

Applicazione di tecniche di machine learning ai log degli accessi della piattaforma Infostud per analisi e predizione di anomalie

Ingegneria dell'informazione, informatica e statistica
Informatica

Anthony Di Pietro
Matricola 1960447



Relatore
Prof. Tolomei Gabriele



Correlatore
Prof. Panizzi Emanuele

Anno Accademico 2022/2023

**Applicazione di tecniche di machine learning ai log degli accessi della piattaforma
Infostud per analisi e predizione di anomalie**
Sapienza Università di Roma

© 2023 Anthony Di Pietro. Tutti i diritti riservati

Questa tesi è stata composta con L^AT_EX e la classe Sapthesis.

Email dell'autore: dipietro.1960447@studenti.uniroma1.it

*Dedicato a
Martina*

*Quando piove sotto gli alberi non piove
Quando fuori smette è sotto gli alberi che piove*

— Tarek Iurcich

Sommario

Le serie temporali multivariate rappresentano un campo di studio di notevole rilevanza in svariati sistemi distribuiti in tutto il mondo. In un certo senso, queste serie di dati delineano il ciclo di vita del sistema. Si può dunque dedurre che attraverso le serie temporali vengono modellati sia i periodi in cui il sistema funziona normalmente, che dualmente i periodi più "anomali".

L'analisi delle serie temporali multivariate, con l'obiettivo di identificare e predire anomalie, non deve solamente tener conto della dipendenza temporale, un problema già complesso di per sé, ma deve anche considerare la correlazione tra i vari segnali all'interno del sistema.

Nel corso degli anni, diversi algoritmi sono stati impiegati per l'analisi di anomalie nelle serie temporali, tra cui SVM[3] (Support Vector Machine), ma con risultati insoddisfacenti. Questi modelli presentano una limitazione fondamentale che compromette la loro efficacia nell'analisi delle serie temporali: la loro incapacità di catturare la correlazione temporale, una caratteristica essenziale per condurre un'analisi accurata di tali serie. Modelli come ARMA[4], invece, riescono a catturare la dipendenza temporale, ma sono per natura monovariati, per cui non sono in grado di osservare l'intercorrelazione dei segnali.

La ricerca sull'individuazione delle anomalie, negli anni recenti, ha fatto molti passi avanti. Algoritmi come Telemanom[1] vengono applicati a sistemi reali, complessi e delicati con ottimi risultati.

L'anno 2019 è stato di particolare interesse per l'analisi di anomalie di serie temporali multivariate, grazie alla pubblicazione dell'articolo scientifico sul modello Multi-Scale Convolutional Recurrent Encoder-Decoder[2] (MSCRED). MSCRED affronta la necessità di misurare non solo l'intercorrelazione tra i dati, ma anche la loro dipendenza temporale.

In questa tesi, verrà applicato MSCRED su un sistema reale che riveste un ruolo di straordinaria importanza per tutta la comunità accademica della Sapienza: il prezioso dataset di Infostud, piattaforma che rappresenta una risorsa inestimabile per la nostra istituzione, essendo un punto di riferimento fondamentale per le attività accademiche e amministrative.

Verranno valutate le prestazioni di MSCRED, confrontando i risultati ottenuti con quelli generati dagli algoritmi OC-SVM, ARMA e Telemanom. È importante sottolineare che MSCRED, grazie alla sua capacità di analisi delle serie temporali multivariate, si prevede porterà a risultati notevolmente superiori rispetto agli altri modelli. Questa analisi rappresenta un passo significativo verso un possibile miglioramento delle operazioni e dell'efficienza del sistema Infostud.

Ringraziamenti

*Desidero esprimere la mia profonda gratitudine al **Professor Gabriele Tolomei**, che mi ha concesso l'opportunità di immergermi nell'ambito della scienza dei dati a livello pratico. La data science rappresenta una disciplina informatica per la quale ho sempre nutrito una profonda passione, e non potevo davvero chiedere di meglio.*

*Non posso non menzionare il contributo fondamentale del **Professor Emanuele Panizzi** e del **Dottor Enrico Bassetti**, che hanno generosamente messo a mia disposizione il dataset relativo ai log degli accessi di Infostud, consentendomi così di applicare modelli su un set di dati concreto ed estremamente interessante.*

*Infine, ma non meno importante, desidero esprimere la mia riconoscenza al collega **Edoardo Gabrielli**, il quale mi ha guidato e sostenuto in tutto il corso del mio tirocinio, fornendomi preziosi consigli e un costante supporto morale. La sua preziosa assistenza è stata fondamentale per il mio percorso di ricerca e studio.*

Indice

1	Introduzione: anomaly detection	1
2	Il dataset di Infostud	5
3	Metodologie di Analisi	11
3.1	Approccio Statistico (Statistical-based)	12
3.1.1	Auto Regressive Moving Average	12
3.2	Approccio di Apprendimento Automatico (Machine Learning-based)	15
3.2.1	One-Class Support Vector Machine	15
3.3	Approccio di Apprendimento Profondo (Deep Learning-based)	17
3.3.1	Telemanom	17
3.3.2	Multi-Scale Convolutional Recurrent Encoder-Decoder	20
4	Valutazione	25
4.1	Metriche di Valutazione	25
4.1.1	Precisione (Precision)	25
4.1.2	Richiamo (Recall)	26
4.1.3	F1-Score (F1)	26
4.2	Valutazione del modello statistico ARMA	26
4.3	Valutazione del modello ML OC-SVM	27
4.4	Valutazione del modello DL Telemanom	27
4.5	Valutazione del modello DL MSCRED	29
5	Conclusioni: un passo avanti verso soluzioni migliori	31

Capitolo 1

Introduzione: anomaly detection

I sistemi complessi sono ubiqi nell'industria moderna e nei servizi dai quali tutti noi traiamo vantaggio quotidianamente. La complessità di tali sistemi i quali, di norma, prevedono la ricezione e l'elaborazione di un certo input, adempiono a task potenzialmente ortogonali e, spesso, processano dati potenzialmente eterogenei ricavati dalla grande quantità di dispositivi e sensori connessi, suscita inquietudini su potenziali anomalie che possono avere fonti e natura varie; che sia per il malfunzionamento di uno dei componenti propri del sistema o dei suoi dispositivi connessi, o deviazioni nella natura del dominio del sistema.

In linea generale, i dati estratti delineano il ciclo di vita del sistema ed è cruciale identificare tempestivamente qualsiasi irregolarità per identificare la radice del problema e contestualmente prevenire o minimizzare perdite; perdite la cui rilevanza può essere di natura finanziaria, dove pochi minuti di malfunzionamenti potrebbero tradursi in danni economici di cifre a sei o sette zeri. In sistemi ancora più delicati, come quelli presenti su un aereomobile, tali perdite potrebbero persino mettere a rischio la vita dei passeggeri e dell'equipaggio di volo.

Di solito, le serie temporali multivariate presentano un certo grado di rumore nelle applicazioni del mondo reale. Pertanto, i modelli concepiti per affrontare il problema della rilevazione di anomalie devono essere sviluppati prendendo in seno tale eventualità. Sarebbe auspicabile che gli algoritmi di detection potessero definire una misura numerica dei punti anomali basata sulla gravità degli eventi che hanno originato tali anomalie. In questo modo, le aziende e gli operatori potrebbero agire in modo relativo e dinamico per risolvere il problema e, potenzialmente, prevenire l'insorgere di un'anomalia prima che essa si manifesti.

I ricercatori hanno esplorato e implementato modelli incentrati sui dati per l'identificazione e l'analisi di anomalie e negli ultimi anni la questione, grazie a tecniche avanzate di machine learning e deep learning e amplificata dalla moltitudine di dati collezionati, è diventata sempre più saliente.

Le anomalie possono essere suddivise in tre macrocategorie:

1. **Anomalie a punto** che corrispondono a singoli dati nel tempo considerati "outlier" poiché si discostano significativamente dal comportamento standard del resto del dataset. La Figura 1.1 illustra un esempio di un dataset in cui si è verificata un'anomalia di tipo puntuale. Possiamo notare che un'osservazione,

quella evidenziata in rosso, ha un valore notevolmente superiore rispetto al resto del dataset.

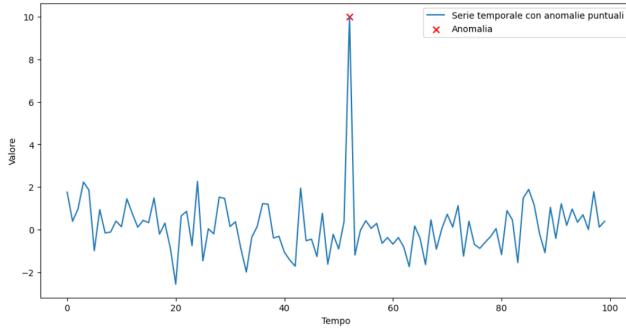


Figura 1.1. Dataset sintetico con anomalia a punto. Il dataset, ad esempio, potrebbe rappresentare il grado di performance di un atleta in un intervallo di cento giorni. L'anomalia puntuale potrebbe indicare un errore nell'inserimento dei dati.

2. Anomalie contestuali le quali rappresentano punti, o sequenze di punti, differenti o distanti dal resto del dataset ma solo se presi in considerazione in un contesto specifico, che sia spaziale o temporale. La Figura 1.2. fornisce un esempio di due anomalie di un dataset sintetico che rappresenta l'indice di temperatura in una finestra temporale lunga due anni. Questi punti, quando considerati nel loro contesto temporale, risultano essere anomalie; tuttavia se analizzassimo complessivamente il dataset senza tener conto dell'ordinamento temporale, essi non sembrerebbero affatto fuori dall'ordinario.

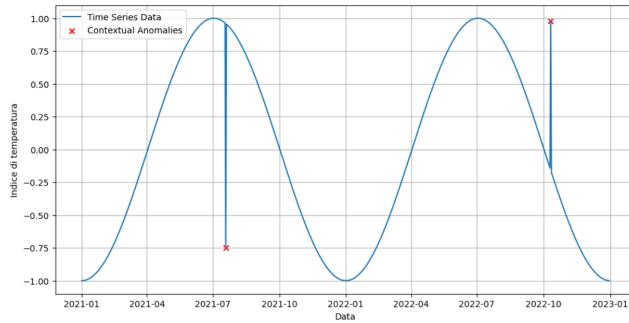


Figura 1.2. Dataset sintetico con anomalie contestuali. Il dataset rappresenta un indice di temperatura lungo due anni. Le anomalie contestuali nell'esempio registrano un grado di temperatura molto più bassa della norma a luglio e una temperatura molto più alta a ottobre.

3. Anomalie collettive si riferiscono a modelli che coinvolgono insiemi di osservazioni divergenti rispetto al resto del dataset. Per esempio, la Figura 1.3. presenta una serie temporale contenente una sottoserie temporale evidenziata in rosso, la quale si differenzia dalle altre sottoserie. In un contesto ipotetico, questa situazione potrebbe rappresentare un contatore bloccato con problemi nell'aggiornamento dei dati.

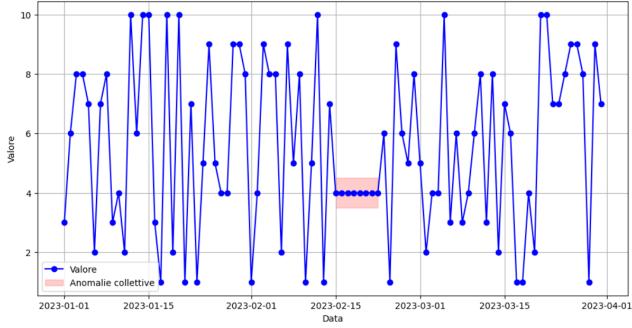


Figura 1.3. Dataset sintetico con anomalie collettive. Il dataset può essere visto come la rappresentazione di un contatore il quale, nel momento in cui avvengono le anomalie collettive, rimane bloccato.

L’individuazione delle anomalie nell’ambito del machine learning, come chiaramente illustrato nei precedenti esempi, è influenzata da una serie di fattori. Se tali fattori non vengono considerati in fase iniziale, potrebbero condurre a un’analisi poco accurata o addirittura errata. Come si sottolinea in modo particolare nella Figura 1.2. e nella Figura 1.3., i modelli devono tener conto della dipendenza temporale delle osservazioni. Nella Figura 1.4. è mostrato un semplice esempio di dipendenza temporale: il prezzo delle azioni a tempo t dipende dal prezzo al tempo $t - 1$ e influenza il prezzo del tempo $t + 1$. I modelli, inoltre, devono essere sviluppati sulla base che i dati su cui saranno addestrati possono vivere in contesti multivariati; è bene che gli algoritmi osservino i vari segnali di un dataset prendendo in considerazione la loro correlazione.

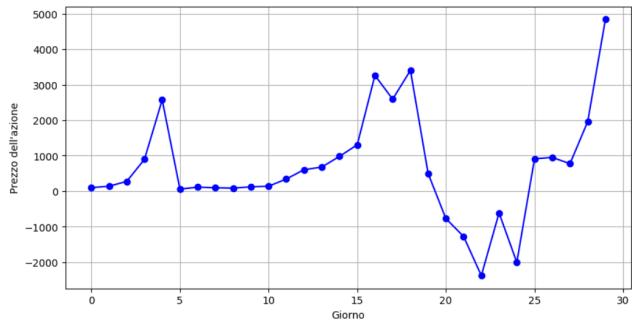


Figura 1.4. Dataset sintetico con dipendenza temporale rappresentante un prezzo di un’azione nel tempo. La dipendenza temporale è dovuta al fatto che il prezzo in un generico tempo t influisce sul prezzo al tempo $t + 1$.

Il problema della rilevazione delle anomalie può essere affrontato mediante un approccio supervisionato, il cui obiettivo è classificare in modo arbitrario le osservazioni anomale da quelle non anomale. La peculiarità di questo approccio è che le anomalie, per essere definite tali, rappresentano una minoranza all’interno del dataset.

Un approccio non supervisionato è preferibile per il problema dell’individuazione delle anomalie. Algoritmi diversi che utilizzano metodologie varie, come quella statistica con Auto Regressive Moving Average[4] (ARMA) e quella del machine

learning per algoritmi come One-Class Support Vector Machine[3] (OC-SVM), o il deep learning con Telemanom[1] e MSCRED[2] cercano di discriminare le anomalie dalle osservazioni non anomale senza la necessità di disporre delle etichette ground-truth, ma basandosi esclusivamente sull'analisi della serie temporale di interesse. Questo metodo, oltre a essere più idoneo, semplifica notevolmente la fase di preparazione dei dati escludendo la costruzione delle etichette.

La tesi è strutturata in cinque capitoli.

1. Nel primo capitolo, quello corrente, viene fatta una piccola paronamica riguardo al problema dell'analisi delle anomalie.
2. Nel secondo capitolo vengono osservate le metriche e la finestra temporale del dataset di Infostud su cui si sono svolti gli esperimenti e vengono giustificate le ragioni per cui è stato scelto quel dataset in particolare.
3. Il terzo capitolo illustra le varie metodologie applicate per la risoluzione del problema dell'anomaly detection al prezioso dataset di Infostud, per auspicabilmente risolvere e tracciare anomalie ai fini del miglioramento della piattaforma più cara a tutta la comunità accademica de La Sapienza. Verranno illustrate le applicazioni di tre approcci diversi: statistico, machine learning e deep learning.
4. Nel capitolo quattro verranno presentate le metriche di valutazione ai fini di giudicare le performance delle soluzioni dei vari algoritmi. Nello stesso capitolo, verranno anche mostrati i risultati dei vari modelli utilizzati.
5. Nell'ultimo capitolo sono riportate e confrontate le conclusioni dei modelli analizzati.

Tutti gli esperimenti citati in questa relazione sono disponibili su GitHub¹ al link riportato a piè di pagina.

¹https://github.com/Pikarz/tirocinio_infostud

Capitolo 2

Il dataset di Infostud

Il dataset preso in analisi, relativo a Infostud, gentilmente fornito dal Professor E. Panizzi e dal Dottor E. Bassetti, riveste un'enorme importanza per l'intera comunità accademica de La Sapienza.

Il dataset rappresenta svariate metriche raccolte dalle piattaforme InfoSapienza e dalle richieste inoltrate attraverso Infostud alla piattaforma esterna GOMP. I dati vengono utilizzati ai fini di monitoraggio e sono raccolti attraverso il software Prometheus.

Il dataset in dettaglio La finestra temporale di osservazione dei dati su cui si sono svolti gli studi è lunga poco più di quattro anni, da gennaio 2019 a febbraio 2023, con una frequenza di campionamento di circa una misurazione ogni 15 secondi per le metriche più dense. Il Dottor Bassetti ha anche tenuto traccia di una buona fetta di anomalie effettivamente verificatesi su Infostud nella stessa finestra temporale, garantendo quindi un prezioso ambiente supervised che ha consentito di avere riscontri numerici sulle performance dei modelli in analisi.

Nel complesso, il dataset è composto da settecentocinquanta file; tali file sono suddivisi per piattaforma, ovvero GOMP e InfoSapienza, e tipologia di metrica. Ciascun file rappresenta circa trenta giorni di osservazioni. Ai fini dell'analisi effettuata con i modelli è stata presa in considerazione la parte dei dati strettamente appartenenti a Infostud.

Di seguito sono riportate in dettaglio le informazioni delle varie tipologie di metriche osservate lato InfoSapienza:

1. **http_responses** Contatore di richieste HTTP in cui ogni segnale rappresenta un codice di stato riscontrato dalla richiesta. Ad esempio, il segnale 200 rappresenta le richieste elaborate con successo (codice di stato OK), mentre 401 rappresenta un errore e comunica che l'autenticazione HTTP non è riuscita (codice di stato Unauthorized). Il contatore è monotono salvo eventuali reset a cadenza non regolare.
2. **requests_total** Contatore delle richieste totali effettuate a InfoSapienza. Anche qui, il contatore è monotono tra due reset.
3. **phoenixws_requests_delay_bucket** Ogni segnale rappresenta un bucket a cui è associato un certo valore *le* (*less or equal*). Il bucket accumula le latenze delle

richieste ai servizi InfoSapienza. Ad esempio, se al tempo t avvengono x richieste a y servizi InfoSapienza e tali richieste vengono risposte con latenza minore o uguale al valore le , allora tale bucket avrà un valore pari a x . È bene notare che i bucket non sono disgiunti e il numero di segnali non è uniforme tra i file. I segnali, in generale, sono sull'ordine delle centinaia.

4. `phoenixws_requests_delay_count` Contatore di richieste processate a servizi. Ogni segnale è un tipo di servizio diverso come, ad esempio, la ricerca degli esami o il login. In genere, questi file contengono diciotto segnali diversi.
5. `phoenixws_requests_delay_sum` Somma delle latenze delle richieste a servizi. I servizi osservati sono gli stessi dei file di tipologia `phoenixws_requests_delay_count`.

In Tabella 2.1. è illustrato un riassunto delle varie metriche del dataset.

Tabella 2.1. Tipologie di metriche lato Infosapienza.

Metrica	Dettagli
<code>http_responses</code>	Contatore in cui ogni segnale è associato a un tipo di richiesta HTTP.
<code>request_total</code>	Contatore totale delle richieste HTTP.
<code>phoenixws_requests_delay_bucket</code>	Insieme di bucket. A ogni bucket è associato un valore le e tale bucket accumula le latenze inferiori o uguali a le .
<code>phoenixws_requests_delay_count</code>	Contatore di richieste processate a servizi. Ogni segnale rappresenta un tipo di servizio.
<code>phoenixws_requests_delay_sum</code>	Somma delle latenze delle richieste a servizi. Ogni segnale rappresenta un tipo di servizio.

Trasformazione delle etichette Le etichette sono state provvedute sottoforma di file. Ogni file rappresenta un certo intervallo in cui si è verificata un'anomalia. I file hanno anche altre informazioni riportate nella Tabella 2.2

Poiché una certa fetta di dataset può contenere più punti all'interno dei vari intervalli di tutti i file delle etichette, è stato creato un dataset binario che mappa un certo timestamp del dataset a un valore di verità: vero se quel timestamp è all'interno di uno degli intervalli dei file delle etichette, ovvero quel timestamp è all'interno di un periodo anomalo del sistema, falso altrimenti.

Tabella 2.2. Le informazioni dei file delle etichette.

Informazione	Significato
<code>Title</code>	Tipo di anomalia, ad esempio ‘Irrangiabilità Infostud’.
<code>Date</code>	Data di inizio dell’anomalia.
<code>resolved</code>	Valore binario che indica se l’anomalia è stata risolta o no.
<code>resolvedWhen</code>	Data di termine dell’anomalia.
<code>Severity</code>	Grado di severità del problema.
<code>affected</code>	Servizi o insiemi di servizi che hanno subito malfunzionamenti

La scelta del tipo di dataset I primi esperimenti si sono basati sui file di tipologia `http_responses`. Questi esperimenti non hanno portato a soluzioni di particolare rilievo per ciascun modello applicato, ma sono stati fondamentali per familiarizzare con il dataset. La seconda trincea di esperimenti si è focalizzata su tre tipologie di file diversi: `phoenixws_requests_delay_bucket`, `phoenixws_requests_delay_count` e `phoenixws_requests_delay_sum`. È banale che i bucket, essendo la categoria predominante nel dataset, hanno comportato un’elevata esposizione dei modelli a tali dati rispetto che alle altre metriche. Difatti gli esperimenti che si sono basati su questi dati non hanno portato soluzioni interessanti.

Dopo questi tentativi iniziali, che hanno comunque contribuito a una migliore comprensione dei dati e dei modelli, è stata presa la decisione, in collaborazione con il relatore, il Professore G. Tolomei, di concentrarsi sugli esperimenti successivi utilizzando un dataset ibrido: abbiamo osservato che, avendo a disposizione la somma delle latenze fatte ai servizi di Infosapienza `phoenixws_requests_delay_sum` e il relativo contatore `phoenixws_requests_delay_count`, era possibile creare il dataset che rappresentasse la latenza media di ogni richiesta per ogni servizio. Si prevedeva che questo approccio potesse portare a risultati migliori nell’analisi e nell’individuazione di anomalie.

La scelta della finestra temporale In virtù delle etichette e del nuovo dataset, dopo un’attenta analisi, gli esperimenti successivi si sono focalizzati su diverse fette temporali che hanno dovuto soddisfare alcune proprietà chiave:

1. i dati devono rientrare in una finestra di osservazione all’interno di quella delle etichette, le quali non sono disponibili per tutto il dataset;
2. la percentuale di anomalie riscontrate deve essere intorno al 5% per garantire al modello di essere addestrato con un numero sufficiente di anomalie, ma non troppe per non rischiare che esso classifichi fluttuazioni anomale come normali;
3. la suddivisione del dataset in insiemi di training, validazione e test deve garantire a tutti e tre i sottoinsiemi di contenere anomalie ed è auspicabile che

la percentuale di anomalie sia quantomeno simile per tutti e tre i sottoinsiemi di dati;

4. i dati devono essere poco vuoti, ovvero non devono contenere troppi valori nulli per preservare il più possibile l'integrità originale dei dati. Questo perché i modelli, per essere addestrati, necessitano di un dataset privo di valori nulli i quali, purtroppo, sono inevitabilmente presenti;
5. per limiti computazionali i dati non devono contenere troppi punti, ma al più devono basarsi su tre mesi di osservazioni.

Tabella 2.3. Riassunto delle proprietà del dataset su cui verranno basati gli esperimenti.

Proprietà	Motivazione
L'intera finestra temporale delle osservazioni prese in considerazione deve essere inclusa in quella delle etichette.	Consente di avere riscontri numerici sulle performance.
Percentuale di anomalie $\approx 5\%$.	Le anomalie devono essere presenti e non devono occupare una grande parte di dataset.
La percentuale di anomalie negli insiemi di training, validation e test deve essere simile e sempre maggiore di zero.	I tre insiemi devono osservare una percentuale simile di anomalie.
I dati non devono contenere molti valori nulli.	Mantiene l'integrità del dataset.
Al massimo tre mesi di osservazioni.	Per soddisfare limiti computazionali.

Pre-processamento dei dati Per la creazione del dataset delle latenze medie è necessaria una divisione elemento per elemento tra i due dataset `phoenixws_requests_delay_sum` e `phoenixws_requests_delay_count`.

Le proprietà da soddisfare, illustrate nella Tabella 2.3., sono varie. Per composizione del dataset, molto problematico è il punto in cui è espressa la necessità che il dataset preso in analisi non debba contenere molti valori nulli, e per far sì che venga rispettato è stato doveroso eliminare le righe e le colonne più vuote dei dataset presi in considerazione.

I dataset più interessanti che hanno soddisfatto tali proprietà, in seno all'eliminazione delle righe e colonne più vuote, sono tre le cui informazioni sono illustrate nella Tabella 2.4.

Tabella 2.4. Statistiche del dataset suddiviso in finestre temporali.

Finestra temporale	Anomalie (%)	Valori nulli (%)
8 agosto - 11 novembre 2019	6%	15%
23 febbraio - 24 marzo 2020	2.1%	16%
23 maggio - 22 giugno 2020	5.6%	14.5%

Tali porzioni di dataset, purtroppo, come illustrato, ancora presentano valori nulli. Per poter permettere ai modelli di essere applicati, è necessario sostituire i valori nulli con un valore che si presti bene al problema. In questi studi, i valori nulli sono stati sostituiti con la media relativa a ogni segnale.

Gli esperimenti sono stati condotti principalmente utilizzando il dataset relativo al periodo dal 23 maggio al 22 giugno 2020, ma tutti i modelli possono essere applicati in maniera analoga a tutti i dataset menzionati, con l'adeguamento degli iperparametri.

Il dataset preso in analisi ha visto la rimozione di un circa un quarto delle righe e un terzo delle colonne meno dense, da diciotto a dodici.

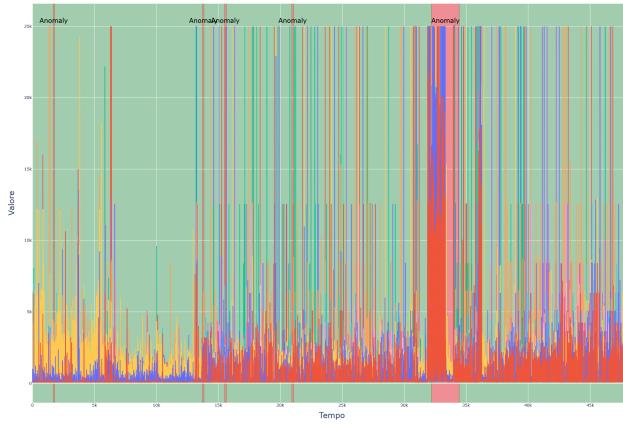


Figura 2.1. Dataset 23 maggio - 22 giugno 2020. Le bande rosse rappresentano le anomalie ground-truth, mentre quelle verdi i periodi non anomali.

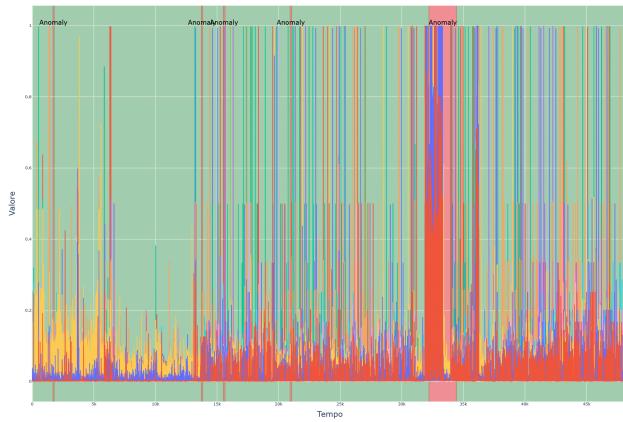


Figura 2.2. Dataset 23 maggio - 22 giugno 2020 normalizzato senza valori vuoti. Le bande rosse rappresentano le anomalie ground-truth, mentre quelle verdi i periodi non anomali.

Visualizzazione del dataset preso in analisi Il dataset su cui si basano gli esperimenti dei capitoli successivi è illustrato in Figura 2.1.

Possiamo osservare le anomalie che si sono presentate nelle aree in cui lo sfondo del grafico è rosso, mentre le aree verdi rappresentano i periodi non anomali. È lampante che, nell'ultima anomalia, quella più grande tra tutte le altre, i segnali

sono molto più alti. Ci aspettiamo che un modello segnali quei periodi come anomali. In Figura 2.2. possiamo osservare il dataset normalizzato con i valori vuoti sostituiti dalla media relativa al segnale associato.

È un privilegio poter affermare che l'applicazione di modelli metodologicamente diversi sul dataset di Infostud, indispensabile per la comunità Sapienza, rappresenta una straordinaria opportunità per gli studi intrapresi dal sottoscritto.

Capitolo 3

Metodologie di Analisi

Ai fini di valutare le performance delle soluzioni dei vari modelli applicati sul dataset che rappresenta la latenza media delle richieste a servizi fatti a Infostud, viene proposto l'utilizzo di tre principali approcci metodologici per l'analisi atta all'individuazione di anomalie nel dataset accademico. Le soluzioni verranno analizzate in virtù delle etichette dapprima esistenti sul dataset:

1. **Approccio Statistico (Statistical-based):** L'approccio statistico si basa sulla tradizionale statistica dei dati. In particolare ARMA[4], il modello utilizzato per fare inferenza sulle anomalie del prezioso dataset di Infostud, modella le relazioni temporali nei dati attraverso l'uso dell'autoregressione e della media mobile, metodi adatti al fine di modellare le relazioni temporali nei dati; il modello identifica tendenze e autocorrelazioni nelle metriche applicate.
2. **Approccio di Apprendimento Automatico (Machine Learning-based):** Nell'ambito dell'apprendimento automatico, è stato scelto l'approccio basato sulle Support Vector Machine (SVM). In particolare, la Support Vector Machine a classe singola (OC-SVM[3]), un algoritmo che si concentra sull'identificazione di dati fuori dal normale comportamento in grado di tracciare confini decisionali creando due regioni che discriminano, nel caso in questione, le anomalie dalle non anomalie.
3. **Approccio di Apprendimento Profondo (Deep Learning-based):** L'apprendimento profondo è particolarmente efficace nell'affrontare problemi di analisi di dati complessi, come le serie temporali. Questo approccio vede applicati due modelli molto recenti: Telemanom[1] e Multi-Scale Convolutional Recurrent Encoder-Decoder (MSCRED[2]); entrambi sono modelli basati su reti neurali e sono progettati con lo specifico obiettivo di individuazione delle anomalie in dati multivariati.

Ciascuno di questi approcci contribuisce in modo significativo all'analisi del dataset indispensabile per tutta la comunità Sapienza, fornendo una varietà di strumenti e metodi per esplorare, valutare e interpretare le informazioni contenute nei dati di Infostud. Nel corso di questo capitolo, verranno esaminate in dettaglio le soluzioni di ciascuno dei modelli presi in considerazione.

3.1 Approccio Statistico (Statistical-based)

3.1.1 Auto Regressive Moving Average

In questa sezione esploriamo l'uso del modello univariato ARMA[4] (Auto Regressive Moving Average) come modello di riferimento che adotta un approccio statistico per valutare e confrontare l'efficacia del modello MSCRED[2]. La scelta del modello ARMA è stata motivata principalmente dalla sua utilizzazione da parte dei ricercatori di MSCRED per valutare le prestazioni di quest'ultimo. In generale, ARMA è un modello ben consolidato e ampiamente utilizzato nella letteratura grazie alla sua semplice implementazione e intuitività.

Tuttavia, la sua semplicità ha un costo: il modello non è in grado di catturare l'intercorrelazione tra i segnali perché è monovariato per natura. Nell'esperimento presentato, per adattarsi alla natura monovariata del modello, è stato selezionato il segnale meno vuoto dal dataset di Infostud descritto nel Capitolo 2, ovvero il segnale con il minor numero, solo dieci, di valori nulli, avendo quindi a disposizione un totale di 48327 osservazioni. Il segnale interessato rappresenta la latenza media delle richieste di login a InfoSapienza.

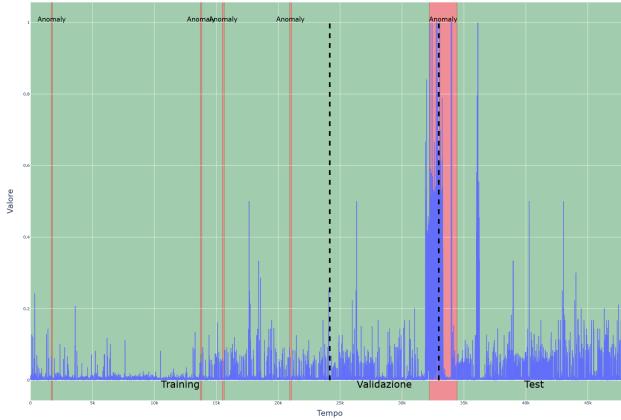
Intuizione ARMA, nelle serie temporali, è utilizzato principalmente per la previsione di dati, ma può essere usato anche per la rilevazione di anomalie. In linea generale, data una previsione di ARMA, i punti in cui la previsione si discosta maggiormente rispetto ai punti originali vengono considerati anomalie.

Più in dettaglio, ARMA effettua una certa previsione P basata sui dati di training. Tale previsione viene confrontata con i valori ground-truth G calcolando i residui $P - G$. Attraverso l'iperparametro t vengono calcolati i confini decisionali c^-, c^+ , ogni punto i è valutato come anomalia se $P_i - G_i \notin [c^-, c^+]$, e non è valutato come tale altrimenti. I confini decisionali c^-, c^+ sono ottenuti come segue: siano μ la media dei valori dell'unico segnale del dataset e sia σ la sua deviazione standard: $c^+ = \mu + t^* \cdot \sigma, c^- = \mu - t^* \cdot \sigma$

Dataset Il dataset di Infostud è stato suddiviso in tre insiemi: training, validation e test, con percentuali rispettivamente pari a 0.5, 0.182, 0.318. Allo scopo di garantire una distribuzione accettabile delle anomalie nei tre insiemi, questa suddivisione atipica è stata necessaria a causa della natura delle anomalie nel dataset, che sono principalmente di tipo collettivo e quindi vicine tra loro sia nel tempo che nello spazio. Nel dataset, come già annunciato, è stato preso in considerazione un solo segnale, quello che rappresenta la latenza media delle richieste di login effettuate a InfoSapienza, i cui dieci punti originariamente nulli sono stati rimossi. Nella Tabella 3.1. sono riportate le informazioni in dettaglio della suddivisione del dataset e nella Figura 3.1. è illustrato il dataset.

Tabella 3.1. Statistiche dataset ARMA (latenza media delle richieste di login).

Dataset	Indice di split	# punti	Anomalie (%)
Training	0.5	24163	2.01 %
Validation	0.182	8796	8.59 %
Testing	0.318	15368	9.59 %

**Figura 3.1.** Dataset delle latenze medie alle richieste di login nel periodo 23 maggio - 22 giugno 2020. Le bande rosse rappresentano le anomalie ground-truth, mentre quelle verdi i periodi non anomali. Le linee tratteggiate delimitano il dataset di training da quello di validazione e quello di validazione dal test.

Iperparametri ARMA, in generale, presenta due iperparametri p, q . p rappresenta l'ordine dell'autoregressione nel modello ARMA. L'autoregressione si riferisce al fatto che il valore corrente della serie temporale dipende dai valori passati della stessa serie temporale, in sintesi p indica quanti periodi temporali passati vengono utilizzati per predire il valore corrente ed è l'iperparametro che fa sì che ARMA possa catturare le dipendenze temporali nella serie temporale. q , invece, rappresenta l'ordine della media mobile nel modello. Essa indica che il valore della corrente serie temporale dipende dai valori passati dei residui, ovvero degli errori previsionali, del modello stesso. Intuitivamente, q indica quanti periodi temporali passati degli errori previsionali vengono utilizzati per predire il valore corrente.

Ai fini dell'anomaly detection, ARMA presenta anche un terzo iperparametro t che, nel caso degli studi effettuati, è attualmente a creare i confini decisionali dei punti non anomali e, dualmente, anomali, definendo la sensibilità del modello.

Per la scelta degli iperparametri è stata effettuata una grid-search di 30 epoch su p, q e t al fine di individuare la combinazione che massimizzasse lo score F1, metrica illustrata nella Sottosezione 4.1.3, sul dataset di validation attraverso il training sul train set.

Formalmente, siano \mathbf{Tr} , \mathbf{Va} i dataset utilizzati per il training e per il validation rispettivamente, e sia $\text{ARMA}_{\mathbf{X}}^{\mathbf{Y}}$ un modello ARMA addestrato su \mathbf{X} che effettua previsioni nei punti \mathbf{Y} e sia T_{10} uno spazio lineare composto da elementi equidistanti

t_i tali che $t_i \in [0.1, 5.0] \forall i \in [10]$:

$$\begin{aligned} p^*, q^*, t^* &= \max_{p,q,t} F1(\text{ARMA}_{\text{Tr}}^{\text{Va}}(p, q, t)) \\ \text{con il vincolo } p, q &\in [30], t \in T_{10} \end{aligned} \quad (3.1)$$

Soluzione L'equazione 3.1 è stata soddisfatta dai seguenti:

- $p^* = 20$
- $q^* = 8$
- $t^* = 0.644$

Poiché l'insieme di validazione era necessario solo ai fini della ricerca degli iperparametri, è stato successivamente accorpato al dataset di training. Dopo l'addestramento del modello sul nuovo dataset con gli iperparametri che hanno generato una soluzione ottimale sul validation, sono state effettuate le previsioni della serie temporale di test, mostrate nella Figura 3.2.

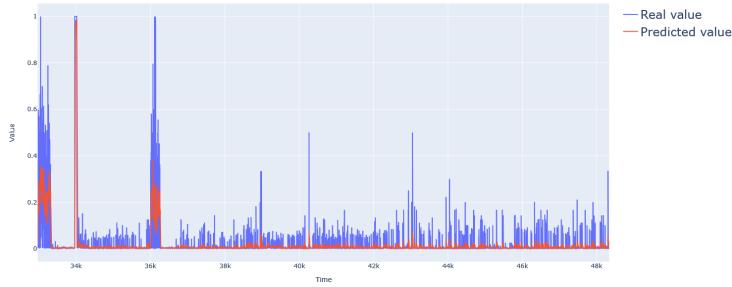


Figura 3.2. Confronto della previsione del modello e le osservazioni originali sull'insieme di test.

È evidente che il modello è in grado di prevedere l'andamento generale della serie temporale per la maggior parte del dataset. Tuttavia, in alcune aree, il modello presenta una performance inferiore rispetto ad altre. Queste aree saranno cruciali per la classificazione dei punti come anomali o non anomali.

Sulla base delle previsioni P , sono stati calcolati i residui rispetto al test set $P - G$ e valutati come punti anomali quelli che soddisfano $p_i - g_i \notin [c^-, c^+] | p_i \in P, g_i \in G$. I residui sono illustrati nella Figura 3.3.

I punti all'interno dell'intervallo definito dai confini inferiore e superiore non sono considerati anomali. Tra questi punti ci sono quelli blu, che non sono anomalie e che il modello non ha considerato come tali (True Negatives), e quelli viola, che il modello ha erroneamente classificato come non anomalie (False Negatives).

Possiamo inoltre osservare che svariati punti in verde sono stati correttamente classificati come anomalie (True Positives), ma sono presenti anche molti punti rossi, ovvero quelli che il modello ha erroneamente considerato anomalie (False Positives).

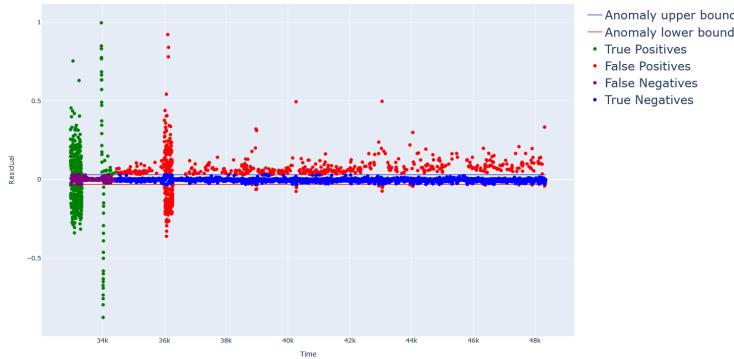


Figura 3.3. I residui del modello ARMA nel test set con i confini decisionali che discriminano la previsione delle anomalie e le non anomalie.

3.2 Approccio di Apprendimento Automatico (Machine Learning-based)

3.2.1 One-Class Support Vector Machine

In questa sottosezione, esaminiamo il modello basato sull'apprendimento automatico One-Class Support Vector Machine[3] (OC-SVM) come altro punto di riferimento per valutare le prestazioni del modello MSCRED[2]. La scelta del modello OC-SVM è motivata dalla sua ampia utilizzazione nella letteratura di riferimento e dal suo impiego da parte dei ricercatori di MSCRED per il confronto delle prestazioni. In generale, gli algoritmi di tipologia Support Vector Machine sono noti per offrire buone prestazioni in diversi contesti. Tuttavia, tendono a soffrire quando si tratta di classificazione di anomalie, poiché le anomalie costituiscono solitamente una piccola parte del dataset, creando uno sbilanciamento tra le classi.

Intuizione OC-SVM, come gli altri algoritmi della famiglia SVM, opera tracciando confini decisionali tra i dati. Nell'ambito della rilevazione delle anomalie, questo si traduce nella definizione di un iperpiano che separa i punti considerati anomali da quelli considerati non anomali.

Dataset Nell'esperimento è stato suddiviso il dataset di riferimento negli insiemi di training, validation e test con le stesse percentuali utilizzate per ARMA, ovvero 0.5, 0.182, 0.318 rispettivamente. La Tabella 3.2. mostra la percentuale di anomalie in ciascun sottoinsieme del dataset originale, mentre nella Figura 3.4. viene evidenziato il dataset partizionato.

Tabella 3.2. Statistiche dataset OC-SVM.

Dataset	Indice di suddivisione	# punti	Anomalie (%)
Training	0.5	24168	2.01%
Validation	0.182	8797	8.60%
Testing	0.318	15372	9.59%

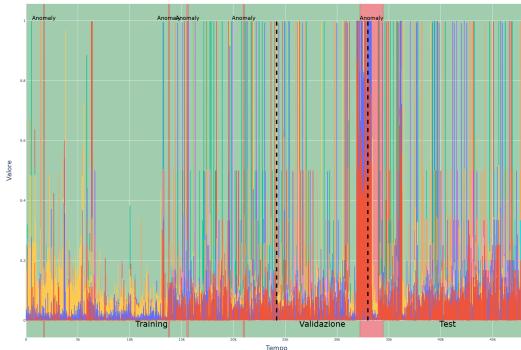


Figura 3.4. Suddivisione dataset negli insiemi di training, validation e test. Le due linee tratteggiate rappresentano, da sinistra a destra rispettivamente la suddivisione tra l'insieme di training e validazione e tra l'insieme di validazione e test.

Tabella 3.3. Valori candidati per gli iperparametri.

Iperparametro	Candidati
ν	0.03, 0.1, 0.25, 0.5, 0.9
Kernel	linear, poly, rbf, sigmoid

Iperparametri Gli iperparametri ν e il tipo di kernel sono stati trovati effettuando una "grid-search" partendo da un insieme di candidati per entrambi gli iperparametri illustrati nella Tabella 3.3. La combinazione di valori (ν^*, Kernel^*) che ha massimizzato lo score F1 sul validation set è stata scelta come configurazione ottimale.

Formalmente, siano \mathbf{Tr}, \mathbf{Va} i dataset utilizzati per il training e per il validation rispettivamente e sia $\text{OC-SVM}_{\mathbf{X}}^{\mathbf{Y}}$ un modello OC-SVM addestrato su \mathbf{X} che effettua previsioni sulle etichette dei punti \mathbf{Y} e sia K l'insieme dei possibili valori che può assumere il kernel, ossia $K = \{\text{linear, poly, rbf, sigmoid}\}$:

$$\nu^*, \text{Kernel}^* = \max_{\nu, \text{Kernel}} F1(\text{OC-SVM}_{\mathbf{Tr}}^{\mathbf{Va}}(\nu, \text{Kernel})) \quad (3.2)$$

con il vincolo $\nu \in [0.03, 0.1, 0.25, 0.5, 0.9], \text{Kernel} \in K$

Soluzione L'equazione 3.2 è stata soddisfatta dai seguenti valori:

- $\nu^* = 0.9$
- $\text{Kernel}^* = \text{sigmoid}$

Dopo aver utilizzato l'insieme di validazione esclusivamente per l'ottimizzazione degli iperparametri, è stato poi unito al dataset di addestramento. Successivamente, il modello è stato addestrato sul nuovo dataset combinato, utilizzando gli iperparametri che hanno prodotto una soluzione ottimale durante la fase di convalida. In Figura 3.5. viene mostrata la soluzione dopo l'applicazione di PCA per fornire una rappresentazione ai primi tre componenti principali.

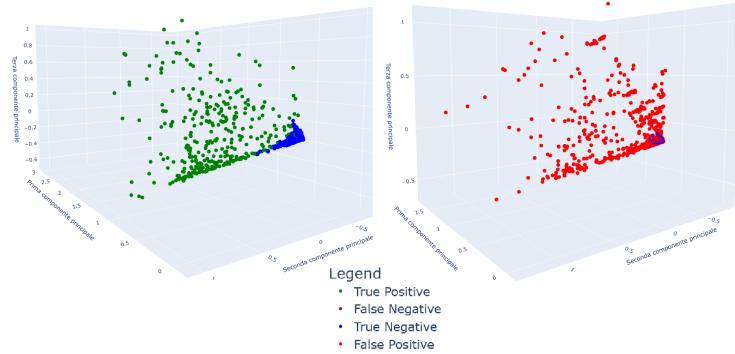


Figura 3.5. Soluzione del modello OC-SVM dopo l'applicazione di PCA ai primi tre componenti principali, garantendo una rappresentazione visibile rispetto a quella originale a dodici dimensioni. A sinistra sono rappresentate le previsioni corrette (TP, TN), mentre a destra quelle errate (FP, FN).

I risultati mostrano chiaramente che il modello ha prestazioni scarse sul dataset di test. I punti verdi a sinistra della Figura 3.5. sono quelli classificati correttamente come anomalie (True Positives) e quelli blu sono quelli classificati correttamente come non anomali (True Negatives). Tuttavia, molti punti sono stati classificati in modo errato, come evidenziato a destra nella stessa figura. I punti viola rappresentano le anomalie erroneamente classificate come normali (False Negatives), mentre i punti rossi sono i punti normali erroneamente classificati come anomalie (False Positives).

3.3 Approccio di Apprendimento Profondo (Deep Learning-based)

3.3.1 Telemanom

Telemanom[1] è una tecnica di rilevazione di anomalie sviluppata dai ricercatori della Nasa nel 2018 ai fini dell'anomaly detection su stazioni e veicoli spaziali. Il modello è stato applicato dai ricercatori sui dati telemetrici dei satelliti e del rover Curiosity e, come MSCRED[2], utilizza reti neurali Long-Short Term Memory (LSTM[5]).

La scelta di utilizzare Telemanom in questa ricerca si è basata sulla necessità di confrontare MSCRED con un modello più recente che sfrutta tecniche avanzate focalizzate sull'apprendimento profondo, concentrandosi esclusivamente sulla rilevazione delle anomalie.

Il modello nasce come miglioramento ai vecchi sistemi di anomaly detection per l'attrezzatura spaziale che, in generale, erano semplicemente basati su valori precisi e predeterminati che, una volta superati, facevano sì che venissero attivate le misure di sicurezza opportune. Telemanom è stato progettato tenendo presente che i dati vivono in un contesto non supervisionato ed è in grado di osservare segnali multipli e analizzare se un certo canale (segnale) si trova in uno stato anomalo o meno individualmente dagli altri segnali. Questo permette di tracciare più facilmente le cause dell'anomalia. Per cui Telemanom tratta un contesto ancora più complesso rispetto a quello in cui vive InfoSapienza, dove le anomalie sono globali per tutti i segnali.

Intuizione Telemanom effettua previsioni creando un modello distinto per ciascun canale di telemetria, questo significa che ogni canale viene trattato in modo indipendente per la previsione. Il modello viene addestrato per predire un certo numero di valori futuri per ciascun canale di telemetria. Durante il processo di previsione, l'errore attuale di previsione $e(t) = y(t) - \hat{y}(t)$, in cui $y(t)$ è il valore ground-truth mentre $\hat{y}(t)$ è il valore previsto dal modello, viene smussato attraverso una media ponderata esponenziale. L'insieme degli errori smussati forma un vettore \mathbf{e}_s .

Dato il vettore \mathbf{e}_s , la soglia di anomalia è calcolata attraverso un approccio che identifica i valori estremi senza supposizioni sulla distribuzione degli errori. Il threshold ϵ è calcolato come segue: $\epsilon = \mu(\mathbf{e}_s) + \mathbf{z}\sigma(\mathbf{e}_s)$ in cui \mathbf{z} è una lista ordinata di valori positivi che rappresentano il numero di deviazioni standard sopra la media $\mu(\mathbf{e}_s)$.

Sia il vettore \mathbf{e}_{seq} che rappresenta gli errori smussati i cui valori $e^{(i)} \geq \epsilon$, ovvero è un vettore che contiene tutti gli errori smussati che hanno il proprio valore maggiore al valore di threshold, la severità dell'anomalia $s^{(i)}$ per ogni punto è calcolata come segue:

$$s^{(i)} = \frac{\max(\mathbf{e}_{seq}^{(i)}) - \arg \max(\epsilon)}{\mu(\mathbf{e}_s) + \sigma(\mathbf{e}_s)}$$

Il rilevamento di anomalie basato sulla previsione dipende in modo significativo dai dati storici utilizzati per calcolare la soglia dinamicamente e valutare gli errori di previsione correnti. Da ciò deduciamo che la mancanza di dati storici può portare a falsi positivi che sono calcolati anomali solo a causa del contesto ristretto in cui essi vengono valutati. Telemanom risolve questo problema applicando una procedura di pruning atta a mitigare i falsi positivi e assume che le anomalie di dimensione simile di solito non si verificano frequentemente nello stesso canale. Queste accortezze aiutano a migliorare la precisione del modello, contribuendo a considerare i comportamenti normali ma rari delle attrezzature spaziali che si verificano a intervalli regolari.

Dataset Telemanom, come già enunciato, prende in considerazione un unico canale alla volta. È quindi necessario suddividere il dataset di Infostud in più parti, una per ogni segnale, per far sì che il modello venga addestrato ed effettui previsioni su tutto il dataset. Gli intervalli anomali associati a ogni canale sono gli stessi, poiché nel caso di InfoSapienza quando avviene un'anomalia, essa è globale per tutti i servizi, quindi per tutti i segnali.

Per l'esperimento è stato utilizzato lo stesso dataset applicato ai modelli precedenti e discusso nel Capitolo 2, ma il cui indice di split è diverso rispetto ai modelli precedentemente analizzati. Questo perché Telemanom, in generale, prevede un'anomalia se il modello osserva come anomalo l'intervallo in cui l'anomalia si verifica. Le varie metriche, nell'implementazione originale che è stata utilizzata per gli esperimenti, vengono quindi basate solo sugli intervalli considerati anomali, non sul numero di punti previsti come anomali come nelle altre applicazioni analizzate. Il dataset di Infostud preso in analisi ha complessivamente cinque intervalli di anomalie distinte e, se prendessimo in considerazione lo split analizzato precedentemente, solo uno sarebbe nell'insieme di test. Per far sì che il modello sia addestrato e testato in maniera più egregia, è stato scelto lo split osservabile nella Tabella 3.4. Il dataset

suddiviso è illustrato nella Figura 3.6., mentre la Figura 3.7. mostra il dataset di addestramento della metrica che rappresenta la latenza media delle richieste di login a InfoSapienza.

Tabella 3.4. Statistiche dataset Telemanom.

Dataset	Indice di split	# punti	# Intervalli anomali
Training	0.43	20784	3
Testing	0.67	27553	2

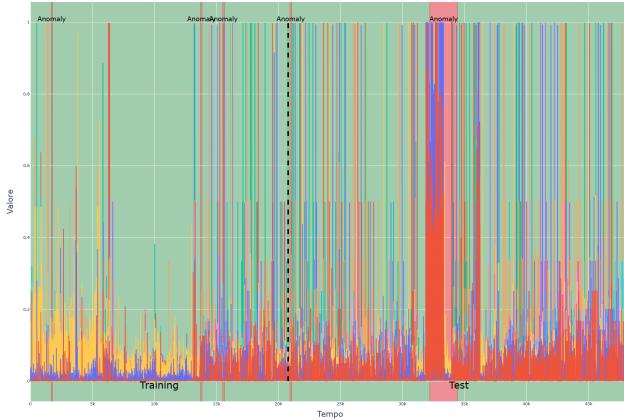


Figura 3.6. Suddivisione dataset negli insiemi di training e test. La linea tratteggiata delimita il dataset di training da quello dedicato al test.

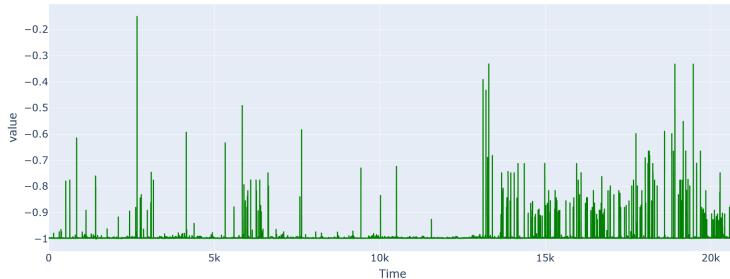


Figura 3.7. Insieme di training di Telemanom.

Non è stato necessario assicurare una parte di dati al validation set perché Telemanom, nella sua implementazione, gestisce da sé la validazione per l'addestramento della sua rete neurale.

Iperparametri Telemanom presenta numerosi iperparametri, tra i quali i più significativi sono riportati nella Tabella 3.5., insieme ai valori empiricamente scelti per questo esperimento.

Tabella 3.5. Descrizione e valori degli iperparametri Telemanom.

Iperparametro	Informazioni	Valore
<i>batch_size</i>	Indica il numero di valori da considerare in ogni batch durante l'addestramento.	100
<i>window_size</i>	Specifica il numero di batch precedenti da utilizzare nel calcolo dell'errore smussato.	110
<i>smoothing_perch</i>	Determina l'indice di smussamento nel calcolo degli errori.	0.1
<i>layers</i>	Vettore bidimensionale che rappresenta il numero di neuroni negli strati nascosti della rete neurale.	[100, 100]
<i>l_s</i>	Numero di passi temporali precedenti a quello attuale su cui verrà basata la previsione futura.	7
<i>n_predictions</i>	Indica il numero di valori futuri da prevedere.	35
<i>validation_split</i>	Specifica la percentuale del dataset di training che verrà usato come validazione per l'addestramento della rete neurale.	0.2

3.3.2 Multi-Scale Convolutional Recurrent Encoder-Decoder

MSCRED[2] è un recente modello avanzato che si concentra sull'identificazione di anomalie all'interno di serie temporali multivariate.

Scelta del modello MSCRED è stato scelto per diverse ragioni; come attestano gli autori del paper, è in grado di catturare informazioni a diverse scale temporali, grazie all'incorporazione di strati convoluzionali. Inoltre, la sua architettura di codifica-decodifica è adatta per l'identificazione di pattern complessi. Fondamentalmente, il modello è stato sviluppato con l'obiettivo specifico di gestire l'intercorrelazione dei segnali e la dipendenza temporale tra di essi. Lo scopo principale è valutare se questo approccio avanzato possa migliorare significativamente il rilevamento delle anomalie rispetto a modelli più tradizionali come ARMA[4] e OC-SVM[3].

Le controversie sul modello L'implementazione originale, disponibile su GitHub¹, non è stata accolta con entusiasmo dalla critica online. Purtroppo, di per sé, il codice non è molto intuitivo e sono presenti dei riferimenti agli ambienti locali dello sviluppatore originale. Inoltre, nell'implementazione originale, le etichette che rappresentano le anomalie nel dataset non sono state fornite, rendendo impossibile replicare i risultati del paper. Tutto questo ha fatto suscitare dubbi riguardo all'effettiva efficacia del modello. Nella repository GitHub² relativa agli studi affrontati in questa relazione viene allegata la versione del codice revisionata e rinnovata a fondo dal sottoscritto con cui sono stati effettuati gli esperimenti.

¹<https://github.com/7fantasysz/MSCRED>

²https://github.com/Pikarz/tirocinio_infostud

Il framework del modello Per quanto riguarda il modello in sé, le "signature matrices" sono una parte fondamentale di MSCRED e svolgono un ruolo chiave nell'analisi delle serie temporali. Queste matrici sono utilizzate per rappresentare le intercorrelazioni tra diverse coppie di serie temporali in un segmento specifico della serie, una proprietà critica che riflette lo stato del sistema.

Le signature matrices sono matrici $n \times n$ dove n è il numero di segnali della serie temporale analizzata. Una certa signature matrix M^t è costruita attraverso il prodotto interno di due serie temporali all'interno del segmento interessato. Formalmente, date due serie temporali $\mathbf{x}_i^w = (x_i^{t-w}, x_i^{t-w-1}, \dots, x_i^t)$ e $\mathbf{x}_j^w = (x_j^{t-w}, x_j^{t-w-1}, \dots, x_j^t)$ appartenenti allo stesso segmento X^w , la loro correlazione $m_{ij}^t \in M^t$ è calcolata secondo l'equazione 3.3.

$$m_{ij}^t = \frac{\sum_{\delta=0}^w x_i^{t-\delta} x_j^{t-\delta}}{w} \quad (3.3)$$

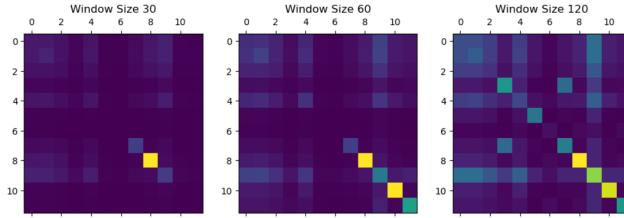


Figura 3.8. Signature matrices.

Banalmente, le signature matrices sono matrici simmetriche. Nell'esperimento, l'intervallo tra due segmenti è $gap_time = 30$ e vengono costruite $s = 3$ signature matrices con grandezze diverse pari a $win_sizes = [30, 60, 120]$ punti temporali. Nella Figura 3.8. è illustrato un esempio di tre signature matrices su cui si basano i risultati dell'esperimento riportati nella Sezione 4.5.

In sintesi, le signature matrices vengono concatenate e il tensore risultante $\chi^{t,0} \in \mathbb{R}^{n \times n \times s}$ viene fornito in input a vari strati convoluzionali. Sia $\chi^{t,l}$ la feature map del livello l -esimo, attraverso un attention based ConvLSTM[5] vengono aggiornati gli strati nascosti delle feature map $\mathcal{H}^{t,l}$ in $\hat{\mathcal{H}}^{t,l}$. In seguito, attraverso un decodificatore convoluzionale, le feature map ottenute al passo precedente vengono decodificate e vengono ricostruite le signature matrices facendo, essenzialmente, il processo inverso: la feature map $\hat{\mathcal{H}}^{t,l}$ viene data in input a una rete neurale deconvoluzionale e l'output, la feature map $\hat{\chi}^{t,l}$, è concatenato con l'output del precedente layer convoluzionale. La concatenazione è poi data in input al prossimo strato deconvoluzionale. L'output finale $\hat{\chi}^{t,0}$ denota le signature matrices ricostruite. La funzione di loss di MSCRED, osservabile nell'equazione 3.4, osserva la differenza tra le signature matrices originali e quelle ricostruite e, nel corso delle epoche di training, punta a minimizzare tale funzione di loss.

$$\mathcal{L}_{MSCRED} = \sum_t \sum_{c=1}^s \|\chi_{::,c}^{t,0} - \hat{\chi}_{::,c}^{t,0}\|_F^2 \quad (3.4)$$

Il dataset La suddivisione del dataset negli insiemi di training, validazione e test ha mantenuto gli indici di suddivisione usati nell'esperimento di OC-SVM, osservabile nella Tabella 3.2.

Iperparametri MSCRED, come già anticipato nei paragrafi precedenti, presenta diversi iperparametri, tra cui *win_sizes* che rappresenta la larghezza delle signature matrices, *gap_time*, che determina la distanza tra i segmenti della serie temporale, ovvero identifica quanto scorre ogni finestra a ogni passo, e *s* che è il numero signature matrices. Inoltre, l'iperparametro *step_max* rappresenta il numero di feature maps precedenti concatenate dal codificatore convoluzionale e *thred_b* e *threshold* regolano la sensibilità del modello nel determinare le anomalie. In particolare, *thred_b* rappresenta un valore di soglia specifico utilizzato dal modello per valutare numericamente l'indice di anomalia degli elementi all'interno dei dati, associando a essi un certo punteggio di anomalia. *threshold*, invece, è attivo a valutare le prestazioni complessive del modello e stabilisce una soglia oltre la quale i punteggi di anomalia vengono etichettati come anomalie.

Dopo un'attenta analisi sono stati scelti per l'esperimento gli iperparametri riportati nella Tabella 3.6. che hanno garantito la migliore soluzione empirica. I restanti iperparametri *thred_b* e *threshold*, sono stati ottimizzati tramite una grid-search atta a massimizzare la metrica F1.

Tabella 3.6. Iperparametri MSCRED (parte 1.)

Iperparametro	Valore
<i>win_sizes</i>	30, 60, 120
<i>s</i>	3
<i>step_max</i>	20
<i>gap_time</i>	30

Formalmente, siano \mathbf{Tr} , \mathbf{Va} i dataset utilizzati per il training e per il validation rispettivamente, e sia $\text{MSCRED}_{\mathbf{X}}^{\mathbf{Y}}$ un modello MSCRED addestrato con gli iperparametri della Tabella 3.6. su \mathbf{X} che effettua previsioni nei punti \mathbf{Y} , e sia *thred_bs50* uno spazio lineare costituito da elementi equidistanti thred_b_i tali che $\text{thred_b}_i \in [1e-11, 1e-6] \forall i \in [50]$ e sia *Thres* la lista esaustiva dei possibili threshold, ovvero dei vari anomaly score V_{score} , generati dal modello per ogni punto:

$$\text{thred_b}^*, \text{threshold}^* = \max_{\substack{\text{thred_b}, \\ \text{threshold}}} F1(\text{MSCRED}_{\mathbf{Tr}}^{\mathbf{Va}}(\text{thred_b}, \text{threshold})) \quad (3.5)$$

con il vincolo $\text{thred_b} \in \text{thred_bs50}$, $\text{threshold} \in V_{\text{score}}$

Tabella 3.7. Iperparametri MSCRED (parte 2.)

Iperparametro	Valore
<i>thred_b</i> *	6.326
<i>threshold</i> *	134.747

Gli iperparametri che hanno soddisfatto l'equazione 3.5 sono riportati nella Tabella 3.7.

È bene notare che un numero di passaggi temporali pari a gap_time vengono collassati in uno singolo dopo la computazione di MSCRED, quindi data una serie temporale che contiene x osservazioni, la soluzione generata avrà $x \bmod gap_time$ osservazioni.

Capitolo 4

Valutazione

In questo capitolo vengono analizzate le metriche di valutazione prese in considerazione ai fini dell'analisi del dataset reale delle richieste a servizi fatte a Infostud, la piattaforma su cui si fonda la carriera accademica di tutti gli affiliati all'università La Sapienza.

4.1 Metriche di Valutazione

Nel contesto dell'analisi condotta sui dati osservati della piattaforma Infostud, è fondamentale valutare numericamente le soluzioni dei modelli applicati ai fini della valutazione delle loro performance. A tale scopo, vengono utilizzate tre metriche importanti: la precisione (precision), il richiamo (recall) e la F1-score (F1) al fine di misurare l'efficacia delle soluzioni in modo rigoroso. Queste metriche sono essenziali per comprendere in modo completo ed esaustivo l'efficacia degli approcci metodologici applicati.

4.1.1 Precisione (Precision)

La precisione è una metrica che misura quanto un modello è accurato nel predire gli esempi positivi, o anomali nel contesto dell'anomaly detection, rispetto a tutti i casi che il modello ha classificato come positivi. Per calcolare la precisione, viene valutata la frazione di predizioni positive fatte dal modello che sono effettivamente corrette.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Dove:

1. TP (True Positives) rappresenta il numero di casi positivi, o anomali, correttamente classificati dal modello.
2. FP (False Positives) sono i casi negativi erroneamente classificati come positivi, o anomali, dal modello.

La precisione fornisce un riscontro numerico che, se massimizzato, rende il modello affidabile quando afferma che un caso è positivo. Tuttavia, la precisione da sola potrebbe non fornire una visione completa delle prestazioni di un modello.

4.1.2 Richiamo (Recall)

Il richiamo misura la capacità di un modello di individuare tutti gli esempi positivi correttamente. In altre parole, indica quanto il modello fornisce soluzioni che correttamente identificano gli esempi positivi. Il richiamo viene calcolato attraverso la frazione di predizioni positive, o anomale, fatte dal modello rispetto al totale dei casi positivi effettivi.

$$\text{Recall} = \frac{TP}{TP + FN}$$

Dove:

1. TP sono i True Positives, definiti nella Sottosezione 4.1.1
2. FN (False Negatives) rappresenta il numero di casi positivi, o anomali, erroneamente classificati come negativi dal modello.

Un alto valore di richiamo indica che il modello ha un'ottima capacità di individuare gli esempi positivi, ma non tiene in considerazione il numero di casi falsi positivi.

4.1.3 F1-Score (F1)

L'F1-score è una metrica che combina la precisione e il richiamo in un valore che tiene conto sia dei falsi positivi che dei falsi negativi. La metrica bilancia la precisione e il richiamo. La formula per calcolare l'F1-score è la seguente:

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

L'F1-score, quindi, mira a cercare un compromesso tra i valori di precisione e richiamo.

4.2 Valutazione del modello statistico ARMA

Come è stato già osservato, il modello statistico ARMA racchiude in modo discreto il comportamento del dataset e indica come anomalie i casi in cui la previsione del modello si discosta molto dai dati reali. In particolare, gli iperparametri scelti hanno riportato uno score F1 di 0.750 sul dataset di validation, mentre le metriche osservate sull'insieme di test sono riportate nella Tabella 4.1.

Tabella 4.1. Risultati ARMA.

Soluzione ARMA sul test set	
Metriche	
	Precisione: 0.332
	Recall: 0.248
	F1-score: 0.284

I risultati mostrano che ARMA riesce a catturare la tendenza generale del modello e molti andamenti anomali vengono predetti come tali. La soluzione non è ottimale, ma fornisce comunque una buona base per la valutazione di MSCRED.

4.3 Valutazione del modello ML OC-SVM

Il metodo che si basa sul machine learning, One-Class Support Vector Machine, arricchisce l'analisi e lo studio delle anomalie sul dataset di Infostud fornendo una soluzione molto buona sull'insieme di validazione con uno score F1 pari a 0.716, e una discreta sull'insieme di test, come mostrato nella Figura 4.2.

Tabella 4.2. Soluzione OC-SVM.

Soluzione OC-SVM sul test set	
Metriche	Precisione: 0.331
	Recall: 0.290
	F1-score: 0.309

I risultati mostrano che OC-SVM è in grado di produrre buoni risultati durante la fase di validazione ma fallisce durante la fase di test. Tuttavia, è importante notare che il modello è stato applicato in un contesto avverso, caratterizzato da un forte sbilanciamento tra le classi. Nonostante l'ambiente sfavorevole, questa soluzione è un punto di riferimento ragionevole per valutare le prestazioni del modello MSCRED.

4.4 Valutazione del modello DL Telemanom

Gli iperparametri ottimali applicati al dataset interessato hanno generato una buona soluzione come evidenziato nella Tabella 4.3.

Tabella 4.3. Risultati Telemanom.

Soluzione Telemanom sul test set	
Metriche	Precisione: 1.0
	Recall: 0.5
	F1-score: 0.66

La soluzione sembra ottima, notevolmente migliore rispetto a quelle dei modelli precedenti, tuttavia non è totalmente valida nella nostra applicazione e il motivo è semplice: Telemanom gestisce le anomalie, e calcola metriche su esse, basandosi solo sugli intervalli anomali come già citato nella Sottosezione 3.3.1. Gli intervalli anomali ground-truth nell'esperimento appena riportato sono solo due: uno grande che dura considerevolmente nel tempo, mentre l'altro molto più piccolo, come si evince nella Figura 4.1., in cui sono illustrati i valori reali e i valori previsti dal modello del dataset che rappresenta la latenza media delle richieste di login a Infostud. Telemanom identifica sempre e solo l'intervallo più grande come anomalo nei modelli dei vari canali, mentre quello più piccolo non è mai rilevato. Ciò ha portato a una soluzione che sembra ottima, ma che in realtà, per struttura stessa del modello e dei dati che osserva, sta overfittando il dataset. Nella Figura 4.2. viene illustrato il grafico degli errori smussati e_s riferente allo stesso segnale precedente.

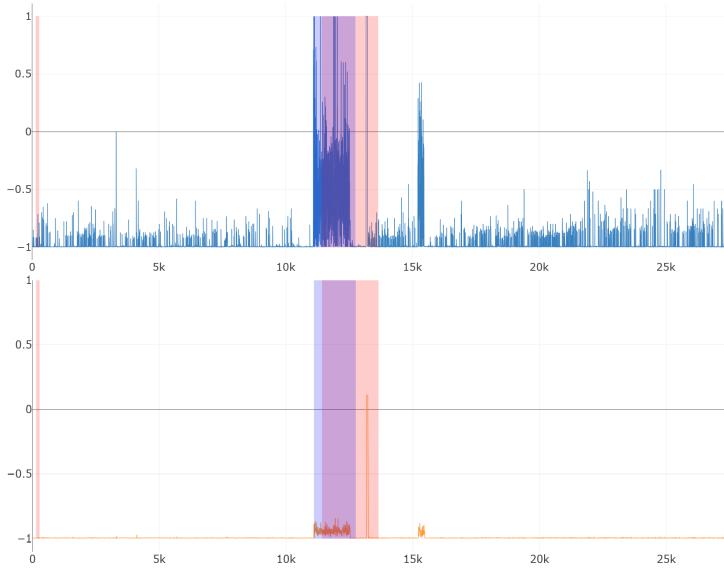


Figura 4.1. Dati reali y (sopra) e previsione di Telemanom \hat{y} (sotto) della latenza media delle richieste login a InfoSapienza. Le bande rosse rappresentano periodi anomali ground-truth, la banda blu è l'intervallo che il modello giudica anomalo.

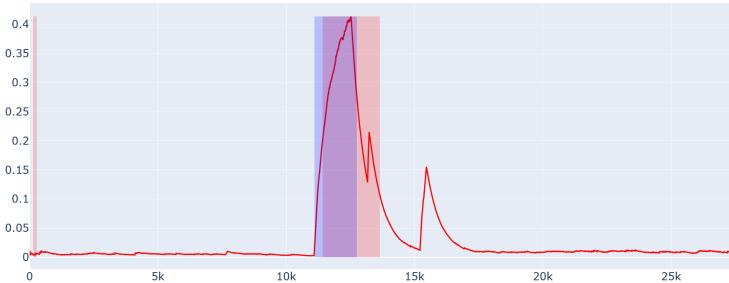


Figura 4.2. Errore smussato tra i valori reali e la previsione del modello e_s dei dati che rappresentano la latenza media delle richieste login a InfoSapienza.

Durante il mio studio di Telemanom ho compreso che il modello, in un contesto in cui sono presenti più anomalie che non durano molto nel tempo non performa in maniera altrettanto ottimale. Per riuscire a ottenere un discreto punteggio di recall, viene generato un grande numero di falsi positivi; i risultati rimangono coerenti a ciò che gli autori originali hanno descritto nel paper: Telemanom, purtroppo, genera molti falsi positivi, e non sembra che le sue capacità di pruning, nell'ambito di InfoSapienza, riescano a mitigare molto il problema.

In sintesi Telemanom, purtroppo, non si presta bene nel contesto della rilevazione di anomalie del dataset di Infostud. Il risultato, seppur ottimo, non viene ritenuto valido perché genera una soluzione che, in una situazione più naturale, sarebbe stata molto diversa.

4.5 Valutazione del modello DL MSCRED

Dopo 7 epoche di addestramento, MSCRED ha proposto la soluzione riportata in Figura 4.3. per l'insieme di validazione. Evidenziamo come MSCRED riesca a percepire l'andamento improvvisamente anomalo della serie temporale estratta dal dataset di Infostud, generando anomaly score elevati per ogni punto corrispondente a uno dei ventisei delle anomalie ground-truth, evidenziate dall'area rossa. In questo caso, la soluzione ha portato a un risultato perfetto, ottenendo uno score F1 pari a 1.0 e affermando l'ipotesi sollevata nel Capitolo 2: il dataset che rappresenta le latenze medie è un ottimo indicatore per quanto riguarda l'analisi e l'individuazione delle anomalie.

Riguardo all'insieme di test, MSCRED ha ottenuto risultati leggermente inferiori ma propone comunque una soluzione ottima mostrata in Figura 4.4. Le metriche risultanti dalla soluzione sono riportate nella Tabella 4.4.

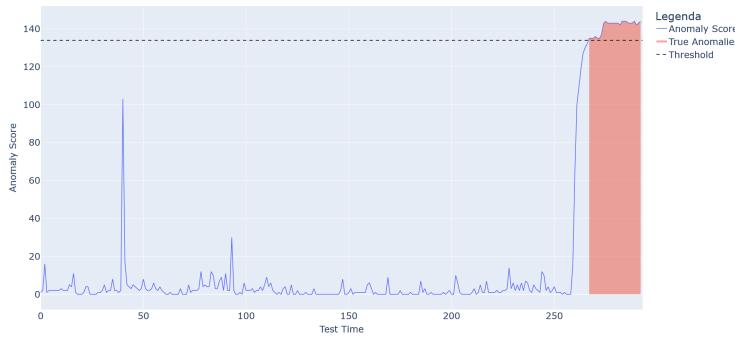


Figura 4.3. Soluzione MSCRED sul validation set.

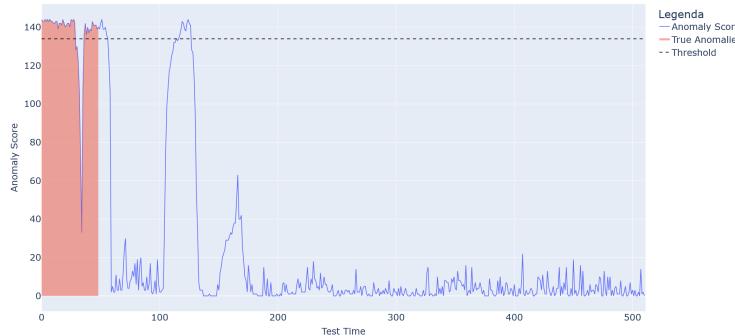


Figura 4.4. Soluzione MSCRED sul test set.

Tabella 4.4. Risultati MSCRED.

Soluzione MSCRED sul test set

Metriche	Precisione: 0.711
	Recall: 0.857
	F1-score: 0.777

Conclusioni I risultati dimostrano in maniera cristallina che MSCRED fornisce soluzioni notevolmente migliori rispetto agli altri metodi presi in analisi. Il modello, inoltre, riesce a catturare l'intercorrelazione dei segnali e la dipendenza temporale che hanno le osservazioni vicine nello spazio, e quindi nel tempo, delle serie temporali multivariate. MSCRED offre una misura numerica della severità delle anomalie anziché una semplice classificazione binaria, soddisfacendo così tutti i requisiti per un algoritmo di rilevamento delle anomalie discussi nel Capitolo 1. e permettendo all'anomaly detection di compiere un grande passo avanti.

D'altro canto, però, assumendo che il dataset di Infostud preso in analisi rispetti la proprietà IID, MSCRED sembra non offrire soluzioni altrettanto ottimali su serie temporali molto lunghe, ma fa sì che alcune anomalie pesino molto più di altre e non garantisce più una netta differenza tra i periodi anomali e non anomali. Questa supposizione è basata su delle prove fatte durante il percorso di studio di MSCRED: più è lunga la serie temporale, più MSCRED sembra performare in maniera peggiore. Tale affermazione, però, richiede studi più approfonditi per poter essere dimostrata o smentita.

Capitolo 5

Conclusioni: un passo avanti verso soluzioni migliori

Nell’ambito degli studi affrontati, sono stati esplorati diversi algoritmi per la rilevazione di anomalie sulla serie temporale multivariata estratta dalla piattaforma indispensabile per tutta la nostra comunità accademica. Gli esperimenti hanno fornito preziose informazioni sulle prestazioni di questi algoritmi e sulle loro potenzialità in questo complesso e affascinante dominio di applicazione; tali soluzioni, auspicabilmente, potrebbero essere analizzate per far sì che vi siano miglioramenti sulle operazioni e sul tempo di risposta e di processamento dei servizi richiesti, potenzialmente garantendo un ambiente migliore a tutti coloro che, giornalmente, beneficiano della piattaforma universitaria Infostud. È stato un onore per il sottoscritto poter interagire con dati di tale rilevanza e di così stretto coinvolgimento.

ARMA: il modello consolidato Nonostante ARMA[4] (Auto Regressive Moving Average) non sia stato originariamente progettato per il rilevamento di anomalie, rimane un pilastro tra gli algoritmi tradizionali. ARMA è un modello robusto ed efficace, basato su semplici metodi statistici ma adattabile a numerosi contesti della data science, e dimostra solidità anche nella rilevazione di anomalie.

OC-SVM: una soluzione sfacciata L’algoritmo dall’approccio metodologico del machine learning visto in questi studi è stato OC-SVM[3]. Sebbene questo algoritmo fosse molto sfavorito per il contesto preso in considerazione, visto l’intrinseco sbilanciamento tra le classi nell’anomaly detection, ha comunque offerto una soluzione degna di interesse. OC-SVM è un algoritmo flessibile e ci ha garantito un’ottima base su cui misurare le performance future.

Telemanom: la non-soluzione Purtroppo Telemanom[1], nonostante sia un modello molto recente che si focalizza sull’anomaly detection attraverso il deep learning, offre una soluzione che overfittà i dati. L’overfit non è tanto dovuto agli iperparametri, quanto a come il modello gestisce le anomalie e a come esse sono naturalmente distribuite nel dataset preso in analisi. La soluzione di Telemanom, seppur buona numericamente, non è da prendere in considerazione.

MSCRED: un promettente avanzamento MSCRED[2] (Multi-Scale Convolutional Recurrent Encoder-Decoder) è un modello relativamente giovane che si basa sul deep learning. Le analisi illustrate suggeriscono che MSCRED, grazie alla sua capacità di catturare l'intercorrelazione tra i segnali e la loro dipendenza temporale, emerge come un candidato di spicco per futuri sviluppi nel campo dell'anomaly detection. Il modello ha dimostrato ottime performance senza richiedere un eccessivo tuning degli iperparametri, lasciando intravedere un promettente futuro.

Riassunto: Le Performance degli Algoritmi La Tabella 5.1. riassume le performance dei vari algoritmi esaminati mediante le metriche principali prese in considerazione.

Tabella 5.1. Performance dei modelli analizzati

Modello	Precisione	Recall	F1-score
OC-SVM	0.331	0.290	0.309
ARMA	0.332	0.248	0.284
Telemanom¹	1.0	0.5	0.66
MSCRED	0.711	0.857	0.777

Il futuro dell'anomaly detection L'obiettivo per il futuro che ci attende è quello di sviluppare soluzioni sempre più accurate, che possano identificare le anomalie nei sistemi complessi attribuendo un valore numerico che identifichi lo stato, anomalo o meno, del sistema. Ciò garantirà non solo la possibilità di prendere atto prontamente delle anomalie, ma renderà possibile anche la loro prevenzione, contribuendo così a migliorare la sicurezza e minimizzando i periodi di down dei sistemi in una varietà di settori industriali, in cui spesso vi è in ballo anche la vita umana.

¹La soluzione non è valida perché il modello non è adatto al contesto di InfoSapienza

Bibliografia

- [1] Christopher Laporte, Ian Colwell, Kyle Hundman, Tom Soderstrom, Valentino Constantinou
Telem anom: Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding.
 Disponibile al link <https://arxiv.org/abs/1802.04431>, 2018.
 arXiv:1802.04431v3 [cs.LG] 6 Jun 2018.
- [2] Chuxu Zhang, Dongjin Song, Yuncong Chen, Xinyang Feng
MSCRED: A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data.
 Disponibile al link <https://dl.acm.org/doi/10.1609/aaai.v33i01.33011409>, 2019.
 The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19).
- [3] Larry M. Manevitz, Malik Yousef
One-Class SVMs for Document Classification.
 J. Mach. Learn. Res. 2(Dec):139–154.
- [4] Peter J. Brockwell, Richard A. Davis
An Introduction to Time Series and Forecasting.
- [5] Xingjian SHI, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, Wang-chun Woo.
Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting.
 NIPS, 802–810.