossero state incluse. Questo step finale ha convalidato tutte le keyword che sono state individuate inizialmente e non è stato necessario effettuare modifiche alla query di ricerca.

3.3 Sorgenti di ricerca

Una volta definita la query di ricerca, sono stati selezionati i database da utilizzare durante l'esecuzione della ricerca. La corretta identificazione di tali database è fondamentale per avere una revisione della letteratura di successo. Per questo motivo, sono stati selezionati i primi tre database di ricerca, vale a dire:

- IEEEExplore (<http://ieeexplore.ieee.org>);

- Google Scholar (<https://scholar.google.com>);
- ResearchGate (<https://www.researchgate.net>)

Questi database sono utilizzati per condurre revisioni sistematiche della letteratura e garantiscono una copertura completa della ricerca pubblicata, consentendoci quindi di accedere all'intero set di articoli .

3.4 Criteri di esclusione e inclusione

I criteri di esclusione e inclusione consentono la selezione di risorse che affrontano le domande di ricerca di una revisione sistematica della letteratura. Nel contesto di questo lavoro di tesi, sono stati identificati e applicati i seguenti criteri di "Inclusione/ Esclusione".

1. **Criteri di esclusione.**Le risorse che soddisfacevano i seguenti vincoli sono state escluse dallo studio:

- Articoli non scritti in inglese;
- Documenti duplicati;
- Papers il cui testo non era disponibile;
- Scarsa comprensibilità;
- Fonte non specificata;
- Non coerente con gli argomenti trattati.

Usando questi filtri, potremmo escludere tutti i risultati preliminari della ricerca, ad esempio, evitare di considerare un articolo simile più volte.

2. **Criteri di inclusione.**Nello studio sono stati inclusi documenti che:

- esplicitano i principi necessari ad applicare MLOps;
- espongono le pratiche o tool MLOps adottate nei sistemi MLOps;
- trattano di pratiche o tool MLOps che soddisfano le proprietà MLOps;

3.5 Valutazione della qualità

Prima di procedere con l'estrazione dei dati necessari per affrontare le domande di ricerca, sono state valutate la qualità e la completezza delle risorse recuperate per scartare i documenti

che non hanno fornito dettagli sufficienti per essere utilizzati nello studio. In particolare, è stata definita una checklist che comprende le seguenti domande:

- D1. Le proprietà che permettono di supportare i sistemi ML tramite MLOps sono chiaramente definite?
- D2. Le pratiche o tool di MLOps adoperate nei sistemi ML trattati nei documenti sono chiaramente definite?
- D3. Le fasi del ciclo di vita di una pipeline di MLOps sono chiaramente definite?
- D4. Le pratiche o tool di MLOps citate negli articoli sono validate?

Ad ogni domanda si può rispondere come "Sì", "Parzialmente", "No". Successivamente, è stato associato un valore numerico per ogni etichetta per valutare meglio la qualità e completezza di ciascuna fonte: l'etichetta "Sì" è stata associata al valore '1', "Parzialmente" a '0.5', "No" a '0'. Il punteggio di qualità complessivo è stato calcolato sommando il punteggio delle risposte alle due domande e sono stati accettati gli articoli con un punteggio di qualità di almeno 1.

3.6 Estrazione dei dati

Una volta identificata la serie finale di fonti da considerare, sono state estratte le informazioni rilevanti per rispondere alle domande di ricerca. In particolare, è stata definita una tabella divisa in tre figure, (**Figura 3.1**), (**Figura 3.2**) e (**Figura 3.3**), nelle quali per ogni documento, preso in considerazione, si è riposto alle domande di ricerca e alle domande di valutazione della qualità.

ID Paper	RQ1	RQ2	RQ3	D1	D2	D3	D4
P1	<ul style="list-style-type: none"> • Automazione CI/CD • Orchestrazione del flusso di lavoro • Riproducibilità • Versione • Formazione e valutazione continue • Cicli di retroazione 	<ul style="list-style-type: none"> • Continuous Integration (CI) • Continuous Deployment (CD) • Continuous Training (CT) • TensorFlow • ModelOps • KubeFlow • Kubernetes 	<ul style="list-style-type: none"> • La pratica CI supporta le proprietà: Automazione CI/CD • La pratica CD supporta le proprietà: Automazione CI/CD • La pratica TensorFlow supporta le proprietà: Versione • La pratica CT supporta le proprietà: Formazione continua • La pratica ModelOps supporta le proprietà: Orchestrazione • La pratica Kubernetes supporta le proprietà: Cicli di retroazione • La pratica KubeFlow supporta le proprietà: Riproducibilità 	0.5	1	0.5	1
P2	<ul style="list-style-type: none"> • Automazione CI/CD • Orchestrazione del flusso di lavoro • Collaborazione • Tracciamento/registrazione • Monitoraggio continuo 	<ul style="list-style-type: none"> • Continuous Integration (CI) • Continuous Deployment (CD) • TensorFlow • MLFlow • KubeFlow • Kubernetes 	<ul style="list-style-type: none"> • La pratica CI supporta le proprietà: Automazione CI/CD • La pratica CD supporta le proprietà: Automazione CI/CD e Formattazione continua • La pratica MLFlow supporta le proprietà: Monitoraggio, Tracciabilità, Orchestrazione 	0.5	1	0	0.5
P3	<ul style="list-style-type: none"> • Riproducibilità • Versione • Tracciamento/registrazione • Monitoraggio continuo 	<ul style="list-style-type: none"> • TensorFlow • KubeFlow • Kubernetes • Apache AirFlow 	<ul style="list-style-type: none"> • La pratica TensorFlow supporta le proprietà: Versione • La pratica Kubernetes supporta le proprietà: Monitoraggio • La pratica Apache AirFlow supporta le proprietà: Riproducibilità 	0.5	0.5	0.5	0.5
P4	<ul style="list-style-type: none"> • Automazione CI/CD • Monitoraggio continuo 	<ul style="list-style-type: none"> • Continuous Integration (CI) • Continuous Deployment (CD) 	<ul style="list-style-type: none"> • La pratica CI supporta le proprietà: Automazione CI/CD e Formattazione continua • La pratica CD supporta le proprietà: Automazione CI/CD 	0.5	1	0	0
P5	<ul style="list-style-type: none"> • Automazione CI/CD • Orchestrazione del flusso di lavoro • Riproducibilità • Versione • Collaborazione • Formazione e valutazione continue • Tracciamento/registrazione • Monitoraggio continuo • Cicli di retroazione 	<ul style="list-style-type: none"> • Continuous Integration (CI) • Continuous Deployment (CD) • TensorFlow • MLFlow • KubeFlow • Kubernetes • Apache AirFlow • Jenkins 	<ul style="list-style-type: none"> • La pratica CI supporta le proprietà: Automazione CI/CD • La pratica CD supporta le proprietà: Automazione CI/CD • La pratica TensorFlow supporta le proprietà: Versione • La pratica MLFlow supporta le proprietà: Monitoraggio, Tracciabilità • La pratica Kubernetes supporta le proprietà: Cicli di retroazione • La pratica Jenkins supporta le proprietà: Collaborazione • La pratica Apache AirFlow supporta le proprietà: Formazione continua, Riproducibilità, Orchestrazione 	1	0.5	0.5	1
P6	<ul style="list-style-type: none"> • Automazione CI/CD • Orchestrazione del flusso di lavoro • Versione • Collaborazione 	<ul style="list-style-type: none"> • Continuous Integration (CI) • Continuous Deployment (CD) • ModelOps • KubeFlow • Kubernetes • Jenkins 	<ul style="list-style-type: none"> • La pratica CI supporta le proprietà: Automazione CI/CD • La pratica CD supporta le proprietà: Automazione CI/CD • La pratica ModelOps supporta le proprietà: Collaborazione, Orchestrazione • La pratica KubeFlow supporta le proprietà: Versione • La pratica Jenkins supporta le proprietà: Formazione continua 	0.5	1	0.5	1
P7	<ul style="list-style-type: none"> • Orchestrazione del flusso di lavoro • Riproducibilità • Versione • Collaborazione • Tracciamento/registrazione 	<ul style="list-style-type: none"> • TensorFlow • MLFlow 	<ul style="list-style-type: none"> • La pratica TensorFlow supporta le proprietà: Versione • La pratica MLFlow supporta le proprietà: Tracciabilità, Riproducibilità, Orchestrazione 	0.5	0.5	1	0.5

Figura 3.1: Questa tabella riporta le informazioni estratte dei primi sette paper

P8	<ul style="list-style-type: none"> Orchestrazione del flusso di lavoro Riproducibilità Versione Collaborazione Formazione e valutazione continue Tracciamento/registrazione 	<ul style="list-style-type: none"> Continuous Integration (CI) Continuous Deployment (CD) Continuous Training (CT) TensorFlow MLFlow ModelOps KubeFlow Kubernetes Apache AirFlow Jenkins 	<ul style="list-style-type: none"> La pratica TensorFlow supporta le proprietà: Versione La pratica CT supporta le proprietà: Formazione continua La pratica ModelOps supporta le proprietà: Collaborazione, Orchestrazione La pratica MLFlow supporta le proprietà: Monitoraggio, Tracciabilità, Riproducibilità La pratica KubeFlow supporta le proprietà: Versione La pratica Jenkins supporta le proprietà: Automatizzazione CI/CD 	1	0.5	0.5	0.5
P9	<ul style="list-style-type: none"> Formazione e valutazione continue Tracciamento/registrazione 	<ul style="list-style-type: none"> KubeFlow Apache AirFlow 	<ul style="list-style-type: none"> La pratica KubeFlow supporta le proprietà: Formazione continua 	0.5	0	0	0.5
P10	<ul style="list-style-type: none"> Versione Collaborazione Formazione e valutazione continue Monitoraggio continuo Cicli di retroazione 	<ul style="list-style-type: none"> Continuous Deployment (CD) 	<ul style="list-style-type: none"> La pratica CD supporta le proprietà: Formazione continua 	0.5	1	0.5	1
P11	<ul style="list-style-type: none"> Orchestrazione del flusso di lavoro Riproducibilità Collaborazione Tracciamento/registrazione Monitoraggio continuo Cicli di retroazione 	<ul style="list-style-type: none"> Continuous Integration (CI) Continuous Deployment (CD) TensorFlow KubeFlow Jenkins 	<ul style="list-style-type: none"> La pratica ModelOps supporta le proprietà: Collaborazione, Monitoraggio, Orchestrazione La pratica KubeFlow supporta le proprietà: Cicli di retroazione 	1	0.5	0.5	1
P12	<ul style="list-style-type: none"> Automazione CI/CD Versione Tracciamento/registrazione Monitoraggio continuo 	<ul style="list-style-type: none"> Continuous Integration (CI) Continuous Deployment (CD) Continuous Training (CT) TensorFlow MLFlow ModelOps KubeFlow Kubernetes 	<ul style="list-style-type: none"> La pratica CI supporta le proprietà: Automazione CI/CD e Formazione continua La pratica CD supporta le proprietà: Automazione CI/CD La pratica TensorFlow supporta le proprietà: Versione La pratica MLFlow supporta le proprietà: Tracciabilità La pratica KubeFlow supporta le proprietà: Monitoraggio 	0.5	1	0.5	0.5
P13	<ul style="list-style-type: none"> Automazione CI/CD Orchestrazione del flusso di lavoro Riproducibilità Versione Collaborazione Tracciamento/registrazione Monitoraggio continuo 	<ul style="list-style-type: none"> Continuous Integration (CI) Continuous Deployment (CD) Continuous Training (CT) 	<ul style="list-style-type: none"> La pratica CI supporta le proprietà: Automazione CI/CD La pratica CD supporta le proprietà: Automazione CI/CD 	1	1	0.5	0.5

Figura 3.2: Questa tabella riporta le informazioni estratte dei paper da P8 a P13

P14	<ul style="list-style-type: none"> • Automazione CI/CD • Orchestrazione del flusso di lavoro • Versione • Collaborazione • Formazione e valutazione continue • Tracciamento/registrazione • Monitoraggio continuo • Cicli di retroazione 	<ul style="list-style-type: none"> • Continuous Integration (CI) • Continuous Deployment (CD) • Continuous Training (CT) • Kubernetes 	<ul style="list-style-type: none"> • La pratica CI supporta le proprietà: Automazione CI/CD • La pratica CD supporta le proprietà: Automazione CI/CD • La pratica CT supporta le proprietà: Formazione Continua • La pratica Kubernetes supporta le proprietà: Monitoraggio, Cicli di retroazione, Orchestrazione 	1	1	1	1
P15	<ul style="list-style-type: none"> • Automazione CI/CD • Orchestrazione del flusso di lavoro • Riproducibilità • Versione • Collaborazione • Formazione e valutazione continue • Tracciamento/registrazione • Monitoraggio continuo • Cicli di retroazione 	<ul style="list-style-type: none"> • Integration Patterns • CD4ML 	<ul style="list-style-type: none"> • La pratica CI supporta le proprietà: Automazione CI/CD e Formazione continua • La pratica CD supporta le proprietà: Automazione CI/CD • La pratica MLFlow supporta le proprietà: Monitoraggio, Tracciabilità, Riproducibilità, Orchestrazione • La pratica Kubernetes supporta le proprietà: Cicli di retroazione • La pratica Jenkins supporta le proprietà: Collaborazione 	1	0.5	0.5	0.5
P16	<ul style="list-style-type: none"> • Versione • Collaborazione • Formazione e valutazione continue • Monitoraggio continuo • Cicli di retroazione 	<ul style="list-style-type: none"> • Continuous Integration (CI) • Continuous Deployment (CD) • Continuous Training (CT) • TensorFlow • MLFlow • ModelOps • KubeFlow • Kubernetes • Apache AirFlow • Jenkins 	<ul style="list-style-type: none"> • La pratica CD4ML supporta le proprietà: Formazione continua • La pratica Integration Patterns supporta le proprietà: Automatizzazione CI/CD, Riproducibilità 	0.5	1	0.5	0.5
P17	<ul style="list-style-type: none"> • Automazione CI/CD • Orchestrazione del flusso di lavoro • Versione • Collaborazione • Formazione e valutazione continue • Tracciamento/registrazione • Monitoraggio continuo 	<ul style="list-style-type: none"> • Continuous Integration (CI) • Continuous Deployment (CD) • Continuous Training (CT) • TensorFlow • Integration Patterns • CD4ML • MLFlow • ModelOps • KubeFlow • Kubernetes • Apache AirFlow • Jenkins 	<ul style="list-style-type: none"> • La pratica CI supporta le proprietà: Automazione CI/CD • La pratica CD supporta le proprietà: Automazione CI/CD • La pratica TensorFlow supporta le proprietà: Versione • La pratica CT supporta le proprietà: Formazione continua • La pratica ModelOps supporta le proprietà: Collaborazione • La pratica MLFlow supporta le proprietà: Tracciabilità • La pratica Kubernetes supporta le proprietà: Monitoraggio • La pratica Jenkins supporta le proprietà: Orchestrazione 	0.5	0.5	1	1
P18	<ul style="list-style-type: none"> • Orchestrazione del flusso di lavoro • Riproducibilità • Versione • Collaborazione • Monitoraggio continuo • Cicli di retroazione 	<ul style="list-style-type: none"> • Continuous Integration (CI) • Continuous Deployment (CD) • TensorFlow • ModelOps • KubeFlow • Kubernetes • Jenkins 	<ul style="list-style-type: none"> • La pratica TensorFlow supporta le proprietà: Versione • La pratica ModelOps supporta le proprietà: Collaborazione, Monitoraggio • La pratica Kubernetes supporta le proprietà: Cicli di retroazione • La pratica KubeFlow supporta le proprietà: Orchestrazione 	0.5	0	0.5	1

Figura 3.3: Questa tabella riporta le informazioni estratte dei paper da P14 a P18

Successivamente, sono stati generati dei grafici per ogni RQ riportati nel capitolo **Analisi Dei Risultati**.

3.7 Analisi dei dati

In questa sezione viene chiarito come i dati, una volta estratti, sono stati sintetizzati, combinati e analizzati per rispondere alle tre domande di ricerca(RQ).

- **Sintesi dei dati.**

I dati sono stati sintetizzati prendendo in considerazione solo la parte dei paper che soddisfano l'esigenze di ricerca. Quindi, sono stati considerati per ogni documento:

- Le pratiche o tool MLOps e le eventuali spiegazioni delle piattaforme considerate.
- Le proprietà MLOps e se tali proprietà sono soddisfatte dalle pratiche e tool considerate precedentemente.

- **Combinazione dei dati.**

I dati sono stati combinati andando ad inserire quest'ultimi in una tabella (si divide in tre parti (**Figura 3.1**), (**Figura 3.2**) e (**Figura 3.3**)), nella quale sono riportate le domande di ricerca (**RQ1**, **RQ2** e **RQ3**).

- **Analisi dei dati.**

I dati, infine, sono stati analizzati per rispondere alle domande di ricerca (**RQ1**, **RQ2** e **RQ3**):

- Per rispondere alla domanda di ricerca (**RQ1**) sono stati analizzati i dati che si riferiscono alle proprietà MLOps. Le proprietà trovate sono:

1. **Automazione CI/CD ;**
2. **Orchestrazione del flusso di lavoro;**
3. **Riproducibilità;**
4. **Versione;**
5. **Collaborazione;**
6. **Formazione e valutazione continue sul machine learning;**
7. **Tracciamento/registrazione dei metadati ML;**
8. **Monitoraggio continuo;**
9. **Cicli di retroazione.**

- Per rispondere alla domanda di ricerca (**RQ2**) sono stati analizzati i dati che si riferiscono alle pratiche o tool MLOps. Le pratiche o tool trovate sono:

1. **Continuous Integration (CI);**
2. **Continuous Deployment (CD);**
3. **Continuous Training (CT);**
4. **TensorFlow;**
5. **Integration Patterns ;**
6. **CD4ML;**
7. **MLFlow;**
8. **ModelOps;**
9. **KubeFlow;**
10. **Kubernetes;**
11. **Apache AirFlow;**

12. Jenkins;

- Per rispondere alla domanda di ricerca (**RQ3**) sono state analizzate le pratiche e tool MLOps che soddisfano le proprietà MLOps. Possiamo fare riferimento alla tabella (**Figura 4.6**), che mostra quale proprietà viene soddisfatta dalle pratiche o tool MLOps.

3.8 Esecuzione del processo di ricerca

Una volta definiti i blocchi fondamentali della revisione sistematica della letteratura, si è proceduto alla sua esecuzione. In particolare, il processo di esecuzione ha funzionato come segue:

- A. Eseguiamo la query di ricerca sui tre database selezionati. La query di ricerca ha prodotto un totale di 1.920 risultati utilizzando Google Scholar come motore di ricerca principale e in un secondo momento ai database di articoli accademici come IEEEExplore e ResearchGate. Il primo passaggio è stato completato scaricando tutti i documenti candidati e archiviandoli in un ambiente locale per un'indagine più rapida.
- B. Ciascuno dei documenti candidati è passato alla fase successiva, che consisteva nell'applicazione dei criteri di esclusione. In questa fase sono stati scansionati tutti i paper e sono stati applicati i filtri. Per prima cosa si è considerato il titolo, l'abstract e le parole chiave di ciascun articolo per decidere se scartarli. Complessivamente sono state escluse 1.762 documenti e, quindi, 158 sono passate alla fase successiva.
- C. Sono stati considerati i criteri di inclusione, i quali sono stati applicati ai 158 documenti restati. A differenza del passaggio precedente, l'inclusione è stata valutata considerando l'intero articolo e non solo il titolo, l'abstract e le parole chiave. Come risultato di questa procedura, sono stati scartati 140 fonti, portando a un numero finale di 18 documenti inclusi nella revisione sistematica.
- D. La fase successiva riguarda l'applicazione della valutazione della qualità. Questa è stata una fase particolarmente critica, poiché sono stati valutati gli articoli in base alla loro chiarezza o alla disponibilità di informazioni sufficienti per rispondere alle nostre domande di ricerca.
- E. Infine, si è proceduto con l'estrazione dei dati. La maggior parte delle informazioni richieste per rispondere alle domande di ricerca era piuttosto facile da raccogliere.

I dati relativi all'esecuzione dell'analisi della letteratura sono disponibili e accessibili online².

²<https://github.com/DamianaBuono/Tesi-MLOps-.git>

L'obiettivo di tale studio è quello di rispondere alle domande di ricerca RQ tramite l'utilizzo dei documenti considerati dalla capitolo precedente, nel quale sono state identificate le proprietà e le pratiche o tool di MLOps che gli autori hanno applicato al loro problema.

4.1 RQ1. Le proprietà necessarie per supportare i sistemi ML tramite MLOps

In primo luogo, sono state identificate le proprietà di MLOps. Tramite un'analisi approfondita dei vari documenti, identificando nove proprietà. Mediante quest'analisi è stato generato un grafico **Figura 4.1**, il quale riporta tutte le proprietà trovate e la presenza di utilizzo di una determinata proprietà in ogni documento attraverso una percentuale.

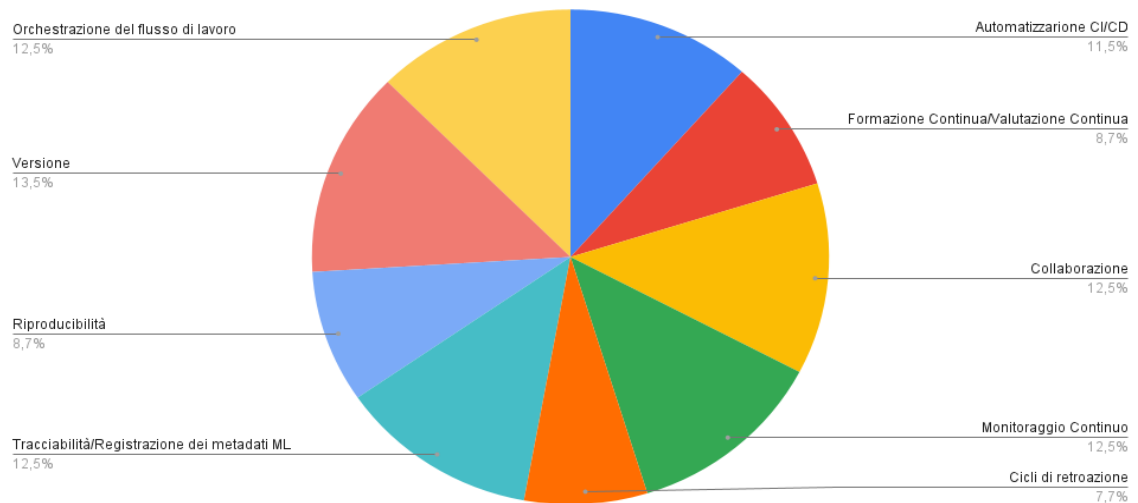


Figura 4.1: Grafico delle proprietà MLOps.

Si nota che la proprietà che viene citata in più fonti è **Versione** con il 13.5%, poi abbiamo con il 12.5% **Collaborazione**, **Monitoraggio continuo**, **Tracciabilità** e **Orchestrare del flusso di lavoro**, **Automatizzazione CI/CD** con l'11.5%, con 8,7% abbiamo **Riproducibilità** e **Formazione/Valutazione Continua**, mentre la proprietà che viene meno menzionata è **Cicli di Retrazione** con il 7.7%. La tabella (Figura 4.2) la quale riporta tutte le proprietà MLOps trovate e la loro definizione.

Proprietà MLOps	Definizione
Automazione CI/CD	L'automazione CI/CD fornisce integrazione continua, distribuzione continua e distribuzione continua. Esegue le fasi di compilazione, test, consegna e distribuzione. Fornisce un rapido feedback agli sviluppatori in merito al successo o al fallimento di determinati passaggi, aumentando così la produttività complessiva.
Orchestrazione del flusso di lavoro	L'orchestrazione del flusso di lavoro coordina le attività di una pipeline del flusso di lavoro ML in base a grafici aciclici diretti (DAG). I DAG definiscono l'ordine di esecuzione delle attività considerando relazioni e dipendenze.
Riproducibilità	La riproducibilità è la capacità di riprodurre un esperimento ML e ottenere esattamente gli stessi risultati
Versione	Il versioning garantisce il versioning di dati, modello e codice per consentire non solo la riproducibilità, ma anche la tracciabilità.
Collaborazione	La collaborazione garantisce la possibilità di lavorare in modo collaborativo su dati, modello e codice. Oltre all'aspetto tecnico, questo principio sottolinea una cultura del lavoro collaborativa e comunicativa che mira a ridurre i silos di dominio tra ruoli diversi.
Formazione e valutazione continue sul machine learning	Formazione continua significa riqualificazione periodica del modello ML sulla base di nuovi dati sulle funzionalità. La formazione continua è abilitata attraverso il supporto di un componente di monitoraggio, un ciclo di feedback e una pipeline del flusso di lavoro ML automatizzata. La formazione continua include sempre una corsa di valutazione per valutare il cambiamento nella qualità del modello.
Tracciamento/registrazione dei metadati ML	I metadati vengono tracciati e registrati per ogni attività del flusso di lavoro ML orchestrata. Il tracciamento e la registrazione dei metadati sono necessari per ogni iterazione del processo di addestramento (ad es. data e ora dell'addestramento, durata e così via), inclusi i metadati specifici del modello, ad es. i parametri utilizzati e le metriche delle prestazioni risultanti, derivazione del modello: dati e codice utilizzati, per garantire la piena tracciabilità delle esecuzioni sperimentali.
Monitoraggio continuo	Il monitoraggio continuo implica la valutazione periodica dei dati, del modello, del codice, delle risorse dell'infrastruttura e delle prestazioni del servizio del modello.
Cicli di retroazione	Cicli di feedback multipli sono necessari per integrare gli approfondimenti dalla fase di valutazione della qualità nel processo di sviluppo o ingegnerizzazione (ad esempio, un ciclo di feedback dalla fase di progettazione del modello sperimentale alla precedente fase di progettazione delle caratteristiche). Un altro ciclo di feedback è richiesto dal componente di monitoraggio (ad esempio, osservando le prestazioni del servizio del modello) allo scheduler per consentire il riaddestramento.

Figura 4.2: Proprietà MLOps.

4.2 RQ2. Le pratiche e tool di MLOps adoperate nei sistemi ML

In secondo luogo, sono state identificate le pratiche e i tool MLOps utilizzati nei sistemi ML. Tramite un'analisi approfondita dei vari documenti, identificando dodici pratiche/tool. Mediante quest'analisi è stato generato un grafico **Figura 4.3**, il quale riporta tutte le pratiche e tool trovate e la presenza di utilizzo di una determinata pratica/tool in ogni documento attraverso una percentuale.

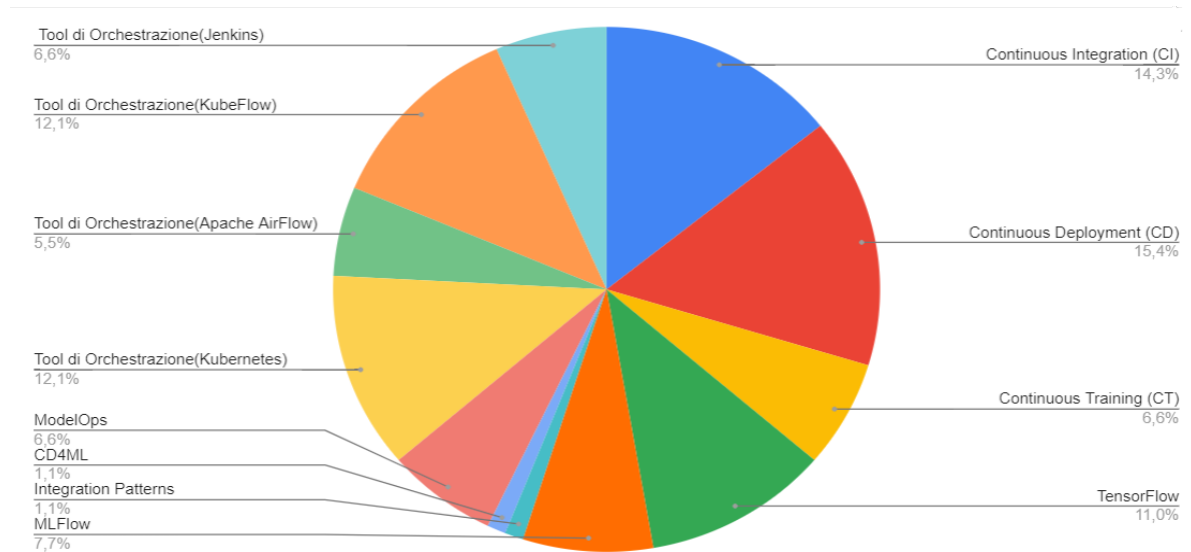


Figura 4.3: Grafico delle pratiche/tool MLOps.

Si nota che la pratica che viene riportata in più fonti è **Continuous Deployment (CD)** con il 15.4% ,poi abbiamo **Continuous Integration (CI)** con il 14.3 %, con il 12.1% abbiamo **KubeFlow** e **Kubernetes**,**TensorFlow**,**MLFlow** con il 7.7%, con il 6,6% abbiamo **Jenkins**, **ModelOps** e **Continuous Training (CT)**, **Apache AirFlow** con il 5.5%, mentre le pratiche che vengono meno menzionate sono **Integration Patterns** e **CD4ML** con il 1.1%. La tabella (**Figura 4.4**) la quale riporta tutte le proprietà MLOps trovate e la loro definizione.

Pratiche/Tool MLOps	Definizione
Continuous Deployment (CD)	È una pratica DevOps che si riferisce alla fase che viene subito dopo: quando c'è un nuovo artefatto di build (generato da CI), viene attivato un rilascio e l'artefatto viene distribuito nell'ambiente desiderato. Se l'intero processo è automatizzato, si parla di distribuzione continua [12].
Continuous Integration (CI)	È una pratica DevOps che consiste nell'avere la fase di sviluppo del software completamente automatizzata, con la convalida del codice che si verifica non appena le modifiche vengono confermate e unite in un sistema di controllo della versione [12].
KubeFlow	È un toolkit ML per distribuire pipeline ML in Kubernetes. La distribuzione dei carichi di lavoro può essere eseguita localmente, on-premise o in ambienti cloud. Ogni passo della pipeline ML adatta la configurazione scegliendo la piattaforma e servizio richiesto [8].
TensorFlow	È una libreria software open source per l'apprendimento automatico (<i>machine learning</i>), che fornisce moduli sperimentati e ottimizzati, utili nella realizzazione di algoritmi per diversi tipi di compiti percettivi e di comprensione del linguaggio.
Jenkins	È un server di automazione open source autonomo che può essere utilizzato per automatizzare le attività relative alla creazione, al test, alla consegna e alla distribuzione del software. Può essere utilizzato come applicazione autonoma all'interno di un servlet container/application server [8].
ModelOps	È un nuovo framework e piattaforma per la gestione del ciclo di vita end-to-end degli artefatti delle applicazioni AI. Uno dei componenti chiave di ModelOps è un linguaggio di astrazione del dominio con supporto di prima classe per gli artefatti comuni nelle soluzioni AI. ModelOps consente inoltre di integrare facilmente i controlli di qualità nel ciclo di vita di un'applicazione AI.
Continuous Training (CT)	È nuovo componente della pipeline e univoco per MLOps. La pipeline di addestramento continuo (CT) riaddestra automaticamente il modello.
Apache AirFlow	È una piattaforma open source per lo sviluppo programmatico, la pianificazione e il monitoraggio dei flussi di lavoro come grafi aciclici diretti per creare istanze di attività e flussi di lavoro. Le pipeline sono implementate come script Python, che rappresentano le strutture grafiche (ad esempio attività e dipendenze) come codice [8].
Kubernetes	È una piattaforma portatile, estensibile e open-source per la gestione di carichi di lavoro e servizi containerizzati, in grado di facilitare sia la configurazione dichiarativa che l'automazione.
Integration Patterns	Permette di minimizzare le dipendenze tra i vari sistemi eterogenei che usano formati di dati diversi prevedendo un livello di "indirizzamento" che disaccoppi i formati dati eterogenei da un formato dati canonico e indipendente da una specifica applicazione.
CD4ML	È un approccio di ingegneria del software in cui un team interfunzionale produce applicazioni di machine learning basate su codice, dati e modelli in incrementi piccoli e sicuri che possono essere riprodotti e rilasciati in modo affidabile in qualsiasi momento, in brevi cicli di adattamento.
MLFlow	È una piattaforma open source per la gestione il ciclo di vita ML end-to-end. Si compone di quattro componenti: Monitoraggio, progetti, modelli e registro dei modelli.

Figura 4.4: Pratiche e Tool MLOps.

4.3 RQ3. Le pratiche o tool che permettono di soddisfare le proprietà MLOps

In terzo luogo, sono state identificate le pratiche e tool riportati nei documenti considerati che permettono di soddisfare le proprietà MLOps. Tramite un'analisi approfondita dei vari documenti sono stati identificati dodici pratiche e tool trovati nella RQ2, le quali soddisfano le proprietà MLOps. Mediante quest'analisi è stato generato un grafico **Figura 4.5**, nel quale vengono riportate tutte le pratiche e tool che soddisfano le proprietà MLOps e la presenza di utilizzo delle proprietà per una determinata pratica/tool in ogni documento attraverso una percentuale.

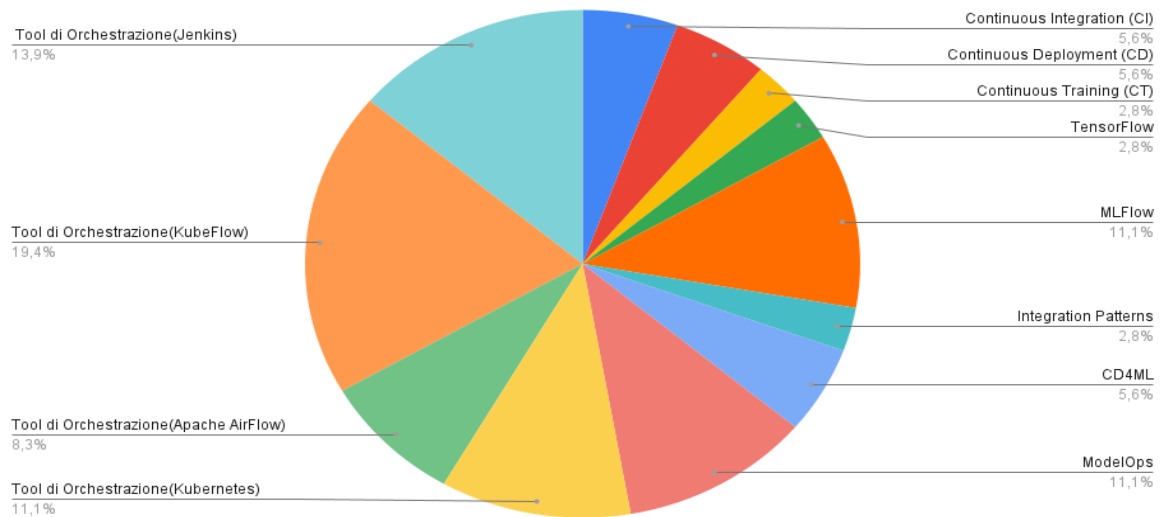


Figura 4.5: Pratiche/Tool che soddisfano le proprietà MLOps .

Si nota che il tool che soddisfa più proprietà è **KubeFlow** con il 19.4%, poi abbiamo **Jenkins** con il 13.9%, con l'11.1% abbiamo **ModelOps** e **Kubernetes**, **MLFlow**, **Apache AirFlow** con il 8.3%, con il 5.6% abbiamo **CD4ML**, **Continuous Deployment (CD)** e **Continuous Integration (CI)**, mentre le pratiche o tool che soddisfano meno proprietà sono **Continuous Training (CT)**, **TensorFlow** e **Integration Patterns** con il 2.8%. Inoltre, nella tabella (**Figura 4.6**) per ogni proprietà MLOps vengono rappresentate le pratiche/ tool riscontrati, i quali sono identificati tramite la parola "SI".

	Proprietà: Automatizzazione CI/CD	Proprietà: Formazione /Valutazione Continua	Proprietà: Collaborazione	Proprietà: Monitoraggio Continuo	Proprietà: Cicli di retroazione	Proprietà: Tracciabilità	Proprietà: Riproducibilità	Proprietà: Versione	Proprietà: Orchestrazione del flusso di lavoro
Continuous Integration (CI)	SI	SI							
Continuous Deployment (CD)	SI	SI							
Continuous Training (CT)		SI							
TensorFlow								SI	
MLFlow				SI		SI	SI		SI
Integration Patterns	SI								
CD4ML		SI					SI		
ModelOps		SI	SI	SI					SI
Kubernetes		SI		SI	SI				SI
Apache AirFlow		SI					SI		SI
KubeFlow		SI		SI	SI	SI	SI	SI	SI
Jenkins	SI	SI	SI		SI				SI

Figura 4.6: Tabella che riporta per ogni proprietà le pratiche/tool riscontrate.

Conclusioni e Sviluppi Futuri

Questo articolo riporta una revisione sistematica della letteratura sulla ricerca di pratiche, tool e proprietà MLOps utilizzate dagli autori. Sono state approfondite una serie di punti di vista di ricerca, che mirano a valutare lo stato dell'arte attuale rispetto alle tipiche pratiche/tool MLOps trattate, le proprietà MLOps trattate e affrontare le pratiche o tool che soddisfano le proprietà MLOps. I risultati di questa analisi sistematica lasciano emergere che la proprietà MLOps più utilizzata è Versione con il 13.5%, mentre la proprietà che viene meno menzionata è Cicli di Retrazione con il 7.7% (**Figura 4.1**). Inoltre, la pratica più utilizzata dagli autori in letteratura è Continuous Deployment (CD) con il 15.4% , mentre le pratiche che vengono meno menzionate sono Integration Patterns e CD4ML con il 1.1% (**Figura 4.3**). Infine, il tool che soddisfa più proprietà è KubeFlow con il 19.4% , mentre le pratiche o tool che soddisfano meno proprietà sono Continuous Training (CT), TensorFlow e Integration Patterns con il 2.8% (**Figura 4.5**).

Nel complesso, questo lavoro di tesi fornisce i contributi elencati di seguito:

- Una revisione sistematica della letteratura che riassume le attuali conoscenze sull'uso delle pratiche o tool e proprietà MLOps.
- L'identificazione dei limiti, delle questioni aperte e delle sfide dello stato dell'arte, che i ricercatori potrebbero utilizzare per definire le prossime fasi di ricerca per migliorare il supporto fornito agli sviluppatori.

5.1 Sviluppi Futuri

Il nostro futuro programma di ricerca è guidato dalle considerazioni e dalle implicazioni di questa revisione sistematica della letteratura. Il lavoro di tesi mira infatti a eseguire analisi empiriche per confrontare l'efficacia delle soluzioni attualmente disponibili, inoltre, per migliorare quanto descritto è possibile eseguire nuove ricerche che mirano ad approfondire tale studio, così da ampliare maggiormente l'insieme delle pratiche, tool e proprietà MLOps trattate in letteratura.

I risultati di questo lavoro di tesi hanno fatto emergere i diversi benefici derivanti dall'applicazione di MLOps. L'utilizzo di queste pratiche all'interno dei propri sistemi può garantire un'efficace comunicazione tra i membri dei teams di sviluppo, fornire monitoraggio continuo nelle varie fasi, tracciabilità e valutazione continua. Un possibile tool che si potrebbe applicare ed essere migliorato sarebbe **KubeFlow** (con il 19.4% nei risultati della RQ3) essendo un dei tool che supporta più proprietà MLOps.

Ringraziamenti

Ringrazio il mio relatore Fabio Palomba, il dottore Giammaria Giordano e il dottore Gilberto Recupito per avermi dato la possibilità di partecipare a questo progetto e per avermi guidato nella fase più importante del mio percorso accademico.

Un ringraziamento speciale alla mia famiglia. Le parole non possono esprimere quanto sono grata a mia madre e mio padre per tutti i sacrifici che hanno fatto per me. Il loro incoraggiamento è stato ciò che mi ha maggiormente sostenuta nel conseguimento di questo obiettivo, dandomi forza e coraggio per attraversare i periodi di difficoltà.

Ringrazio mia nonna, per l'affetto che non ha mai fatto mancare, per essere stata sempre orgogliosa di me e per le tante preghiere fatte ad ogni esame.

Ringrazio mia sorella per essere stata il mio punto di riferimento per questi tre anni universitari, e anche una guida dato che senza il suo sostegno e incoraggiamento non sarei mai arrivata dove sono oggi. Soprattutto grazie per avermi amato anche quando non mi amavo e grazie per avermi aiutato quando ne avevo più bisogno.

Ringrazio Vincent, per avermi aiutata in alcuni momenti di difficoltà.

Ringrazio Antonella, la mia amica da sempre e per sempre. Grazie per l'autostima che riesci ad infondermi e per avermi sempre tenuto la mano in questo percorso e in questi mesi difficili.

Ringrazio Rosalia, la mia cara amica per avermi sempre supportato e per esserci sempre stata per me, e che sicuramente in questi giorni per me speciali continua a farlo da lassù.

Ringrazio i mie colleghi e amici Valentina, Alessandra, Martina, Elena, Siria, Pio, Francesco, Fabio, Severino e Alberto con i quali ho trascorso bellissimi momenti del mio percorso universitario e condiviso le gioie, le ansie e le difficoltà in questi tre anni.

Ringrazio tutti i miei zii, le mie zie, le mie cugine e mio cugino per l'affetto dimostrato, in particolare a zio Carmine e mio cugino Damiano per i numerosi passaggi da casa all'università.

Grazie a tutti, senza di voi non ce l'avrei mai fatta.

Bibliografia

- [1] Yue Zhou, Yu Yu e Bo Ding; *Towards MLOps: A Case Study of ML Pipeline Platform*, 2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE), l' 01 Marzo 2021, DOI: 10.1109/ICAICE51518.2020.00102.
- [2] SATVIK GARG, PRADYUMN PUNDIR, GEETANJALI RATHEE, PK GUPTA, SOMIA GARG AND SARANSH AHLAWAT. *On Continuous Integration / Continuous Delivery for Automated Deployment of Machine Learning Models using MLOps*, pubblicato su: 2021 IEEE Fourth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE), il 03 Marzo 2022, DOI: 10.1109/AIKE52691.2021.00010.
- [3] JENS FRID AND ANDREAS HELLANDER *The machine learning life cycle and the cloud: implications for drug discovery*, Ricevuto il 22 Marzo 2021, Accettato il 17 Maggio 2021, pubblicato il: 31 Maggio 2021, DOI: 10.1080/17460441.2021.1932812.
- [4] SERGIO MORESCHINI, FRANCESCO LOMIO, DAVID HASTBACKA E DAVIDE TAIBI; *MLOps for evolvable AI intensive software systems*, pubblicato su: 2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), il 21 Luglio 2022, DOI: 10.1109/SANER53432.2022.00155.
- [5] DOMINIK KREUZBERGER, NIKLAS KUHL E SEBASTIAN HIRSCHL. *Machine Learning Operations (MLOps): Overview, Definition, and Architecture*, Presentato il 4 maggio 2022 (versione1), Ultima Revisione: 14 Maggio 2022 (versione3), DOI: 10.48550/arXiv.2205.02302.
- [6] NIRANJAN D R E MOHANA. *Jenkins Pipelines: A Novel Approach to Machine Learning Operations (MLOps)*, pubblicato su: 2022 International Conference

- on Edge Computing and Applications (ICECAA), del 08 Novembre 2022, DOI: 10.1109/ICECAA55415.2022.9936252.
- [7] MATEI ZAHARIA, ANDREW CHEN, AARON DAVIDSON, ALI GHODSI, SUE ANN HONG, ANDY KONWINSKI, SIDDHARTH MURCHING, TOMAS NYKODYM, PAUL OGILVIE, MANI PARKHE, FEN XIE, COREY ZUMAR: *Accelerating the Machine Learning Lifecycle with MLflow*, pubblicato su: IEEE Data EngBull, il 3 Aprile 2018.
- [8] FABIO PALOMBA, GILBERTO RECUPITO, FABIANO PECORELLI, GEMMA CATOLINO, SERGIO MORESCHINI, DARIO DI NUCCI E DAMIAN A TAMBURRI; *A Multivocal Literature Review of MLOps Tools and Features*, pubblicato a Giugno 2022.
- [9] DAMIAN A. TAMBURRI; *Sustainable MLOps: Trends and Challenges*, Pubblicato su: 2020 22nd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), il 24 Febbraio 2021, DOI: 10.1109/SYNASC51798.2020.00015.
- [10] SASU MAKINEN, HENRIK SKOGSTROM, EERO LAAKSONEN E TOMMI MIKKONN; *Who Needs MLOps: What Data Scientists Seek to Accomplish and How Can MLOps Help?*, Pubblicato su: 2021 IEEE/ACM 1st Workshop on AI Engineering - Software Engineering for AI (WAIN), l'08 Luglio 2021, DOI: 10.1109/WAIN52551.2021.00024.
- [11] BO LI, PENG QI, BO LIU, SHUAI DI, JINGEN LIU, JIQUAN PEI, JINFENG YI E BOWEN ZHOU; *Trustworthy AI: From Principles to Practices*, presentato il 4 Ottobre 2021, DOI: 10.48550/arXiv.2110.01167.
- [12] BEATRIZ M. A. MATSUI E DENISE H. GOYA; *MLOps: A Guide to its Adoption in the Context of Responsible AI*, pubblicato su: 2022 IEEE/ACM 1st International Workshop on Software Engineering for Responsible Artificial Intelligence (SE4RAI), il 30 Giugno 2022.
- [13] DHAYA SINDHU BATTINA; *AN INTELLIGENT DEVOPS PLATFORM RESEARCH AND DESIGN BASED ON MACHINE LEARNING*, Sr.Data Engineer & Department of Information Technology USA, pubblicato a Marzo 2019.
- [14] MEENU MARY JOHN, HELENA HOLMSTROM OLSSON E JAN BOSCH. *Towards MLOps: A Framework and Maturity Model*, Pubblicato su: 2021 47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), il 27 Ottobre 2021, DOI: 10.1109/SEAA53835.2021.00050.

- [15] RAKSHITH SUBRAMANYA, SEPPÖ SIERLA E VALERIY VYATKIN; *From DevOps to MLOps: Overview and Application to Electricity Market Forecasting*, pubblicato il 30 settembre 2022, DOI: 10.3390/app12199851.
- [16] TUOMAS GRANLUND, ALEKSI KOPPONEN, VLAD STIRBU, LALLI MYLLYAHÖ E TOMMI MIKKONEN; *MLOps Challenges in Multi-Organization Setup: Experiences from Two Real-World Cases*, Pubblicato su: 2021 IEEE/ACM 1st Workshop on AI Engineering - Software Engineering for AI (WAIN), l' 08 luglio 2021, DOI: 10.1109/WAIN52551.2021.00019.
- [17] ENRICO SALVUCCI; *MLOps - Standardizing the Machine Learning Work-flow*, UNIVERSITÀ DI BOLOGNA 'CAMPUS DI CESENA I Sessione di Laurea Anno Accademico 2020-2021.
- [18] W. HUMMER, V. MUTHUSAMY, T. RAUSCH, P. DUBE, K. EL MAGHRAOUI, A. MURTHI E P. OUM; *Modelops: Cloud-based lifecycle management for reliable and trusted AI*, pubblicato su: 2019 IEEE International Conference on Cloud Engineering (IC2E), 2019, pp. 113–120, DOI: 10.1109/IC2E.2019.00025.
- [19] IOANNIS KARAMITSOS, SAEED ALBARHAMI AND CHARALAMPOS APOSTOLOPOULOS; *Applying DevOps Practices of Continuous Automation for Machine Learning*, pubblicato il 13 July 2020, <https://www.mdpi.com/2078-2489/11/7/363>.