ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

SCUOLA DI INGEGNERIA E ARCHITETTURA

DIPARTIMENTO DI

INGEGNERIA DELL'ENERGIA ELETTRICA E DELL'INFORMAZIONE

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA DELL'ENERGIA ELETTRICA

TESI DI LAUREA

In

STRUMENTAZIONE E METODI PER LE MISURE SU SISTEMI ELETTRICI - M

Manutenzione predittiva dei giunti mediante tecnologie di Intelligenza Artificiale per la prevenzione dei guasti nelle reti di distribuzione in MT

CANDIDATO:

Virginia Negri

RELATORE:

Prof. Roberto Tinarelli

CORRELATORI:

Prof. Roberta Calegari

Prof. Alessandro Mingotti

Prof. Lorenzo Peretto

Anno Accademico 2021/2022

Sessione II

Indice

1. Introduzione 1
2. Giunti nelle reti di distribuzione 3
   2.1. Introduzione 3
   2.2. Struttura 4
   2.3. Modi di guasto 8
   2.4. Parametri caratteristici 10
3. Intelligenza Artificiale 13
   3.1. Concetti fondamentali 13
   3.2. Algoritmi di machine learning 15
      3.2.1. Algoritmi di apprendimento supervisionato 16
         3.2.1.1. Logistic Regression (LR) 16
         3.2.1.2. Support Vetor Machine (SVM) 17
         3.2.1.3. K-Nearest Neighbors (KNN) 18
         3.2.1.4. Decision Tree 19
         3.2.1.5. Random Forest 20
         3.2.1.6. Artificial Neural Network (ANN) 21
      3.2.2. Metriche di valutazione 23
         3.2.2.1. Accuratezza 23
         3.2.2.2. Matrice di confusione 23
         3.2.2.3. Precisione e recall 25
4. Stato dell'arte 26
   4.1. Systematic Literature Review (SLR) 26
   4.2. Metodo 27
      4.2.1. Research question 27
      4.2.2. Ricerca bibliografica 27
      4.2.3. Criteri di inclusione ed esclusione 28
      4.2.4. Analisi dei dati 29
   4.3. Risultati 32
5. Descrizione dei test 35
   5.1. Test su dataset di pubblico dominio 36
   5.2. Generazione di dati sintetici 37
   5.3. Test su dataset sintetico 43
      5.3.1. Numero di campioni nella fase di apprendimento 44

5.3.2. Ripetibilità 45
5.3.3. Distribuzione dei dati 45
6. Risultati 47
6.1. Test su dataset di pubblico dominio 47
6.2. Test su dataset sintetico 51
6.2.1. Numero di campioni nella fase di apprendimento 56
6.2.2. Ripetibilità 60
6.2.3. Distribuzione dei dati 64
7. Conclusioni 70

Bibliografia 72

1. Introduzione

Lo scopo di questo elaborato è implementare e valutare tecnologie di intelligenza artificiale per svolgere manutenzione predittiva sui giunti installati nelle linee elettriche in cavo, contribuendo quindi alla prevenzione dei guasti nella rete di distribuzione di media tensione.

È infatti noto che i giunti siano, insieme alle terminazioni, i punti deboli della rete elettrica e che la loro manutenzione sia un intervento complesso. Questo richiede infatti lunghi tempi di intervento durante i quali vi è un'interruzione della fornitura elettrica e inoltre comporta costi onerosi a carico del gestore della rete. Di conseguenza, svolgere manutenzione condition-based su questi componenti risulta essere un'azione importante, al fine di prevenirne i guasti e ottenere un risparmio in termini di tempo e denaro.

Per manutenzione predittiva si intende un tipo di manutenzione che ha l'obiettivo di valutare lo stato di salute del componente ed eventualmente effettuare un intervento di manutenzione nel caso in cui questo sia necessario. A differenza della manutenzione correttiva, questo tipo di manutenzione fa sì che l'intervento venga predisposto prima di un guasto imminente, evitando quindi l'incorrere di quest'ultimo. La manutenzione predittiva si differenzia poi da quella correttiva poiché l'intervento è pianificato in relazione all'effettiva situazione del componente e non a priori, evitando lo spreco di risorse e mezzi ancora in grado di svolgere il proprio compito.

Per svolgere manutenzione preventiva vengono utilizzate tecniche di condition monitoring al fine di osservare le prestazioni dell'attrezzatura durante il suo normale funzionamento e individuare eventuali anomalie per risolverle prima che diano luogo a malfunzionamenti. Nel caso specifico dei giunti, la manutenzione condition-based può essere effettuata selezionando ed osservando delle grandezze di tipo elettrico e non, le quali sono considerate di interesse per la valutazione dell'integrità del componente.

A tale scopo questo elaborato tratterà i giunti in termini di struttura, funzione e modi di guasto, per definirne le cause e i parametri caratteristici e comprendere gli andamenti delle grandezze di interesse e le relazioni che intercorrono fra queste e i vari meccanismi di guasto, al fine della prevenzione di quest'ultimo. Sulla base di queste considerazioni, sarà poi generato sinteticamente un dataset che simuli diverse misurazioni sui giunti delle grandezze considerate di interesse, al fine di ottenere sufficienti dati per procedere alla fase implementativa e di test.

Per lo sviluppo del modello predittivo saranno utilizzate tecnologie di intelligenza artificiale e in particolare algoritmi di machine learning: tramite una Revisione Sistematica della Letteratura (SLR) si valuterà lo stato dell'arte e tramite opportuni criteri saranno selezionati sei algoritmi, che verranno implementati e confrontati in termini di prestazioni.

In conclusione saranno presentati i risultati ottenuti e sarà valutata l'effettiva applicabilità dell'intelligenza artificiale per svolgere manutenzione predittiva sui giunti, arrivando anche a comprendere quali algoritmi siano più efficienti per tale scopo.

2. Giunti nelle reti di distribuzione

2.1. Introduzione

L’estensione delle reti in cavo è in costante aumento per supportare lo sviluppo della rete elettrica, specialmente nelle zone urbane dove non è possibile realizzare linee aree, garantendo maggior affidabilità e minor inquinamento visivo rispetto a queste ultime.

Per l’installazione di nuove linee, l’estensione di linee preesistenti o la loro manutenzione si fa uso di giunti, necessari perché i cavi di linea hanno lunghezza limitata e fissata in base alle loro caratteristiche, quali diametro, peso e possibilità di trasporto. Tipicamente i giunti sono installati ogni 500 – 800 m di linea e di conseguenza il loro numero può essere considerevole, specialmente se consideriamo linee di distribuzione estese.

Per i motivi sopra detti, nelle linee di distribuzione in cavo si fa ampio uso di giunti, che però sono spesso soggetti a guasti e di difficile manutenzione: questi componenti sono infatti sollecitati da diversi tipi di stress, quali stress meccanici, termici, elettrici ed ambientali, e si trovano a diversi metri di profondità nel terreno.

Una delle attuali sfide tecnologiche di questo campo è quella di aumentare l’affidabilità di questi componenti, al fine di ridurre il numero di guasti nella rete di distribuzione: è infatti generalmente noto che siano la principale causa di guasto nelle linee di media tensione e che il modo di guasto più frequente sia il cedimento dell’isolante elettrico.

Se consideriamo la vita media di un cavo, questa è di circa 30-40 anni, mentre quella di un giunto è di circa 7-8 anni. Inoltre, è risaputo che modalità di installazione, condizioni di posa e caratteristiche dell’ambiente circostante, quali tipo di terreno, umidità e temperatura, possano drasticamente velocizzarne l’invecchiamento, in aggiunta ad eventuali difetti di produzione preesistenti.

In caso di guasto di un giunto elettrico, è necessario disalimentare la porzione di rete in cui questo è installato fino al momento della sua sostituzione,

comportando un intervento di manutenzione oneroso, un disservizio agli utilizzatori ed una conseguente penale a carico del gestore della rete.
Risulta quindi importante svolgere manutenzione predittiva sui giunti, al fine di prevenirne i guasti ed ottenere un risparmio in termini di tempo e denaro.

2.2. Struttura

Per giunto si intende un punto della rete elettrica dove due o più cavi sono uniti fra loro. Questo collegamento è realizzato in modo da garantire continuità elettrica nel punto di connessione, isolamento elettrico verso l'esterno, supporto meccanico e protezione fisica del cavo su cui sono installati.
In base alla funzione, al tipo di collegamento, al tipo di cavo e ai materiali costruttivi vengono identificati diversi tipi di giunti, mentre il loro design è tipicamente basato sui valori di corrente e tensione a cui sono sottoposti e sulle condizioni dell'ambiente dove sono installati.
In particolare, vi sono due macrocategorie di giunti: giunti restringibili a caldo e giunti restringibili a freddo.
I giunti restringibili a caldo sono rappresentati in figura 2.1 e sono tipicamente costituiti di un materiale polimerico (gomma-plastica) termo-restringente. Una volta installati sul cavo, necessitano di una fonte di calore, come per esempio una fiamma ossidrica, per aderire al cavo.
I giunti restringibili a freddo sono rappresentati in figura 2.2. e sono tipicamente costituiti di gomma siliconica che viene espansa su di un nucleo rigido a spirale. Una volta installati sul cavo, la spirale viene estratta e il giunto aderisce gradualmente al cavo.

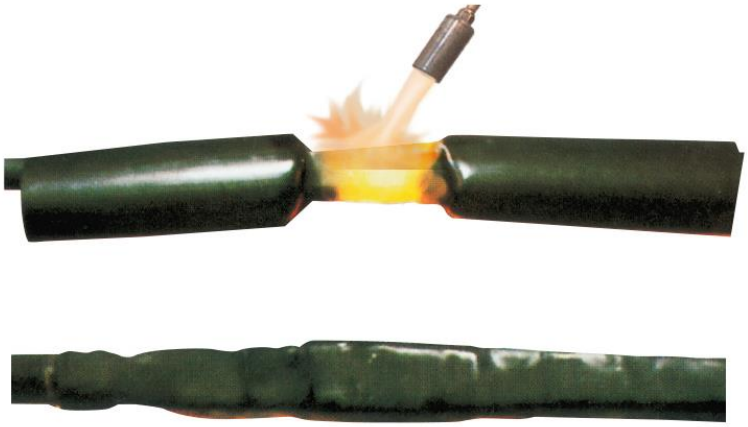


Figura 2.1 Giunto restringibile a caldo. Estratto da https://www.raytech.it/en/product/low-voltage/joint/heat-shrink/glv

Figura 2.2 Giunto restringibile a freddo. Estratto da https://rodantech.com/product/tef-cold-shrink-tube/

Per quanto riguarda il materiale costituente, la gomma siliconica risulta avere ottime proprietà isolanti ed elevata elasticità, caratteristica invece assente nel polimero costituente i giunti restringenti a caldo. Oltre a questo, le due tipologie di giunti citate si differenziano per ulteriori aspetti: in primis, i giunti a freddo sono prefabbricati all'interno di aziende tramite processi ad alta pressione ed alta

temperatura e nel luogo di installazione necessitano soltanto della rimozione del nucleo a spirale, azione limitata che non intacca quindi la qualità raggiunta e controllata dal processo produttivo. I giunti a caldo invece richiedono un'installazione più complessa, che può portare più facilmente all'insorgere di difetti nel componente e che non può essere svolta per esempio in luoghi dove non si può utilizzare la fiamma ossidrica.

Inoltre, per come sono realizzati ed installati, i giunti a freddo si adattano perfettamente al cavo, seguendone le deformazioni termiche e limitando la formazione di vuoti o l'ingresso di acqua e lasciando inalterate le prestazioni isolanti del componente. Per quel che riguarda i giunti a caldo, questi sono privi di elasticità e di conseguenza hanno minor resistenza alle sollecitazioni meccaniche e cicli termici possono portare all'insorgere di difetti all'interfaccia cavo-giunto. I giunti restringibili a freddo risultano quindi essere quelli più utilizzati per la loro facilità di installazione (tipicamente richiedono un terzo del tempo richiesto da quelli a caldo), la loro versatilità e le loro elevate prestazioni in termini di isolamento e protezione meccanica del cavo.

Figura 2.3 – Struttura di un giunto restringibile a freddo. Estratto da A. Ghaderi, A. Mingotti, F. Lama, L. Peretto and R. Tinarelli, "Effects of Temperature on MV Cable Joints Tan Delta Measurements", in IEEE Transactions on Instrumentation and Measurement, vol. 68, no. 10, pp. 3892-3898, Oct. 2019, doi: 10.1109/TIM.2019.2900131.

La generica struttura di un giunto a freddo è rappresentata in figura 2.3, dove vengono fondamentalmente identificati quattro strati a partire da quello più esterno: guaina a freddo, schermo metallico, strato isolante in gomma siliconica e strati semiconduttivi, connettore metallico.

Come detto in precedenza, la guaina restringibile a freddo garantisce l'isolamento elettrico e la protezione da sollecitazioni meccaniche.

Lo schermo metallico è costituito da sottili fili conduttori intrecciati fra loro e ha la funzione di collegamento del potenziale di terra fra le due parti di cavo unite dal giunto, per confinare il campo elettrico all'interno del giunto e del cavo e, inoltre, limitare i disturbi dall'esterno.

Lo strato isolante in gomma siliconica è preceduto e seguito da strati di materiale semiconduttivo al fine di uniformare il campo elettrico al quale il componente è sollecitato.

Il connettore metallico realizza il collegamento meccanico fra i due conduttori e contribuisce alla protezione meccanica di quest'ultimi.

2.3. Modi di guasto

Come già espresso nel paragrafo precedente, i giunti sono componenti importanti da monitorare: sono infatti la principale causa di guasto nelle reti di media tensione. Il modo di guasto più frequente per questi accessori è il cedimento dell'isolante elettrico, che tipicamente viene identificato come unico modo di guasto in seguito ad usura e/o invecchiamento dell'isolante stesso. In realtà questa degradazione è un fenomeno molto complesso, che può essere scaturito da diverse cause e a sua volta scaturire svariate conseguenze, tutte fortemente interconnesse.

In particolare possono essere identificati come modi di guasto: breakdown dell'isolamento principale o dell'isolante del cavo, thermal breakdown nel dielettrico, guasto nel punto di collegamento dei conduttori, nella guaina e all'interfaccia cavo-giunto. Come principali cause vi sono l'instaurazione di scariche parziali e lo sviluppo del treeing elettrico o del water treeing, fenomeni alquanto interconnessi e a loro volta conseguenze (ed in alcuni casi anche cause stesse) di: sovratensioni, surriscaldamento, corrente eccessiva, presenza di difetti o vuoti nell'isolante, sollecitazioni dovute a cicli termici, stress meccanici, ingresso di acqua o umidità a causa di danni alla guaina del cavo, thermal runaway, rilassamento o cedimento meccanico dei componenti, rottura o cortocircuito di conduttori elettrici e diretto danneggiamento [1, 2, 3]. Inoltre altre cause di guasto sono l'invecchiamento, la scarsa fabbricazione, la presenza di difetti di produzione e di danni meccanici per scarsa resistenza meccanica del rivestimento esterno, la posa impropria o un'installazione non corretta.

In tabella 2.1 sono riportati i diversi modi di guasto e alcune delle rispettive cause.

Tabella 2.1 – Modi di guasto e rispettive cause nei giunti in media tensione

MODO DI GUASTO	CAUSE DI GUASTO
Electric breakdown dell'isolamento principale	Usura dell'isolamento, causato da: - Electrical treeing (provocato da instaurazione di scariche parziali), causato per esempio da vuoti per cicli termici - Sovratensioni - Water treeing - Surriscaldamento - Scarica all'interfaccia
Breakdown dell'isolante del cavo	Usura dell'isolamento, causato da: - Electrical treeing (provocato da instaurazione di scariche parziali), causato per esempio da vuoti per cicli termici, scorretto taglio dello schermo o taping, difetti o sporcizia sulla superficie dell'isolante - Water treeing provocato dall'ingresso di acqua e umidità - Cicli termici (es. migrazione impregnante isolamento) - Sovraccarico - Stress meccanici - Ingresso di acqua e umidità nel cavo per danni alla guaina Eccessiva corrente, causata da: - Difetti nella guaina - Difetti nel conduttore
Thermal breakdown dell'isolante	Degrado dell'isolante - Corrente eccessiva - Thermal runaway causata per esempio dall'ingresso di acqua e la conseguente ossidazione dei fili dell'armatura - Stato del suolo Aumento del calore dissipato - Degrado del collegamento fra conduttori per cicli di lavoro
Guasto nel punto di collegamento dei conduttori	Posizionamento improprio del core Collegamento errato per errata crimpatura dei conduttori Vuoti nel collegamento meccanico Spessore irregolare nel build-up del tape isolante Eccessiva corrente di servizio
Guasto nella guaina	Ingresso di acqua e umidità

Guasto all'interfaccia cavo-giunto	Usura dell'isolamento - Presenza di vuoti causati per esempio da cicli termici o sovraccarichi - Sovraccarico - Electrical treeing o arco elettrico (provocati dall'instaurazione di scariche parziali), causati per esempio da spessore irregolare nel build-up del tape isolante, diminuzione della pressione all'interfaccia difetti nell'interfaccia cavo-giunto, difetti nell'isolante del cavo - Water treeing causato dall'ingresso di acqua e umidità per danni alla guaina - Rilassamento meccanico per invecchiamento - Surriscaldamento - Eccessiva corrente

2.4. Parametri caratteristici

Al fine di prevenire i guasti sui giunti vi sono svariati parametri utili allo scopo di monitoraggio, quali fattori ambientali come per esempio temperatura, pressione ed umidità e fattori elettrici come per esempio corrente, tensione, campo elettrico, scariche parziali e tan delta.

In particolare, le scariche parziali possono essere misurate per classificarle [4], localizzarne la sorgente e identificare il punto di guasto [5] o per stimare lo stato di invecchiamento dell'isolante del giunto [6, 7], mentre è possibile rilevare un guasto o un difetto sulla base di misure di corrente [3], pressione e temperatura [8] o campo elettrico [9] o valutare la qualità del dielettrico calcolandone il tan delta [10, 11].

Un altro aspetto importante da considerare nella valutazione di questi parametri è che risultano tutti fortemente interconnessi: in letteratura sono infatti presenti molteplici studi che indagano le relazioni fra le grandezze elettriche e non di questi componenti. Il fine ultimo che accomuna queste differenti ricerche è quello di conoscere a fondo i processi che possono instaurarsi all'interno dei giunti al fine di prevenirne i guasti.

In [12] è investigata la relazione fra temperatura e tan delta ed è osservato come quest'ultima grandezza aumenti al diminuire della temperatura. In particolare, per la misurazione del tan delta è utilizzato il circuito presentato in [11] e mostrato in figura 2.4 e, nella sua versione equivalente, in figura 2.5.

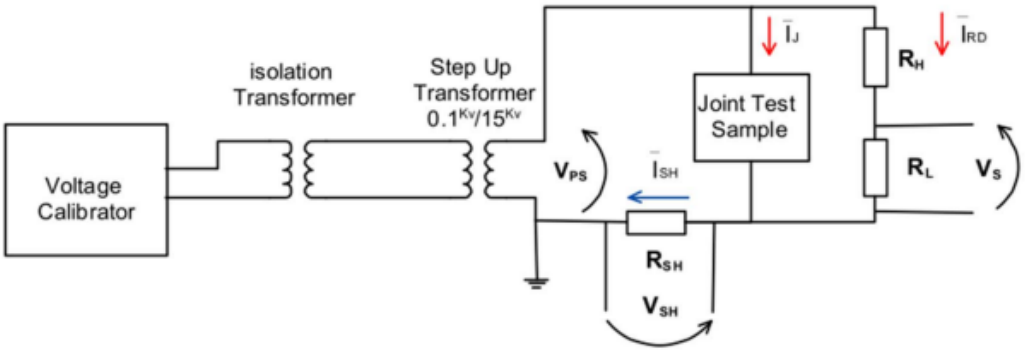


Figura 2.4 – Circuito per la misurazione automatica del Tan Delta. Estratto da A. Mingotti, A. Ghaderi, L. Peretto, R. Tinarell, F. Lama, "Test Setup Design, and Calibration for Tan Delta Measurements on MV Cable Joints", 2018 IEEE 9th International Workshop on Applied Measurements for Power Systems (AMPS), 2018, pp. 1-5.

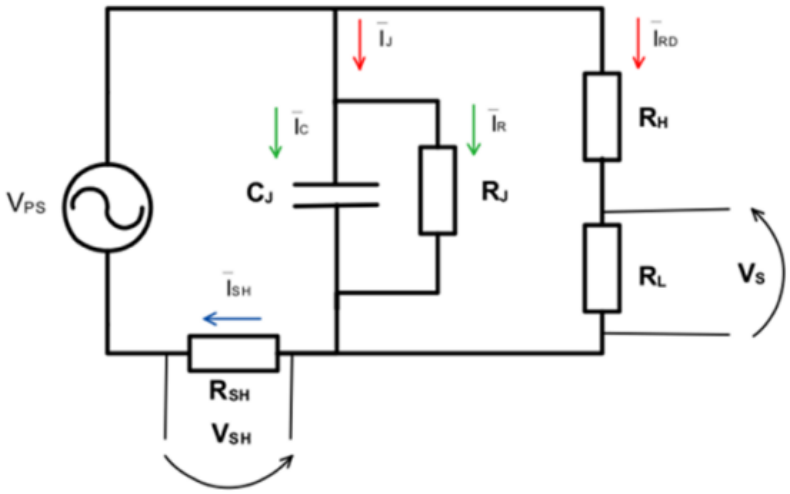


Figura 2.5 – Circuito equivalente per la misurazione automatica del Tan Delta. Estratto da A. Mingotti, A. Ghaderi, L. Peretto, R. Tinarell, F. Lama, "Test Setup Design, and Calibration for Tan Delta Measurements on MV Cable Joints", 2018 IEEE 9th International Workshop on Applied Measurements for Power Systems (AMPS), 2018, pp. 1-5.

In [13] la relazione sopra citata è indagata in modo più approfondito e, a partire da misurazioni di tan delta, è calcolata l'impedenza equivalente del giunto, al fine di valutarne resistenza e reattanza e capire come queste cambino al variare

della temperatura del giunto. In conclusione è infatti reso noto che la diminuzione del tan delta in relazione ad un aumento di temperatura è dovuta al fatto che, seppur la reattanza aumenti, la resistenza ha una crescita maggiore e prevale quindi sul risultato.

Il tan delta è anche valutato in relazione alla pressione: in [14] tre giunti di diverse caratteristiche sono utilizzati per investigare la correlazione esistente fra queste due grandezze. In particolare, il circuito di misura per il tan delta è quello menzionato precedentemente e mostrato in figura 2.4 e 2.5 e le variazioni di pressione sono indotte da morsetti opportunamente regolati. I test in laboratorio così effettuati portano ad osservare che il tan delta diminuisce a seguito di un aumento di pressione, poiché quest'ultimo comporta una modifica all'interfaccia fra l'isolante del cavo e l'isolante del giunto, con una conseguente riduzione della componente tangenziale del campo elettrico.

Infine, un ulteriore legame che esiste fra i parametri caratteristici è quello fra pressione e temperatura e in [15, 16], mediante la sollecitazione di giunti a cicli termici, si ottiene che un aumento di temperatura porta ad un amento di pressione.

È quindi dimostrato che la temperatura ambiente influenzi le prestazioni dei giunti e, inoltre, che il loro tasso di guasto sia maggiore nei mesi estivi, quando le temperature sono più elevate [17].

Come già espresso in precedenza, tutti i parametri sopra menzionati risultano essere fortemente interconnessi e lo studio delle relazioni che li legano è estremamente complesso. Una comprensione approfondita di questi aspetti è però fondamentale per valutare l'integrità dei giunti e delle loro parti costituenti e l'analisi dell'evoluzione temporale delle grandezze di interesse è quindi uno strumento importante per la diagnostica e la prevenzione dei guasti in questi componenti.

3. Intelligenza artificiale

3.1. Concetti fondamentali

L'intelligenza artificiale è una disciplina informatica che convenzionalmente nasce nel 1956 [18] e che studia la progettazione di sistemi hardware e programmi software che simulino processi e aspetti dell'intelligenza umana, quali il ragionamento, l'apprendimento, l'adattamento, la pianificazione, la percezione visiva o spazio-temporale ed anche la creatività.

A seconda del principio alla base dello sviluppo di tecnologie di intelligenza artificiale, questa disciplina viene suddivisa in intelligenza artificiale forte e debole: la prima si fonda sull'idea che le macchine siano in grado di sviluppare una coscienza propria e autonoma, in grado anche di superare le capacità di apprendimento umano, mentre la seconda ritiene che le macchine riescano a svolgere i processi richiesti senza una vera e propria coscienza delle loro attività, determinando quindi un sistema capace di svolgere una o più funzioni umane complesse, ma privo di una reale intelligenza e coscienza umana.

In generale, i sistemi di intelligenza artificiale lavorano valutando una grande quantità di dati al fine di trovarvi correlazioni e patterns ed usarli per fare predizioni o valutazioni su dichiarazioni future.

Le tecnologie di intelligenza artificiale e gli ambiti applicativi sono molteplici, rendendo questa disciplina in grado di avere impatti importanti su diversi aspetti della realtà che ci circonda. Le applicazioni principali sono l'implementazione di chatbot, uno strumento capace di offrire assistenza in maniera continuativa e rispondere alle domande degli utenti, la computer vision, un campo che studia i meccanismi per comprendere ad alto livello il contenuto di immagini o video, il Natural Language Processing (NLP), che si pone l'obiettivo di favorire la comprensione, l'analisi e la rappresentazione del linguaggio naturale, i recommendation system, per produrre raccomandazioni all'utente in base alle sue interazioni e interessi, l'Intelligent Data Processing (IDP), che è la classe più ampia in termini di applicazioni e vi rientrano tutte le soluzioni che prevedono

l’estrapolazione di informazioni a partire da dati e, infine, soluzioni fisiche come veicoli autonomi, robot autonomi e oggetti intelligenti. I campi di applicazione possono essere per esempio quello medico, delle infrastrutture e trasporti, industriale, della filiera agricola e alimentare, delle amministrazioni pubbliche e dei servizi.

Il machine learning è un campo (o un’applicazione) dell’intelligenza artificiale che si basa sulla capacità dei sistemi di apprendere in maniera automatica a partire dai dati senza essere esplicitamente programmati, migliorando le proprie prestazioni dopo lo svolgimento di un compito o il completamento di un’azione. Infatti, gli algoritmi di machine learning possono aumentare la propria efficienza modificando i propri processi per adattarli ai dati che analizzano tramite strategie matematico-computazionali, senza l’utilizzo di equazioni e modelli matematici predeterminati.

Il recente e repentino sviluppo del machine learning è dovuto fondamentalmente a due aspetti abilitanti, quali la disponibilità di grandi quantità di dati e la vasta capacità di calcolo che è fornita dagli strumenti oggi disponibili.

Una delle principali tecniche di machine learning è il deep learning, ambito nel quale le tecniche di apprendimento automatico si basano su reti neurali artificiali: strutture costituite di unità elementari chiamate neuroni o nodi, collegate fra loro e organizzate in diversi strati con l’obiettivo di simulare l’architettura del cervello umano. Il deep learning ha infatti lo scopo di automatizzare parti del processo di elaborazione dei dati, eliminando alcune delle azioni umane necessarie dagli algoritmi standard di machine learning. Questo procedimento di ottimizzazione automatica è attuabile in virtù della capacità delle reti neurali artificiali di modificare i valori dei propri parametri in funzione dei dati elaborati.

In conclusione, il machine learning è un modo per realizzare l’intelligenza artificiale tramite l’apprendimento automatico e il deep learning è un campo del machine learning, che si basa sulla riproduzione della capacità di pensiero del cervello umano utilizzando reti neurali artificiali.

3.2. Algoritmi di machine learning

Gli algoritmi di machine learning vengono suddivisi in base alla modalità di apprendimento che richiedono: in particolare, l'approccio può essere quello di apprendimento supervisionato, non supervisionato e per rinforzo [19].

L'apprendimento supervisionato prevede l'apprendimento a partire da dati etichettati, cioè coppie di input-output, al fine di capire la relazione che vi intercorre ed utilizzarla per fornire l'output non noto a priori per un generico input. L'implementazione di questi algoritmi avviene fondamentalmente tramite due fasi: la prima è detta fase di apprendimento ed è la fase nella quale si fornisce all'algoritmo una prima parte del dataset, detta training set, costituita dalle coppie input-output e utilizzata per trovarne correlazioni e modellizzazione. La seconda fase è quella di test e prevede l'applicazione dell'algoritmo alla seconda parte del dataset, detta validation set, al fine di confrontare l'output ottenuto dall'algoritmo con quello corretto e valutare l'accuratezza del sistema. Gli algoritmi di machine learning tramite apprendimento supervisionato vengono utilizzati nella risoluzione di due macrocategorie di problemi: problemi di classificazione e problemi di regressione. Nel primo caso l'output assume valori discreti, mentre nel secondo caso assume valori continui. I problemi di classificazione si dividono a loro volta in problemi di classificazione binaria, nei quali l'output può assumere due valori (0 – 1) e problemi di classificazione multi-classe, nei quali l'output può assumere un numero di valori pari a quello delle classi del problema.

L'apprendimento non supervisionato prevede invece che l'apprendimento sia applicato a dati non etichettati o non strutturati, che devono essere analizzati dall'algoritmo stesso al fine di indentificarvi relazioni o pattern. Un esempio di tecnica che si basa su questo tipo di apprendimento è il clustering o analisi dei gruppi, che prevede appunto la selezione e il raggruppamento di elementi omogenei all'interno di un insieme eterogeneo di dati.

Infine, l'apprendimento per rinforzo prevede che durante l'apprendimento il sistema interagisca con un ambiente dinamico all'interno del quale deve

raggiungere un obiettivo. Man mano che l'algoritmo esplora il dominio del problema riceve dei riscontri (feedback) in termini di ricompense o punizioni, al fine di essere indirizzato verso il goal.

3.2.1. Algoritmi di apprendimento supervisionato

Come detto nella sezione precedente, gli algoritmi di apprendimento supervisionato permettono la risoluzione di problemi di classificazione o di regressione e, a seconda dell'obiettivo, richiedono una diversa implementazione.

Per semplicità, in questo elaborato sono trattati algoritmi applicati alla classificazione binaria. Il dataset risulta quindi essere un insieme di coppie input-output e i possibili valori di output sono 0 – 1.

I principali algoritmi di questa categoria sono: Logistic Regression (LR), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Decision Tree, Random Forest e Reti Neurali Artificiali (ANN). Ogni algoritmo ha una sua predisposizione a determinati campi applicativi, varie tipologie di dati e strutture e, infine, la tendenza ad avere prestazioni migliori con l'utilizzo di più o meno campioni in fase di apprendimento.

L'obiettivo di questo elaborato è infatti quello di indagare quali algoritmi si prestino meglio al dataset di interesse, caratterizzato da misurazioni sui giunti delle grandezze di interesse e dalla label riguardo l'eventuale presenza di un guasto.

3.2.1.1. Logistic Regression (LR)

L'algoritmo di regressione logistica è un tipo di modello statistico basato sull'utilizzo della funzione logistica o sigmoidea, rappresentata in figura 3.1. La funzione sigmoidale associa ai valori reali di input un valore di output compreso fra 0 e 1.

Considerando un dataset di training composto da n vettori X_i di m elementi $x_{i,1}, \dots, x_{i,m}$, l'algoritmo individua un vettore peso W_i da associare al vettore X_i e calcola il valore z ottenuto dal prodotto del vettore di ingresso e del vettore peso. Il vettore $Z = \{z_1, \dots, z_n\}$ viene convertito in vettore di output tramite la funzione sigmoidale. Per la scelta dei valori dei vettori peso si utilizza il metodo statistico della massima verosimiglianza.

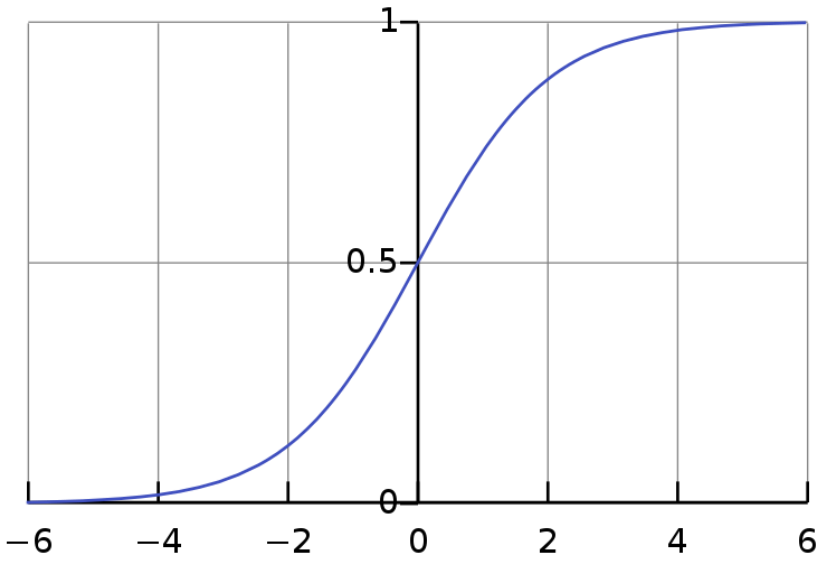


Figura 3.1 – Funzione sigmoidale. Estratto da https://netai.it/guida-rapida-alle-funzioni-di-attivazione-nel-deep-learning/#page-content

3.2.1.2. Support Vetor Machine (SVM)

L'algoritmo Support Vector Machine si basa sulla costruzione di un iperpiano ottimo che riesca a classificare correttamente i dati. Nel caso di classificazione binaria, questo iperpiano corrisponde ad una retta che separa il piano bidimensionale in due parti, equivalenti alle due classi di output.

L'iperpiano ottimo è quello che va a massimizzare la distanza, detta margine, fra gli elementi più vicini appartenenti alle due diverse classi. Gli elementi più vicini sono detti vettori di supporto e sono

appunto gli elementi più prossimi alla zona di separazione delle due classi, cioè i valori classificabili con maggiore difficoltà.

La rappresentazione dell'iperpiano e la massimizzazione del margine per la scelta ottimale del confine decisionale sono mostrati in figura 3.2.

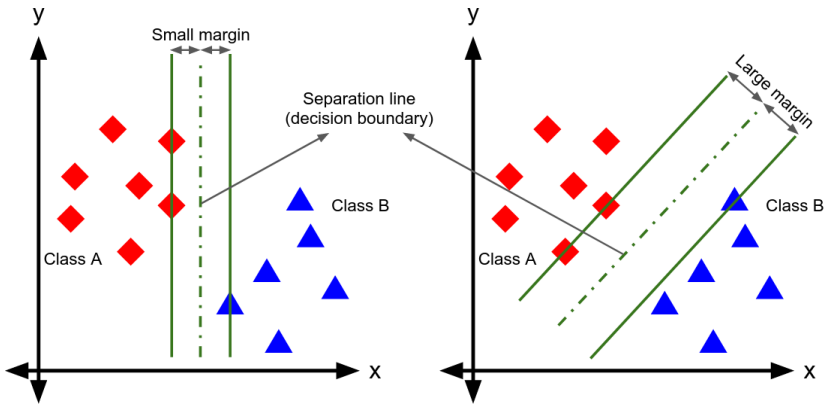


Figura 3.2 – Algoritmo SVM per classificazione binaria. Estratto da https://medium.com/it-paragon/grid-search-f24a73a8a0ac

3.2.1.3. K-Nearest Neighbors (KNN)

L'algoritmo K-Nearest Neighbors classifica i dati in base alla loro somiglianza e distanza da quelli precedenti, assumendo che dati simili saranno collocati in posizioni vicine nel piano di rappresentazione.

Il labeling di nuovi dati viene effettuato cercando all'interno del training set le K istanze più simili (i K vicini) e assegnando come valore di output quello con la frequenza maggiore all'interno di questo gruppo di K elementi.

Il calcolo della distanza fra i punti può essere svolto in modi diversi, utilizzando per esempio quella euclidea, di Manhattan, di Minkowski o di Hamming.

In questo elaborato si è scelto per semplicità di considerare come distanza quella euclidea e di fissare il numero di vicini a cinque.
Un esempio del funzionamento dell'algoritmo per classificazione binaria è mostrato in figura 3.3.

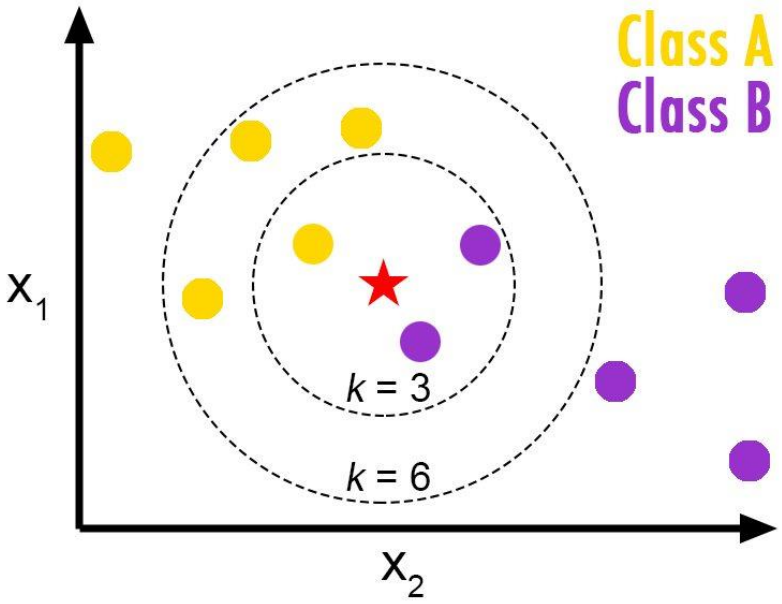


Figura 3.3 – Algoritmo KNN per classificazione binaria. Estratto da https://matlab1.com/knn-classifier/

3.2.1.4. Decision Tree

Il Decision Tree è un algoritmo che si basa sulla costruzione di un albero grafico di decisione, dove i nodi rappresentano i punti nei quali i dati vengono separati e le foglie rappresentano i risultati intermedi o finali di tale separazione. Ogni nodo è una funzione condizionale che verifica l'esistenza di una proprietà per il dato che si sta valutando. Il processo consiste quindi in una sequenza di test che inizia dal nodo iniziale più in alto e procede verso il basso.
L'obiettivo dell'algoritmo è quello di trovare la funzione di split ottimale, cioè quella che ad ogni passo nell'albero di decisione divide i dati in due gruppi che siano internamente il più omogenei possibili, in modo da minimizzare la difficoltà di classificazione.

Tipicamente la scelta della migliore funzione di separazione è fatta sulla base di diverse metriche e per i problemi di classificazione quelle più utilizzate sono entropia e indice di Gini.
Una generica rappresentazione della struttura di un Decision Tree è mostrata in figura 3.4.

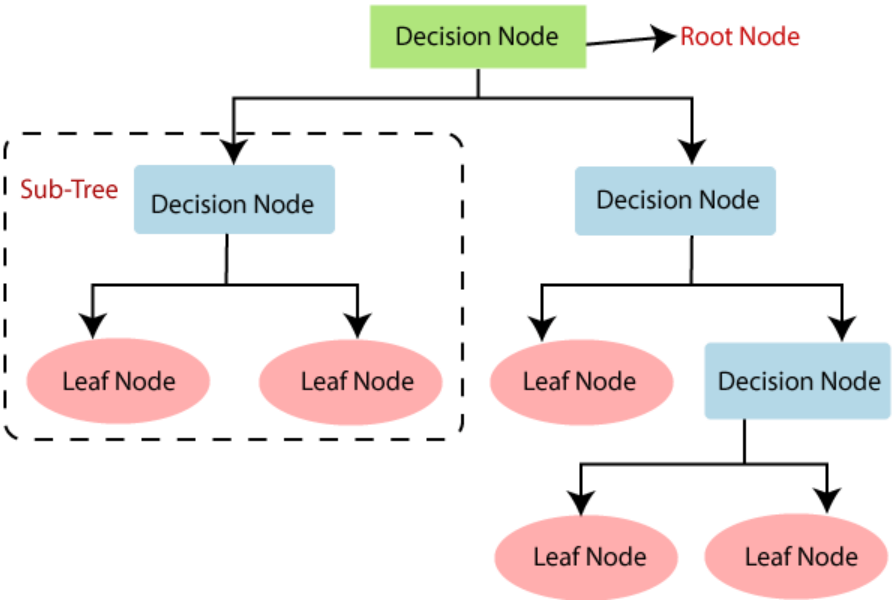


Figura 3.4 – Algoritmo Decision Tree per classificazione binaria. Estratto da https://www.devops.ae/decision-tree-classification-algorithm/

3.2.1.5. Random Forest

L'algoritmo Random Forest è ottenuto dall'aggregazione di più alberi di decisione tramite bagging. Il bagging è una tecnica di machine learning basata su apprendimento ensable, un tipo di apprendimento che utilizza modelli multipli per aumentare le prestazioni predittive. Il bagging prevede infatti l'utilizzo di più classificatori, aventi tutti lo stesso peso nell'ottenimento del risultato comune e ottenuti tramite addestramento su diversi dataset, ottenuti dal dataset iniziale tramite campionamento casuale con rimpiazzo (bootstrap).

Nel caso dell'algoritmo Random Forest, il risultato finale è ottenuto scegliendo quello con maggiore frequenza fra tutti quelli prodotti da tutti gli alberi che compongono la foresta. In figura 3.5 è riportata una schematizzazione di tale algoritmo.

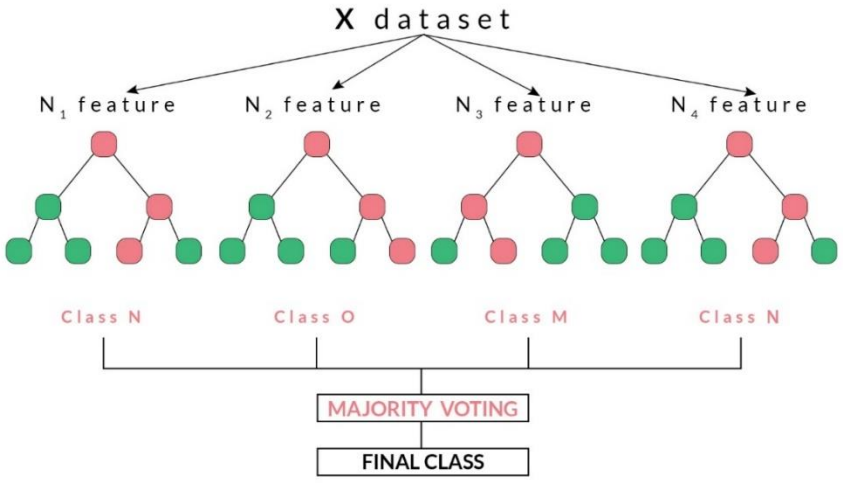


Figura 3.5 Random Forest. Estratto da https://cnvrg.io/random-forest-regression/

3.2.1.6. Artificial Neural Network (ANN)

La rete neurale artificiale (ANN) è una struttura costituita da unità elementari, dette nodi o neuroni. Queste unità funzionali sono organizzate in strati (strutture verticali) e, a seconda del numero di strati, la rete è detta monostrato o multistrato. Nel caso di reti multistrato, tutti i neuroni di uno strato sono connessi con tutti i neuroni di quello successivo e, a seconda di come le informazioni si propagano, le reti neurali vengono suddivise in reti feedforward o reti recurrent.

Ogni neurone è caratterizzato da una funzione di attivazione, che rappresenta la relazione fra segnale di ingresso e segnale di uscita del neurone, e da un valore di bias che realizza uno shift sul valore di ingresso del neurone. Gli input del neurone, eccetto per il primo

strato di ingresso, sono gli output dei neuroni dello strato precedente moltiplicati per dei pesi, che rappresentano l'intensità delle connessioni fra i neuroni.

L'obiettivo di una rete neurale artificiale è quello di modellizzare la relazione ingresso uscita, variando i valori dei propri pesi tramite opportune tecniche di apprendimento nella fase di training dell'algoritmo.

Per semplicità in questo elaborato tratteremo l'algoritmo Multi-layer Perceptron (MLP), la cui struttura è rappresentata in figura 3.6. Il percettrone multistrato è una rete multistrato feedforward con funzione di attivazione lineare a gradino e tecnica di apprendimento di error backpropagation, che prevede ad ogni iterazione l'aggiornamento dei valori dei pesi in modo da minimizzare l'errore ottenuto dalla rete.

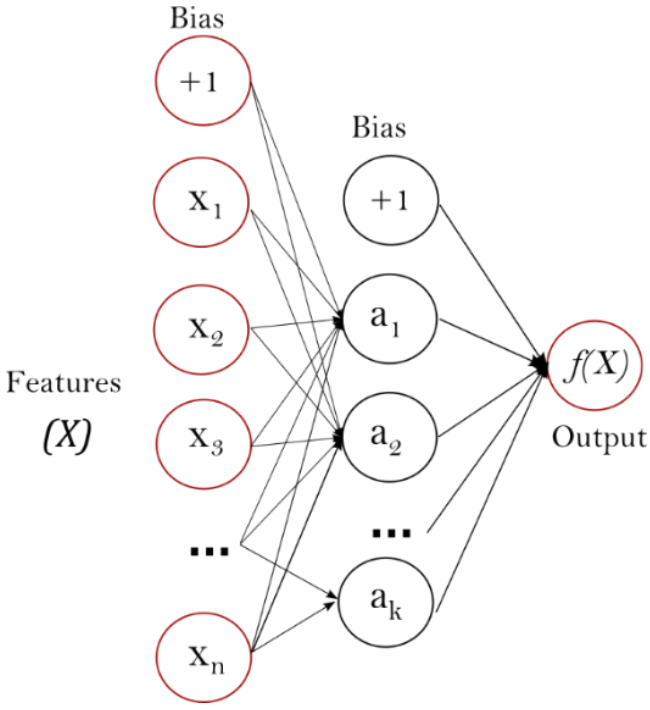


Figura 3.6 Struttura della rete neurale artificiale Multi-Layer Perceptron. Estratto dahttps://scikit-learn.org/stable/modules/neural_networks_supervised.html

3.2.2. Metriche di valutazione

Le previsioni fornite dagli algoritmi di machine learning hanno tutte un carattere probabilistico e un modello di questo tipo non potrà mai produrre risultati sempre corretti. È quindi importante che ne siano misurate le prestazioni, al fine di ottenere un algoritmo che fornisca esiti il più corretti possibili.

A tal proposito esistono diverse metriche utilizzabili per quantificare l'esattezza di un modello e le sue effettive prestazioni, quali accuratezza, matrice di confusione, precisione e recall.

3.2.2.1. Accuratezza

L'accuratezza è definita come il rapporto fra il numero di previsioni corrette e il numero totale di campioni nel validation set. È quindi un numero compreso fra 0 e 1 e spesso può essere anche espressa in percentuale.

Questa metrica è sicuramente la prima da prendere in considerazione, in quanto fornisce un'indicazione chiara e facilmente valutabile sulle prestazioni del modello. D'altra parte non tiene però conto della natura dei risultati del modello, ma considera soltanto quanti di questi siano corretti, indipendentemente dal loro valore effettivo e da quello predetto.

3.2.2.2. Matrice di confusione

La matrice di confusione, a differenza dell'accuratezza, valuta i risultati in base al loro valore corretto e a quello effettivamente predetto. In particolare, si possono identificare quattro categorie di risultati: vero positivo, vero negativo, falso positivo, falso negativo. Un risultato è vero positivo (o vero negativo) se nel validation set è

etichettato come positivo (o negativo) e viene correttamente predetto dal modello, mentre un falso positivo (o falso negativo) è un risultato che viene predetto positivo (o negativo), ma in realtà è negativo (o positivo). I falsi positivi sono quindi i così detti errori di prima specie, mentre i falsi negativi sono quelli di seconda specie.

Una generica matrice di confusione può essere rappresentata come mostrato in tabella 3.1. Le predizioni corrette sono quelle evidenziate in verde, quindi la somma di risultati veri negativi (TN) e veri positivi (TP), mentre quelle errate sono evidenziate in rosso e sono la somma di falsi negativi (FN) e falsi positivi (FP). A partire dalla matrice di confusione, l'accuratezza può quindi essere calcolata come mostrato nella formula 3.1.

Tabella 3.1 – Generica matrice di confusione

		Valori predetti	
		Negative (0)	**Positive (1)**
Valori reali	**Negative (0)**	True Negative (TN)	False Positive (FP)
	Positive (1)	False Negative (FN)	True Positive (TP)

$$Accuratezza = \frac{TP + TN}{TP + TN + FP + FN} \qquad 3.1$$

3.2.2.3. Precisione e recall

A partire dalle definizioni di vero positivo, vero negativo, falso positivo e falso negativo, le metriche di precisione e recall sono espresse rispettivamente nelle formule 3.2 e 3.3. Entrambe queste metriche possono assumere valori compresi fra 0 e 1.

$$Precisione = \frac{TP}{TP + FP} \tag{3.2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.3}$$

La precisione è quindi la frazione di risultati predetti come positivi e che sono correttamente positivi; infatti, è il rapporto fra il numero di positivi predetti correttamente e il numero di totale di positivi predetti, ma non necessariamente corretti. Questa metrica si differenzia quindi dall'accuratezza perché, se quest'ultima esprime la vicinanza del modello all'effettivo risultato, la precisione quantifica invece la consistenza del risultato, ignorando l'effettivo raggiungimento dell'obiettivo.

La metrica di recall rappresenta invece la frazione di positivi che vengono identificati tali dal modello; infatti, è il rapporto fra il numero di positivi predetti correttamente e il numero totale dei risultati effettivamente positivi.

4. Stato dell’arte

Come espresso nei capitoli precedenti, i giunti sono la principale causa di guasto nelle reti in cavo di media tensione e la loro riparazione è un intervento oneroso in termini di tempo e denaro, risultando quindi essere componenti importanti da monitorare al fine della prevenzione dei guasti nella rete di distribuzione.

Per questi motivi in letteratura possono essere trovati vari studi riguardo ai giunti, ai loro modi di guasto e alla relazione che intercorre fra questi ultimi ed i diversi parametri considerati caratteristici. A seconda del loro scopo, questi articoli scientifici possono essere fondamentalmente suddivisi in due macrogruppi: analisi dei fenomeni fisici che hanno portato al guasto e monitoraggio online e offline del giunto. Per quanto riguarda l’analisi dei fenomeni fisici che possono provocare un guasto in questi componenti, questa è stata trattata nel paragrafo 2.3, mentre i parametri utili al monitoraggio online e offline dei giunti sono stati discussi nel paragrafo 2.4.

In merito all’utilizzo di tecnologie di intelligenza artificiale per l’obiettivo di monitoraggio sopra citato, si è proceduto ad una valutazione delle tecniche presenti in letteratura tramite lo svolgimento di una Revisione Sistematica della Letteratura (SLR).

4.1. Systematic Literature Review (SLR)

La SLR è un tipo di revisione realizzato attraverso un approccio rigoroso, articolato in diverse fasi e con l’obiettivo di ridurre i possibili errori sistematici e/o casuali. Una revisione tradizionale non è infatti esplicita nel definire i metodi di selezione, sintesi ed interpretazione utilizzati e di conseguenza risulta difficilmente riproducibile. La SLR prevede invece chiare definizioni di titolo, obiettivo, strategia di ricerca, criteri inclusivi e/o esclusivi e una lista esaustiva di tutti gli studi trovati, risultando quindi essere un’analisi trasparente e riproducibile.

Le revisioni sistematiche della letteratura sono necessarie nella valutazione di un elevato numero di pubblicazioni e ricerche scientifiche, evitando bias di pubblicazione e sintetizzando quindi un risultato il più oggettivo possibile, eliminando anche eventuali conflitti di interesse.
Generalmente le fasi che costituiscono una SLR sono le seguenti: stesura di un protocollo di ricerca, definizione dei criteri di inclusione ed esclusione, ricerca bibliografica, valutazione della qualità degli studi e riassunto critico dei risultati ottenuti. La stesura del protocollo di ricerca prevede l'identificazione di obiettivo e research questions, mentre la ricerca bibliografica consiste nell'applicare la stessa strategia a diversi database.

4.2. Metodo

L'obiettivo della revisione sistematica della letteratura trattata in questo elaborato è quello di trovare quali algoritmi di machine learning siano già stati usati per svolgere monitoraggio online e/o offline dei giunti, al fine di valutarli in relazione al caso di studio proposto.

4.2.1. Research question

L'obiettivo sopra detto può essere espresso dalla seguente research question, utilizzata per definire il protocollo di ricerca della SLR realizzata.

(RQ) Quali algoritmi di machine learning sono stati utilizzati nella prevenzione dei guasti nei giunti per cavi elettrici?

4.2.2. Ricerca bibliografica

La ricerca bibliografica è stata svolta manualmente e consultando i seguenti motori di ricerca e database:

- Scopus - https://www.scopus.com/home.uri
- IEEEXplore - https://ieeexplore.ieee.org/Xplore/home.jsp
- ScienceDirect - https://www.sciencedirect.com/

Ognuna di queste fonti è stata interrogata tramite le seguenti combinazioni di parole chiave, articolate a partire dalla research question RQ.

(KW1)	Failure prediction machine learning cable joint
(KW2)	Aging condition assessment machine learning cable joint
(KW3)	Condition-based maintenance machine learning cable joint
(KW4)	Condition monitoring diagnostics machine learning cable joint
(KW5)	Health index analysis machine learning cable joint

4.2.3. Criteri di inclusione ed esclusione

Per ogni ricerca sono stati presi in considerazione i primi 100 risultati e sono stati esclusi gli articoli pubblicati prima del 1° gennaio 2018 e quelli scritti in lingua diversa dall'inglese.
In un secondo momento, sono stati esaminati abstract e testi integrali degli articoli inizialmente trovati, per escludere quelli che non trattavano i giunti elettrici e, successivamente, quelli che non citavano esplicitamente algoritmi di machine learning e, infine, sono stati rimossi gli articoli ripetuti.

4.2.4. Analisi dei dati

Inizialmente sono stati selezionati 165 articoli, distribuiti come mostrato nella tabella 4.1.

Tabella 4.1 – Risultati iniziali delle ricerche nei rispettivi motori di ricerca/database

No	Motore di ricerca / database	Risultati iniziali
1	Scopus	71
2	IEEEXplore	63
3	ScienceDirect	31
Numero totale di articoli		165

A seguito dell'applicazione dei criteri di esclusione descritti al paragrafo 4.2.3, quali la rimozione di articoli non inerenti ai giunti e nei quali non venivano citati specifici algoritmi di machine learning, sono stati ottenuti 40 paper, così come mostrato in tabella 4.2.

Tabella 4.2 – Risultati dell'applicazione dei criteri di esclusione

No	Motore di ricerca / database	Risultati
1	Scopus	18
2	IEEEXplore	19
3	ScienceDirect	3
Numero totale di articoli		40

Infine, in tabella 4.3 sono mostrati gli articoli selezionati dopo la rimozione dei duplicati.

Tabella 4.3 – Dettagli degli articoli finali selezionati

ID	Anno	Autori	Titolo
1	2022	Chien-Kuo Chang et al. [20]	Application of Pulse Sequence Partial Discharge Based Convolutional Neural Network in Pattern Recognition for Underground Cable Joints
2	2022	Marco Bindi et al. [21]	Applications of Machine Learning Techniques for the Monitoring of Electrical Transmission and Distribution lines
3	2021	Chien-Kuo Chang and Bharath Kumar Boyanapalli [22]	Assessment of the Insulation Status Aging in Power Cable Joints Using Support Vector Machine
4	2021	P. L. Lewin et al. [23]	Avoiding Medium Voltage Cable Joint Failure: Development of a Real-Time Prognostic Tool
5	2019	Chien-Kuo Chang et al. [24]	Decision Tree Rules for Insulation Condition Assessment of Pre-molded Power Cable Joints with Artificial Defects
6	2022	Sara Mantach et al. [25]	Deep Learning in High Voltage Engineering: A Literature Review
7	2021	Jeong H. Choi et al. [26]	Detection of Series Faults in High-Temperature Superconducting DC Power Cables Using Machine Learning

8	2020	Nathalie Morette et al. [27]	Feature extraction and ageing state recognition using partial discharges in cables under HVDC
9	2021	Rakesh Sahoo and Subrata Karmakar [28]	Health Index Prediction of Underground Cable System using Artificial Neural Network
10	2022	Joel Yeo et al. [29]	Identification of Partial Discharge Through Cable-Specific Adaption and Neural Network Ensemble
11	2021	Wong Jee Keen Raymond et al. [30]	Noise invariant partial discharge classification based on convolutional neural network
12	2021	Norfadilah Rosle et al. [31]	Partial Discharges Classification Methods in XLPE Cable: A Review
13	2021	Xiaohua Zhang et al. [32]	Review on Detection and Analysis of Partial Discharge along Power Cables
14	2022	Mohammed Omer Alhusin et al. [33]	Weather and Seasonal Effects on Medium Voltage Underground Cable Joint Failures

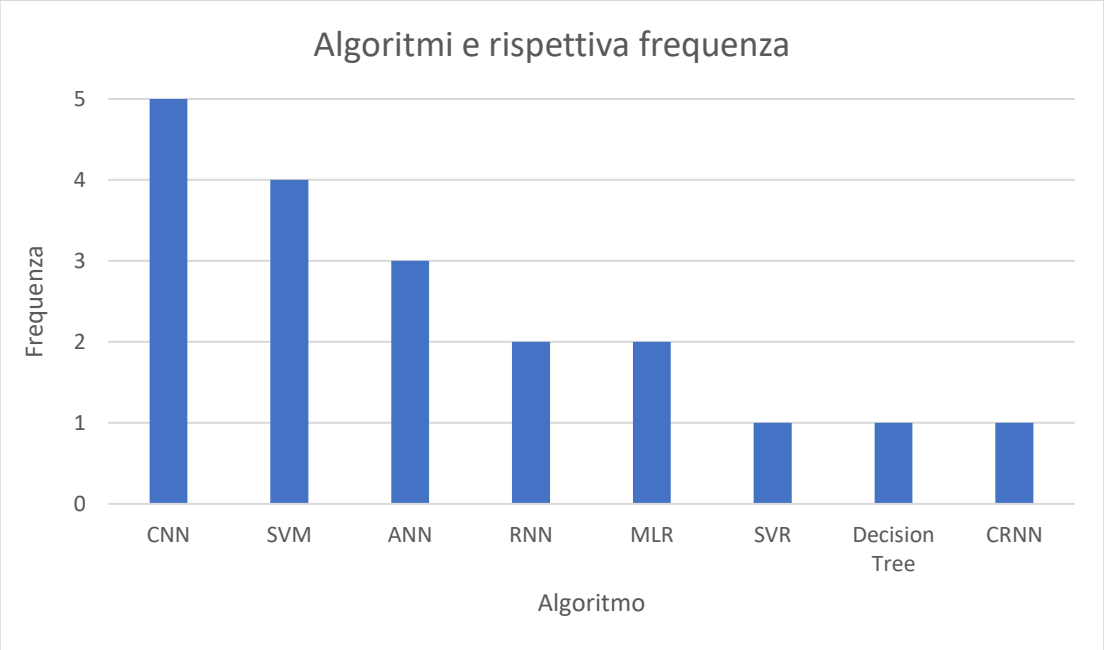
4.3. Risultati

A partire dagli articoli scientifici ottenuti dalla revisione sistematica della letteratura precedentemente descritta, sono stati trovati gli algoritmi illustrati nella tabella 4.4 e nel grafico 4.1.

Tabella 4.4 – Classifica degli algoritmi citati negli articoli selezionati e rispettive frequenze

No	Algoritmi citati negli articoli selezionati	Frequenza
1	Convolutional Neural Network (CNN)	5
2	Support Vector Machine (SVM)	4
3	Artificial Neural Network (ANN)	3
4	Recurrent Neural Network (RNN)	2
5	Multiple Linear Regression (MLR)	2
6	Support Vector Regression (SVR)	1
7	Decision Tree	1
8	Convolutional Recurrent Neural Network (CRNN)	1

Grafico 4.1 – Algoritmi citati e rispettiva frequenza nei 14 articoli selezionati



Dai risultati sopra mostrati si può notare come le reti neurali convoluzionali (CNN) siano le più utilizzate. Questa specifica tipologia di reti neurali feedforward è ispirata all'organizzazione della corteccia visiva ed è principalmente usata nel riconoscimento di immagini. In generale, queste reti neurali trovano applicazione in diversi campi e, date le loro proprietà matematiche e computazionali, riescono ad adattarsi a svariate tipologie di dataset e fornire ottimi risultati.

Riguardo all'algoritmo SVM, questo risulta largamente utilizzato sia per problemi di classificazione sia per quelli di classificazione. Inoltre, in letteratura sono disponibili svariate applicazioni di questo metodo, in campo elettrico e non. Questa fruibilità, in aggiunta all'alta efficienza offerta da questo algoritmo, lo rendono una delle scelte più popolari nel campo del machine learning.

Le generiche reti neurali e la loro variante di RNN, per i motivi già detti riguardo alle specifiche CNN, sono considerate un ottimo strumento per l'implementazione di tecnologie di intelligenza artificiale perché sono strutture estremamente performanti. Dall'altro lato, data la loro elevata complessità,

necessitano di studi approfonditi per essere usate nel modo più efficace possibile.

Per quanto riguarda l'algoritmo MLR, questo è uno dei più semplici e più conosciuti algoritmi di machine learning per la risoluzione di problemi di regressione. Dato che in questo elaborato si è scelto per semplicità di focalizzarsi sulla classificazione, questo algoritmo non è stato ulteriormente approfondito.

Infine, l'algoritmo SVR può essere considerato come un tipo del più generico SVM, il Decision Tree è decisamente un algoritmo di facile comprensione, seppur abbia una struttura complessa, e dalle buone prestazioni, soprattutto nella classificazione binaria. Infine, le reti neurali convoluzionali e ricorrenti (CRNN), sono un'ulteriore tipologia delle generiche ANN già trattate in precedenza.

5. Descrizione dei test

Il caso di studio proposto in questo elaborato è quello di implementare e testare algoritmi di machine learning opportunamente selezionati per l'analisi di misurazioni effettuate sui giunti elettrici, al fine di valutarne l'integrità ed effettuare manutenzione predittiva sulla base dei risultati ottenuti dall'algoritmo.

A tale scopo, sulla base di considerazioni sullo stato dell'arte e dall'osservazione di giunti in servizio o in laboratorio, sono state selezionate alcune delle grandezze caratteristiche di questi componenti.

Per quanto riguarda invece gli algoritmi di machine learning, a partire da considerazioni circa i risultati ottenuti dalla revisione sistematica della letteratura e da ulteriori valutazioni, ne sono stati selezionati sei, elencati qui di seguito:

- Logistic Regression (LR)
- K-Nearest Neighbors (KNN)
- SVM
- ANN
- Decision Tree
- Random Forest

In particolare, l'algoritmo LR è stato scelto perché risulta essere uno degli algoritmi più semplici nel campo del machine learning, ma comunque caratterizzato da estrema efficienza; è quindi comunemente considerato un ottimo punto di partenza, per poi procedere alla valutazione di algoritmi più complessi.

Allo stesso modo, anche l'algoritmo KNN risulta essere di facile comprensione, seppur porti a risultati più competitivi rispetto a quello precedentemente citato. Di conseguenza, si è scelto di valutarne l'adattabilità a questo caso di studio, anche a seguito di considerazioni in merito all'applicabilità di questo algoritmo a svariati tipi di dati e alla sua velocità di calcolo.

Per quanto riguarda l'algoritmo SVM e il Decision Tree, questi sono stati selezionati perché presenti e già trattati nello stato dell'arte, insieme alle diverse tipologie di reti neurali artificiali: in merito a queste, si è scelto di implementare la tipologia più

semplice di rete neurale (ANN), senza approfondire quelle convoluzionali (CNN), ricorrenti (RNN) e l'ibrido fra le due (CRNN). In particolare è stato scelto di implementare il Percettrone Multi Livello, descritto nel paragrafo 3.2.1.6.
Infine, si è incluso nei test anche l'algoritmo Random Forest, in quanto estensione del Decision Tree precedentemente citato.
Per l'implementazione degli algoritmi sopra detti, è stata utilizzata la libreria Scikit Learn (sklearn) in linguaggio Python, la cui documentazione è disponibile al seguente link: https://scikit-learn.org/stable/.

5.1. Test su dataset di pubblico dominio

Inizialmente si è proceduto a testare gli algoritmi su un dataset di pubblico dominio, al fine di capire il funzionamento della libreria Python utilizzata e avere un'idea iniziale delle sue prestazioni.
Il dataset utilizzato è Pima Indians Diabetes Database, le cui variabili sono svariati parametri medici e il cui target è 0-1, a seconda che la diagnosi di diabete sia positiva o negativa. Il dataset contiene 768 istanze e in tabella 5.1 ne sono mostrate le prime righe con l'aggiunta di intestazione.

Tabella 5.1 – Intestazione e prime righe del dataset Pima Indians Diabetes Database. Estratto da https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

Numero di gravidanze	Glucosio	Pressione sanguigna	Spessore della pelle	Insulina	BMI	Funzione del diabete	Età	Risultato
6	148	72	35	0	33,6	0,627	50	1
1	85	66	29	0	26,6	0,351	31	0
8	183	64	0	0	23,3	0,672	32	1
1	89	66	23	94	28,1	0,167	21	0
0	137	40	35	168	43,1	2,288	33	1

Tale dataset è stato scelto per la sua somiglianza a quello di interesse per questo caso di studio, essendo un dataset caratterizzato da variabili numeriche per classificazione binaria con esito numerico (0-1).

Per quanto riguarda la fase implementativa, si è scelto di non modificare i parametri tipici degli algoritmi della libreria, al fine di concentrarsi sulle effettive prestazioni in relazione al dataset utilizzato. Inoltre, per la fase di training è stato usato l'80 % del dataset, mentre il restante 20 % è stato usato per la fase di test.

Per ogni algoritmo sono state poi calcolate le metriche di accuratezza, matrice di confusione, precisione e recall, trattate nel paragrafo 3.2.2 e i cui risultati sono riportati al paragrafo 6.1.

5.2. Generazione di dati sintetici

Al fine di indagare quali algoritmi siano applicabili nello svolgimento di manutenzione predittiva sui giunti, è necessario disporre di un dataset di

misurazioni sui componenti stessi, che, opportunamente fornite all'algoritmo, ne permettano l'apprendimento e il test. Data la mancanza di un numero sufficiente di misure reali su questi componenti, si è proceduto alla generazione sintetica di un dataset, permettendo quindi sia la valutazione degli algoritmi in relazione ad uno specifico tipo di dataset, sia in relazione a specifiche caratteristiche del dataset stesso, quali numero di complessivo di campioni, numero di campioni per la fase di apprendimento e per la fase di test e, infine, distribuzione dei dati.
In un primo momento è stato quindi necessario selezionare le grandezze di interesse, cioè le variabili dell'algoritmo. Come descritto nel paragrafo 2.3, diverse cause possono portare al guasto dei giunti, che tipicamente si manifesta con il cedimento dell'isolante elettrico, e svariati parametri caratteristici sono coinvolti. In questo caso di studio si è scelto di porre l'attenzione su tre grandezze, quali corrente, tan delta e temperatura. Tale scelta è motivata considerando le misurazioni più significative ottenibili da un giunto e limitandone il numero a tre, per tenere conto delle difficolta di implementazione di un sistema distribuito di misura capace di acquisizioni multiple. A queste tre grandezze è stata aggiunta anche l'età dei giunti, misurata in anni di servizio.
Una volta selezionate le grandezze di interesse, ne sono stati fissati degli intervalli di variazione per la loro generazione sintetica e delle soglie al fine di ottenere dei criteri per determinare la presenza o meno di un guasto nel componente.
In particolare, in tabella 5.2 sono mostrati gli intervalli di variabilità delle grandezze di interesse: tali valori sono da intendere come limiti superiori e inferiori nel caso di generazione randomica a partire dall'ipotesi di distribuzione uniforme dei dati, mentre sono da considerare come limiti a tre-sigma nel caso di generazione randomica a partire dall'ipotesi di distribuzione normale, trattate nel dettaglio in seguito. La scelta degli intervalli di variabilità così fissati è motivata dalle seguenti ragioni: per quanto riguarda la temperatura, il range è stato deciso considerando che i giunti sono tipicamente installati sottoterra e la variazione della temperatura del suolo è tipicamente limitata, a differenza di quella che può avere l'aria. La variabilità del tan delta è stata fissata ipotizzando

una qualità media delle proprietà dell'isolante e per la corrente la scelta è stata fatta a partire da un generico datasheet di un cavo di media tensione, fissando come limite superiore il valore di portata massima e come limite inferiore zero. Infine, per scegliere l'intervallo degli anni di vita, l'età media di un componente elettrico è di 30 anni e questo valore è stato quindi selezionato come limite superiore.

Per quanto riguarda invece i criteri di guasto, nelle tabelle 5.3, 5.4 e 5.5 sono mostrate le soglie per poter aver un guasto, da eccedere rispettivamente da una quantità, due quantità o da tutte e tre le quantità. Per quanto riguarda invece gli anni di servizio, questi forniscono in ultima istanza l'esito positivo (1) o negativo (0) di guasto, a seconda che il tasso di guasto, ottenuto dalla curva a vasca da bagno implementata e mostrata in figura 5.1, sia rispettivamente superiore o inferiore al valore di 0.5.

Tabella 5.2 – Limiti superiori e inferiori di variazione delle grandezze di interesse

Quantità	**Limite inferiore**	**Limite superiore**
Temperatura (°C)	5	45
Tan delta (-)	10^{-5}	10^{-1}
Corrente (A)	0	300
Età (anni)	0	30

Tabella 5.3 – Limiti superiori che una quantità di un'istanza deve superare per avere un guasto

Quantità	**Limite superiore**
Temperatura (°C)	40
Tan delta (-)	10^{-2}
Corrente (A)	300

Tabella 5.4 – Limiti superiori che due quantità di un'istanza devono superare per avere un guasto

Quantità	Limite superiore
Temperatura (°C)	30
Tan delta (-)	10^{-3}
Corrente (A)	150

Tabella 5.5 – Limiti superiori che tutte e tre le quantità di un'istanza devono superare per avere un guasto

Quantità	Limite superiore
Temperatura (°C)	40
Tan delta (-)	10^{-2}
Corrente (A)	300

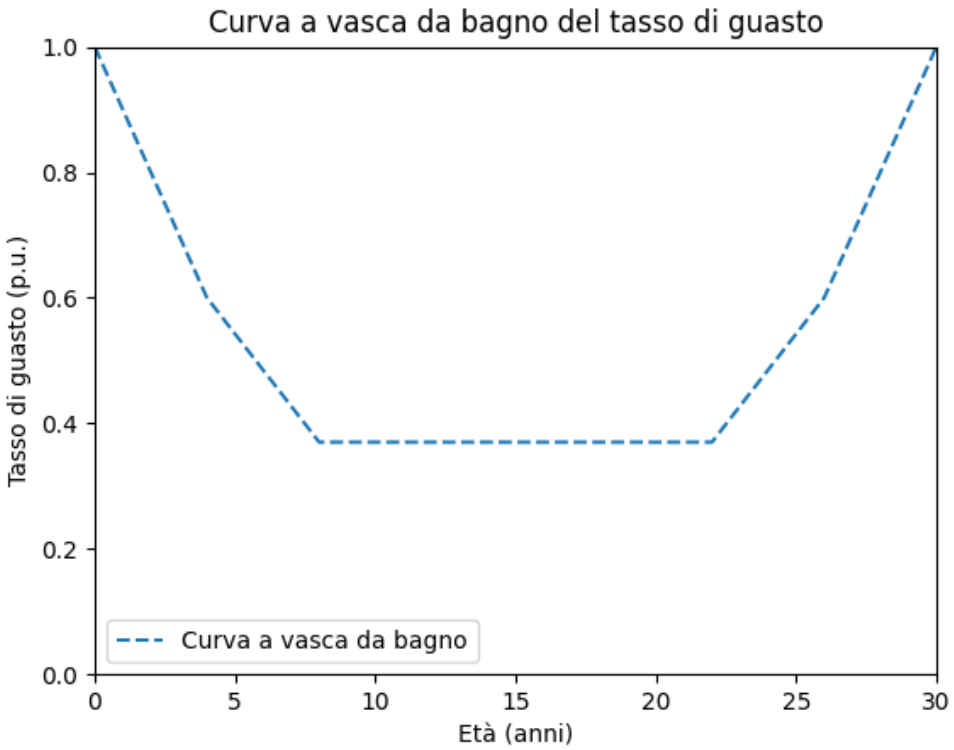


Figura 5.1 – Approssimazione della curva a vasca da bagno per la determinazione della presenza di guasto

Una volta stabiliti i criteri sopra citati, si è proceduto all'effettiva generazione del dataset, il cui procedimento è riassunto nello schema a blocchi rappresentato in figura 5.2.

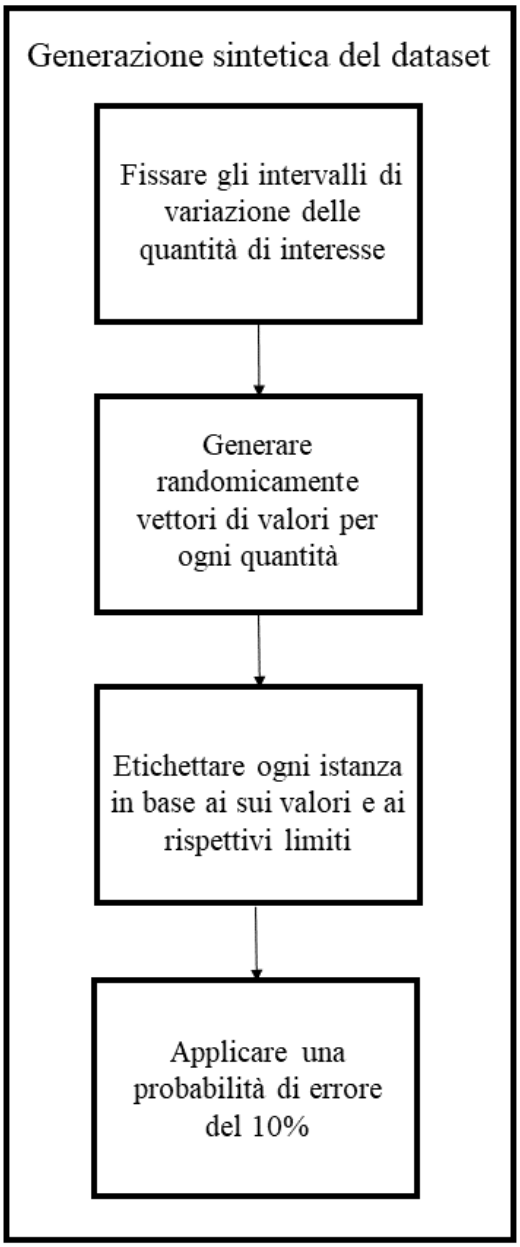


Figura 5.2 – Schema a blocchi della generazione sintetica del dataset

Per quanto riguarda le prime due fasi, gli intervalli decisi sono quelli precedentemente illustrati e la generazione casuale è stata effettuata utilizzando specifiche funzioni del modulo Random della libreria Numpy, ottenendo quattro vettori di valori, uno per ogni grandezza.

Per la terza fase, un quinto vettore è stato inizializzato e popolato di 0 o 1 a seconda dei valori delle grandezze e dei limiti imposti.

Infine, i risultati così ottenuti sono stati elaborati introducendo una probabilità del 10 % di essere sbagliati. In altre parole, un decimo dei risultati viene casualmente modificato da 0 a 1 o da 1 a 0, a seconda del proprio valore iniziale. Questa scelta è data dalla volontà di voler simulare l'incertezza che sempre influenza le misurazioni reali e che in questo caso di studio, data la generazione randomica, non era insita nei dati di ingresso. Non essendo questa nota, in questo studio preliminare è stata arbitrariamente posta pari al 10 %.

I cinque vettori così ottenuti sono stati inseriti come colonne in un file csv, ottenendo il dataset di interesse, di cui intestazione e un esempio delle prime righe sono mostrate in tabella 5.6.

Tabella 5.6 – Intestazione e prime righe di un esempio del dataset sinteticamente generato

Tan delta (-)	**Temperatura (°C)**	**Corrente (A)**	**Età (anni)**	**Risultato**
0,0045	9,170	238,421	17	0
0,049	23,725	132,801	25	0
0,080	18,085	123,281	6	0
0,036	14,876	235,686	25	0
0,070	29,843	23,033	14	0

5.3. Test su dataset sintetico

Analogamente ai test sul dataset di pubblico dominio precedentemente descritti, anche in questo caso si è proceduto al calcolo iniziale delle metriche di accuratezza, matrice di confusione, precisione e recall, i cui risultati sono riportati al paragrafo 6.3. Per questa valutazione iniziale si è ipotizzata una distribuzione uniforme delle grandezze di interesse e fissato il numero di

campioni per la fase di apprendimento pari a 1000 e quello per la fase di test pari a 100000. Per quanto riguarda la scelta del numero di istanze per la fase di test, la scelta è data dal fatto che un operatore di sistema distribuisce più di 10 migliaia di giunti per cavi elettrici all'anno, di conseguenza, avere informazioni su circa il 10 % di essi, è assolutamente accettabile. Riguardo alla scelta del numero di istanze per la fase di test, questo è stato fissato intenzionalmente alto per concentrarsi solo sulla valutazione del numero di campioni utilizzati per la fase di training.

Su questo dataset si è poi proceduto anche ad effettuare ulteriori simulazioni, quali la valutazione delle metriche sopra citate in relazione al numero di campioni nella fase di apprendimento, l'analisi della ripetibilità e l'ipotesi di distribuzione normale, di seguito descritti.

5.3.1. Numero di campioni nella fase di apprendimento

Durante la fase di generazione sintetica, non esiste nessun metodo teorico per determinare a priori il numero di campioni necessari durante la fase di apprendimento per ottenere una specifica accuratezza. La relazione fra quest'ultimo parametro e l'estensione del dataset dipende infatti dall'algoritmo utilizzato e dalle caratteristiche del dataset stesso.

Si è quindi proceduto al calcolo dell'accuratezza e delle metriche di precisione e recall al variare del numero di campioni forniti nella fase di apprendimento, al fine di osservarne la relazione e stabilire una soglia minima di campioni necessari per ottenere l'efficienza voluta.

Per ogni algoritmo sono state svolte sette simulazioni, variando il numero di campioni all'interno dell'intervallo da 100 a 10000 e tenendo costante e pari a 100000 il numero di campioni della fase di test, al fine di focalizzarsi sul numero di istanze dedicate all'apprendimento. Inoltre, si è ipotizzata una distribuzione uniforme dei dati. I risultati ottenuti sono illustrati al paragrafo 6.2.1.

5.3.2. Ripetibilità

Questo test ha l'obiettivo di assicurare la ripetibilità dell'algoritmo. Infatti, dato il carattere casuale del dataset, è importante valutare che, al variare dei valori sinteticamente generati che costituiscono le istanze di ingresso degli algoritmi, questi non cambino significativamente il loro risultato. A questo scopo, tramite uno script opportunamente implementato, per ogni algoritmo sono state eseguite 500 generazioni sintetiche di dataset e per ognuna di queste è stata calcolata l'accuratezza del risultato. Durante tutte le simulazioni, il numero di campioni per la fase di apprendimento è stato fissato a 10000 e quello della fase di test al valore di 100000. Inoltre, la distribuzione dei dati ipotizzata è quella uniforme.

I risultati ottenuti da questo test sono dettagliatamente descritti al paragrafo 6.2.2.

5.3.3. Distribuzione dei dati

Come specificato nei paragrafi precedenti, tutti i test sopra descritti sono stati svolti ipotizzando una distribuzione uniforme dei dati di ingresso.

Questa scelta è motivata dal fatto che la GUM [34], in mancanza di informazioni riguardo all'effettiva distribuzione dei dati, suggerisce l'utilizzo di quella uniforme, che risulta quindi essere una buona soluzione per caratterizzare i dati di ingresso. D'altra parte, se consideriamo l'applicazione reale del caso in esame, una distribuzione uniforme dei dati di ingresso coincide anche con il caso peggiore che si può avere.

Per i motivi sopra detti, questo test ha quindi l'obiettivo di valutare gli algoritmi quando i dati di ingresso sono caratterizzati da una distribuzione normale, i cui parametri sono stati illustrati nel paragrafo 5.2 e decisi in relazione agli intervalli di variabilità delle grandezze di interesse. Si fa presente che sono generate a partire da una distribuzione normale tutte le grandezze caratteristiche eccetto l'età dei giunti, che è ancora caratterizzata

da distribuzione uniforme. Questa decisione è presa in relazione al fatto che, data la caratteristica a curva a vasca da bagno di questa grandezza, utilizzare la distribuzione normale porterebbe ad una ingente riduzione dei casi guasti all'interno del dataset, già presenti in numero limitato rispetto a quelli non guasti.

Il test descritto al paragrafo 5.3.1 è stato quindi ripetuto utilizzando dati caratterizzati da una distribuzione normale e i risultati ottenuti sono riportati al paragrafo 6.2.3.

6. Risultati

In questo capitolo sono riportati i risultati ottenuti dall'esecuzione dei test sopra descritti, suddivisi in quelli ottenuti dal dataset di pubblico dominio Pima Indians Diabetes Database e quelli ottenuti dal dataset generato sinteticamente tramite la procedura descritta al paragrafo 5.2.

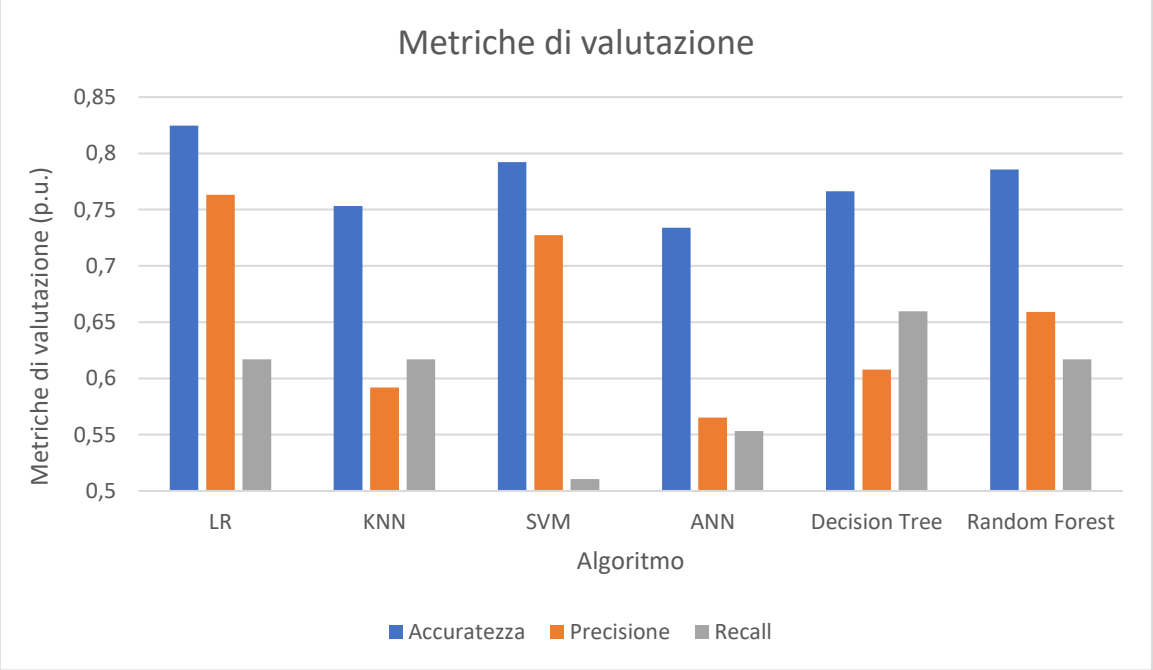
6.1. Test su dataset di pubblico dominio

In tabella 6.1 sono riportati i valori di accuratezza, precisione e recall ottenuti per ogni algoritmo implementato, anche rappresentati nell'istogramma 6.1.

Tabella 6.1 – Metriche di accuratezza, precisione e recall per i sei algoritmi implementati testati sul dataset di pubblico dominio

	Accuratezza	**Precisione**	**Recall**
LR	0,825	0,763	0,617
KNN	0,753	0,593	0,617
SVM	0,792	0,727	0,511
ANN	0,734	0,565	0,553
Decision Tree	0,766	0,608	0,660
Random Forest	0,786	0,659	0,617

Grafico 6.1 – Metriche di accuratezza, precisione e recall per i sei algoritmi implementati testati sul dataset di pubblico dominio



Le matrici di confusione per i sei algoritmi implementati sono invece mostrate nelle figure da 6.1 a 6.6.

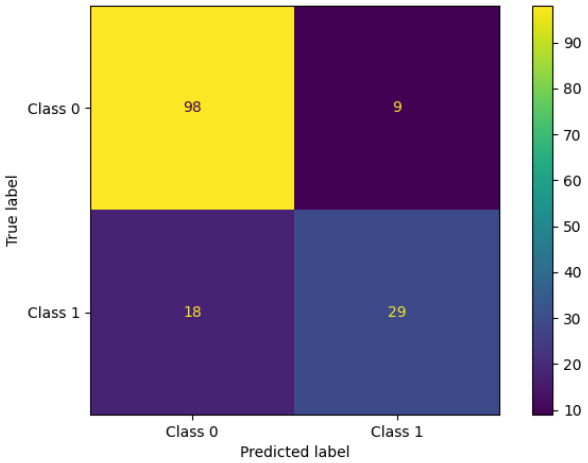


Figura 6.1 – Matrice di confusione dell'algoritmo LR testato sul dataset di pubblico dominio

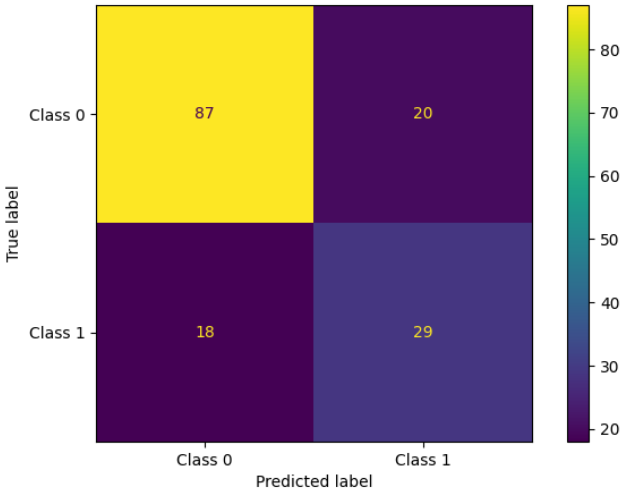


Figura 6.2 – Matrice di confusione dell'algoritmo KNN testato sul dataset di pubblico dominio

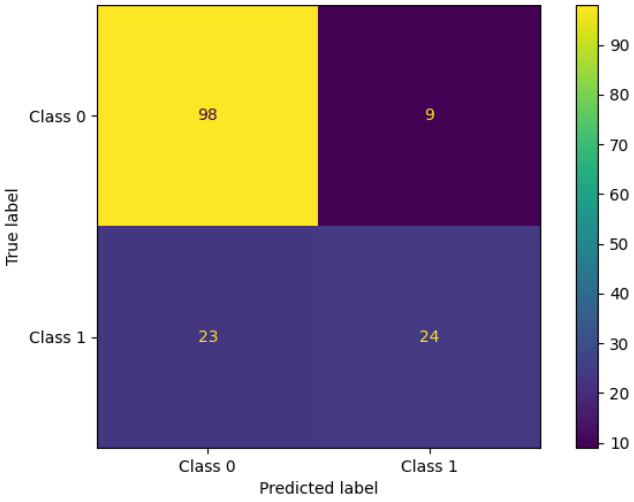


Figura 6.3 – Matrice di confusione dell'algoritmo SVM testato sul dataset di pubblico dominio

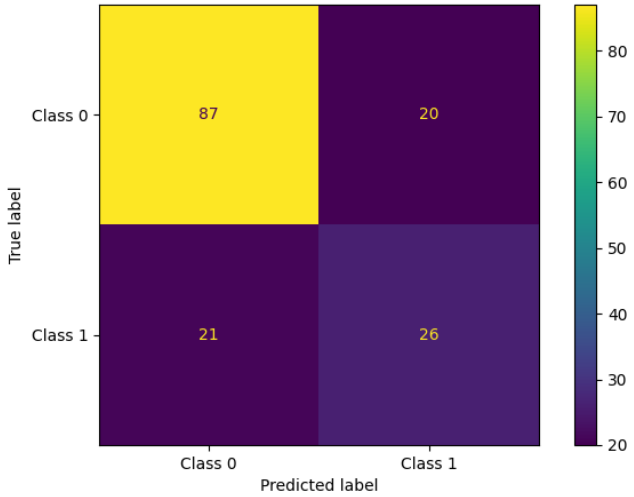


Figura 6.4 – Matrice di confusione dell'algoritmo ANN testato sul dataset di pubblico dominio

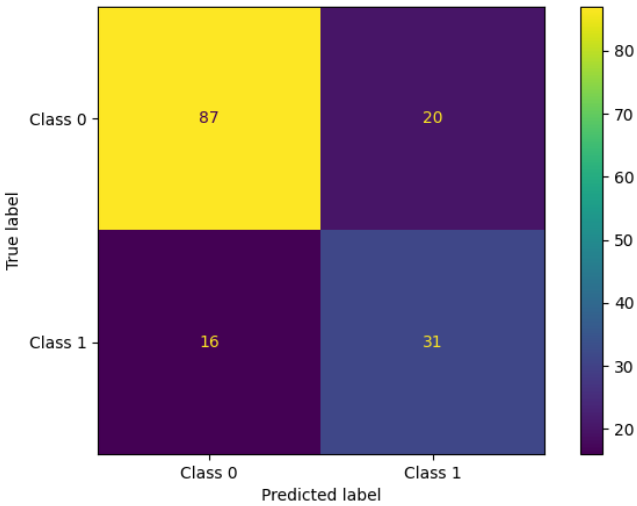


Figura 6.5 – Matrice di confusione dell'algoritmo Decision Tree testato sul dataset di pubblico dominio

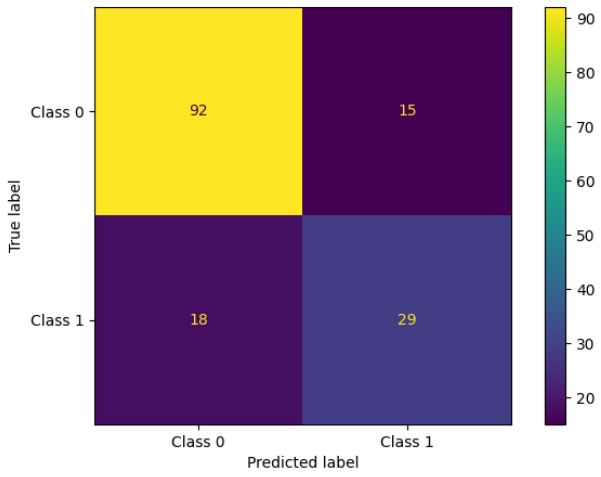


Figura 6.6 – Matrice di confusione dell'algoritmo Random Forest testato sul dataset di pubblico dominio

A partire da questi test preliminari sul dataset di pubblico dominio si può osservare che l'algoritmo più accurato risulta essere il LR, mentre quello meno accurato ANN. Gli stessi risultati si ottengono anche valutando la metrica di precisione, mentre per quello che riguarda quella di recall le prestazioni migliori si hanno per l'algoritmo Decision Tree e quelle peggiori per SVM.

6.2. Test su dataset sintetico

In tabella 6.2 sono riportati i valori di accuratezza, precisione e recall ottenuti per ogni algoritmo implementato, anche rappresentati nell'istogramma 6.2. In tabella 6.3 si riportano inoltre i numeri esatti di casi guasti e non guasti all'interno del dataset, al fine di fornire uno strumento in più per la valutazione dei risultati sopra citati.

Tabella 6.2 – Metriche di accuratezza, precisione e recall per i sei algoritmi implementati testati sul dataset sintetico

	Accuratezza	**Precisione**	**Recall**
LR	0,900	0,000	0,000
KNN	0,890	0,102	0,013
SVM	0,900	0,000	0,000
ANN	0,900	0,000	0,000
Decision Tree	0,784	0,098	0,142
Random Forest	0,897	0,091	0,003

Grafico 6.2 – Metriche di accuratezza, precisione e recall per i sei algoritmi implementati testati sul dataset sintetico

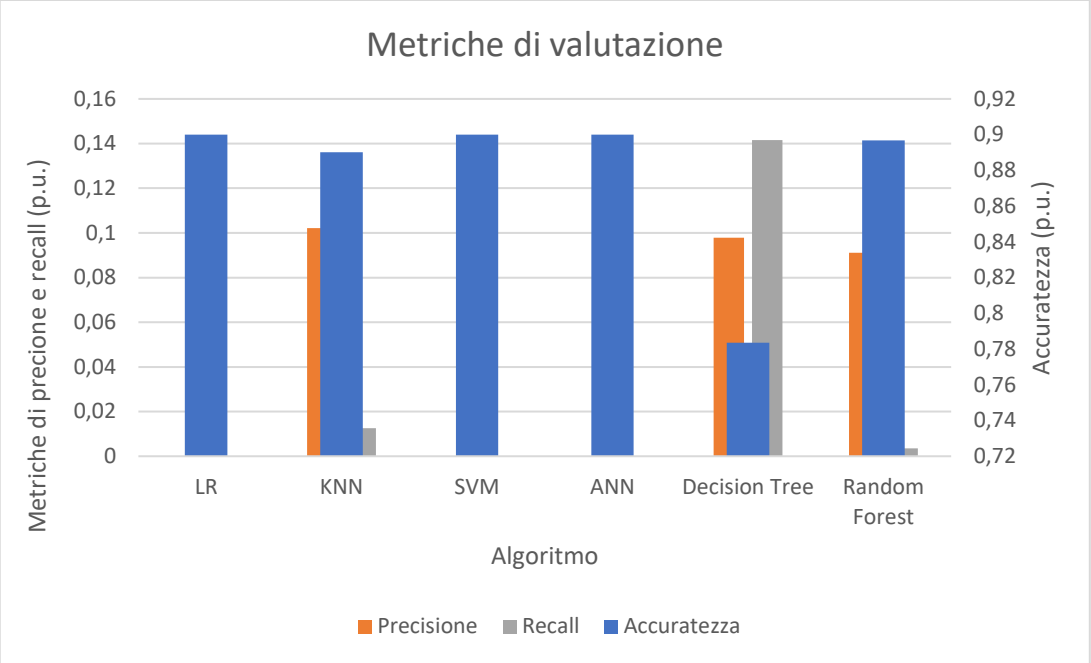


Tabella 6.3 – Numero dei casi guasti e non guasti all'interno della porzione di dataset sintetico utilizzata per la fase di apprendimento della simulazione generica

Numero totale di campioni nella fase di apprendimento	**Casi non guasti (etichettati come 0)**	**Casi guasti (etichettati come 1)**
1000	876	124

Dai risultati ottenuti possiamo osservare un'elevata accuratezza per tutti gli algoritmi, ma basse prestazioni in termini di precisione e recall. Per quanto riguarda la prima metrica, questa è nulla in metà degli algoritmi e arriva al massimo al valore di 0,102 per l'algoritmo KNN. Riguardo alla metrica di recall, anche questa è nulla nella metà degli algoritmi e il suo valore massimo è pari a 0,142 per l'algoritmo Decision Tree.

Questo sbilanciamento nei risultati è presumibilmente dovuto allo sbilanciamento intrinseco nel dataset sintetico. Infatti, osservando la tabella 6.3, si nota come l'87,6 % dei campioni utilizzati per la fase di learning degli

algoritmi siano etichettati come output 0 (caso non guasto). Questa predominanza di casi non guasti riflette le condizioni reali che si hanno sul campo, confermando quindi la validità del dataset generato sinteticamente.

Le matrici di confusione per i sei algoritmi implementati sono invece mostrate nelle figure da 6.7 a 6.12 e confermano i risultati sopra ottenuti: concentrandosi per esempio sulla matrice dell'algoritmo LR, riportata in figura 6.7, i casi non guasti e correttamente predetti (cella 0,0) sono pari al numero totale di casi non guasti all'interno del validation set. Questo significa che tutti i casi non guasti sono stati riconosciuti correttamente dall'algoritmo, infatti la cella 1,0 è pari a zero. Dall'altro lato, i casi guasti e correttamente predetti (cella 1,1) sono pari a zero, mentre quelli predetti in modo scorretto sono pari al numero totale di casi guasti all'interno del dataset, cioè tutti i casi guasti sono stati predetti in modo errato dall'algoritmo. Nonostante ciò, l'accuratezza risulta comunque elevata perché quasi il 90 % delle istanze del validation set è stato predetto correttamente, viceversa le metriche di precisione e recall hanno valori particolarmente limitati.

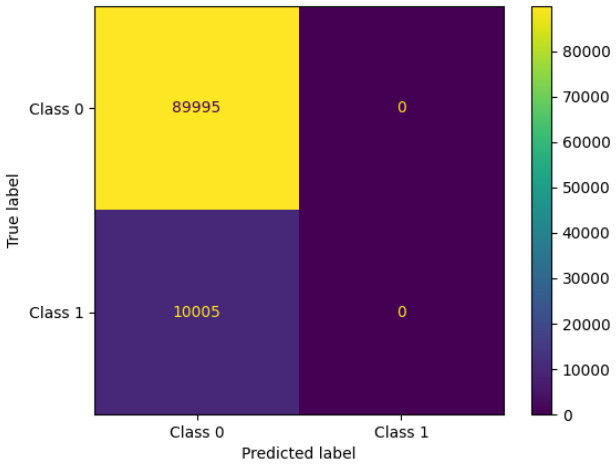


Figura 6.7 – Matrice di confusione dell'algoritmo LR testato sul dataset sintetico

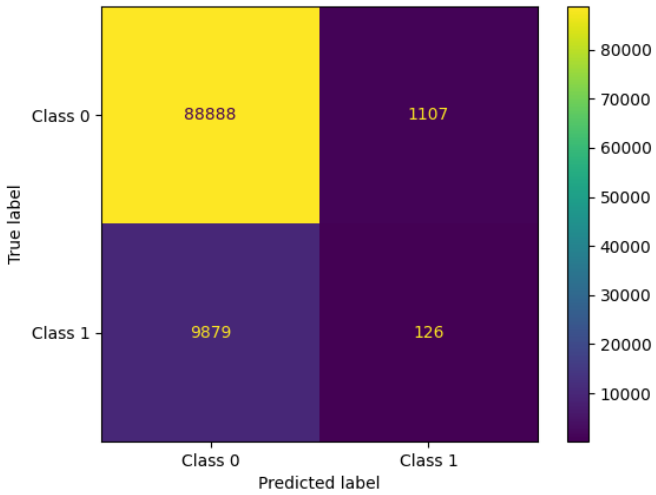


Figura 6.8 – Matrice di confusione dell'algoritmo KNN testato sul dataset sintetico

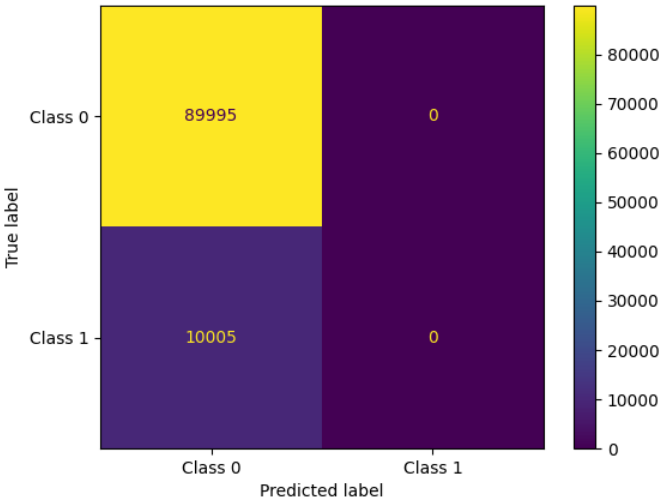


Figura 6.9 – Matrice di confusione dell'algoritmo SVM testato sul dataset sintetico

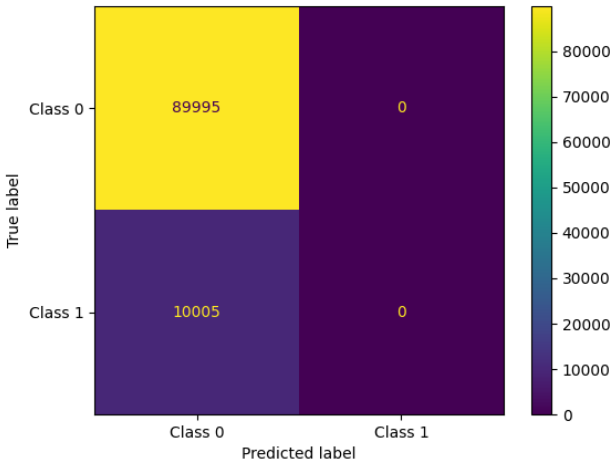


Figura 6.10 – Matrice di confusione dell'algoritmo ANN testato sul dataset sintetico

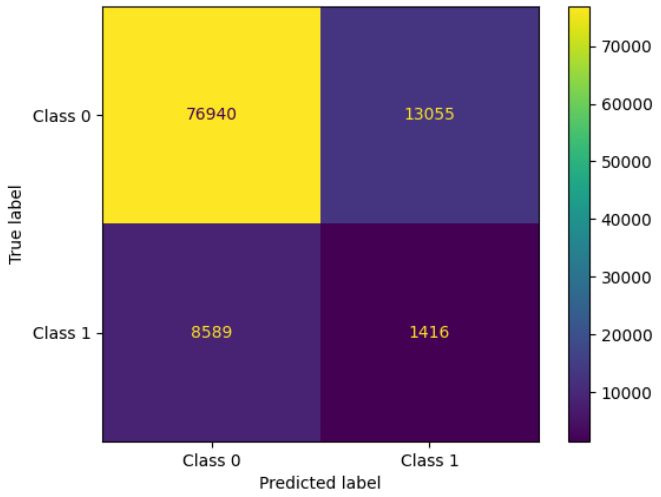


Figura 6.11 – Matrice di confusione dell'algoritmo Decision Tree testato sul dataset sintetico

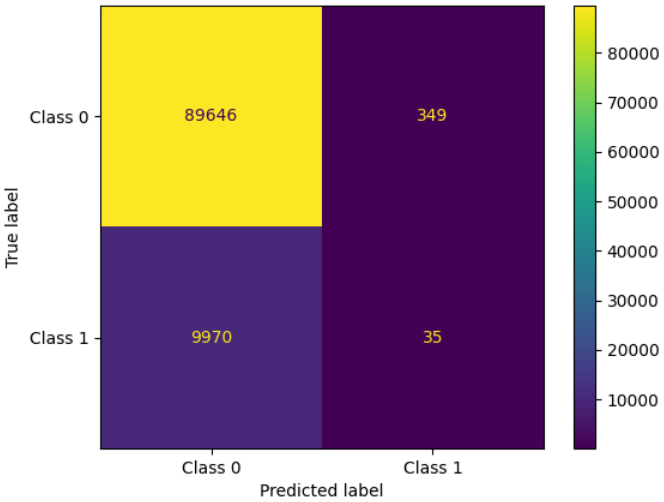


Figura 6.12 – Matrice di confusione dell'algoritmo Random Forest testato sul dataset sintetico

6.2.1. Numero di campioni nella fase di apprendimento

I risultati delle simulazioni descritte al paragrafo 5.3.1 sono riportati qui di seguito. In particolare, le metriche di accuratezza, precisione e recall al variare del numero di campioni sono rispettivamente riportate nelle tabelle 6.4, 6.5 e 6.6. Ai fini della lettura dei risultati, in tabella 6.7 è stato anche riportato il numero di istanze etichettate come non guaste in relazione al numero complessivo di campioni utilizzati per la fase di apprendimento degli algoritmi.

Tabella 6.4 – Accuratezza al variare del numero di campioni nella fase di apprendimento

	100	**200**	**500**	**1000**	**2000**	**5000**	**10000**
LR	0,899	0,901	0,899	0,899	0,900	0,899	0,904
KNN	0,898	0,901	0,887	0,891	0,895	0,893	0,899
SVM	0,899	0,901	0,899	0,899	0,900	0,899	0,904
ANN	0,899	0,901	0,895	0,899	0,900	0,899	0,904
Decision Tree	0,803	0,781	0,789	0,803	0,819	0,800	0,819
Random Forest	0,898	0,895	0,892	0,896	0,899	0,898	0,903

Tabella 6.5 – Precisone al variare del numero di campioni nella fase di apprendimento

	100	**200**	**500**	**1000**	**2000**	**5000**	**10000**
LR	0,000	0,000	0,000	0,000	0,000	0,000	0,000
KNN	0,107	0,146	0,099	0,102	0,108	0,106	0,109
SVM	0,000	0,000	0,000	0,000	0,000	0,000	0,000
ANN	0,000	0,000	0,105	0,000	0,000	0,000	0,000
Decision Tree	0,101	0,098	0,102	0,098	0,097	0,099	0,095
Random Forest	0,042	0,111	0,103	0,090	0,077	0,100	0,121

Tabella 6.6 – Recall al variare del numero di campioni nella fase di apprendimento

	100	**200**	**500**	**1000**	**2000**	**5000**	**10000**
LR	0,000	0,000	0,000	0,000	0,000	0,000	0,000
KNN	0,000	0,001	0,015	0,010	0,008	0,008	0,007
SVM	0,000	0,000	0,000	0,000	0,000	0,000	0,000
ANN	0,000	0,000	0,005	0,000	0,000	0,000	0,000
Decision Tree	0,119	0,148	0,139	0,116	0,098	0,120	0,104
Random Forest	0,000	0,009	0,009	0,003	0,001	0,001	0,001

Tabella 6.7 – Numero dei casi guasti e non guasti all'interno della porzione di dataset sintetico utilizzata per la fase di apprendimento per il calcolo delle metriche di accuratezza, precisione e recall

Numero totale di campioni nella fase di apprendimento	**Casi non guasti (etichettati come 0)**	**Casi guasti (etichettati come 1)**
100	91	9
200	177	23
500	442	58
1000	897	103
2000	1805	195
5000	4490	510
10000	9072	928

I dati sopra riportati sono anche mostrati nei grafici 6.3, 6.4 e 6.5, dove sono rispettivamente rappresentati gli andamenti delle tre metriche di accuratezza, precisione e recall in funzione del numero di campioni utilizzati nella fase di apprendimento.

Grafico 6.3 – Accuratezza degli algoritmi implementati al variare del numero di campioni nella fase di learning

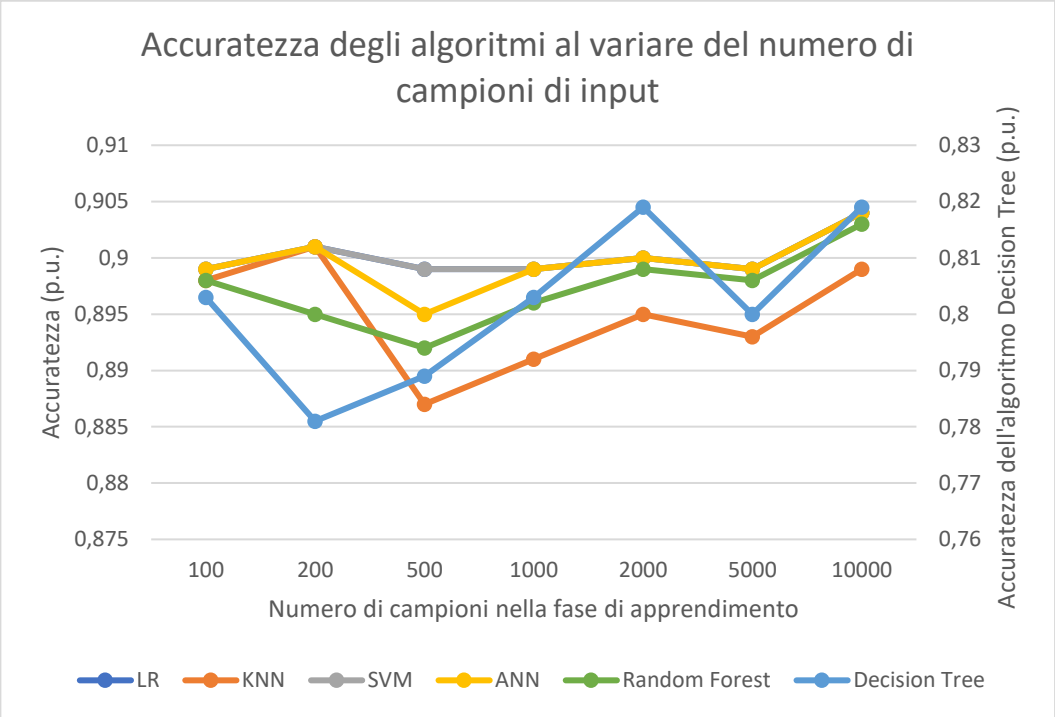


Grafico 6.4 – Precisione degli algoritmi implementati al variare del numero di campioni nella fase di learning

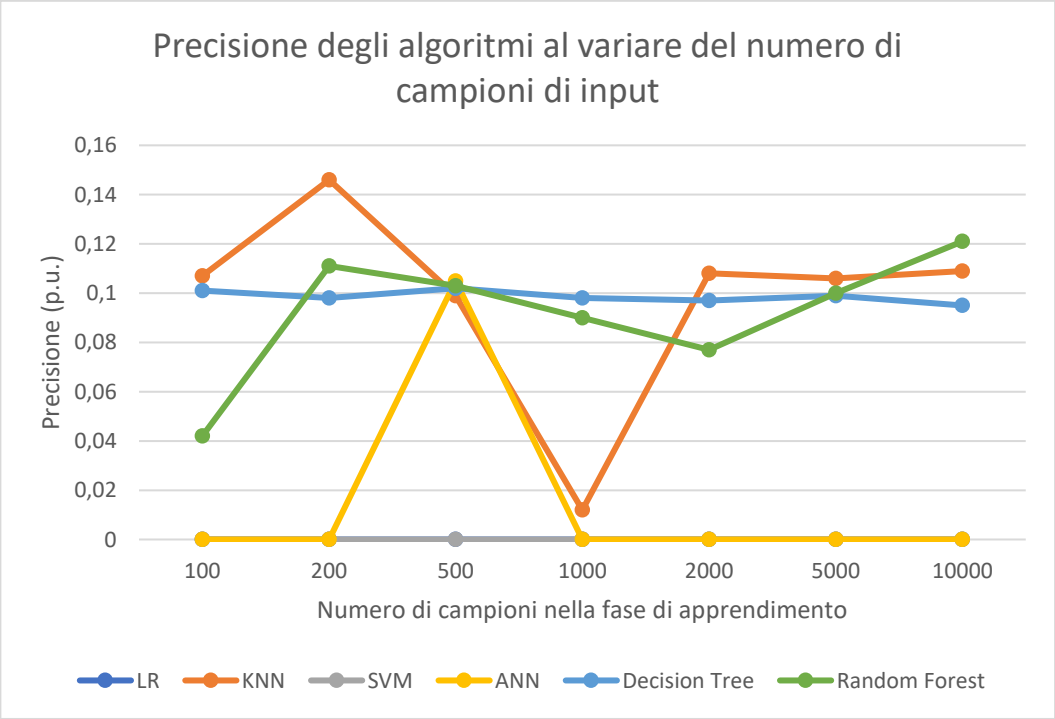
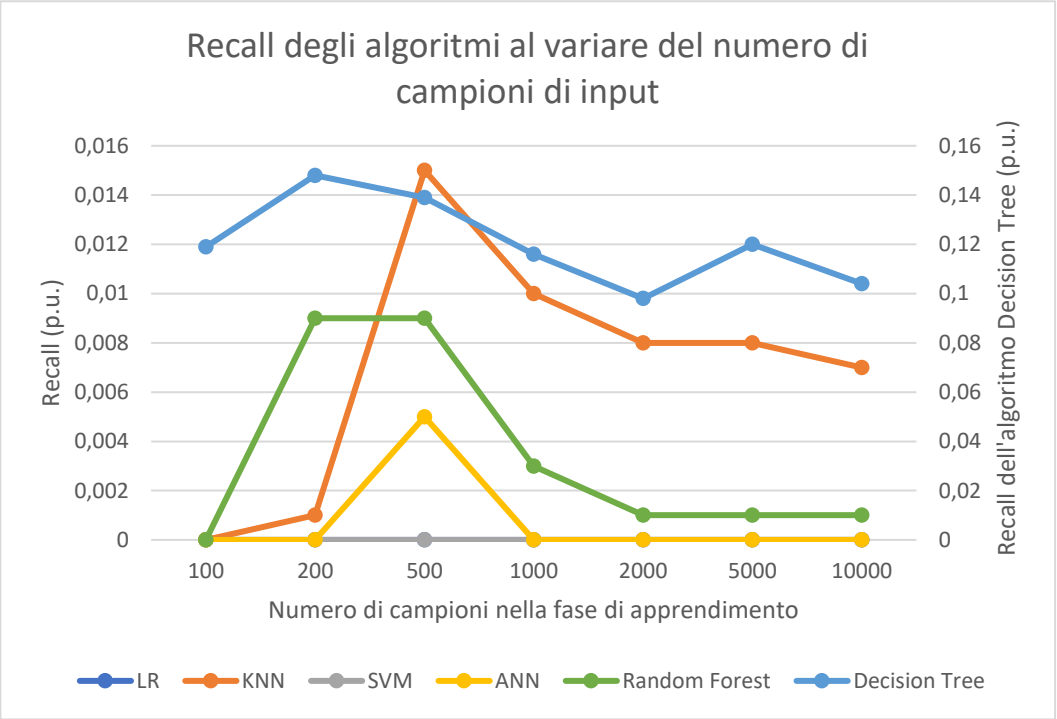


Grafico 6.5 – Recall degli algoritmi implementati al variare del numero di campioni nella fase di learning



Per quanto riguarda le metriche di accuratezza, precisione e recall, possono essere fatte valutazioni analoghe a quelle del paragrafo precedente, dato lo squilibrio all'interno del dataset sintetico. Concentrandosi invece sugli andamenti, si può osservare che l'accuratezza raggiunge prestazioni accettabili e più o meno stabili attorno al valore di 0,85 quando il numero di campioni di input è 1000, che, come già espresso in precedenza, è un numero di totalmente accettabile dal punto di vista della disponibilità di informazioni.

6.2.2. Ripetibilità

I risultati della verifica della ripetibilità degli algoritmi, effettuati così come descritto nel paragrafo 5.3.2, sono riportati nei grafici 6.6, 6.7, 6.8, 6.9, 6.10, 6.11. In particolare, per ogni algoritmo si riporta la distribuzione delle accuratezze ottenute e, in tabella 6.8, valori di media e deviazione standard. Tali risultati confermano la ripetibilità degli algoritmi.

Grafico 6.6 – Distribuzione dell'accuratezza per 500 esecuzioni dell'algoritmo LR

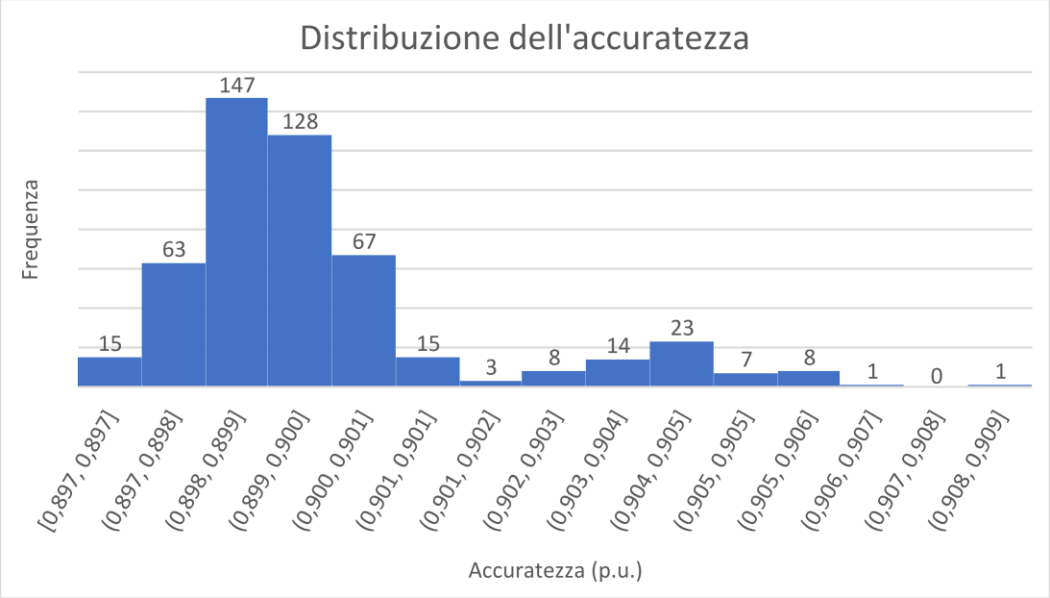


Grafico 6.7 – Distribuzione dell'accuratezza per 500 esecuzioni dell'algoritmo KNN

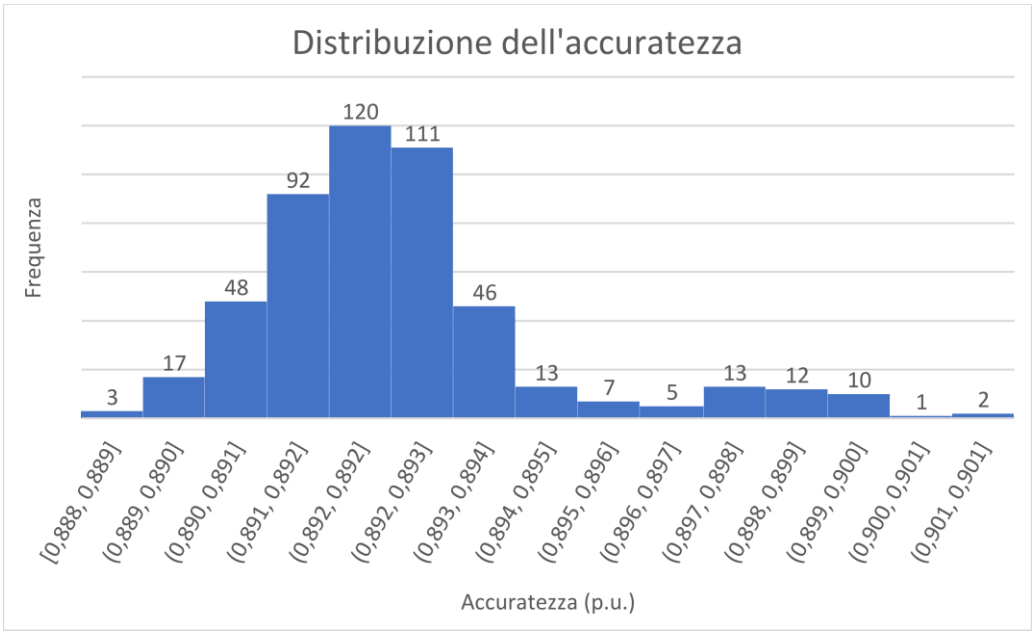


Grafico 6.8 – Distribuzione dell'accuratezza per 500 esecuzioni dell'algoritmo SVM

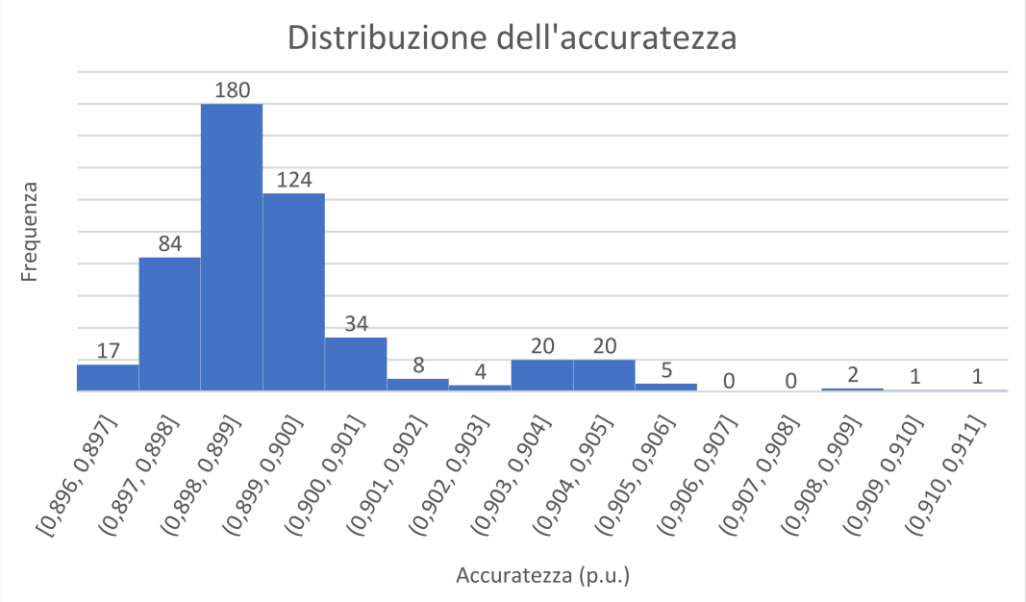


Grafico 6.9 – Distribuzione dell'accuratezza per 500 esecuzioni dell'algoritmo ANN

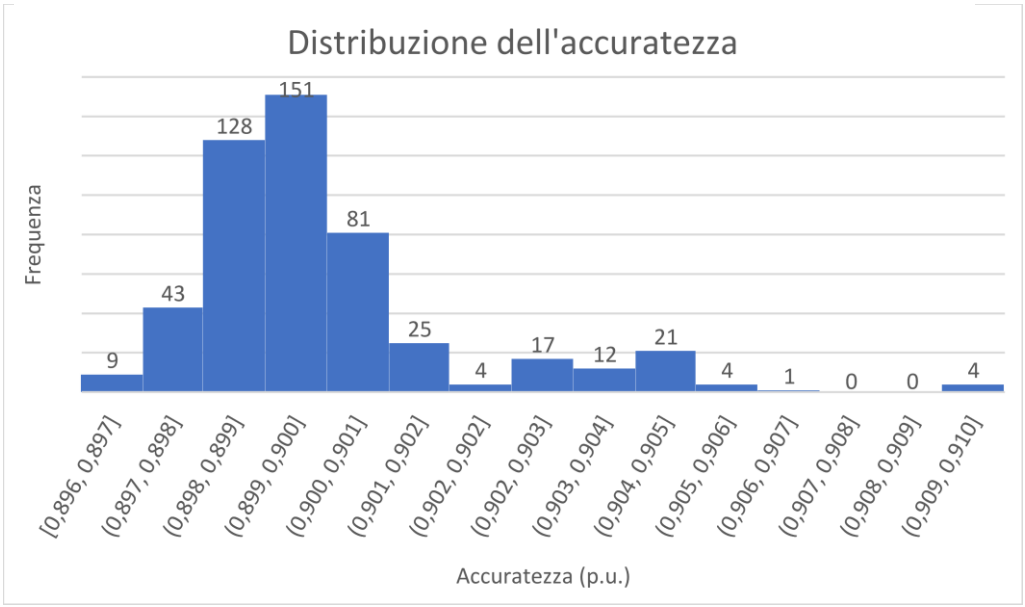


Grafico 6.10 – Distribuzione dell'accuratezza per 500 esecuzioni dell'algoritmo Decision Tree

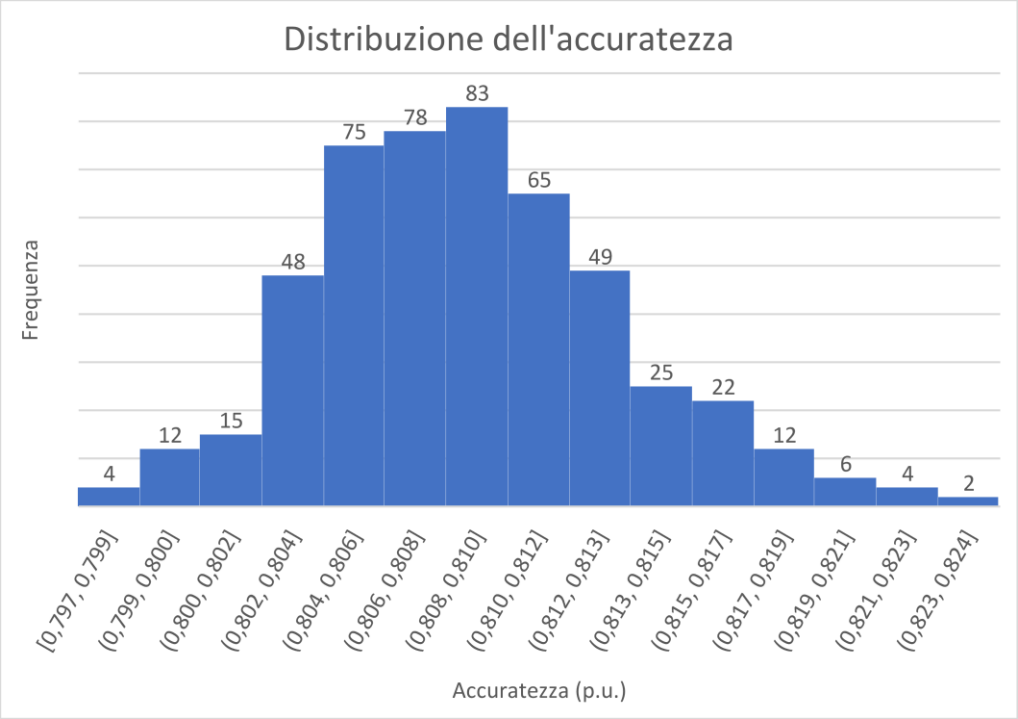


Grafico 6.11 – Distribuzione dell'accuratezza per 500 esecuzioni dell'algoritmo Random Forest

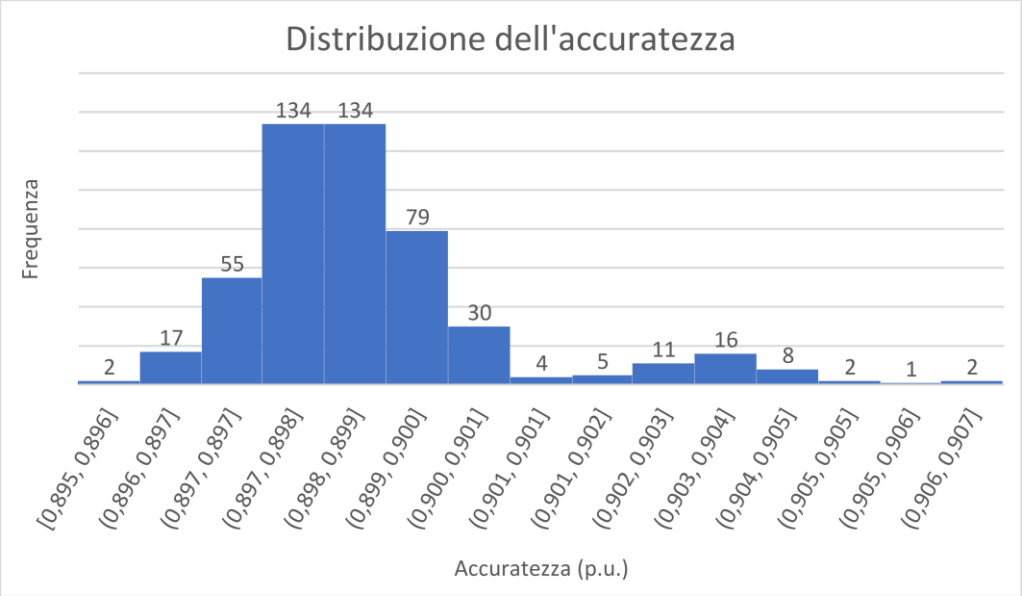


Tabella 6.8 – Valori di media e deviazione standard per le 500 esecuzioni degli algoritmi

	Media	**Deviazione Standard**
LR	0,900	0,002
KNN	0,893	0,002
SVM	0,900	0,002
ANN	0,900	0,002
Decision Tree	0,809	0,005
Random Forest	0,899	0,002

6.2.3. Distribuzione dei dati

I risultati del test sulla distribuzione dei dati, descritto al paragrafo 5.3.3 e costituito dalle simulazioni descritte al paragrafo 5.3.1. sono riportati qui di seguito. In particolare, le metriche di accuratezza, precisione e recall al variare del numero di campioni sono rispettivamente riportate nelle tabelle 6.9, 6.10 e 6.11. Ai fini della lettura dei risultati, in tabella 6.12 è stato anche riportato il numero di istanze etichettate come non guaste in relazione al numero complessivo di campioni utilizzati per la fase di apprendimento degli algoritmi.

Tabella 6.9 – Accuratezza al variare del numero di campioni nella fase di apprendimento

	100	**200**	**500**	**1000**	**2000**	**5000**	**10000**
LR	0,900	0,899	0,899	0,900	0,899	0,900	0,900
KNN	0,899	0,895	0,885	0,893	0,892	0,896	0,893
SVM	0,900	0,899	0,899	0,900	0,899	0,900	0,900
ANN	0,900	0,899	0,899	0,900	0,898	0,900	0,900
Decision Tree	0,817	0,780	0,806	0,804	0,812	0,820	0,804
Random Forest	0,896	0,891	0,896	0,897	0,898	0,898	0,899

Tabella 6.10 – Precisione al variare del numero di campioni nella fase di apprendimento

	100	**200**	**500**	**1000**	**2000**	**5000**	**10000**
LR	0,077	0,000	0,000	0,000	0,000	0,000	0,000
KNN	0,071	0,086	0,093	0,111	0,110	0,117	0,087
SVM	0,000	0,000	0,000	0,000	0,000	0,000	0,000
ANN	0,000	0,200	0,000	0,138	0,125	0,000	0,000
Decision Tree	0,104	0,101	0,105	0,100	0,106	0,095	0,098
Random Forest	0,098	0,089	0,118	0,108	0,136	0,069	0,139

Tabella 6.11 – Recall al variare del numero di campioni nella fase di apprendimento

	100	**200**	**500**	**1000**	**2000**	**5000**	**10000**
LR	0,000	0,000	0,000	0,000	0,000	0,000	0,000
KNN	0,001	0,004	0,015	0,009	0,009	0,006	0,007
SVM	0,000	0,000	0,000	0,000	0,000	0,000	0,000
ANN	0,000	0,000	0,000	0,001	0,001	0,000	0,000
Decision Tree	0,109	0,149	0,122	0,120	0,114	0,092	0,117
Random Forest	0,004	0,008	0,004	0,003	0,002	0,001	0,001

Tabella 6.12 – Numero dei casi guasti e non guasti all'interno della porzione di dataset sintetico utilizzata per la fase di apprendimento per il calcolo delle metriche di accuratezza, precisione e recall

Numero totale di campioni nella fase di apprendimento	**Casi non guasti (etichettati come 0)**	**Casi guasti (etichettati come 1)**
100	90	10
200	175	25
500	450	50
1000	902	98
2000	1816	184
5000	4542	458
10000	9014	986

I valori delle metriche di accuratezza, precisione e recall al variare del numero di istanze per la fase di apprendimento sono rispettivamente mostrati nei grafici 6.12, 6.13 e 6.14.

Grafico 6.12 – Accuratezza degli algoritmi implementati al variare del numero di campioni nella fase di learning

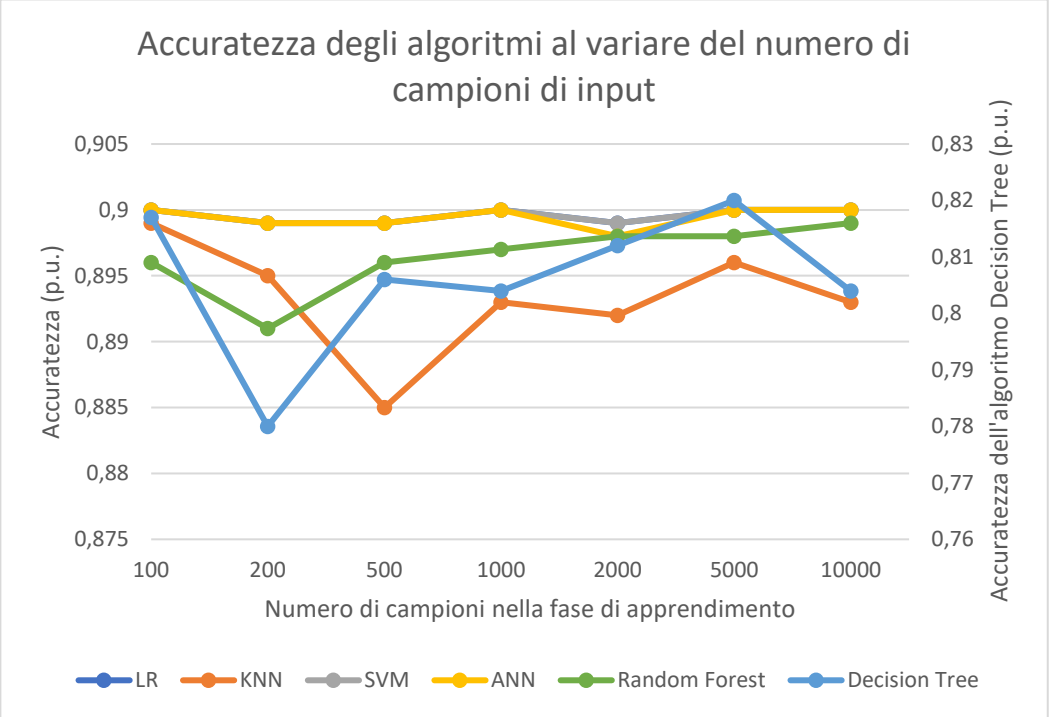


Grafico 6.13 – Precisione degli algoritmi implementati al variare del numero di campioni nella fase di learning

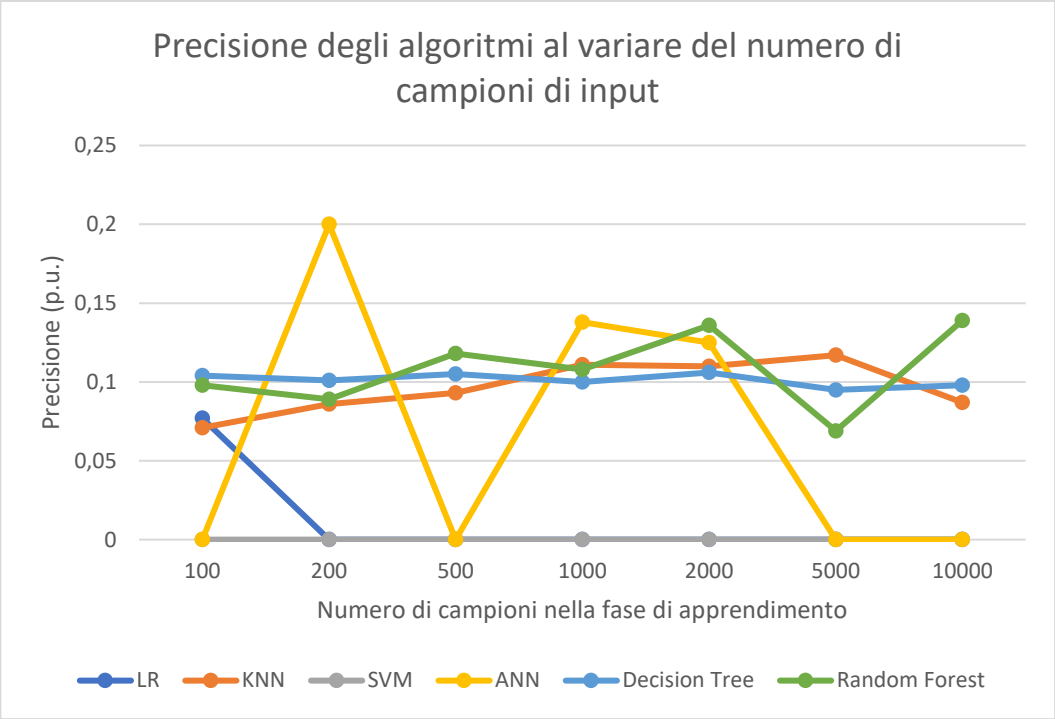
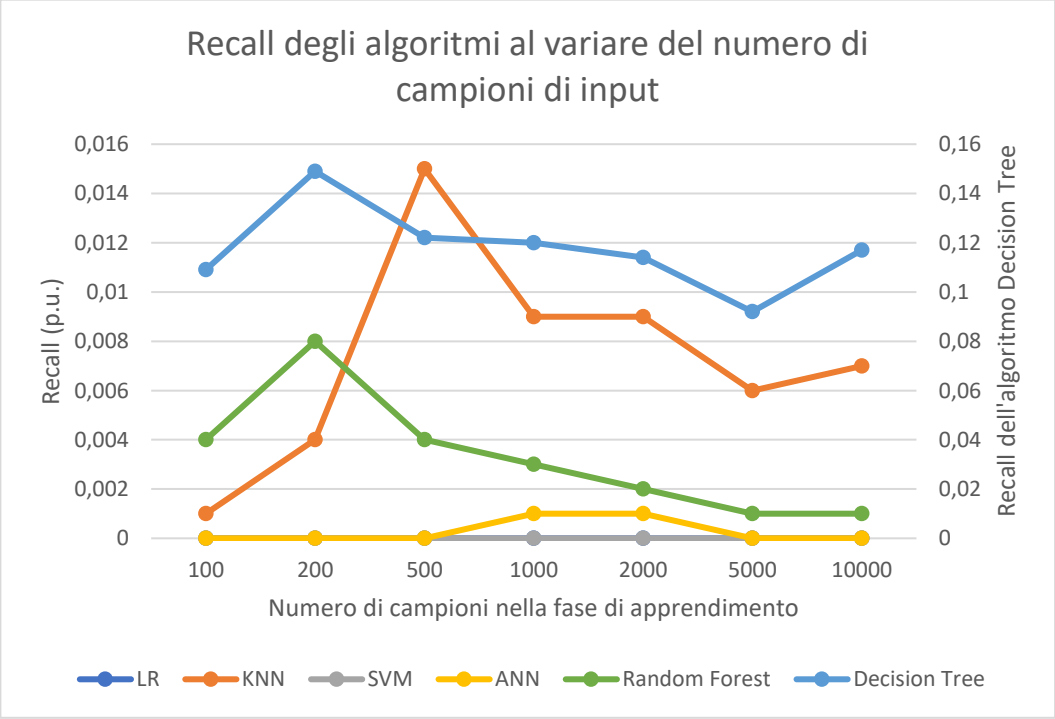


Grafico 6.14 – Recall degli algoritmi implementati al variare del numero di campioni nella fase di learning



Dai grafici si può osservare come, ipotizzando di avere dati di ingresso distribuiti secondo la normale, le prestazioni degli algoritmi aumentino. In particolare, considerando un numero di campioni di ingresso pari a 1000, l'accuratezza ha valori compresi fra 0,89 e 0,90. Per quanto riguarda la precisione l'algoritmo SVM continua ad avere valori sempre nulli e l'algoritmo ANN oscilla fra lo zero e valori diversi da zero, mentre tutti gli altri algoritmi si stabilizzano attorno ad una precisione dello 0,1. La metrica di recall risulta più elevata e attorno al valore di 0,12 per l'algoritmo di Decision Tree, mentre per tutti gli altri algoritmi è quasi nulla.

Nonostante la distribuzione normale porti a prestazioni più elevate, è importante osservare che non è comune avere dati di ingresso che siano caratterizzati da questo tipo di distribuzione.

7. Conclusioni

L’utilizzo di cavi interrati in media tensione è oggi diffuso in tutte le reti europee e consente agli operatori di evitare i problemi che derivano dall’uso di cavi e linee aeree; vegetazione, eventi climatici avversi ed animali sono le cause più frequenti di guasto per questo tipo di linee di distribuzione. Per questo motivo i cavi interrati risultano essere una scelta usuale in questo campo. Tuttavia, oltre ai benefici sopra citati, il loro uso porta all’insorgere di ulteriori problematiche: i cavi interrati sono infatti soliti a guastarsi in corrispondenza di terminazioni o giunti. In particolare, questi ultimi sono considerati i punti deboli della rete di distribuzione e in letteratura sono investigate diverse metodologie e tecniche per la comprensione dei fenomeni che portano questi componenti a guastarsi. Il verificarsi di un guasto in un giunto è infatti un evento importante dal punto di vista delle sue conseguenze e della sua gestione a carico dell’ente responsabile.

In questo elaborato vengono illustrati i giunti, ponendo particolare attenzione ai loro modi di guasto, allo studio della propagazione di quest’ultimo e dei parametri coinvolti, allo scopo di utilizzare queste conoscenze per realizzare e valutare una prima applicazione di intelligenza artificiale per svolgere manutenzione predittiva su questi componenti. In particolare, dopo aver selezionato sei algoritmi di machine learning a partire dalla letteratura disponibile, questi sono stati utilizzati per l’elaborazione di un dataset contenente misurazioni sui giunti. Tale dataset è stato precedentemente generato in modo sintetico e questa scelta è stata data dal fatto che, in questo primo approccio, non si avevano a disposizione sufficienti misurazioni reali. Inoltre, nell’ottica di un effettivo sviluppo del sistema trattato, era importante valutare in prima istanza quali algoritmi fossero più prestanti e quali fossero le caratteristiche da tenere in considerazione a tale scopo.

Gli algoritmi sono stati inizialmente eseguiti su un dataset di pubblico dominio e hanno fornito risultati accettabili, soprattutto considerata la limitatezza del dataset a meno di 800 istanze e il fatto che, come parametri dei diversi algoritmi, sono stati lasciati quelli tipici.

In seguito, gli stessi algoritmi sono stati utilizzati per elaborare il dataset sintetico e, da una prima valutazione delle metriche di accuratezza, precisione, recall e matrice di confusione, si osservano elevati valori di accuratezza, ma scarse prestazioni nelle restanti metriche. Questo sbilanciamento nei risultati è conseguente a quello presente nel dataset generato sinteticamente.

Dagli ulteriori tre test effettuati, quali accuratezza al variare del numero di campioni di input, ripetibilità e distribuzione dei dati è emerso che: avere 1000 campioni di input risulta un buon compromesso fra accuratezza ottenuta e disponibilità di informazioni sul campo, la ripetibilità dell'algoritmo è confermata e, infine, avere dati di ingresso caratterizzati da distribuzione normale aumenta le prestazioni degli algoritmi.

Confrontando questi risultati con quelli della revisione sistematica della letteratura, si può osservare che ogni algoritmo di machine learning potrebbe potenzialmente prestarsi ad un determinato caso di studio e che è quindi opportuno che ognuno sia valutato in modo approfondito, al fine di poter essere adattato al meglio ai dati e al problema che si sta considerando.

In conclusione, è generalmente noto che l'intelligenza artificiale sia uno strumento fondamentale in svariati campi e in questo elaborato è proposta una prima applicazione di alcuni fra i più conosciuti algoritmi di machine learning per l'elaborazione di misure ottenute dai giunti elettrici, al fine di effettuare manutenzione predittiva condition-based su questi componenti. Dai risultati emergono le elevate potenzialità di questa tecnologia ai fini della prevenzione dei guasti nelle reti di distribuzione, soprattutto se considerata la semplicità degli algoritmi, che li rende facilmente implementabili in tutte le sale di controllo degli operatori di sistema.

Bibliografia

[1] K. Rana, B. Dasgupta, *Analysis of HT cable and joint failures and associated design modifications*, CESC Ltd Third-Party Damage to Underground and Submarine Cables, CIGRE Working Group B1.21, December 2009.

[2] *Analysis of joint & termination failures on extruded dielectric cables*, IEEE Transaction on Power Apparatus and Systems, Vol. PAS-103, No. 12, December 1984.

[3] X. Dong, Y. Yuan, Z. Gao, C. Zhou, P. Wallace, B. Alkali, B. Sheng, H. Zhou, *Analysis of cable failure modes and cable joint failure detection via sheath circulating current*, Electrical Insulation Conference, Philadelphia, June 2014.

[4] Y. Jiang et al., *Partial Discharge Pattern Characteristic of HV Cable Joints with Typical Artificial Defect*, 2010 Asia-Pacific Power and Energy Engineering Conference, 2010.

[5] L. Testa, A. Cavallini, G. C. Montanari, A. Makovoz, *On-line partial discharges monitoring of high voltage XLPE / fluid-filled transition joints*, Electrical Insulation Conference, Annapolis, June 2011.

[6] B. Rajalakshmi, L. Kalaivani, *Analysis of Partial Discharge in underground cable joints*, International Conference on Innovations in Information, Embedded and Communciation Systems, Coimbatore, March 2015.

[7] R. N. Wu, C. K. Chang, *The Use of Partial Discharges as an Online Monitoring System for Underground Cable Joints*, IEEE Transaction on Power Delivery, vol. 24, no. 3, 2011, pp. 1585-1591.

[8] D. Fournier and L. Lamarre, *Effect of pressure and temperature on interfacial breakdown between two dielectric surfaces*, in Proc. Annu. Rep., Conf. Elect. Insul. Dielectr. Phenomena, Oct. 1992, pp. 229–235.

[9] X. Zhou, J. Cao, S. Wang, Y. Jiang, T. Li, and Y. Zou, *Simulation of electric field around typical defects in 110kV XLPE power cable joints*, in Proc. Int. Conf. Circuits Devices Syst., Chengdu, China, Sep. 2017, pp. 21–24.

[10] N. Permal, C. K. Chakrabarty, A. R. Avinash, T. Marie, H. S. Abd Halim, *Tangent Delta Extraction of Cable Joints for Aged 11kV Underground Cable System*, International Conference on Advances in Electrical, Electronic and System Engineering, Putrajaya, Nov. 2016.

[11] A. Mingotti, A. Ghaderi, L. Peretto, R. Tinarell, F. Lama, *Test Setup Design, and Calibration for Tan Delta Measurements on MV Cable Joints*, 2018 IEEE 9th International Workshop on Applied Measurements for Power Systems (AMPS), 2018, pp. 1-5.

[12] A. Ghaderi, A. Mingotti, F. Lama, L. Peretto, R. Tinarelli, *Effects of Temperature on MV Cable Joints Tan Delta Measurements*, in IEEE Transactions on Instrumentation and Measurement, vol. 68, no. 10, Oct. 2019, pp. 3892-3898.

[13] A. Mingotti, A. Ghaderi, R. Tinarelli, L. Peretto, *Analysis of MV Cable Joints Equivalent Impedance and its Variation vs. Temperature*, Proceedings of the 23rd IMEKO TC-4, IMEKO, 2019, pp. 68 – 72.

[14] A. Ghaderi, A. Mingotti, L. Peretto, R. Tinarelli, *Effects of Mechanical Pressure on the Tangent Delta of MV Cable Joints*, in IEEE Transactions on Instrumentation and Measurement, vol. 68, no. 7, July 2019, pp. 2656-2658.

[15] R. Di Sante, A. Ghaderi, A. Mingotti, L. Peretto, R. Tinarelli, *Test Bed Characterization for the Interfacial Pressure vs. Temperature Measurements in MV Cable-Joints*, 2019 II Workshop on Metrology for Industry 4.0 and IoT (MetroInd4.0&IoT), 2019, pp. 186-190.

[16] R. Di Sante, A. Ghaderi, A. Mingotti, L. Peretto, R. Tinarelli, *Effects of Thermal Cycles on Interfacial Pressure in MV Cable Joints*, Sensors (Basel, Switzerland), 2019.

[17] R. Jongen, E. Gulski, J. Smit, *Failures of medium voltage cable joints in relation to the ambient temperature*, 20th International Conference on Electricity Distribution, Prague, June 2009.

[18] *Treccani*, disponibile al link

[19] *IntelligenzaArtificialeItalia*, disponibile al link

[20] C. -K. Chang, H. -H. Chang and B. K. Boyanapalli, *Application of Pulse Sequence Partial Discharge Based Convolutional Neural Network in Pattern Recognition for Underground Cable Joints*, in IEEE Transactions on Dielectrics and Electrical Insulation, vol. 29, no. 3, pp. 1070-1078, June 2022, doi: 10.1109/TDEI.2022.3168328.

[21] M. Bindi, A. Luchetta, L. Paolucci, F. Grasso, S. Manetti and M. C. Piccirilli, *Applications of Machine Learning Techniques for the Monitoring of Electrical Transmission and Distribution lines*, 2022 18th International Conference on Synthesis, Modeling, Analysis and

Simulation Methods and Applications to Circuit Design (SMACD), 2022, pp. 1-4, doi: 10.1109/SMACD55068.2022.9816290.

[22] C. -K. Chang and B. K. Boyanapalli, *Assessment of the Insulation Status Aging in Power Cable Joints Using Support Vector Machine*, in IEEE Transactions on Dielectrics and Electrical Insulation, vol. 28, no. 6, pp. 2170-2177, December 2021, doi: 10.1109/TDEI.2021.009783.

[23] P. L. Lewin, T. Coleman, N. Koumbari, Y. Liu and S. Christou, *Avoiding Medium Voltage Cable Joint Failure: Development of a Real-Time Prognostic Tool*, 2021 IEEE Electrical Insulation Conference (EIC), 2021, pp. 181-184, doi: 10.1109/EIC49891.2021.9612392.

[24] C. -K. Chang, C. -S. Lai and R. -N. Wu, *Decision tree rules for insulation condition assessment of pre-molded power cable joints with artificial defects*, in IEEE Transactions on Dielectrics and Electrical Insulation, vol. 26, no. 5, pp. 1636-1644, Oct. 2019, doi: 10.1109/TDEI.2019.008208.

[25] Mantach, S., Lutfi, A., Moradi Tavasani, H., Ashraf, A., El-Hag, A., & Kordi, B. (2022). *Deep Learning in High Voltage Engineering: A Literature Review*, Energies, 15(14), 5005.

[26] J. H. Choi, C. Park, P. Cheetham, C. H. Kim, S. Pamidi and L. Graber, *Detection of Series Faults in High-Temperature Superconducting DC Power Cables Using Machine Learning*, in IEEE Transactions on Applied Superconductivity, vol. 31, no. 5, pp. 1-9, Aug. 2021, Art no. 4800609, doi: 10.1109/TASC.2021.3055156.

[27] Nathalie Morette, Thierry Ditchi, Yacine Oussar, *Feature extraction and ageing state recognition using partial discharges in cables under HVDC*, Electric Power Systems Research,

Volume 178, 2020, 106053, ISSN 0378-7796, https://doi.org/10.1016/j.epsr.2019.106053.

[28] R. Sahoo and S. Karmakar, *Health Index Prediction of Underground Cable System using Artificial Neural Network*, 2021 1st Odisha International Conference on Electrical Power Engineering, Communication and Computing Technology(ODICON), 2021, pp. 1-4, doi: 10.1109/ODICON50556.2021.9429013.

[29] J. Yeo et al., *Identification of Partial Discharge Through Cable-Specific Adaption and Neural Network Ensemble*, in IEEE Transactions on Power Delivery, vol. 37, no. 3, pp. 1598-1607, June 2022, doi: 10.1109/TPWRD.2021.3093670.

[30] Wong Jee Keen Raymond et al., *Noise invariant partial discharge classification based on convolutional neural network*, Measurement, Volume 177, 2021, 109220, ISSN 0263-2241, https://doi.org/10.1016/j.measurement.2021.109220.

[31] N. Rosle, N. A. Muhamad, M. N. K. H. Rohani and M. K. M. Jamil, *Partial Discharges Classification Methods in XLPE Cable: A Review*, in IEEE Access, vol. 9, pp. 133258-133273, 2021, doi: 10.1109/ACCESS.2021.3115519.

[32] Zhang, X.; Pang, B.; Liu, Y.; Liu, S.; Xu, P.; Li, Y.; Xie, Q., *Review on detection and analysis of partial discharge along power cables*, Energies 2021, 14, 7692.

[33] M. O. Alhusin, A. Sahoo, I. Nikolakakos, H. Alhammadi, A. Alsahi and M. Alhammadi, *Weather and Seasonal Effects on Medium Voltage Underground Cable Joint Failures*, 2022 IEEE International Conference on Environment and Electrical Engineering and 2022 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe), 2022, pp. 1-6, doi: 10.1109/EEEIC/ICPSEurope54979.2022.9854584.

[34] BIPM, IEC, IFCC, ISO, IUPAC, IUPAP, OIML, *Guide to the Expression of Uncertainty in Measurement*, ISO, 1995.