Anomaly Detection in Presenza di Concept Drift Ricorrente

Tesi di Laurea Magistrale in Ingegneria Informatica

Salvatore Drago

Relatore: Prof. Giuseppe Lo Re

Correlatori: Ing. Pierluca Ferraro

UNIVERSITÀ DEGLI STUDI DI PALERMO
DIPARTIMENTO DI INGEGNERIA

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ANOMALY DETECTION IN PRESENZA DI CONCEPT DRIFT RICORRENTE

Tesi di Laurea di
Dott. Salvatore Drago

Relatore:
Ch.mo Prof. Giuseppe Lo Re

Correlatore:
Ing. Pierluca Ferraro

Abstract

Molte applicazioni di crowdsensing oggi si basano su algoritmi di apprendimento applicati a flussi di dati per classificare accuratamente le informazioni e gli eventi di interesse in ambienti intelligenti. A tal proposito uno degli ambiti più studiati in letteratura riguarda il rilevamento di anomalie incrementale. Tale interesse deriva dall'importanza del problema per molti domini applicativi, come la sicurezza informatica o le frodi bancarie. Quando si lavora con i dati in streaming si presta attenzione ai vincoli temporali e di memoria che scaturiscono dalla necessità di analizzare grandi flussi di dati in tempo reale. Tuttavia si deve tenere in considerazione un altro aspetto: le proprietà statistiche dei dati in ingresso possono cambiare in modo inaspettato. Di conseguenza, la definizione di dati anomali e normali può variare nel tempo e i modelli di apprendimento automatico possono dover essere riaddestrati. Questo problema è noto come concept drift (deriva concettuale) ed è stato spesso ignorato dai sistemi di rilevamento delle anomalie, con un conseguente significativo degrado delle prestazioni dei sistemi che fanno uso di tali algoritmi. Inoltre, la distribuzione statistica dei dati passati tende spesso a ripetersi e quindi i vecchi modelli di apprendimento potrebbero essere riutilizzati, evitando costose fasi di riaddestramento su nuovi dati, con conseguente spreco di risorse computazionali. In questo lavoro di tesi si propone un sistema ibrido di rilevamento delle anomalie per i dati in streaming negli ambienti intelligenti, che tiene conto del concept drift e minimizza il numero di modelli di apprendimento automatico che devono essere riaddestrati quando vengono rilevati cambiamenti nella distribuzione dei dati in entrata. Il sistema è multi-livello e si basa su due diversi moduli

di rilevamento del concept drift e su un ensemble di modelli di rilevamento delle anomalie. È stata condotta un'ampia valutazione sperimentale, utilizzando due insiemi di dati reali e uno sintetico; i risultati mostrano le elevate prestazioni raggiunte dal sistema utilizzando metriche comuni come F1-score e accuratezza.

Contents

Introduzione

Recentemente, con la diffusione delle tecnologie IoT legate alle smart city e ad altri scenari di ambiente intelligente, la quantità di informazioni raccolte attraverso dispositivi mobili dotati di sensori e connessione a Internet sta crescendo in modo esponenziale. I dati provenienti dai sensori degli smartphone possono anche essere combinati con informazioni di alto livello fornite direttamente dagli esseri umani per rilevare e monitorare in modo proattivo fenomeni complessi del mondo reale [1], come previsto dal paradigma del Mobile Crowdsensing (MCS) [2]. Allo stesso tempo, molte aziende private e pubbliche hanno bisogno di tecniche di apprendimento automatico efficienti per analizzare enormi quantità di dati in streaming, consentendo loro di fare previsioni in tempo reale e di supportare le loro decisioni basate su tali dati [3]. Per rispondere a questa esigenza è necessario tenere conto di due aspetti fondamentali: l'enorme quantità di dati da analizzare e la loro natura variegata. In primo luogo, considerando che i dispositivi degli utenti generano continuamente nuovi dati che devono essere raccolti e analizzati in tempo reale, le metodologie esistenti di fusione dei dati e di rilevamento delle anomalie devono essere riadattate per elaborare dati in streaming [4]. Inoltre, molti dei dati raccolti e analizzati sono legati a fenomeni naturali e al comportamento umano, ad esempio il rilevamento di anomalie nei flussi video di sorveglianza [5], il monitoraggio dell'inquinamento ambientale nelle città intelligenti [6], il monitoraggio delle condizioni del manto stradale [7], e il monitoraggio della salute [8]. Oltre ad essere inevitabilmente influenzati dal rumore e, potenzialmente, da manomissioni da parte di utenti egoisti [9, 10], questi dati sono anche soggetti al problema del concept drift [11]. Ciò significa che le proprietà statistiche dei dati di input o della variabile obiettivo che il modello sta cercando di prevedere possono cambiare nel tempo in modo improvviso e inaspettato. Ad esempio, si consideri un sistema di rilevamento delle anomalie che analizza le immagini estratte dai flussi video delle telecamere di sorveglianza degli aeroporti. L'obiettivo del sistema è classificare i viaggiatori in base a determinate caratteristiche, come l'abbigliamento e la copertura del viso. Per esempio, esso viene addestrato per etichettare come sospette le persone

il cui volto è stranamente coperto, in modo da nascondere tratti somatici utili al riconoscimento stesso del soggetto in questione. La situazione di emergenza causata dalla pandemia COVID-19 ha portato a un improvviso aumento delle persone che indossano le mascherine e che di conseguenza hanno il volto coperto. Come dovrebbe comportarsi il sistema in questo caso, se indossare una mascherina non è più un comportamento anormale? Siamo in presenza di concept drift e il sistema deve essere riaddestrato di conseguenza.

Inoltre, cosa succede se l'emergenza sanitaria termina e le mascherine non sono più comunemente utilizzate? Si tratta di un "concetto" che si ripete e quando ciò si verifica il sistema deve tornare al suo funzionamento originale. Molti degli approcci tradizionali alla fusione dei dati e al rilevamento delle anomalie nei sistemi di crowdsensing ignorano completamente il problema del concept drift [12], ma questo solleva seri problemi di accuratezza e affidabilità in qualsiasi scenario reale in cui esso si verifichi effettivamente, che richiederebbero un riaddestramento tempestivo ed efficiente dei modelli di apprendimento automatico in tempo reale adottati. Infatti, quando si verifica un concept drift, un modello addestrato sui dati passati potrebbe non essere in grado di analizzare correttamente i nuovi valori in arrivo, con conseguenti previsioni imprecise e scarsi risultati decisionali. Se il modello di analisi dei dati non gestisce esplicitamente il concept drift, dovrà essere riaddestrato manualmente con i nuovi dati, inoltre, finché ciò non avviene, il sistema subisce un drastico calo delle prestazioni, che potrebbe avere conseguenze catastrofiche in scenari critici come il rilevamento di anomalie nella gestione dell'energia [13], nelle transazioni bancarie online o nel rilevamento di intrusioni [14] in infrastrutture di rete critiche. Questo continuo riaddestramento del modello, a sua volta, comporta uno spreco di tempo e risorse di rete e di calcolo.

In questo lavoro si propone un sistema di rilevamento delle anomalie non supervisionato in presenza di concept drift ricorrente nei flussi di dati per ambienti intelligenti. L'obiettivo principale è ridurre al minimo il numero di modelli di apprendimento automatico che devono essere riaddestrati, mantenendo un'accuratezza molto elevata nel rilevamento delle anomalie, anche nel caso di concept drift improvviso e ricorrente. L'architettura del sistema è multi-livello e sfrutta due moduli di rilevamento del concept drift e un insieme di modelli appositamente addestrati per il rilevamento delle anomalie. L'approccio proposto sfrutta il fatto che, anche in presenza di concept drift, i "vecchi concetti" ricorrono spesso ciclicamente. In questo modo, viene mantenuta una cronologia di modelli precedentemente addestrati che potrebbero essere riutilizzati in futuro, nello spirito della sostenibilità e dell'uso efficiente delle risorse disponibili [15], che è particolarmente prezioso in scenari mobili in cui tali risorse sono limitate e il loro

utilizzo deve essere ridotto il più possibile [16]. L'unico caso in cui viene addestrato un nuovo modello è quando nessuno dei vecchi modelli si adatta ai nuovi dati, cioè quando è avvenuto un concept drift verso un "concetto" mai osservato prima. Tuttavia, anche nei rari casi in cui è necessario addestrare un nuovo modello, il sistema lo farà in modo efficiente utilizzando solo l'ultima finestra di dati ricevuta. Non a caso, i modelli a finestra sono i più utilizzati in letteratura per la gestione dei dati in streaming. Per convalidare l'approccio del sistema proposto, esso è stato ampiamente valutato con una serie di esperimenti eseguiti su tre diversi set di dati. I risultati mostrano che il sistema è in grado di rilevare gli outlier con un'elevata accuratezza, riducendo al minimo il numero di modelli da addestrare.

I principali contributi di questo lavoro possono essere riassunti come segue:

- Viene proposto un nuovo sistema ibrido di rilevamento delle anomalie che tiene conto direttamente del concept drift ricorrente.
- Viene adattata la nota tecnica di rilevamento delle anomalie LOF per renderla più adatta all'analisi di flussi di dati in tempo reale.
- Si riduce il più possibile il numero di modelli di rilevamento delle anomalie che devono essere addestrati, pur mantenendo alte le prestazioni.
- Il sistema viene ampiamente valutato con due diversi set di dati reali e uno sintetico.

Il resto di questa tesi è organizzato come segue. Il Capitolo 1 e il Capitolo 2 sono dedicati rispettivamente allo stato dell'arte delle tecniche di Anomaly Detection e al problema del Concept Drift. Nel Capitolo 3 viene descritto nel dettaglio il sistema proposto. Il capitolo 4 è dedicato alle conclusioni finali.

Chapter 1

Anomaly Detection

Negli ultimi anni molti lavori si sono concentrati sullo studio delle tecniche di Anomaly Detection (rilevamento delle anomalie), con l'obiettivo di identificare gli outlier in un set di dati [17]. Un outlier è un'osservazione talmente diversa dalle altre da sembrare generata da un meccanismo diverso da quello dei dati normali. Questa definizione si basa su considerazioni statistiche e presuppone che i dati normali seguano un meccanismo di generazione comune. Le osservazioni che si discostano da questo meccanismo sono considerate outlier (Fig. 1.1). In molti casi, le tecniche di Anomaly Detection sono uno step all'interno di compiti più vasti. Gli outlier vengono identificati per poterli escludere dalle fasi di analisi successiva, in quanto possono alterare i valori statistici di interesse, oppure vengono identificati per essere analizzati più nel dettaglio, come nel caso degli Intrusion Detection System (IDS). In altri campi sono invece gli outlier ad avere importanza, come nei sistemi di rilevamento di frodi finanziarie, di diagnosi medica e di e-voting [18].

La sezione 1.1 descrive le diverse tipologie di anomalia. La sezione 1.2 riassume quelle che sono le sfide generali da affrontare per l'identificazione degli outlier. Nella sezione 1.3 si descrivono le principali tecniche presenti in letteratura, suddividendole in base a determinati criteri. Infine la sezione 1.4 descrive un campo di applicazione delle tecniche di Anomaly Detection molto studiato in letteratura, ovvero quello degli Intrusion Detection System (IDS) [19].

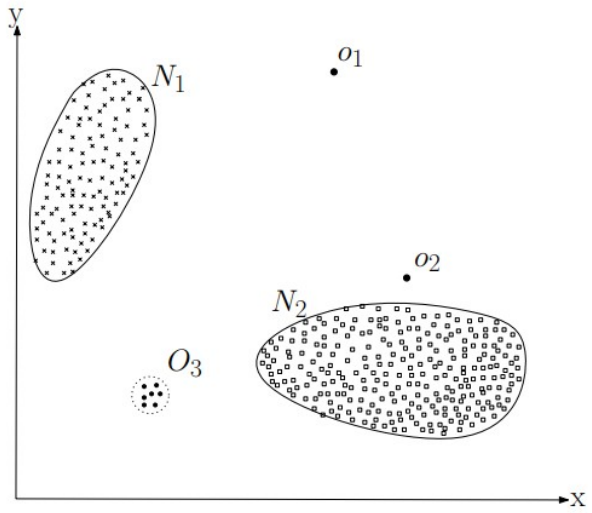

Figure 1.1: Esempio bidimensionale di dati anomali. Immagine tratta da [17]

1.1 Tipi di anomalia

La natura dell'anomalia da trattare è un aspetto che incide sulla tecnica di rilevamento, per questo motivo in letteratura spesso si distinguono tre categorie di anomalie:

- **anomalie puntuali**: quando una singola istanza dei dati può essere considerata anomala rispetto al resto dei dati. Questo è il tipo più frequente e semplice di anomalia. Alcune anomalie che rientrano in questa tipologia sono riportate nella Fig. 1.1.

- **anomalie contestuali**: per individuare questo tipo di anomalia occorre considerare la relazione tra il punto analizzato e il suo vicinato. L'istanza dei dati può essere anomala in un contesto specifico ma non in un altro. La nozione di contesto deriva dalla natura dei dati e dalla loro struttura e deve far parte della formulazione del problema. I dati sono definiti da due tipi di attributi:

  - contestuali: fanno parte di questi attributi quelli utilizzati per determinare il vicinato di quell'istanza, per esempio, latitudine e longitudine di una località in un insieme di dati spaziali.
  - comportamentali: descrivono il comportamento di quell'istanza. Ad esempio, in un insieme di dati spaziali, la temperatura media annuale di una località.

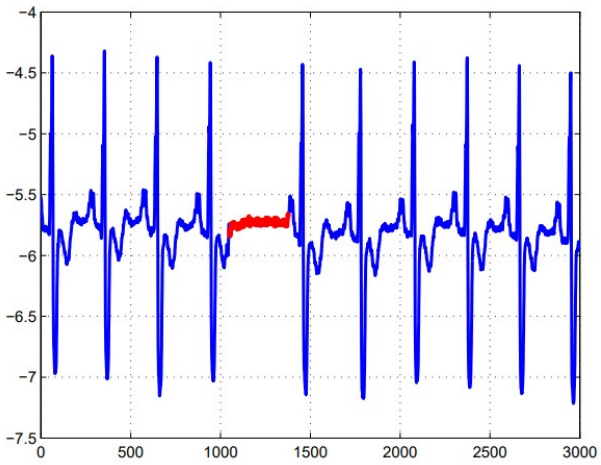

Figure 1.2: Anomalia collettiva corrispondente a una contrazione prematura atriale in un elettrocardiogramma umano. Immagine tratta da [17]

- **anomalie collettive o dei dati sequenziali**: contrariamente alle anomalie puntuali, l'anomalia va ricercata su un insieme di istanze correlate. Le singole istanze che compongono un'anomalia collettiva possono non essere anomale di per sé, ma il loro comportamento collettivo lo è. Quello che conta è la forma multidimensionale delle sequenze di punti ordinati nel tempo. Un tipico esempio è quello delle anomalie negli elettrocardiogrammi umani (Fig. 1.2). Un problema di rilevamento di anomalie collettive può essere trasformato in un problema di anomalie contestuali incorporando le informazioni sul contesto. Le tecniche utilizzate in questo caso possono essere costruite su caratteristiche estratte dall'analisi del segnale (ad es. FFT, correlazione, ecc.).

1.2 Sfide principali nell'Anomaly Detection

La definizione di outlier, seppur molto semplice e intuitiva, nasconde molte insidie. L'approccio, più semplice, che direttamente o indirettamente segue quasi la maggior parte delle tecniche di Anomaly Detection, consiste nel definire una o più regioni che rappresentino il comportamento normale e nel dichiarare anomala qualsiasi osservazione nei dati che non appartiene a tale regione.

Diversi fattori rendono questo approccio più impegnativo del previsto [17]:

- definire con precisione delle regioni che raccolgono tutti i possibili comportamenti normali

non è semplice: il confine tra "normale" e "anomalo" non è sempre preciso, e alcune osservazioni possono trovarsi molto vicine a quel confine;

- i dati possono avere elevata dimensionalità. Ciò causa sparsità e riduce la significatività di definizioni di prossimità, distanza, densità e vicinato su cui alcune tecniche di analisi si basano. Questo problema è anche noto come la maledizione della dimensionalità;
- spesso i dati contengono rumore che tende a essere simile alle anomalie effettive;
- la nozione esatta di anomalia è diversa per i vari domini applicativi. Di conseguenza, applicare una tecnica sviluppata per un dominio ad un altro non è semplice;
- solitamente è difficile reperire dati etichettati per l'addestramento/validazione dei modelli di rilevamento delle anomalie supervisionati;
- quando le anomalie sono il risultato di azioni malevole, gli avversari malintenzionati tentano di far apparire le azioni malevole come normali, rendendo così più difficile il compito di definire il comportamento normale.

1.3 Tassonomia delle tecniche di Anomaly Detection

Una prima suddivisione delle tecniche di Anomaly Detection riguarda ovviamente la natura **supervisionata** o **non supervisionata** dell'algoritmo di apprendimento. Il primo caso deriva dall'idea di sfruttare il training-set per costruire un modello di predizione che possa classificare un certo elemento come outlier o meno.

In uno scenario non supervisionato l'algoritmo deve individuare gli outliers solamente sulla base dei dati a disposizione e sull'assunzione che le istanze di dati normali siano nettamente superiori in numero alle istanze delle anomalie.

Le tecniche supervisionate generalmente risultano più efficienti di quelle non supervisionate perchè la conoscenza posseduta attraverso il training-set viene usata per affinare il processo decisionale ma soffrono di un problema ulteriore, descritto in precedenza, sul reperimento dei dati etichettati. Gli approcci non supervisionati sono più applicabili proprio perché l'unica cosa che richiedono per poter essere eseguiti sono i dati da analizzare.

Le tecniche supervisionate si possono ricondurre a classificatori binari (anomalo/non anomalo) realizzati principalmente attraverso: classificatore bayesiano, support vector machine (SVM), alberi decisionali, classificatore Random Forest, reti neurali profonde [20, 21, 22, 23].

Le tecniche di rilevamento delle anomalie non supervisionate più utilizzate in letteratura appartengono principalmente ad una di queste tre macro-categorie:

- **approcci statistici**: gli approcci statistici [25] presuppongono che i dati siano stati generati secondo una specifica distribuzione. Gli outlier saranno quindi quei punti che hanno una bassa probabilità di essere stati generati secondo quella distribuzione. Essi si dividono in:
  - **parametrici**: devono avere una conoscenza preventiva della distribuzione dei dati.
  - **non parametrici**: apprendono dall'insieme dei dati per ottenere la distribuzione sottostante.
- **metodi basati sulla distanza**: nei metodi basati sulla distanza [26], gli outlier sono modellati come punti isolati dai dati normali, senza fare alcuna ipotesi sulla loro distribuzione. Tali approcci, a loro volta si dividono in:
  - **metodi basati sul clustering e su sue proprietà**: i metodi di clustering suddividono il set di dati in diversi cluster in base alla somiglianza tra i dati. Il cluster più distante o il cluster con la densità più bassa può essere considerato un cluster di anomalia [27, 28].
  - **metodi basati sui vicini più prossimi**: tali metodi (nearest-neighbors) determinano i vicini di un'osservazione calcolando la distanza tra tutte le osservazioni del set di dati. Un'anomalia è considerata l'osservazione che è più **distante** dai suoi k vicini più prossimi [29] o con **densità** minore rispetto ad essi.
- **metodi basati sull'isolamento**: i metodi basati sull'isolamento [30, 31], invece, cercano di isolare le osservazioni outlier dal set di dati, poiché i dati anomali sono considerati molto diversi dai dati normali e si presume che rappresentino una piccola parte dell'intero set di dati.

Le tecniche elencate precedentemente sono **statiche** in quanto determinano la presenza di outlier una volta che tutti i record sono disponibili nel set di dati e quindi possono funzionare solo a posteriori.

Tuttavia, negli ultimi anni con il progredire della ricerca nell'ambiente dei flussi di dati, il problema di come segnalare in modo efficiente quando ci si trova di fronte ad una anomalia è stato studiato con grande interesse. Le tecniche che affrontano questo problema vanno sotto il nome di **online anomaly detection**, **anomaly detection in data stream** o **rilevamento incrementale degli outlier**.

1.3.1 Anomaly Detection in Data stream

Al contrario delle tecniche statiche, le tecniche di **rilevamento incrementale degli outlier** [34] possono lavorare in tempo reale con i dati in streaming, identificando gli outlier non appena un nuovo record viene ricevuto dal sistema. La necessità di analizzare grandi flussi di dati in tempo reale aggiunge ulteriori sfide a quelle già elencate dell'Anomaly Detection statico:

- **vincolo temporale**: i dati in streaming da analizzare possono susseguirsi più o meno velocemente. Di conseguenza, un algoritmo che lavora con tale tipologia di dati deve elaborare le osservazioni in arrivo in un tempo limitato. Questo comporta anche che i dati in arrivo devono essere esaminati meno volte possibile.

- **vincolo di memoria**: dal momento che i flussi di dati sono potenzialmente infiniti, gli algoritmi devono lavorare all'interno di una memoria limitata, memorizzando il minor numero possibile di osservazioni in entrata e di informazioni statistiche sui dati elaborati e visti fino a quel momento.

- **concept drift**: in molti domini il comportamento normale/anomalo continua a evolversi e una nozione attuale di comportamento normale/anomalo potrebbe cambiare o non essere sufficientemente rappresentativa in futuro. Quest'ultimo aspetto, spesso viene trascurato, ma se si verifica porta ad un drastico decadimento delle prestazioni del sistema di cui l'algoritmo di Anomaly Detection fa parte. Questo aspetto viene ulteriormente approfondito nel Capitolo 2 ed affrontato dall'architettura proposta nel Capitolo 3.

Modelli a finestra

Una tecnica ampiamente utilizzata per lavorare con i dati in streaming è quella del "modello a finestra". Tale tecnica consente di trattare il flusso di dati a batch, e lavorare volta per volta sul singolo batch, riconducendosi ad una applicazione statica dell'algoritmo o del modello utilizzato.

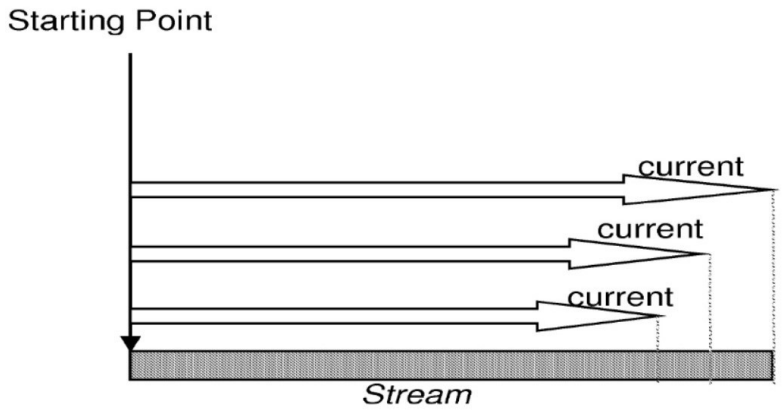

Figure 1.3: Landmark Window. Immagine tratta da [35]

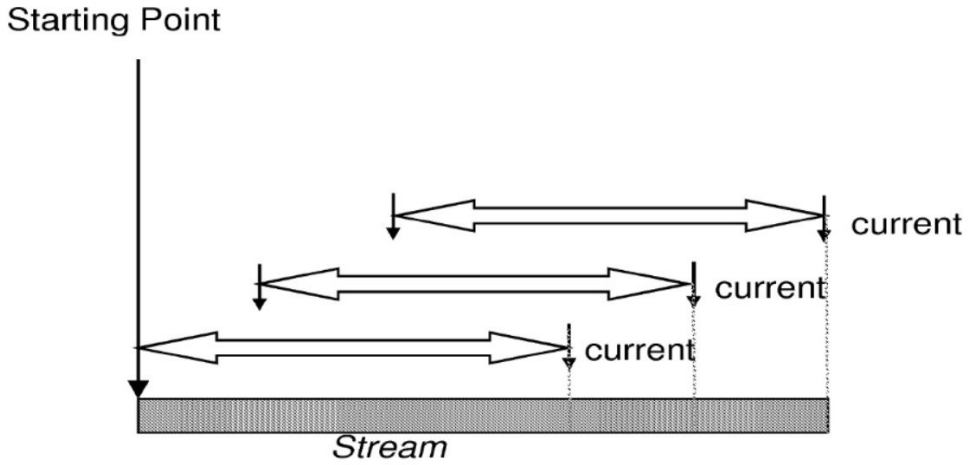

Figure 1.4: Sliding Window. Immagine tratta da [35]

Sono stati proposti diversi modelli a finestra utili per mantenere una parte dei contenuti del flusso di dati in memoria [35]:

- **Landmark Window**: la finestra raccoglie i dati a partire da un dato iniziale detto landmark fino al dato che è arrivato più di recente. La finestra, quindi, cresce di dimensioni fino a quando il landmark stesso non viene spostato.
- **Sliding Window**: i dati vengono raccolti in una finestra di dimensione fissa scorrevole. Ogni volta che arriva un nuovo dato esso viene incluso nella finestra a discapito del dato più vecchio che era presente nella finestra stessa.
- **Damped Window**: simile al modello sliding windows ma ai dati viene assegnato un peso

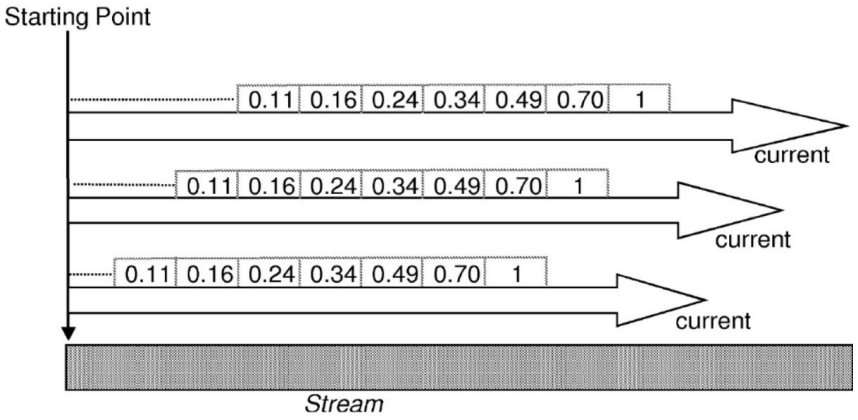

Figure 1.5: Damped Window. Immagine tratta da [35]

decrescente rispetto il loro arrivo. I dati più vecchi presenti nella finestra avranno pesi più piccoli rispetto ai dati più recenti.

La tipologia di modello a finestra che si adotta, la dimensione di tale finestra e il momento in cui tale finestra di dati viene data in input al modello di machine learning sono fortemente vincolati al tipo di sistema che si vuole realizzare e ai vincoli temporali e di memoria a cui il sistema deve sottostare.

Oltre ad aiutare il progettista del sistema a rispettare i vincoli temporali e di memoria, i modelli a finestra fanno parte anche di molte tecniche che servono per riconoscere il Concept Drift (Capitolo 2).

Le tecniche di Anomaly Detection che lavorano con i flussi di dati sono adattamenti delle tecniche tradizionali di rilevamento delle anomalie, sopra elencate, tramite l'applicazione dei modelli a finestre. Tuttavia, la natura statica dell'algoritmo rende difficoltoso un adattamento ottimale allo scenario dei flussi di dati.

La Fig. 1.6 sintetizza vantaggi e svantaggi delle varie tecniche con particolare attenzione a quegli aspetti che influiscono sull'adattamento delle suddette tecniche ai flussi di dati.

1.4 Anomaly Detection nella sicurezza delle reti: IDS Anomaly-based

Negli ultimi decenni si è assistito ad un esponenziale aumento dell'utilizzo di Internet e delle reti, sia pubbliche che private, nonché all'informatizzazione di molti processi aziendali,

Approcci	Vantaggi	Svantaggi
Approcci statistici	I metodi non parametrici sono adattati al contesto del flusso di dati.	• I metodi parametrici sono difficili da applicare al flusso di dati. • I metodi non parametrici possono essere utilizzati solo per flussi di dati a bassa dimensionalità.
Clustering	Adattato per l'identificazione dei cluster.	Non ottimizzato per l'identificazione di singole anomalie.
Nearest-neighbors	Metodi basati sulla distanza: • adattati per il rilevamento delle anomalie globali.	Metodi basati sulla distanza: • non sono adatti a densità non omogenee. • hanno un costo computazionale elevato per flusso di dati ad alta dimensionalità.
	Metodi basati sulla densità: • adatti al rilevamento di anomalie locali. • più efficienti dei metodi basati sulla distanza.	Metodi basati sulla densità: • hanno un'elevata complessità. • non sono efficaci per flussi di dati ad alta dimensionalità.
Basati sull'isolamento	• hanno un minore consumo di CPU, tempo e memoria. • sono efficienti per il rilevamento delle anomalie.	• dipendono in larga misura dalla finestra e dalle scelte di politica di aggiornamento del modello. • difficili da adattare a dati categorici.

Figure 1.6: Confronto tra gli approcci di Anomaly Detection in Data Stream. Immagine tratta da [36]

con sistemi web-based. Contemporaneamente sono aumentati attacchi informatici alle reti, ai computer e ai sistemi informativi. Tali attacchi violano le politiche di sicurezza informatica ovvero la riservatezza, l'integrità e la disponibilità (CIA). Di pari passo è, dunque, cresciuto l'interesse verso i sistemi di Intrusion Detection System (IDS) e di Intrusion Prevention System (IPS).

Un sistema di rilevamento delle intrusioni è un sistema software o hardware per automatizzare tale rilevamento [14]. Inoltre, un sistema di prevenzione delle intrusioni (IPS) è un sistema che dispone di tutte le capacità degli IDS e può tentare di bloccare eventuali attacchi. I sistemi studiati in letteratura sono spesso sia di rilevamento che di prevenzione (IDPS) ma spesso ci si riferisce loro semplicemente come IDS.

La Fig. 1.7 cerca di individuare, rispetto alla letteratura, le principali caratteristiche degli Intrusion Detection [19]:

- **Tempestività** (Timeliness).
- **Strategia di Rilevamento** (Detection Strategy).
- **Implementazione del Sistema** (System Deployment).

- **Fonte dei dati** (Data Source).

Un sistema IDS può essere classificato rispetto alle tecniche che utilizza in corrispondenza a queste caratteristiche.

L'aspetto più importante è, senza dubbio, quello riguardante la metodologia di rilevamento. Le metodologie di rilevamento delle intrusioni sono classificate in quattro principali categorie:

- **rilevamento basato sulle firme** (Signature-based Detection) [37, 38, 39]: una firma è un modello o una stringa che corrisponde a un attacco o a una minaccia nota (pattern). Questo metodo consiste nel confrontare tali pattern con gli eventi catturati per riconoscere eventuali intrusioni. Poiché utilizza le conoscenze accumulate da specifici attacchi la SD è nota anche come Knowledge-based Detection. Il vantaggio principale di queste tecniche è la semplicità e l'efficacia con cui rileva gli attacchi noti. Tuttavia, è inefficace nel rilevare attacchi sconosciuti e varianti di attacchi noti, e risulta difficile mantenere aggiornate le firme/pattern. Ciò è molto pericoloso visto che tutto quello che non corrisponde ad un pattern viene considerato traffico benigno.

- **rilevamento basato sulle anomalie** (Anomaly-based detection) [40, 41, 42]: si utilizzano tecniche di anomaly detection supervisionate e non per creare dei profili comuni di utilizzo della rete, da parte degli utenti benigni, derivati dal monitoraggio di attività regolari. I dati che risultano anomali derivano da attività illegittime in quanto avranno caratteristiche diverse dai dati che hanno formato i profili di utilizzo regolare. Questa tecnica è efficace per rilevare vulnerabilità nuove e impreviste e facilita il rilevamento di un attacco basato sull'abuso di privilegi. L'approccio è conservativo in quanto viene riconosciuto come anomalo tutto quello che non è conforme ai dati che formano i profili di attività regolari. I principali svantaggi [19] riguardano il decadimento dell'accuratezza dei profili regolari a causa della loro natura variabile. Il sistema diventa non disponibile durante la ricostruzione dei profili di comportamento, che consiste nel riaddestramento del sistema con i dati più recenti (nuovi profili di comportamento).

- **analisi del protocollo stateful** (Stateful protocol analysis) [43, 44]: la parola stateful indica che questa tipologia di IDS possono conoscere e tracciare gli stati del protocollo. Il processo SPA assomiglia a quello degli AD ma la differenza è consistente: AD adotta profili pre-appresi specifici per quella rete o quell'host, mentre SPA dipende da profili

generici sviluppati dal fornitore per protocolli specifici. In generale, i modelli di protocollo di rete in SPA si basano su standard internazionali come ad esempio l'IETF. SPA è noto anche come rilevamento basato su specifiche. I vantaggi riguardano la possibilità di riconoscere e tracciare gli stati del protocollo e distinguere sequenze inaspettate di comandi. Tuttavia, risultano più difficili da realizzare rispetto alle altre due tipologie anche per le notevoli risorse consumate per tracciare e analizzare gli stati del protocollo in esame. Risulta comunque impossibile ispezionare gli attacchi che sembrano comportamenti benigni del protocollo. Per questo motivo è la tipologia di IDS meno trattata in letteratura e si presenta spesso in IDS ibridi come complemento alle altre tipologie.

- **rilevamento ibrido** [45, 46]:la maggior parte degli IDS utilizza diverse metodologie per fornire un rilevamento più esteso e accurato. Ad esempio, AD e SD sono metodi complementari. Quando il primo rileva un'anomalia rispetto al comportamento normale il secondo può confrontarla con un pattern noto.

Gli IDS basati sulle anomalie sono sicuramente i più diffusi oltre ad essere presenti in quasi tutti gli IDS ibridi. É anche l'approccio più intuitivo per rilevare un'intrusione. L'intrusione genera dati anomali rispetto ai dati estratti durante il comportamento normale della rete.

Si noti, come, il loro principale svantaggio, discusso precedentemente, sia riconducibile al problema del **concept drift** per i modelli di anomaly detection che trattano i dati in streaming. Gli IDS anomaly-based vengono addestrati su un dataset che contiene record che descrivono uno o più profili di comportamento normale della rete. Successivamente il sistema riceverà dati in ingresso, che corrispondono a quello che sta succedendo nella rete, che dovrà confrontare con i profili appresi alla ricerca di un outlier. La nozione di comportamento normale/anomalo può cambiare nel tempo. In questo caso specifico può cambiare, per diversi motivi, il profilo di comportamento della rete. Quando ciò accade, la maggior parte degli IDS, devono essere riaddestrati su dati generati del nuovo comportamento della rete, lasciando quest'ultima vulnerabile agli attacchi durante questa fase. Adottare una tecnica di Anomaly Detection che tratta esplicitamente il problema del concept drift può evitare il drastico decadimento dell'accuratezza del sistema e ridurre al minimo la necessità di intervento diretto.

In questo lavoro viene modificata una nota tecnica statica di rilevamento delle anomalie, nota come Local Outlier Factor (LOF) [26] per adattarla all'analisi dei dati in streaming, come verrà spiegato di seguito.

Il LOF rientra nei metodi basati sulla densità dei vicini più prossimi. Tale adattamento vuole

correggere lo svantaggio dell'elevato costo computazionale di tale tecnica e rispondere anche ad alcune sfide dell'Anomaly Detection (Sezione 1.2), soprattutto al problema dell'evoluzione del comportamento normale/anomalo (concept drift).

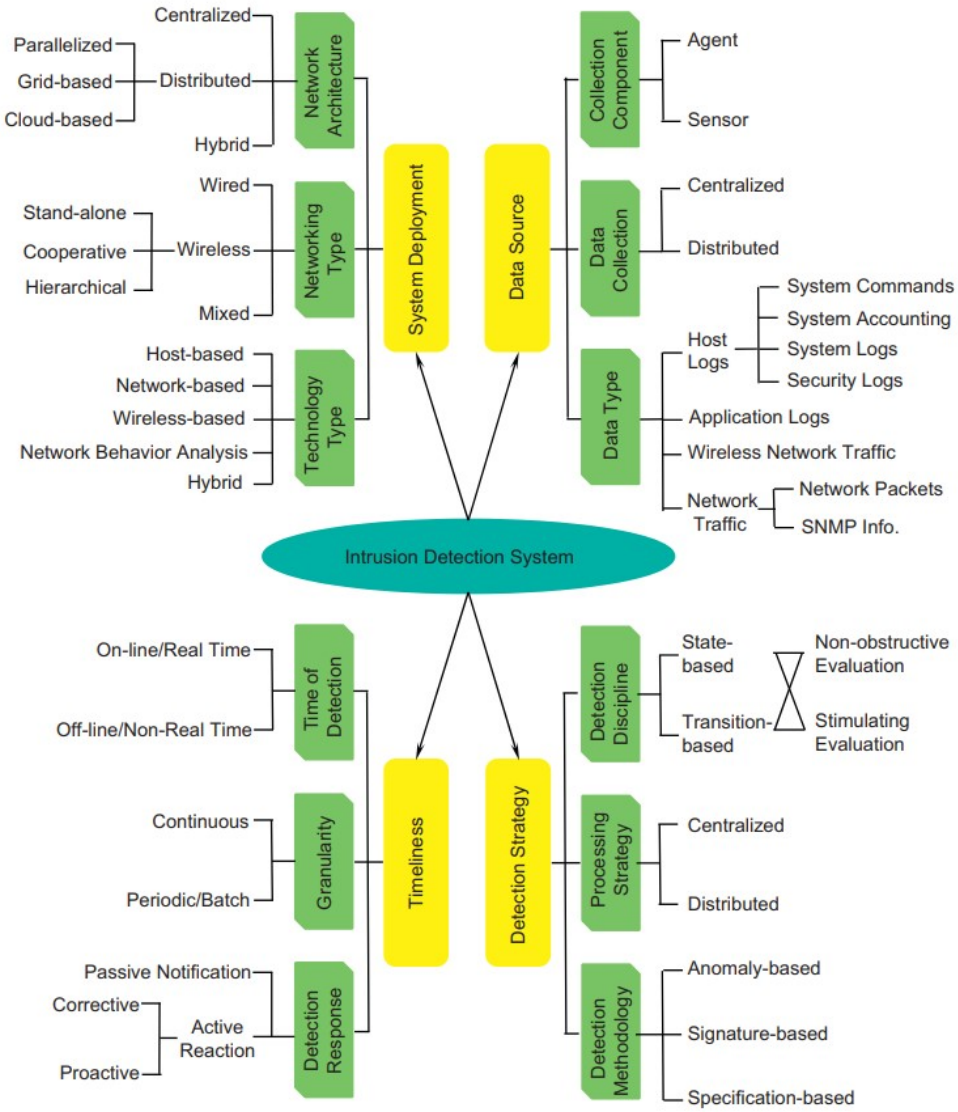

Figure 1.7: Tassonomia IDS. Immagine tratta da [19]

Chapter 2

Concept Drift

Si parla di concept drift (o deriva concettuale) [47, 11] quando la distribuzione dei dati di input e/o di output cambia nel tempo in modo inaspettato. Formalmente il concept drift è definito come segue:

$$\exists t : P_t(X,y) \neq P_{t+1}(X,y)$$

Visto che $P_t(X,y) = P_t(X) \times P_t(y|X)$ il concept drift può avere diverse origini, come mostrato in Fig. 2.1:

a) Il primo tipo di drift [48], noto come drift virtuale o drift dello spazio delle caratteristiche, si verifica quando cambia solo la distribuzione dei dati di input e non ci sono spostamenti nelle previsioni, come mostrato in Fig. 2.3-a ($P_t(X) \neq P_{t+1}(X)$ con $P_t(y|X) = P_{t+1}(y|X)$).

b) Nel secondo tipo di drift, Fig. 2.3-b, chiamato drift effettivo o drift del confine decisionale, la distribuzione dei dati di ingresso rimane invariata, mentre le previsioni cambiano, causando un'alterazione del confine decisionale ($P_t(X) = P_{t+1}(X)$ con $P_t(y|X) \neq P_{t+1}(y|X)$).

c) Tuttavia, nel mondo reale è più probabile che questi due tipi di drift si verifichino contemporaneamente: la Fig. 2.3-c mostra il terzo tipo di drift, che si verifica quando la distribuzione dei dati di ingresso e le previsioni cambiano insieme ($P_t(X) \neq P_{t+1}(X)$ con $P_t(y|X) \neq P_{t+1}(y|X)$).

Un'altra classificazione spesso utilizzata in letteratura per il concept drift riguarda il modo in

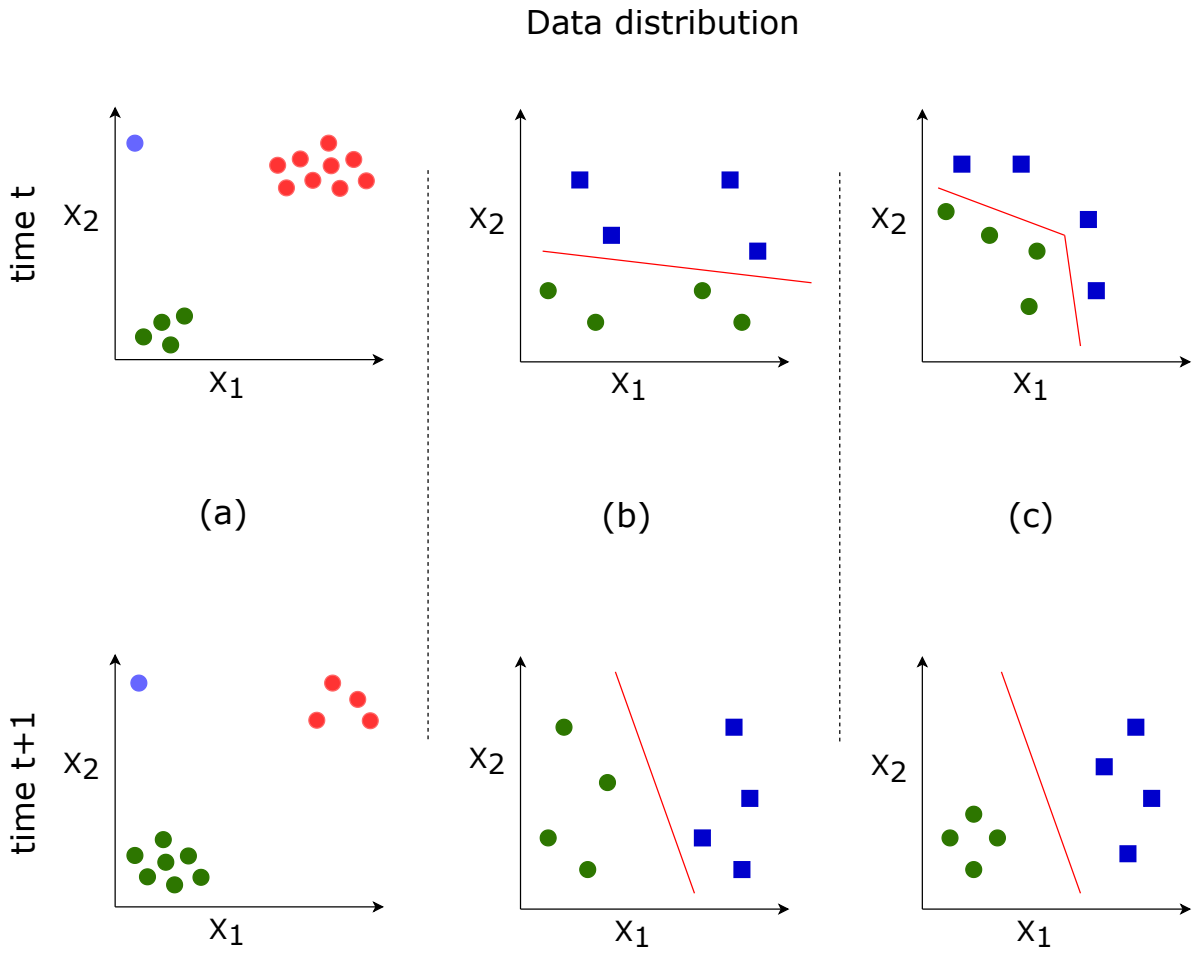

Figure 2.1: Fonti di Concept Drift.

cui la distribuzione dei dati cambia nel tempo. In generale, si distinguono quattro tipi di possibili drift, come mostrato in Fig. 2.2:

- Il drift potrebbe essere improvviso (*sudden*), ad esempio quando un sensore viene sostituito con un altro calibrato in modo diverso, oppure quando si verifica un malfunzionamento improvviso o un evento inaspettato, come lo scoppio di una pandemia;
- Il drift potrebbe essere incrementale (*incremental*), con una serie di "concetti intermedi" che si susseguono, riflettendo il caso di preferenze dell'utente che cambiano nel tempo o l'usura continua del sensore, con conseguente perdita di precisione;
- Il drift potrebbe essere graduale (*gradual*), se il "nuovo concetto" si alterna inizialmente a

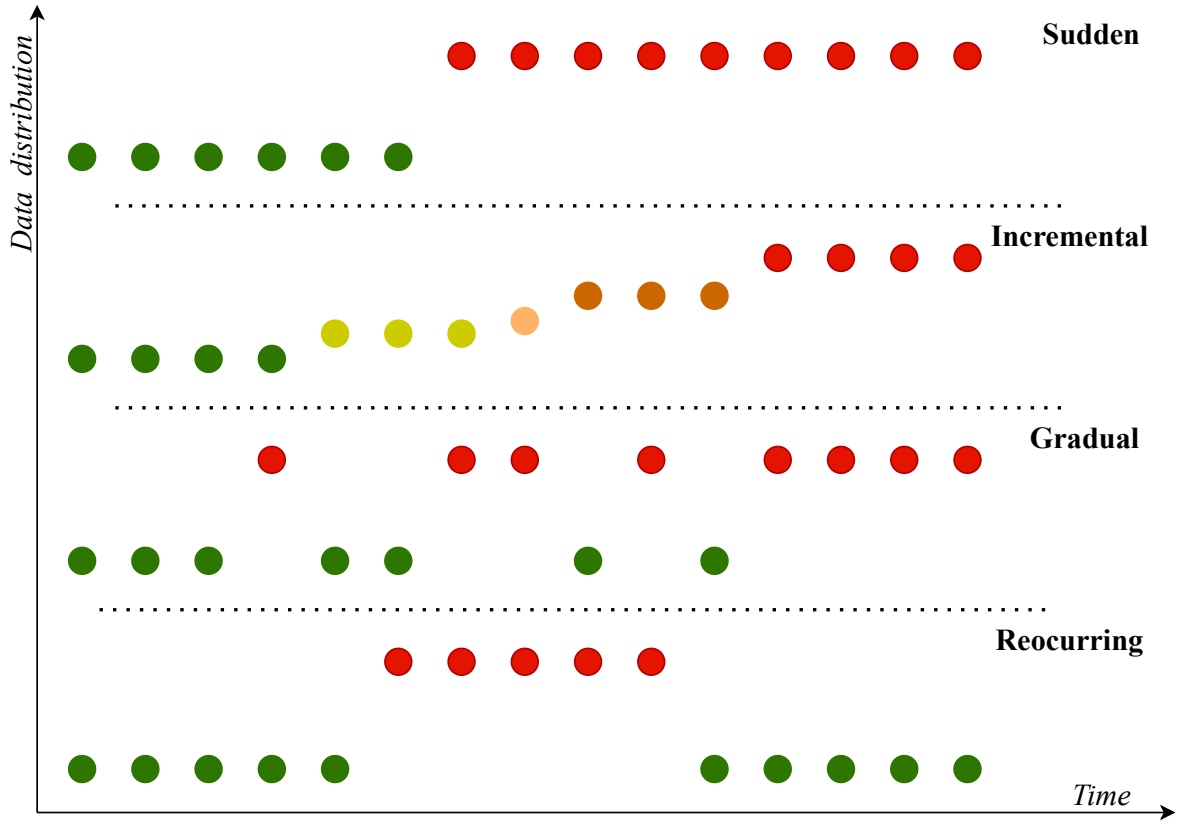

Figure 2.2: Tipi di Concept Drift.

quello precedente, per poi stabilizzarsi e infine prevalere;

- Il drift potrebbe essere "ricorrente" (*reoccurring*), se "vecchi concetti" ricorrono ciclicamente, ad esempio in base a cambiamenti stagionali o eventi periodici.

La Fig. 2.3 schematizza quello che è il flusso di lavoro più utilizzato, ad oggi, dai lavori presenti in letteratura che trattano online machine learning attenzionando esplicitamente il problema del concept drift.

Nel resto di questo capitolo si discuterà dei principali framework e algoritmi di Concept drift detection e di Concept drift adaptation rispettivamente nelle sezioni 2.1 e 2.2.

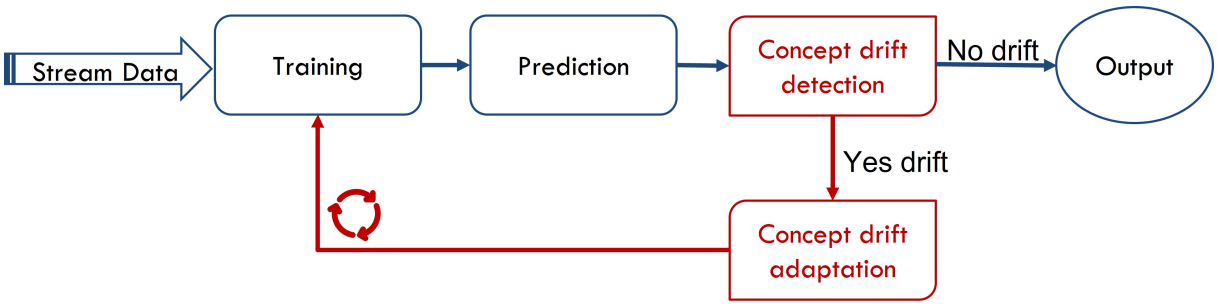

Figure 2.3: Framework per la gestione del Concept Drift.

2.1 Concept Drift Detection

Con Concept drift detection ci si riferisce a tutte quelle tecniche e meccanismi che consentono di decidere quando la differenza tra $P_t(X,y)$ e $P_{t+1}(X,y)$ diventa abbastanza grande da poter essere ricondotta con molta probabilità al verificarsi di un concept drift.

Un framework generale per il rilevamento del concept drift contiene quattro fasi [47]:

1. **Recupero dei dati**: in questa fase rientrano problematiche su come trattare il flusso di dati in arrivo. Spesso si lavora su finestre scorrevoli di dati di dimensione fissa o variabile, che contengono "dati storici" e "nuovi dati". Una singola istanza di dati non può contenere abbastanza informazioni per dedurre la distribuzione complessiva, bisogna organizzare i dati in batch per formare un modello significativo.

2. **Modellazione dei dati**: in questa fase si eseguono delle elaborazioni sui dati al fine di estrarre dai dati grezzi caratteristiche chiave o informazioni più significative al fine dell'individuazione del concept drift. Alcuni esempi di queste elaborazioni sono riduzione della dimensionalità o riduzione della dimensionalità del campione. Questa fase è facoltativa. Molte delle tecniche utilizzate in questa fase nascono per essere applicate sull'intero set di dati; quindi, è utile riflettere sul significato di tali procedure applicate su finestre di dati che contengono solo una determinata parte dei campioni come stabilito nella fase precedente.

3. **Calcolo delle statistiche del test di ipotesi**: si tratta di definire una misura di dissimilarità o stima della distanza accurata e robusta. Questa fase è la più delicata ed importante ed è ancora una questione aperta in letteratura. L'output di questa fase è l'input della fase successiva.

4. **Test di ipotesi**: si utilizza un test di ipotesi per valutare la significatività statistica della misura di dissimilarità calcolata nella fase precedente. Si vuole capire, cioè, quanto è probabile che il cambiamento sia stato causato dal concept drift e non dal rumore o dalla selezione casuale del campione. Alcuni test d'ipotesi più comunemente utilizzati sono: stima della distribuzione delle statistiche del test, bootstrapping, test di permutazione, e identificazione dei limiti basata sulla disuguaglianza di Hoeffding.

Rispetto al framework precedentemente discusso, soprattutto per quanto riguarda le statistiche di test che applicano (Fase 3), gli algoritmi di Concept drift detection possono essere classificati in tre categorie:

- **Rilevamento del concept drift basato sul tasso di errore**: come suggerisce il nome fanno parte di questa categoria gli algoritmi che prendono in input l'output dei predittori di base. La Fig. 2.3, in effetti, è più adatta a descrivere il flusso di lavoro di questa specifica categoria di algoritmi. Tuttavia, questa tipologia di rilevamento è quella più diffusa e discussa in letteratura. L'idea è che un aumento o una diminuzione nel tasso di errore dei predittori può essere sintomo di un avvenuto concept drift nei dati in ingresso. Se si dimostra che un aumento o una diminuzione del tasso di errore è statisticamente significativo, viene attivato il processo di Concept drift adaptation. Sono stati sviluppati diversi algoritmi di questa categoria soprattutto variando il test di ipotesi eseguito [49, 50, 51, 52, 53, 54, 55]. Tre di questi, oltre ad essere tra i più rappresentativi per la categoria, si differenziano anche per la fase di recupero dei dati:

  - **Drift Detection Method (DDM)** [51]: in questo algoritmo, la fase di recupero dati è implementata da una finestra temporale di riferimento, come mostrato nella Fig. 2.4. DDM rileva se il tasso di errore online complessivo che si ottiene sulle valutazioni dei nuovi dati è aumentato significativamente rispetto al tasso di errore sui dati storici. La finestra dei nuovi dati cresce man mano che nuovi dati sono disponibili per la valutazione.
  - **Statistical Test of Equal Proportions Detection (STEPD)** [55]: per ogni timestamp esistono due finestre. La variazione del tasso di errore viene rilevata confrontando la finestra temporale più recente con la finestra temporale complessiva, come mostrato in Fig. 2.5. La dimensione della finestra dei nuovi dati è un parametro che deve essere definito dall'utente.

  - **Adaptive Windowing (ADWIN)** [52]: come STEPD utilizza due finestre, ma a differenza di quest'ultimo non richiede all' utente di definire in anticipo le dimensioni delle finestre, ma solo di specificare la dimensione totale n di una finestra W "sufficientemente grande". L'algoritmo decide le dimensioni delle due finestre tagliando W in diversi punti e analizzando la media di una statistica calcolata sulle due finestre ottenute. Se il valore assoluto della differenza tra le due medie supera una soglia predefinita, viene rilevato un cambiamento in quel punto e tutti i dati precedenti vengono scartati.

- **Rilevamento del concept drift basato sulla distribuzione dei dati**: gli algoritmi di questa categoria [56, 57, 58, 59] utilizzano una metrica di distanza per quantificare la dissimilarità tra la distribuzione dei dati storici e quella dei nuovi dati. Come sempre, la dissomiglianza deve essere statisticamente significativa affinché il sistema attivi un processo di Concept drift adaptation. Questi algoritmi affrontano il concept drift analizzando direttamente la distribuzione dei dati in input. Ciò permette di identificare con precisione il momento, la posizione e la gravità del drift (la metrica utilizzata per confrontare due finestre di dati riflette proprio l'intensità di tale drift) a discapito di un costo computazionale più elevato rispetto agli algoritmi basati sul tasso di errore, che di contro, non permettono di avere una stima diretta della gravità perché si concentrano principalmente sul monitoraggio delle prestazioni del sistema di apprendimento, non sui cambiamenti del "concetto" stesso. Inoltre, questi algoritmi di solito richiedono agli utenti di definire la dimensione e il comportamento della finestra dei dati storica e della finestra dei nuovi dati. La strategia comunemente utilizzata prevede due finestre scorrevoli con la finestra temporale dei dati storici che rimane fissa fino al rilevamento del drift e la finestra dei nuovi dati che scorre ogni qual volta che un nuovo dato è disponibile, come mostrato in Fig. 2.6.

- **Rilevamento del concept drift basato su test di ipotesi multipli**: gli algoritmi di rilevamento del drift con test di ipotesi multipli [60, 61, 62] applicano tecniche menzionate nelle due categorie precedenti con la novità che usano test di ipotesi multipli per rilevare il concept drift in modi diversi, con l'idea di rendere il rilevamento finale più robusto. Questi algoritmi possono essere suddivisi in test di ipotesi multipli paralleli (Fig. 2.7) e test di ipotesi multipli gerarchici (Fig. 2.8).

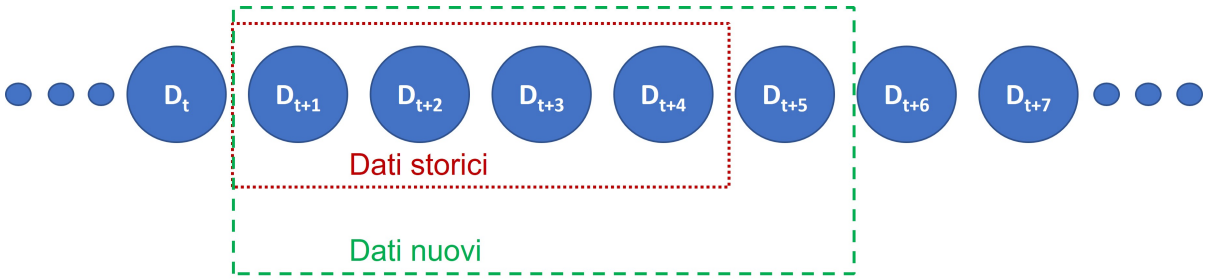

Figure 2.4: Il punto di partenza della finestra è fisso, mentre il punto finale della finestra viene esteso dopo la ricezione di una nuova istanza di dati.

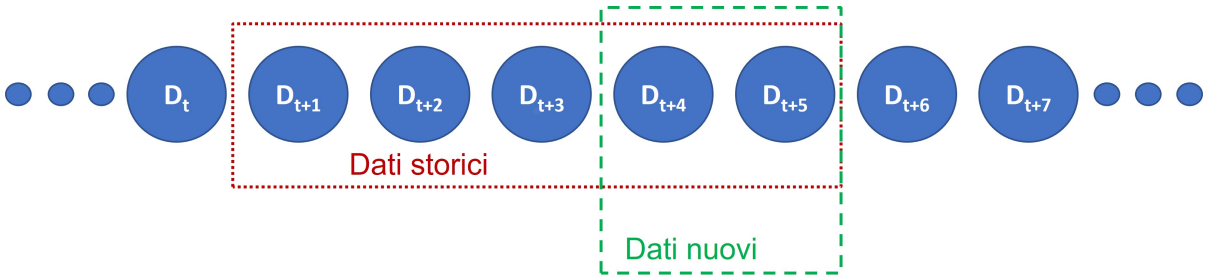

Figure 2.5: Due finestre temporali per il rilevamento del concept drift. La nuova finestra di dati deve essere definita dall'utente.

In alcuni casi, come accennato precedentemente discutendo delle tecniche di rilevamento basate sulla distribuzione dei dati, insieme al rilevamento del concept drift si vogliono estrarre alcune informazioni riguardo:

- **Il momento del concept drift**: L'informazione più banale del rilevamento del concept drift riguarda l'identificazione del tempo t in cui essa avviene. Ricordando la formula del concept drift:

$$\exists t : P_t(X,y) \neq P_{t+1}(X,y)$$

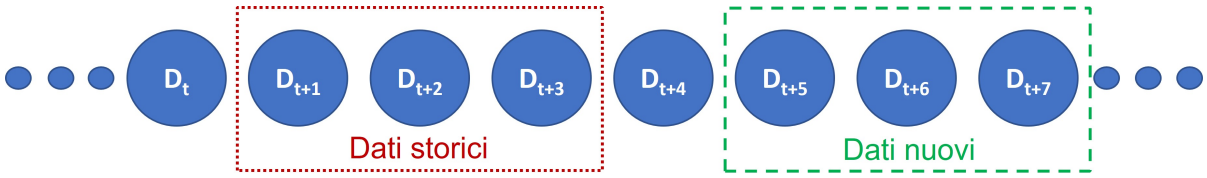

Figure 2.6: Due finestre temporali scorrevoli, di dimensioni fisse. La finestra dei dati storici rimarrà fissa, mentre la finestra dei nuovi dati continuerà a muoversi.

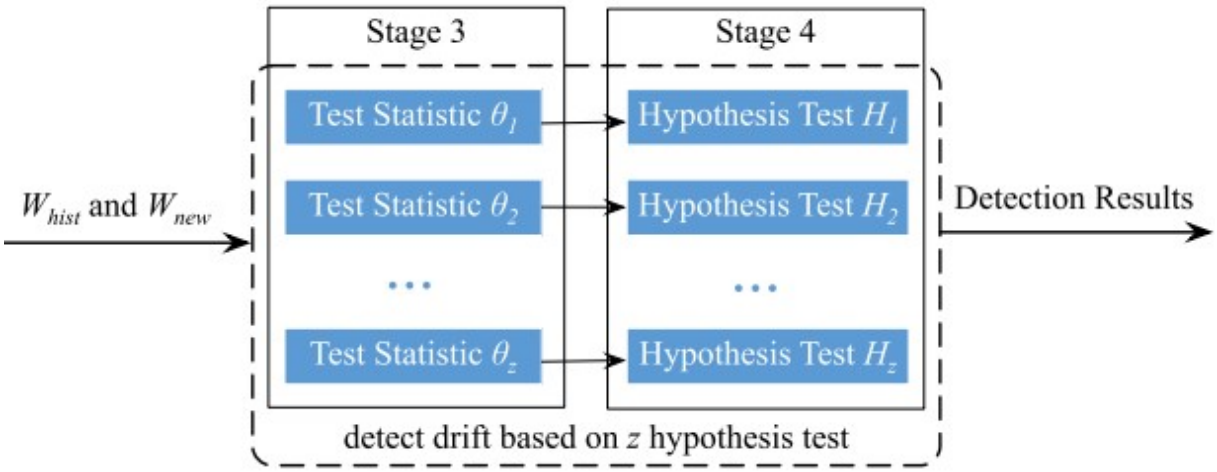

Figure 2.7: Test di ipotesi multipli paralleli. Immagine tratta da [47]

la variabile t rappresenta proprio l'istante temporale in cui il concept drift avviene. L'identificazione precisa del momento in cui esso si verifica è fondamentale per il processo di adattamento di un sistema di apprendimento; un ritardo o un falso allarme porteranno il sistema di apprendimento a non seguire i "nuovi concetti". Come si può vedere dalla Fig. 2.9 ci sarà sempre un certo ritardo temporale tra il momento in cui il concept drift avviene e il momento in cui esso viene rilevato. Questo perché i primi dati appartenenti ad un "nuovo concetto" non saranno in un numero sufficiente all'interno della finestra attuale dei dati per far sì che essa si discosti abbastanza da quella dei dati storici.

- **Gravità del concept drift**: La gravità del concept drift si può misurare come somiglianza o distanza tra il "nuovo concetto" e quello precedente, come mostrato nella Fig. 2.9. Formalmente, la gravità del concept drift può essere rappresentata come:

$$\Delta = \delta(P_t(X,y), P_{t+1}(X,y))$$

dove δ è una funzione per misurare la discrepanza di due distribuzioni di dati e t è il timestamp in cui si è verificata il concept drift. Maggiore è il valore di Δ maggiore è la gravità del concept drift.

- **Posizione del concept drift**: Per posizione del concept drift si intende l'individuazione di quelle regioni di conflitto tra un "nuovo concetto" e quello precedente. Le regioni di

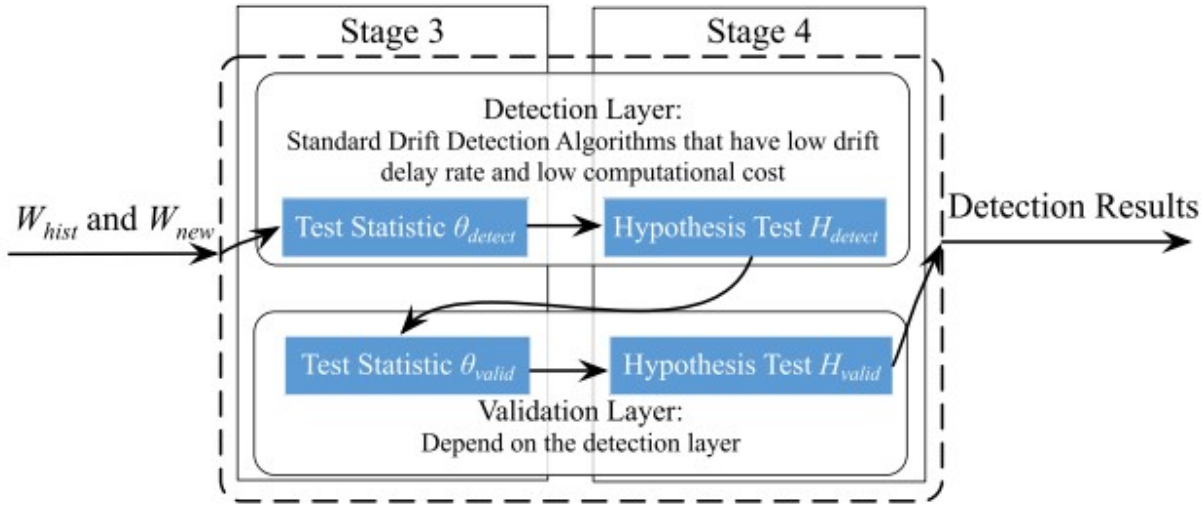

Figure 2.8: Test di ipotesi multipli gerarchici. Immagine tratta da [47]

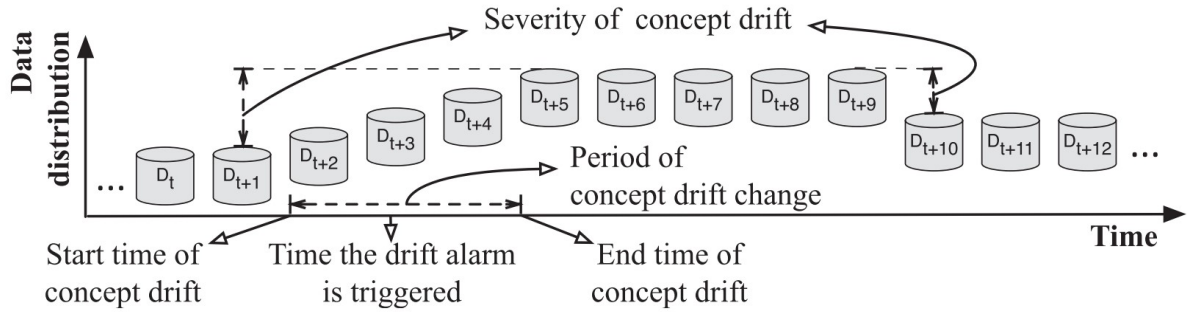

Figure 2.9: Informazioni relative al concept drift. Immagine tratta da [47]

drift sono individuate trovando regioni nello spazio delle caratteristiche dei dati X in cui $P_t(X,y)$ e $P_{t+1}(X,y)$ hanno una differenza significativa.

Tali informazioni (Fig. 2.9), quando possibile, vengono recuperate dall'algoritmo di concept drift detection e possono essere fornite alla fase di Concept drift adaptation per adattarsi al meglio.

2.2 Concept Drift Adaptation

Con questo termine, in letteratura, ci si riferisce a tutte le strategie intraprese per rispondere al concept drift una volta che esso è stato individuato dalla fase di Concept drift detection.

Le strategie più diffuse prevedono **il riaddestramento di un nuovo modello** o **il parziale aggiornamento di quello esistente**, quando la predizione viene svolta da un singolo modello, o **una di queste operazioni su uno o più modelli di un ensemble**. Questa fase, ovviamente, è fortemente connessa con la natura del compito che il sistema sta svolgendo, e quindi con la tipologia di modello utilizzato (supervisionato/ non supervisionato, algoritmo di machine learning classico piuttosto che rete neurale o alberi decisionali).

La strategia più adottata ([63, 64, 65]) è quella che prevede un approccio a finestre, come quelli sopra descritti, in modo da avere a disposizione, volta per volta, i dati storici, su cui è stato addestrato il modello corrente, e i nuovi dati. Successivamente quando un rilevatore basato sul tasso di errore (strategia più utilizzata), o sulla distribuzione dei dati, rileva un drift, il modello viene sostituito con uno addestrato sulla finestra dei nuovi dati Fig. 2.10.

Un'alternativa al riaddestramento dell'intero modello è lo sviluppo di un modello che apprende in modo adattivo dai dati in evoluzione. Tali modelli hanno la capacità di aggiornarsi parzialmente quando la distribuzione dei dati sottostanti cambia. Questo approccio è probabilmente più efficiente del riaddestramento quando il drift si verifica solo in regioni locali. Gli algoritmi basati sugli alberi decisionali si adattano bene a questa strategia perché essi intrinsecamente hanno la capacità di esaminare e adattarsi a ogni sotto regione separatamente ([66, 67, 68]).

Più recenti, ma in continuo aumento, sono i lavori che sfruttano i metodi di ensemble per gestire il concept drift. Gli stessi metodi di ensemble hanno recentemente ricevuto molta attenzione nella comunità di ricerca. Essi comprendono un insieme di modelli di base di tipo eterogeneo o con parametri diversi. La valutazione dei dati appena arrivati scaturisce da una combinazione, eseguita tramite determinate regole di voto, delle valutazioni dei classificatori di base. I metodi classici di ensemble conosciuti sono stati estesi per lavorare con dati in streaming e con il concept drift [69, 70, 71, 72]. Le strategie più utilizzate, si riconducono alle due strategie di base che lavorano sul singolo modello, con la differenza che occorre un processo aggiuntivo per decidere quale sia il modello dell'ensemble con le prestazioni peggiori da sostituire o da aggiornare. Oltre a estendere i metodi classici, sono stati sviluppati molti nuovi metodi di ensemble per gestire il concept drift agendo anche sulle tecniche di voto. Un lavoro interessante, che rappresenta tale categoria, è Dynamic Weighted Majority (DWM) [71]. Se l'ensemble sbaglia la classificazione di un'istanza, DWM addestra un nuovo classificatore di base e lo aggiunge all'ensemble. Se un classificatore di base sbaglia a classificare un'istanza, DWM riduce il suo peso di un fattore. Quando il peso di un classificatore di base scende al

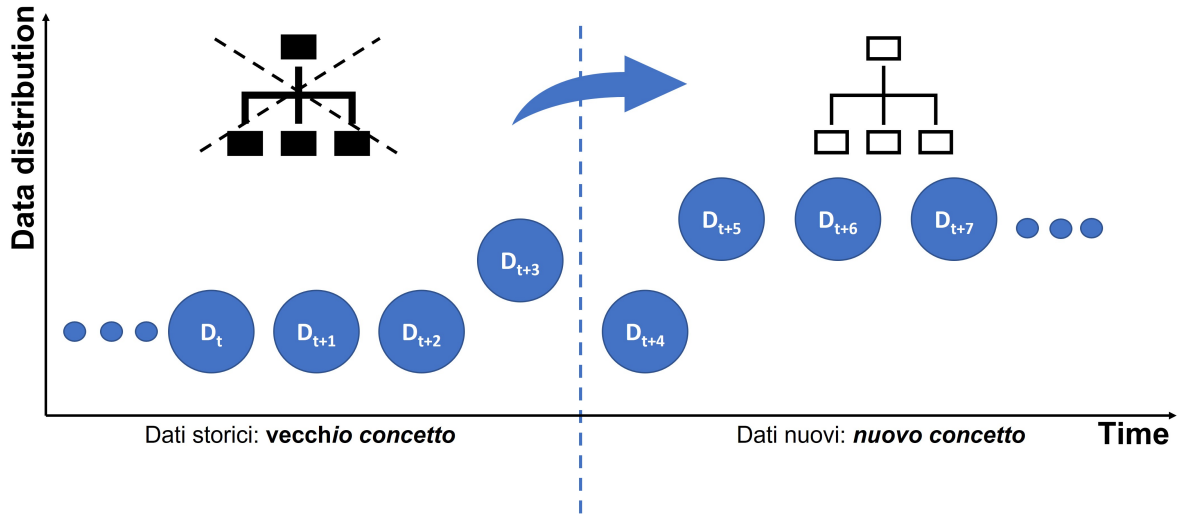

Figure 2.10: Un nuovo modello viene addestrato con i dati più recenti per sostituire il vecchio modello quando viene rilevato un concept drift.

di sotto di una soglia definita dall'utente, DWM lo rimuove dall'ensemble. Un altro lavoro, Learn++NSE [70] modifica il peso dei voti dei classificatori in base al loro tasso di errore di predizione sull'ultimo lotto di dati. Se il tasso di errore del classificatore più recente supera il 50% viene addestrato un nuovo classificatore basato sui dati più recenti. Questo lavoro riesce a gestire il concept drift improvviso, graduale e ricorrente. Non ci sono molti lavori, in letteratura, che si occupano esplicitamente di concept drift ricorrente [73, 74, 75], tuttavia la tecnica con ensemble ha aperto la strada per affrontare questa tipologia di concept drift. Si possono conservare un insieme di modelli che rappresentano vari "concetti", e gestire la predizione finale del sistema agendo sul sistema di ponderazione del voto, facendola influenzare maggiormente dal modello più adatto a gestire il "concetto attuale".

Chapter 3

Architettura Proposta

In questo capitolo viene presentata l'architettura multi-livello del sistema proposto di Online Anomaly detection. Le sezioni che precedono la descrizione vera e propria dell'architettura fungono da guida per capire i criteri secondo cui essa è stata realizzata in modo da inquadrare l'architettura proposta relativamente agli argomenti trattati nel Capitolo 2 e i limiti che riesce a superare rispetto ai sistemi presenti in letteratura.

OMISSIS

Chapter 4

Conclusioni

In questo lavoro si è studiato il fenomeno del concept drift nel contesto dello streaming dei dati per i sistemi intelligenti, evidenziando l'importanza di gestire esplicitamente il concept drift ricorrente. A tal fine, è stato proposto un nuovo sistema che combina le tecniche di rilevamento delle anomalie e del concept drift, utilizzando l'algoritmo LOF statico in modo ibrido per gestire i dati in streaming. Inoltre, è stato proposto un metodo innovativo per affrontare in modo esplicito il concept drift ricorrente, mantenendo una cronologia dei modelli passati e riducendo al minimo il numero di fasi di re-training necessarie, anche nel caso di concept drift improvviso.

Il sistema proposto è caratterizzato da un'architettura multi-livello ed è composto da due moduli di rilevamento del concept drift e da un ensemble di modelli LOF per il rilevamento delle anomalie. Per convalidare l'approccio, il sistema è stato valutato ampiamente con esperimenti multipli su tre set di dati.

I risultati mostrano che il sistema è efficace nel rilevare gli outlier, sia che i dati in ingresso presentino o meno concept drift, raggiungendo valori elevati di accuratezza e di F1-score.

Inoltre, la percentuale di modelli riutilizzati è estremamente elevata, a dimostrazione del fatto che l'approccio adottato è efficace nel gestire il concept drift ricorrente, riducendo al minimo il numero di nuovi modelli da addestrare.

Alcuni sviluppi futuri potrebbero riguardare l'utilizzo di finestre di dati di dimensioni variabili, a seconda della frequenza e della gravità del drift rilevato in un determinato lasso di tempo.

Un altro miglioramento del sistema proposto sarebbe quello di considerare anche un numero dinamico di modelli nella storia, per adattarsi meglio ai periodi in cui il rapporto di riutilizzo dei

modelli è particolarmente alto o basso, o per ideare una politica di sostituzione dei modelli più sofisticata, ad esempio utilizzando un meccanismo di reputazione [76, 77].

Ciò consentirebbe al sistema di ridurre il più possibile la dimensione della finestra e di aumentare il numero di modelli nei periodi in cui si verificano molti concept drift diversi e, viceversa, di aumentare la dimensione della finestra e di ridurre il numero di modelli nei periodi più stabili, con un conseguente aumento complessivo delle prestazioni.

List of Figures

List of Tables

Bibliografia

[1] A. De Paola, P. Ferraro, S. Gaglio, G. Lo Re, M. Morana, M. Ortolani, and D. Peri. "A context-aware system for ambient assisted living". In: *International Conference on Ubiquitous Computing and Ambient Intelligence*. Springer. 2017, pp. 426–438.

[2] Y. Liu, L. Kong, and G. Chen. "Data-oriented mobile crowdsensing: A comprehensive survey". In: *IEEE communications surveys & tutorials* 21.3 (2019), pp. 2849–2885.

[3] P. Ferraro and G. Lo Re. "Designing ontology-driven recommender systems for tourism". In: *Advances onto the Internet of Things*. Springer, 2014, pp. 339–352.

[4] R. Al-amri, R. K. Murugesan, M. Man, A. F. Abdulateef, M. A. Al-Sharafi, and A. A. Alkahtani. "A review of machine learning and deep learning techniques for anomaly detection in IoT data". In: *Applied Sciences* 11.12 (2021), p. 5320.

[5] K. K. Santhosh, D. P. Dogra, and P. P. Roy. "Anomaly detection in road traffic using visual surveillance: A survey". In: *ACM Computing Surveys (CSUR)* 53.6 (2020), pp. 1–26.

[6] S. Ali, T. Glass, B. Parr, J. Potgieter, and F. Alam. "Low Cost Sensor With IoT LoRaWAN Connectivity and Machine Learning-Based Calibration for Air Pollution Monitoring". In: *IEEE Transactions on Instrumentation and Measurement* 70 (2021), pp. 1–11.

[7] V. Agate, F. Concone, and P. Ferraro. "A Resilient Smart Architecture for Road Surface Condition Monitoring". In: *The Proceedings of the International Conference on Smart City Applications*. Springer. 2021, pp. 199–209.

[8] D. Santani, T.-M.-T. Do, F. Labhart, S. Landolt, E. Kuntsche, and D. Gatica-Perez. "DrinkSense: Characterizing Youth Drinking Behavior Using Smartphones". In: *IEEE Transactions on Mobile Computing* 17.10 (2018), pp. 2279–2292.

[9] F. Restuccia, P. Ferraro, S. Silvestri, S. K. Das, and G. Lo Re. "IncentMe: Effective mechanism design to stimulate crowdsensing participants with uncertain mobility". In: *IEEE Transactions on Mobile Computing* 18.7 (2018), pp. 1571–1584.

[10] V. Agate, A. De Paola, G. Lo Re, and M. Morana. "Vulnerability Evaluation of Distributed Reputation Management Systems". In: *InfQ 2016 - New Frontiers in Quantitative Methods in Informatics*. ICST, Brussels, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2016, pp. 1–8.

[11] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang. "Learning under concept drift: A review". In: *IEEE Transactions on Knowledge and Data Engineering* 31.12 (2018), pp. 2346–2363.

[12] A. De Paola, P. Ferraro, S. Gaglio, G. Lo Re, and S. K. Das. "An adaptive bayesian system for context-aware data fusion in smart environments". In: *IEEE Transactions on Mobile Computing* 16.6 (2016), pp. 1502–1515.

[13] A. Timilsina, A. R. Khamesi, V. Agate, and S. Silvestri. "A Reinforcement Learning Approach for User Preference-aware Energy Sharing Systems". In: *IEEE Transactions on Green Communications and Networking* (2021).

[14] V. Agate, F. M. D'Anna, A. De Paola, P. Ferraro, G. Lo Re, and M. Morana. "A Behavior-Based Intrusion Detection System Using Ensemble Learning Techniques." In: *ITASEC*. 2022.

[15] V. Agate, A. R. Khamesi, S. Silvestri, and S. Gaglio. "Enabling peer-to-peer User-Preference-Aware Energy Sharing Through Reinforcement Learning". In: *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*. 2020.

[16] A. De Paola, P. Ferraro, G. Lo Re, M. Morana, and M. Ortolani. "A fog-based hybrid intelligent system for energy saving in smart buildings". In: *Journal of Ambient Intelligence and Humanized Computing* 11.7 (2020), pp. 2793–2807.

[17] V. Chandola, A. Banerjee, and V. Kumar. "Anomaly detection: A survey". In: *ACM computing surveys (CSUR)* 41.3 (2009), pp. 1–58.

[18] V. Agate, A. De Paola, P. Ferraro, G. Lo Re, and M. Morana. "SecureBallot: A secure open source e-Voting system". In: *Journal of Network and Computer Applications* 191 (2021).

[19] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung. "Intrusion detection system: A comprehensive review". In: *Journal of Network and Computer Applications* 36.1 (2013), pp. 16–24.

[20] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. "Estimating the support of a high-dimensional distribution". In: *Neural computation* 13.7 (2001), pp. 1443–1471.

[21] W. Alhakami, A. ALharbi, S. Bourouis, R. Alroobaea, and N. Bouguila. "Network anomaly intrusion detection using a nonparametric Bayesian approach and feature selection". In: *IEEE Access* 7 (2019), pp. 52181–52190.

[22] A. P. Muniyandi, R. Rajeswari, and R. Rajaram. "Network anomaly detection by cascading k-Means clustering and C4. 5 decision tree algorithm". In: *Procedia Engineering* 30 (2012), pp. 174–182.

[23] S. Naseer, Y. Saleem, S. Khalid, M. K. Bashir, J. Han, M. M. Iqbal, and K. Han. "Enhanced network anomaly detection based on deep neural networks". In: *IEEE access* 6 (2018), pp. 48231–48246.

[24] V. Agate, F. Concone, A. De Paola, P. Ferraro, G. Lo Re, and M. Morana. "Bayesian Modeling for Differential Cryptanalysis of Block Ciphers: A DES Instance". In: *IEEE Access* 11 (2023), pp. 4809–4820.

[25] P. D. Domański. "Study on statistical outlier detection and labelling". In: *International Journal of Automation and Computing* 17.6 (2020), pp. 788–811.

[26] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. "LOF: identifying density-based local outliers". In: *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. 2000, pp. 93–104.

[27] C. C. Aggarwal, S. Y. Philip, J. Han, and J. Wang. "A framework for clustering evolving data streams". In: *Proceedings 2003 VLDB conference*. Elsevier. 2003, pp. 81–92.

[28] I. Assent, P. Kranen, C. Baldauf, and T. Seidl. "Anyout: Anytime outlier detection on streaming data". In: *Database Systems for Advanced Applications: 17th International Conference, DASFAA 2012, Busan, South Korea, April 15-19, 2012, Proceedings, Part I 17*. Springer Berlin Heidelberg. 2012, pp. 228–242.

[29] F. Angiulli and F. Fassetti. “Detecting distance-based outliers in streams of data”. In: *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. 2007, pp. 811–820.

[30] Z. Cheng, C. Zou, and J. Dong. “Outlier detection using isolation forest and local outlier factor”. In: *Proceedings of the conference on research in adaptive and convergent systems*. 2019, pp. 161–168.

[31] F. T. Liu, K. M. Ting, and Z.-H. Zhou. “Isolation-based anomaly detection”. In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 6.1 (2012), pp. 1–39.

[32] V. Agate, S. Drago, P. Ferraro, and G. Lo Re. “Anomaly Detection for Reoccurring Concept Drift in Smart Environments”. In: *2022 18th International Conference on Mobility, Sensing and Networking (MSN)*. IEEE. 2022, pp. 113–120.

[33] V. Agate, P. Ferraro, and S. Gaglio. “A Cognitive Architecture for Ambient Intelligence Systems”. In: *AIC*. 2018, pp. 52–58.

[34] J. Gao, W. Ji, L. Zhang, A. Li, Y. Wang, and Z. Zhang. “Cube-based incremental outlier detection for streaming computing”. In: *Information Sciences* 517 (2020), pp. 361–376.

[35] W. Ng and M. Dash. “Discovery of frequent patterns in transactional data streams”. In: *Transactions on large-scale data-and knowledge-centered systems II* (2010), pp. 1–30.

[36] M. U. Togbe, Y. Chabchoub, A. Boly, M. Barry, R. Chiky, and M. Bahri. “Anomalies detection using isolation in concept-drifting data streams”. In: *Computers* 10.1 (2021), p. 13.

[37] A. Kartit, A. Saidi, F. Bezzazi, M. El Marraki, and A. Radi. “A new approach to intrusion detection system”. In: *Journal of theoretical and applied information technology* 36.2 (2012), pp. 284–289.

[38] F. Sabahi and A. Movaghar. “Intrusion detection: A survey”. In: *2008 Third International Conference on Systems and Networks Communications*. IEEE. 2008, pp. 23–26.

[39] S. Axelsson. “Intrusion detection systems: A survey and taxonomy”. In: (2000).

[40] J. Jabez and B. Muthukumar. “Intrusion detection system (IDS): anomaly detection using outlier detection approach”. In: *Procedia Computer Science* 48 (2015), pp. 338–346.

[41] A. Patcha and J.-M. Park. "An overview of anomaly detection techniques: Existing solutions and latest technological trends". In: *Computer networks* 51.12 (2007), pp. 3448–3470.

[42] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez. "Anomaly-based network intrusion detection: Techniques, systems and challenges". In: *computers & security* 28.1-2 (2009), pp. 18–28.

[43] P. Stavroulakis and M. Stamp. *Handbook of information and communication security*. Springer Science & Business Media, 2010.

[44] A. Murali and M. Rao. "A survey on intrusion detection approaches". In: *2005 International Conference on Information and Communication Technologies*. IEEE. 2005, pp. 233–240.

[45] M. Xie, S. Han, B. Tian, and S. Parvin. "Anomaly detection in wireless sensor networks: A survey". In: *Journal of Network and computer Applications* 34.4 (2011), pp. 1302–1325.

[46] S.-S. Wang, K.-Q. Yan, S.-C. Wang, and C.-W. Liu. "An integrated intrusion detection system for cluster-based wireless sensor networks". In: *Expert Systems with Applications* 38.12 (2011), pp. 15234–15243.

[47] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia. "A survey on concept drift adaptation". In: *ACM computing surveys (CSUR)* 46.4 (2014), pp. 1–37.

[48] S. Ramirez-Gallego, B. Krawczyk, S. Garcia, M. Woźniak, and F. Herrera. "A survey on data preprocessing for data stream mining: Current status and future directions". In: *Neurocomputing* 239 (2017), pp. 39–57.

[49] A. Bifet, G. Holmes, B. Pfahringer, and R. Gavalda. "Improving adaptive bagging methods for evolving data streams". In: *Advances in Machine Learning: First Asian Conference on Machine Learning, ACML 2009, Nanjing, China, November 2-4, 2009. Proceedings 1*. Springer. 2009, pp. 23–37.

[50] M. Baena-Garcıa, J. del Campo-Ávila, R. Fidalgo, A. Bifet, R. Gavalda, and R. Morales-Bueno. "Early drift detection method". In: *Fourth international workshop on knowledge discovery from data streams*. Vol. 6. Citeseer. 2006, pp. 77–86.

[51] J. Gama, P. Medas, G. Castillo, and P. Rodrigues. "Learning with drift detection". In: *Brazilian symposium on artificial intelligence*. Springer. 2004, pp. 286–295.

[52] A. Bifet and R. Gavalda. "Learning from time-changing data with adaptive windowing". In: *Proceedings of the 2007 SIAM international conference on data mining*. SIAM. 2007, pp. 443–448.

[53] J. Gama and G. Castillo. "Learning with local drift detection". In: *Advanced Data Mining and Applications: Second International Conference, ADMA 2006, Xi'an, China, August 14-16, 2006 Proceedings 2*. Springer. 2006, pp. 42–55.

[54] G. J. Ross, N. M. Adams, D. K. Tasoulis, and D. J. Hand. "Exponentially weighted moving average charts for detecting concept drift". In: *Pattern recognition letters* 33.2 (2012), pp. 191–198.

[55] K. Nishida and K. Yamauchi. "Detecting concept drift using statistical testing". In: *International conference on discovery science*. Springer. 2007, pp. 264–269.

[56] D. Kifer, S. Ben-David, and J. Gehrke. "Detecting change in data streams". In: *VLDB*. Vol. 4. Toronto, Canada. 2004, pp. 180–191.

[57] T. Dasu, S. Krishnan, S. Venkatasubramanian, and K. Yi. "An information-theoretic approach to detecting changes in multi-dimensional data streams". In: *Proc. Symposium on the Interface of Statistics, Computing Science, and Applications (Interface)*. 2006.

[58] L. Bu, C. Alippi, and D. Zhao. "A pdf-free change detection test based on density difference estimation". In: *IEEE transactions on neural networks and learning systems* 29.2 (2016), pp. 324–334.

[59] L. Bu, D. Zhao, and C. Alippi. "An incremental change detection test based on density difference estimation". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 47.10 (2017), pp. 2714–2726.

[60] C. Alippi and M. Roveri. "Just-in-time adaptive classifiers—Part I: Detecting nonstationary changes". In: *IEEE Transactions on Neural Networks* 19.7 (2008), pp. 1145–1153.

[61] H. Wang and Z. Abraham. "Concept drift detection for streaming data". In: *2015 international joint conference on neural networks (IJCNN)*. IEEE. 2015, pp. 1–9.

[62] Y. Zhang, G. Chu, P. Li, X. Hu, and X. Wu. "Three-layer concept drifting detection in text data streams". In: *Neurocomputing* 260 (2017), pp. 393–403.

[63] S. H. Bach and M. A. Maloof. "Paired learners for concept drift". In: *2008 Eighth IEEE International Conference on Data Mining*. IEEE. 2008, pp. 23–32.

[64] D. Liu, Y. Wu, and H. Jiang. "FP-ELM: An online sequential learning algorithm for dealing with concept drift". In: *Neurocomputing* 207 (2016), pp. 322–334.

[65] S. G. Soares and R. Araújo. "An adaptive ensemble of on-line extreme learning machines with variable forgetting factor for dynamic system prediction". In: *Neurocomputing* 171 (2016), pp. 693–707.

[66] P. Domingos and G. Hulten. "Mining high-speed data streams". In: *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2000, pp. 71–80.

[67] L. Rutkowski, L. Pietruczuk, P. Duda, and M. Jaworski. "Decision trees for mining data streams based on the McDiarmid's bound". In: *IEEE Transactions on Knowledge and Data Engineering* 25.6 (2012), pp. 1272–1279.

[68] L. Rutkowski, M. Jaworski, L. Pietruczuk, and P. Duda. "A new method for data stream mining based on the misclassification error". In: *IEEE transactions on neural networks and learning systems* 26.5 (2014), pp. 1048–1059.

[69] A. Bifet, G. Holmes, and B. Pfahringer. "Leveraging bagging for evolving data streams". In: *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part I 21*. Springer. 2010, pp. 135–150.

[70] R. Elwell and R. Polikar. "Incremental learning of concept drift in nonstationary environments". In: *IEEE Transactions on Neural Networks* 22.10 (2011), pp. 1517–1531.

[71] J. Z. Kolter and M. A. Maloof. "Dynamic weighted majority: An ensemble method for drifting concepts". In: *The Journal of Machine Learning Research* 8 (2007), pp. 2755–2790.

[72] F. Chu and C. Zaniolo. “Fast and light boosting for adaptive mining of data streams”. In: *Advances in Knowledge Discovery and Data Mining: 8th Pacific-Asia Conference, PAKDD 2004, Sydney, Australia, May 26-28, 2004. Proceedings 8*. Springer. 2004, pp. 282–292.

[73] J. Gama and P. Kosina. “Recurrent concepts in data streams classification”. In: *Knowledge and Information Systems* 40.3 (2014), pp. 489–507.

[74] J. B. Gomes, M. M. Gaber, P. A. Sousa, and E. Menasalvas. “Mining recurring concepts in a dynamic feature space”. In: *IEEE Transactions on Neural Networks and Learning Systems* 25.1 (2013), pp. 95–110.

[75] Z. Ahmadi and S. Kramer. “Modeling recurring concepts in data streams: a graph-based framework”. In: *Knowledge and Information Systems* 55.1 (2018), pp. 15–44.

[76] V. Agate, A. De Paola, G. Lo Re, and M. Morana. “A platform for the evaluation of distributed reputation algorithms”. In: *2018 IEEE/ACM 22nd International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*. IEEE. 2018, pp. 1–8.

[77] V. Agate, A. De Paola, G. Lo Re, and M. Morana. “A simulation software for the evaluation of vulnerabilities in reputation management systems”. In: *ACM Transactions on Computer Systems (TOCS)* 37.1-4 (2021), pp. 1–30.