POLITECNICO DI TORINO

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA CIVILE

Tesi di Laurea Magistrale

Individuazione di anomalie nei sistemi di distribuzione delle acque

Relatori :
Prof. *Fulvio Boano*
Ing. *Marco Scibetta* (SMAT s.p.a.)

Candidato:
Carlo Arciuli
matricola: 224363

ANNO ACCADEMICO 2018 - 2019

"Alla mia Nonna . . . Anna"

Abstract

Le aziende che si occupano della gestione e distribuzione delle acque hanno avviato negli ultimi anni un processo di costante ed irreversibile, anche se graduale, trasformazione del proprio modo di operare: non più operazioni locali e manuali, bensì azioni programmate e gestite da remoto in maniera centralizzata e senza un presidio fisico sulle postazioni interessate alla attività di gestione, controllo e distribuzione.

Si tratta di una trasformazione importante, in linea peraltro con le generali tendenze consentite o imposte dai nuovi strumenti e livelli di informatizzazione di tutti i settori di attività.

Uno degli effetti della trasformazione in atto è che la distanza tra l'operatore umano o il supervisore e i processi di produzione e distribuzione dell'acqua sta gradualmente aumentando il che comporta il rischio, concreto e importante, che i guasti nel sistema di rilevazione possano rimanere inosservati nei momenti in cui nessun operatore o supervisore umano monitori i processi. Questo, evidentemente, assume importanza ancor più specifica e rilevante nei sistemi in uso nella gestione degli acquedotti, in cui variazioni dei valori standard di esercizio di apparecchiature quali pompe, valvole, sfioratori ecc. rimangano ignare agli operatori. Tali valori sono spesso correlati a perdite idriche o malfunzionamenti e pertanto molte delle rotture che si verificano lungo le tubazioni, da cui scaturiscono perdite, rimangono inosservate.

Nasce dunque l'esigenza di un sistema che, funzionando in real-time, possa, in maniera automatizzata, ed affidabile, segnalare la presenza di un evento anomalo. Con tale finalità è stata formulata, mediante il software Python, la funzione *Anna*. Tale funzione assume come principio di funzionamento la CUSUM Anomaly Detection (CAD). Si tratta di una tecnica di analisi sequenziale in genere utilizzata per monitorare e rilevare i cambiamenti delle serie su cui è svolta l'analisi.

L'elaborazione della funzione è passata prima da un processo di automatizzazione per il quale, nota la serie d'analisi, la funzione è in grado di individuare il valore ottimale dei parametri da utilizzare durante l'applicazione dell'algoritmo CUSUM.

Successivamente la funzione è stata implementata in modo che essa potesse operare in tempo reale. Infatti, integrando la funzione *Anna* al sistema Supervisory Control And Data Acquisition (SCADA), essa è in grado di acquisire i dati di tutti i componenti connessi al sistema e di produrre come output messaggi di warning che indicano una anomalia connessa al relativo componente.

Dunque tale funzione svolge anche un effetto di *Early Warning* in quanto, una volta riconosciuta l'anomalia, si può agire di conseguenza e con immediatezza, limitando l'impatto dannoso della anomalia stessa e provvedendo a risanare il malfunzionamento, in modo da tornare alle corrette condizioni di esercizio.

Abbreviazioni

ADP: Anomaly-Detection-Problem

SPC: Statistical-Process-Control

CAD: Cumsum-Anomaly-Detection

RLS: Recursive-Least-Squares

IWA: International Water Association

SMAT: Società Metropolitana Acque Torino

PLC: Programmable Logic Controller

SCADA: Supervisory Control And Data Acquisition

EWS: Early Warning System

VAI: Valore adiacente inferiore

VAS: Valore adiacente superiore

HDF: Hierarchical Data Format

CSV: Comma Separated Values

DMA: District Meter Area

MNF: Minimum Night Flow

Indice

Elenco delle tabelle

Elenco delle figure

Capitolo 1

Introduzione

Il problema dell'individuazione delle anomalie in un insieme di dati (Anomaly Detection Problem - ADP) costituisce un settore molto importante nell'ambito delle tecniche di Data Mining. Si tratta di un problema che presenta numerose applicazioni in diversi contesti, come nell'individuazione delle frodi, nella rivelazione di intrusioni in sistemi informatici, nei sistemi di supporto alle diagnosi mediche, nel marketing e in molti altri ancora. La comunità di ricerca ha proposto molte soluzioni, alcune più specifiche per determinati campi applicativi, altre più generiche [1]. Nel corso di questa tesi ci si è focalizzati sulla ricerca di anomalie nei sistemi di distribuzione e gestione delle risorse idriche.

Nella gestione degli acquedotti, le nuove strumentazioni e le sempre più sviluppate conoscenze ed applicazioni informatiche da una parte hanno condotto all'ottimizzazione dei consumi energetici, ed una più efficace gestione dell'acquedotto ma dall'altra hanno determinato una crescente distanza tra l'operatore umano e i processi di produzione e distribuzione dell'acqua. Tale aumento della distanza comporta il rischio, concreto e importante, che i guasti e i differenti comportamenti associati a problemi occorsi nel sistema possano rimanere inosservati, con ogni possibile conseguenza.

Nasce dunque l'esigenza di un sistema che, in real-time e in maniera automatizzata, segnali la presenza di un evento anomalo. Con tale finalità è stata formulata, mediante il software Python, la funzione *Anna*. Essa assume come principio di funzionamento la CUSUM Anomaly Detection (CAD), che è una tecnica di analisi sequenziale utilizzata per monitorare e rilevare i cambiamenti delle serie su cui è svolta l'analisi.

Essendo ogni acquedotto munito di un sistema Supervisory Control And Data Acquisition (SCADA) in grado di acquisire in real-time i dati di tutti i componenti connessi al sistema, integrando ad esso la funzione proposta è possibile ottenere

messaggi di warning indicanti un'anomalia connessa al relativo componente. Così facendo, si è in grado di ridurre drasticamente il periodo di tempo che va dalla nascita alla conoscenza di una potenziale rottura lungo la tubazione (*Unawareness period*).

Dando per assodato che le perdite idriche causano danni via via crescenti con lo scorrere del tempo, a causa del sempre maggior volume d'acqua perso e dei possibili danni causati all'infrastruttura circostante, per i gestori idrici la tempestività di intervento diviene un obiettivo primario da perseguire.

Dunque la funzione formulata svolge anche un effetto di *Early Warning* in quanto, una volta riconosciuta l'anomalia, si può agire di conseguenza e con immediatezza, limitandone l'impatto dannoso e provvedendo a risanare il malfunzionamento, in modo da tornare alle corrette condizioni di esercizio.

Per una completa comprensione dei temi trattati e di come si sia arrivati alla formulazione della funzione, si sintetizza qui la struttura della tesi:

dopo una breve introduzione dell'argomento, nel secondo e nel terzo capitolo oltre a fornire il background in cui il lavoro si colloca, vengono indicate alcune delle premesse teoriche necessarie alla comprensione del funzionamento e dei possibili vantaggi perseguibili dall'integrazione della funzione *Anna* nei sistemi di telecontrollo idrico.

Nel quarto capitolo si riporta il cuore dell'elaborato: partendo da una tecnica sequenziale per il controllo statistico (CUSUM o cumulative sum control chart), l'elaborazione della funzione è passata prima da un processo di automatizzazione per il quale, nota la serie d'analisi, la funzione è in grado di individuare il valore ottimale dei parametri da utilizzare durante l'applicazione dell'algoritmo CUSUM.

Successivamente la funzione è stata implementata in modo che essa potesse operare in real-time. Infine è stata integrata al sistema SCADA della Società Metropolitana Acque Torino (SMAT). In particolare, è stata connessa al sistema di monitoraggio del distretto Belgio. Grazie a questo processo, la funzione è in grado di acquisire i dati connessi al sistema e di produrre, nel momento in cui si verificasse una perdita idrica, messaggi di warning mediante l'accensione di una spia indicante un'anomalia nel sistema.

Nel quinto capitolo sono descritti i potenziali vantaggi sia in termini economici che di risorsa idrica salvata derivanti dalla previa applicazione della funzione alle centrali idriche analizzate nella tesi.

Infine nel sesto capitolo sono riportate le conclusioni dedotte, in cui oltre a fornire una visione d'insieme sull'elaborato, vengono suggeriti possibili spunti di riflessione ed offerte possibili prospettive a chi, in futuro, volesse ulteriormente riprendere ed approfondire la materia.

Capitolo 2

Anomaly Detection Problem

Il problema della rivelazione delle anomalie (Anomaly/Outlier Detection Problem) consiste nell'individuare dei pattern, in un insieme di dati, che differiscono dal normale comportamento aspettato[1]. Tali pattern discordanti potranno essere indicati con nomi diversi, in base al dominio di riferimento: eccezioni, agenti contaminanti, anomalie (anomaly o outlier), aberrazioni, osservazioni discordanti, peculiarità, ecc. E' appena il caso di sottolineare che la rivelazione delle anomalie trova applicazioni estremamente importanti in vari contesti. A titolo di esempio, nella individuazione delle frodi informatiche, i sistemi di rilevazione delle anomalie vengono di solito utilizzati per individuare possibili furti di carte di credito. In questi casi, il malintenzionato inizia di norma ad effettuare un alto numero di acquisti, diversamente da quel che fa normalmente un possessore di carta di credito. Dunque, la individuazione di buying patterns anomali, come mostrato in Figura 2.1, può indicare un possibile furto.

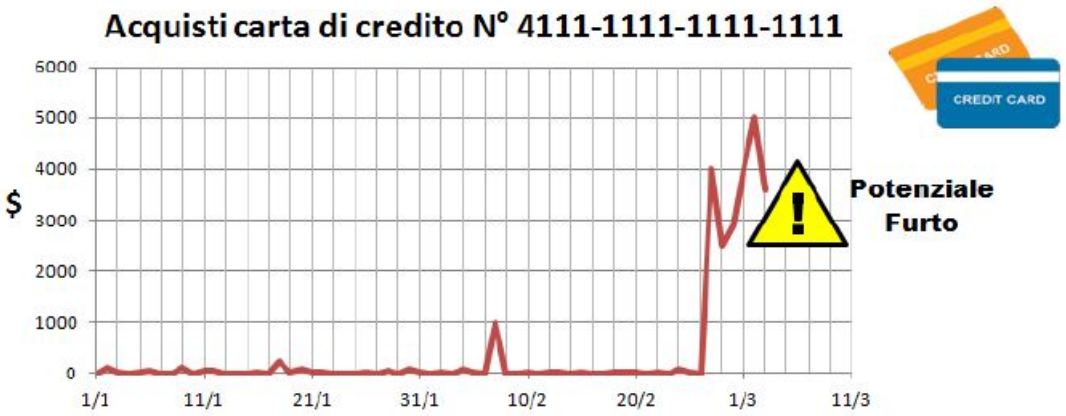


Figura 2.1: Esempio Anomaly Fraud Detection [26].

In medicina, il manifestarsi di sintomi e/o di risultati di analisi strumentali non ordinari può esser indice di patologie in atto di grado piu o meno avanzato. In informatica, nelle procedure di accertamento delle intrusioni nei PC e nelle reti telematiche, vengono individuati i potenziali hackeraggi attraverso il costante monitoraggio delle attività ordinarie ed il loro confronto con i pattern indici dell'ordinario comportamento del sistema. I programmi pirata e gli hacker che si introducono in una rete determinano indici di attività di solito estremamente discordanti dall'ordinario comportamento, il che fa sì che un efficace sistema di outlier detection possa rapidamente rilevare, riconoscere e contrastare gli attacchi. Altre applicazioni significative ed utili, d'altronde, si hanno nell'individuazione e nella soluzione di problemi in ambito di spionaggio industriale, nella sicurezza militare, nell'image processing e, come ampiamente sostenuto e dimostrato anche con questa Tesi, nei sistemi di approvvigionamento idrico.

Douglas M. Hawkins [2] definisce una anomalia (*anomaly o outlier*) come un'osservazione che devia in maniera tanto evidente, rispetto alle altre osservazioni, da alimentare sospetti che essa sia stata generata attraverso un meccanismo differente. Si tratta di una definizione che si basa tutta su considerazioni di tipo statistico partendo dal presupposto che gli oggetti normali seguano un comune o ordinario "meccanismo di generazione" analogo ad un certo processo statistico. Vengono così considerati outlier gli oggetti che si discostano da tale meccanismo predeterminato. Generalizzando tale definizione potremo definire le anomalie come pattern che non seguono il comportamento normalmente atteso. Preliminarmente dunque, volendo porre in essere un tentativo di rivelazione delle anomalie, sarà necessario definire una regione che rappresenti il comportamento normale e dichiarare come outlier ogni osservazione che da essa si differenzi. Ovviamente, svariati fattori rendono questo approccio delicato e complesso. Definire infatti una regione come normale e tale da contenere e rappresentare ogni possibile comportamento ordinario, è evidentemente difficile, perchè il confine tra normale e anomalo non è mai netto e predeterminabile con certezza. Di solito inoltre i pattern che indicano il comportamento o l'andamento normale sono in costante evoluzione e, dunque, la rappresentazione attuale di comportamento normale può non essere più valida o attendibile in un futuro anche immediato o prossimo. Infine, il significato o la definizione di anomalia può essere diversa a seconda dello specifico settore d riferimento. Per esempio, in ambito sanitario una piccola deviazione dalle normali condizioni (si pensi alle variazioni della temperatura corporea) può indicare un'anomalia, laddove nei mercati finanziari le fluttuazioni più o meno significative dei prezzi possono essere considerati normali.

In materia di rilevazione delle anomalie, è evidentemente necessario tener conto anche della disponibilità o meno di dati già classificati come normali o outlier e tali

che possano utilizzarsi come modelli di training o di validazione. In relazione a tutti questi elementi di valutazione, sono nate tecniche diverse di rilevazione e diverse definizioni dell'outlier detection problem. Di tali tecniche o definizioni,alcune sono più generali ed altre più specifiche per particolari settori o domini. Le diverse formulazioni del problema possono dunque dipendere, per quanto abbiamo più sopra sostenuto, dalla natura dei dati, dalla disponibilità o meno di dati già classificati, dal tipo di anomalie che devono essere ricercate e da diversi altri fattori

2.1 Differenti aspetti di un ADP

Questa sezione identifica e discute i diversi aspetti della rilevazione delle anomalie. Come menzionato in precedenza, una specifica formulazione del problema è determinata da diversi aspetti, quali la natura dei dati di input, la possibilità o l'impossibilità di sapere quali dei dati a disposizione sono anomali e quali no, o di conoscere o meno i requisiti indotti dal dominio di applicazione. Tutto ciò giustifica la necessità dell'ampio spettro di tecniche di rilevamento delle anomalie.

2.1.1 Natura dei dati di input

Un aspetto chiave di qualsiasi tecnica di rilevamento delle anomalie è la natura dei dati di input. L'input è generalmente una raccolta di istanze di dati (anche denominate oggetto, record, punto, vettore, campione, osservazione o entità). Ogni istanza di dati può essere descritta utilizzando una serie di attributi (anche denominata variabile, caratteristica, campo o dimensione). Gli attributi possono essere di diversi tipi come binario, categoriale o continuo. Ogni istanza di dati potrebbe essere composta da un solo attributo (*univariato*) o più attributi (*multivariato*). Nel caso di multivariato, gli attributi potrebbero essere dello stesso tipo o potrebbero essere una combinazione di diversi tipi di dati.

I dati di input possono anche essere classificati in base alla relazione presente tra i dati stessi. Alcuni esempi sono i dati in sequenza, dati spaziali e dati grafici. Nei dati di sequenza, le istanze di dati sono linearmente ordinati, ad esempio, dati di serie temporali (time-series data), sequenze di genomi e sequenze di proteine. Nei dati spaziali, ogni istanza di dati è correlata alle istanze adiacenti, ad esempio dati sul traffico veicolare e dati ecologici. Quando i dati spaziali hanno un componente temporale (sequenziale), vengono definiti dati spazio-temporali, ad esempio i dati climatici. Nei dati grafici, le istanze di dati sono rappresentate come vertici in un grafico e sono collegate ad altri vertici tramite bordi.

2.1.2 Tipi di anomalie

Le anomalie posso essere classificate nelle tre seguenti categorie:

Anomalie Puntuali. Il tipo di anomalia più semplice si propone allorchè si tratti di stabilire se un solo, singolo dato possa essere considerato anomalo rispetto a tutti gli altri. Su questo problema si concentra la maggior parte delle tecniche di individuazione delle anomalie. Così ad esempio, in Figura 2.2, si noti come l'istanza evidenziata si discosti nettamente rispetto ai rimanenti dati considerati "normali". Dunque, essa è considerata un'anomalia. Considerando il dominio della Fraud Detection un esempio di tale anomalia può essere una transazione avente una spesa molto più alta rispetto al normale range di spesa associato al titolare della carta di credito, vedi Figura 2.1.

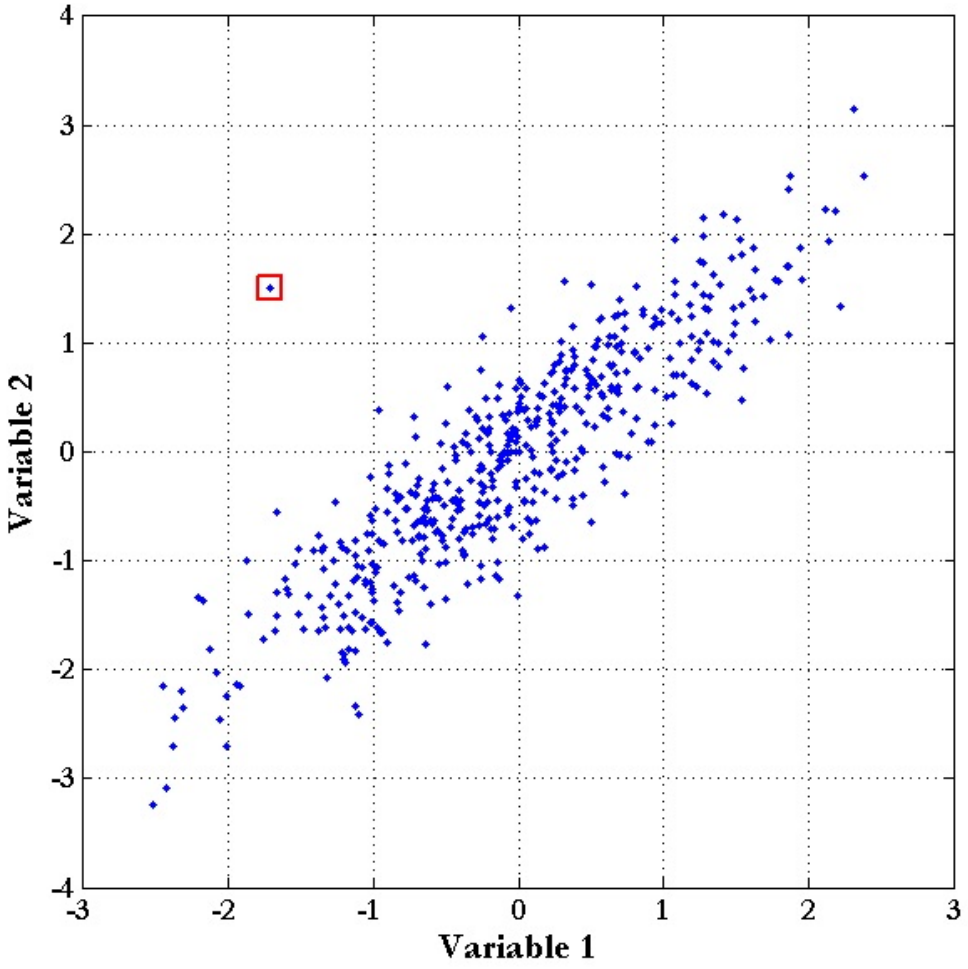


Figura 2.2: Esempio anomalia puntuale in un set a due-dimensioni [1].

Anomalie Contestuali. Come abbiamo detto più sopra, un dato può risultare anomalo in uno specifico contesto, ma del tutto normale in altri. E' evidente che la nozione di contesto è determinata dalla struttura del set di dati e deve essere specificata come parte della formulazione del problema. Ogni istanza di dati viene dunque definita utilizzando due set di attributi:

1. *Attributi di contesto.* Sono usati per determinare il contesto per quel dato. Per esempio, in time-series data, il tempo è un attributo di contesto che determina la posizione di un dato rispetto all'intera sequenza.
2. *Attributi di comportamento.* Definiscono le caratteristiche non contestuali di un dato. Servono per individuare le anomalie all'interno di un contesto. Per esempio, la media delle temperature di una data serie temporale.

In Figura 2.3 è rappresentata una serie temporale di temperature mensili di una data area, nel corso di tre anni. Una temperatura di 35°F può ritenersi normale durante l'inverno (al tempo t_1), ma la stessa temperatura durante l'estate (al tempo t_2) potrebbe essere un anomalia.

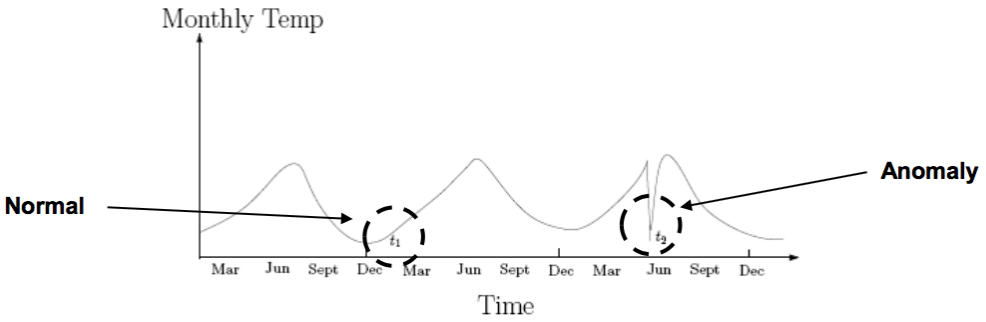


Figura 2.3: Esempio di anomalia contestuale [1].

Anomalie Collettive. Se una raccolta di dati correlati è anomala rispetto l'intero set di dati. Il singolo dato in una anomalia collettiva potrebbe non essere anomalo, ma il verificarsi di esso insieme ad altri correlati costituisce un'anomalia. In Figura 2.4 è riportato il grafico di un elettrocardiogramma. La regione in rosso denota un anomalia poiché si riscontra lo stesso valore per un periodo di tempo troppo lungo. Si noti che il singolo valore, preso da solo, non costituisce un'anomalia.

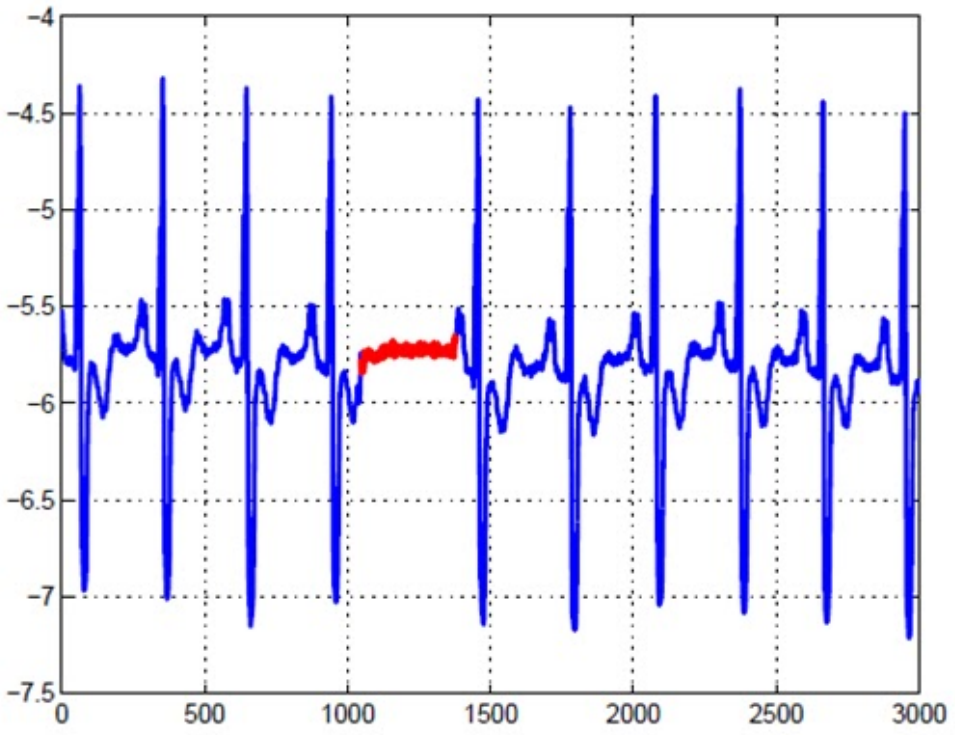


Figura 2.4: Anomalia collettiva corrispondente a una contrazione prematura atriale in un elettrocardiogramma umano [1].

Si noti che mentre le anomalie puntuali possono verificarsi in qualsiasi set di dati, le anomalie collettive possono verificarsi solo in serie in cui i dati sono correlati. Al contrario, la presenza di anomalie contestuali dipende dalla disponibilità di attributi di contesto nei dati. Un'anomalia puntuale o collettiva può anche essere contestuale se analizzata rispetto ad un attributo di contesto. Pertanto un problema di rilevamento di anomalie puntuali o collettive può essere trasformato in un problema di rilevamento di anomalie contestuali incorporando le informazioni di contesto.

2.1.3 Modalità d'azione

Le etichette (labels) associate a un'istanza di dati indicano se quell'istanza è normale o anomala. Si fa presente che ottenere dati etichettati che siano accurati e rappresentativi di tutti i tipi di comportamenti, è spesso proibitivo. L'etichettatura viene spesso eseguita manualmente da un esperto umano e quindi è necessario uno sforzo notevole per ottenere il set di dati di input etichettato. In genere, ottenere un insieme etichettato di istanze di dati anomali che coprano tutti i possibili tipi di comportamento anomalo è anche più difficile rispetto ad ottenere quello relativo ad un comportamento normale. Inoltre, il comportamento anomalo è spesso di natura dinamica, in quanto, ad esempio, potrebbero insorgere nuovi tipi di anomalie per le quali non esistono dati già etichettati. A seconda della disponibilità o meno di dati già classificati come normali o anomali, le tecniche di rilevamento delle anomalie possono operare in una delle tre seguenti modalità:

Supervised Anomaly Detection. Esse assumono di avere a disposizione un *training set,* le cui istanze siano state divise in almeno due distinte classi: normale e anomalia. L'idea alla base di queste tecniche è di costruire un modello di predizione, basato sul training set, per determinare se un futuro elemento possa essere considerato come normale o come anomalia. Si possono avere inoltre classi multiple per le due categorie, in modo da ottenere una suddivisione più accurata.

Semisupervised Anomaly Detection. Esse dispongono invece di un training set contenente solo esempi di dati normali. Il modello di predizione di queste ultime sarà ovviamente meno accurato di quello delle prime, dato che è costruito sulla base di un training set più povero. Tuttavia, le tecniche semisupervisionate sono più applicabili perché non sempre si hanno a disposizione dei campioni di anomalie.

Unsupervised Anomaly Detection. Esse non richiedono la presenza di un traning set e sono quindi, in generale, le più applicabili. Si basano sull'assunzione implicita che le istanze normali siano in numero nettamente superiore rispetto a quello delle anomalie. Se tale ipotesi non risulta essere veritiera, queste tecniche possono soffrire di un numero molto elevato di rivelazioni errate. Queste tecniche sono le più applicabili perché per essere eseguite necessitano solo di un database di dati da analizzare.

In generale le tecniche supervisionate sono più efficienti di quelle non supervisionate, perché la conoscenza posseduta nei training set viene usata per affinare il

processo di ricerca. In molti casi però, i pattern che rappresentano il comportamento il comportamento normale sono in continua evoluzione e una rappresentazione attuale di comportamento normale può essere non valida nel futuro.

2.1.4 Output

Notevole importanza assume anche il modo in cui le anomalie sono segnalate. Tipicamente gli output sono di 2 tipi:

1. *Labels.* Esso è il più semplice e prevede un uscita binaria assegnando un etichetta "normale/anomalia" a ciascun dato in modo da indicare se esso è un outlier o meno.

2. *Scores.* Tali tecniche assegnano un peso che indica quanto un certo dato è un outlier. Più il peso è alto più il dato ha caratteristiche anormali. Il peso può essere calcolato tramite considerazioni sulla sparsità della regione, considerazioni sulle distanze dai vicini o il match con una certa distribuzione di dati.

Le tecniche di rilevamento delle anomalie basate sul punteggio consentono all'analista di utilizzare una soglia specifica per il dominio per selezionare le anomalie più rilevanti. Le tecniche che forniscono etichette binarie alle istanze di test non consentono direttamente agli analisti di fare una tale scelta, sebbene ciò possa essere controllato indirettamente attraverso le scelte dei parametri all'interno di ciascuna tecnica.

2.2 Principali approcci di Anomaly Detection

La comunità di ricerca scientifica ha sviluppato diversi modelli ed approcci di Anomaly Detection. Ogni approccio indica di solito una base di riferimento cui sottostanno i dati "normali". Da tale assunzione si diramano le diverse tecniche appartenenti al medesimo approccio. E' evidente che la preliminare scelta dell'approccio e, subito dopo, della tecnica da utilizzare è fondamentale, atteso che approcci o tecniche diverse e non pertinenti possono condurre a risultati diversi, con ogni possibile conseguenza.

2.2.1 Approcci basati sulla classificazione

Le tecniche che si fondano sulla classificazione sono utilizzate per definire ed apprendere un modello (*classifier*) da un insieme di istanze di dati etichettati (*training*);

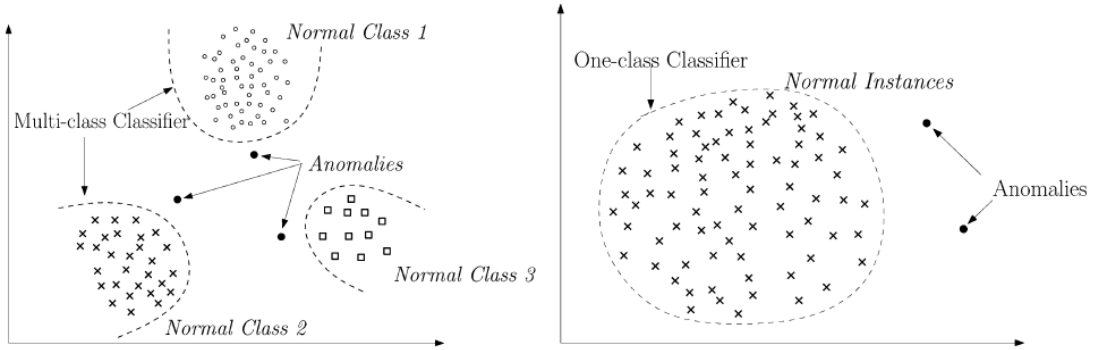
dette tecniche, quindi, classificano un'istanza di test in una delle classi utilizzando il modello appreso (*testing*). Si tratta di tecniche la cui operatività si articola in due fasi: quella del *training* che apprende un *classifier* utilizzando i dati disponibili di allenamento etichettati (cioè già classificati in normali o anomali) e la fase di *testing* che classifica una nuova istanza di test come normale o anomala, utilizzando il *classifier* . Tutte le tecniche di rilevamento delle anomalie basate sulla classificazione operano rispettando la seguente assunzione generale.

Assunzione: un *classifier* che può distinguere tra classi normali e anomale può essere appreso nello spazio delle funzioni specificato.

Sulla base dei dati già classificati disponibili per la fase di *training*, le tecniche di rilevamento delle anomalie basate sulla classificazione possono essere raggruppate in due ampie categorie:

Multi-class. Esse partono dal presupposto che i dati di training contengano in se istanze etichettate appartenenti a più classi normali [3],[4].Si tratta di tecniche che inducono il *classifier* a distinguere tra le varie classi normali e le anomalie. Una istanza di test è considerata anomala, vedi Figura 2.5(a), se non è classificata come normale in nessuna delle classi formate dal *classifier*. Alcune tecniche in questa sottocategoria associano un punteggio di confidenza con la previsione fatta dal *classifier*.

One-class. Esse presuppongono che tutte le istanze normali, nella fase di training, appartengano ad una classe soltanto, delineano un confine attorno alle istanze normali e dichiarano come anomala qualunque istanza di test che ricade al di fuori. Vedi Figura 2.5(b).



(a) Multi-class Anomaly Detection (b) One-class Anomaly Detection

Figura 2.5: Anomaly detection basate sulla classificazione [1].

Esiste una varietà diversificata di tecniche di rilevamento delle anomalie ; si tratta di tecniche che utilizzano diversi algoritmi di classificazione per costruire *classifiers* e ne sono concreti esempi reti neurali, classificatori bayesiani, support vector machines (SVMs). Si tratta di approcci che hanno il vantaggio basarsi su algoritmi potenti che consentono la distinzione tra le diverse classi e che sono caratterizzati da una fase di *testing* molto rapida. Purtroppo però spesso non è possibile ottenere un training set tanto ampio da poter rappresentare in modo attendibile le due distinte classi ed in modo particolare quella delle anomalie, il che rappresenta un grosso limite.

2.2.2 Approcci statistici

Gli approcci di tipo statistico si basano sulla seguente assunzione.

Assunzione: le istanze di dati normali sono generate da una specifica distribuzione e dunque la anomalie saranno quindi i punti che hanno scarsa probabilità di essere stati generati secondo tale modello di distribuzione.

La loro efficacia dipende naturalmente da quanto i dati seguono il modello che descrive la loro distribuzione.

Si hanno molti approcci che differiscono in base al tipo ed al numero di distribuzioni assunte, al numero di variabili (*univariate/multivariate*) ed al tipo di tecniche parametriche o non parametriche di volta in volta usate. Le tecniche parametriche partono dalla conoscenza della distribuzione sottostante nonchè da quella dei parametri corrispondenti, mentre quelle non parametriche non si riferiscono nè alla distribuzione nè ai relativi parametri. Come esempio di tecnica parametrica possiamo citare il test di Grubb. Essa prende in considerazione la distribuzione normale e fa determinare possibili outlier in un dataset univariato. Invece un esempio di tecnica non parametrica è quella basata sulla costruzione e lo studio di istogrammi. Nel caso di dati univariati viene costruito un istogramma basandosi sui differenti valori di una *feature*; ad ogni istanza viene assegnato un punteggio inversamente proporzionale all'altezza (cioè la frequenza) del *bin* a cui essa appartiene. Le istanze con un punteggio alto vengono considerate come outlier. Gli approcci di tipo statistico hanno il vantaggio di basarsi su modelli statistici e forniscono perciò soluzioni giustificabili da un punto di vista statistico. Non va dimenticato neanche che, di solito, *l'anomaly score* fornito è associato ad un intervallo di confidenza che può rappresentare o fornire un'informazione importante. E' però da tener presente lo svantaggio principale derivante e cioè che in molti casi, anche se non tutti, i dati possano esser stati generati da una distribuzione del tutto particolare. Infine è da sottolineare che spesso non è facile individuare la giusta tecnica statistica da utilizzare, in particolare nel caso di dataset di alta dimensionalità.

2.2.3 Approcci basati su nearest-neighbor

Gli approcci nearest-neighbor based utilizzano il concetto di neighborhood (vicinato) di un punto per determinare le anomalie, analizzando di solito i punti più vicini (K-nearest-neighbors K-NN) ad ogni punto, senza fare alcuna assunzione sulla loro distribuzione. In Figura 2.6 è mostrata la differenza tra un punto ritenuto normale e uno anomalo.

Assunzione I dati ordinari sono caratterizzati dalla presenza di neighborhood densi, mentre gli outlier sono relativamente più distanti dai propri vicini e dunque presentano neighborhoods meno densi.

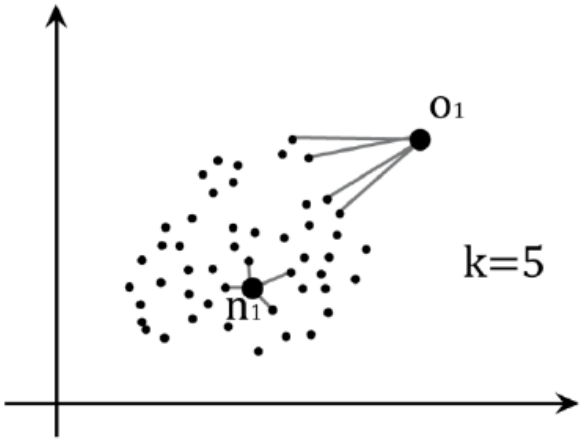


Figura 2.6: K-NN di un punto normale e di un punto anomalo [1].

Le tecniche ricadenti in questa categoria possono essere a loro volta divise in due gruppi:

Distance-based. Nei metodi distance-based si sono succedute ed alternate diverse definizioni di outlier. Knorr e Ng [5] hanno definito gli outlier come i punti per cui si hanno una minor quantità di punti nel dataset, all'interno di un raggio δ . La definizione però non consente di realizzare un attendibile ranking degli outlier ed impone comunque un corretto dimensionamento del valore δ, per poter ottenere risultati utili ed attentibili. Determinata la distanza δ di un punto dai suoi primi k vicini, la scelta del valore k sarà evidentemente determinante per una corretta soluzione. Se il k è troppo piccolo, piccoli gruppi di punti vicini tra loro ma lontani dagli altri dati possono non essere etichettati come outlier. In Figura 2.7, i tre punti raggruppati in basso a destra sono certamente degli outlier, ma se viene usato un k troppo piccolo,

ad esempio k=2, si presenteranno come appartenenti punti normali, proprio perché tra di essi molto vicini.

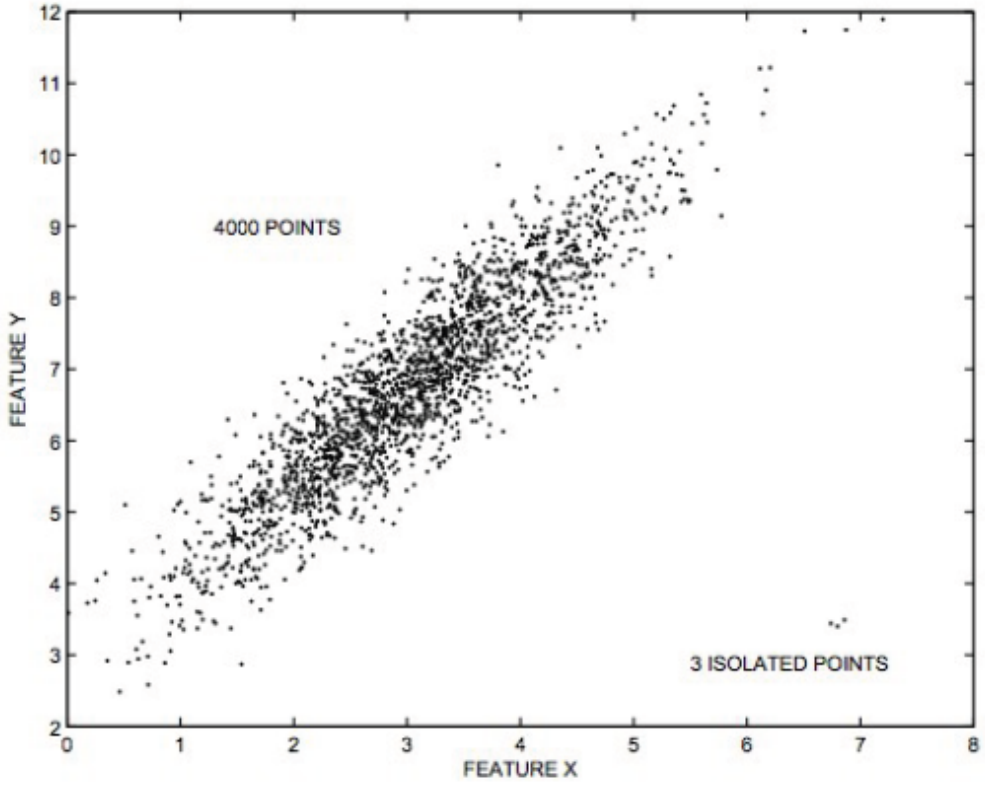


Figura 2.7: Esempio applicazione metodo distance-based [1].

Density-based. I metodi density-based osservano e valutano la densità del neighborhood di ogni punto del dataset e dichiarano come outlier i punti che risiedono in neighborhood a bassa densità. Nascono dall'osservazione che le tecniche distance-based tendono a presentare problemi nel caso siano presenti più zone di dati a differenti densità.

I metodi nearest-neighbor based, oltre a non essere supervisionati e a non dipendere da assunzioni relative alla distribuzione statistica dei dati, hanno il vantaggio di adattarsi a diverse tipologie di dati, richiedendo semplicemente la definizione di una misura di distanza tra le istanze. Inoltre tipicamente non è richiesto che tale funzione di distanza sia una metrica in senso stretto; nella maggior parte dei metodi si assume solamente che sia definita positiva e simmetrica, ma è non obbligatorio il soddisfacimento della disuguaglianza triangolare. Peraltro in alcuni contesti, con

dati particolarmente complessi,la definizione di misure di distanza può essere molto complicata (come, ad es, per i grafici). C'è infine da rilevare che i metodi nearest-neighbor based presentano problemi nel caso in cui i dati normali non abbiano un numero sufficiente di vicini rispetto alle anomalie, o che gli outlier presentino comunque un certo numero di vicini. Nel caso di dataset ad alta dimensionalità, inoltre, il noto problema della "maledizione della dimensionalità" potrà determinare taluni inconvenienti, nel senso che le differenze tra le distanze delle varie coppie punti si assottiglieranno, i dati diventeranno più sparsi, il concetto di neighborhood diventerà poco significativo e quasi tutti i punti potranno considerarsi outlier.

2.2.4 Approcci basati sul clustering

Gli approcci basati sul clustering partizionano i dati in cluster di dimensioni e densità variabile.

Assunzione: i dati normali appartengono a cluster di grandi dimensione e alta densità, mentre gli outlier appartengono a cluster piccoli di bassa densità, o anche a nessun cluster.

Le anomalie saranno quei punti presenti nei cluster di dimensione o densità inferiore un valore di soglia prestabilito. Tipicamente in un approccio di questo tipo, il primo passo è quello di usare un algoritmo di clustering per determinare le regioni più dense. Successivamente si etichettano come candidati i punti appartenenti a cluster piccoli e si calcola la distanza tra questi e i cluster non candidati: se i punti candidati sono lontani da tutti i punti non candidati, allora sono dei veri outlier. Nella Figura 2.8 i punti appartenenti al cluster O_3 verranno identificati come outlier, e lo stesso vale per i punti o_1 e o_2. I cluster N_1 e N_2 vengono invece identificati come classi normali.

Diverse tecniche basate sul clustering richiedono il calcolo della distanza tra una coppia di istanze. Pertanto, a tale riguardo, sono simili alle tecniche nearest-neighbor based. La differenza chiave tra le due tecniche, tuttavia, è che le prime valutano ogni istanza rispetto al cluster a cui appartiene, mentre le seconde analizzano ogni istanza rispetto al suo neighbor locale. Gli approcci clustering-based hanno il vantaggio di poter essere applicati anche a dataset molto complessi, scegliendo degli algoritmi di clustering che supportino tali dati. Inoltre la fase di testing, successiva alla generazione dei cluster su un training set, è tipicamente molto veloce, in quanto il numero dei cluster è inferiore al numero delle singole istanze. Soffrono però del problema di essere strettamente legati alle performance degli algoritmi di clustering e di essere difficilmente ottimizzati per l'outlier detection.

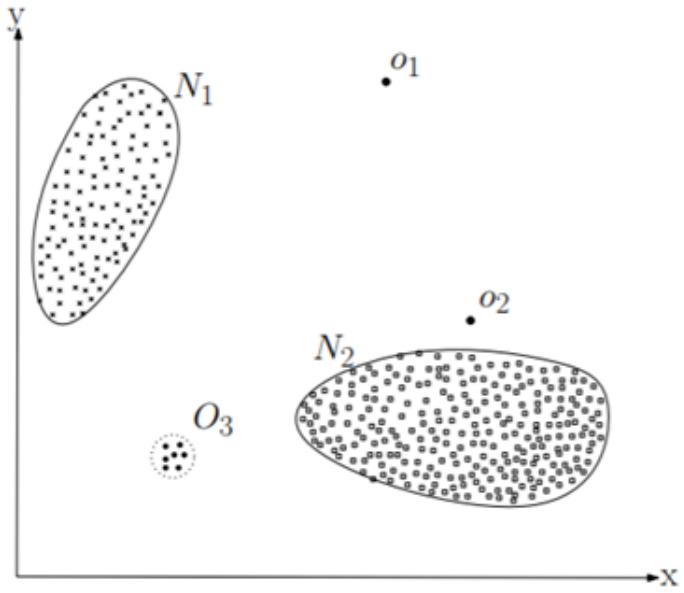


Figura 2.8: Esempio approccio clustering-based [1].

2.2.5 Ulteriori approcci utilizzati

Si riportano infine approcci meno utilizzati ma comunque significativi nel campo dell'anomaly detection.

Approcci basati sulla teoria dell'informazione. Analizzano il contenuto informativo del dataset, utilizzando delle misure come l'entropia e l'entropia relativa. Tali metodi assumono che le anomalie introducano delle irregolarità nel contenuto informativo del dataset.

Approcci basati sull'analisi spettrale. Sono specifici per dataset ad alta dimensionalità ed eseguono una riduzione della dimensionalità, cercando un'approssimazione dei dati tramite una combinazione di attributi, che meglio catturano la variabilità delle istanze. L'idea è quella di determinare dei sottospazi in cui le anomalie siano più facili da identificare.

Approcci grafici. Basati sull'osservazione degli oggetti come punti mappati in uno spazio multidimensionale, un tipo di analisi svolta dall'operatore umano, e per questo motivo è soggettiva. Inoltre, la complessità di questo tipo di analisi cresce notevolmente all'aumentare della dimensionalità dello spazio da osservare.

2.3 Metodo CUSUM

Il controllo statistico del processo (Statistical process control - SPC) é un metodo di controllo della qualità, basato su un approccio di tipo statistico, per monitorare e controllare un processo. Ciò contribuisce a migliorare l'efficienza del processo, producendo più prodotti conformi alle specifiche. L'SPC può essere applicato a qualsiasi processo in cui sia possibile misurare l'output di "prodotto conforme". Gli strumenti chiave utilizzati nell'SPC comprendono grafici di esecuzione, tabelle di controllo, attenzione al miglioramento continuo e progettazione di esperimenti.

Nel controllo statistico della qualità, assume particolare importanza il CUSUM (o cumulative sum control chart). Si tratta di una tecnica di analisi sequenziale sviluppata da E. S. Page dell'Università di Cambridge che viene in genere utilizzata per monitorare e rilevare i cambiamenti delle serie su cui è svolta l'analisi [6]. La CUSUM Anomaly Detection (CAD) è stata annunciata in Biometrika nel 1954.

Il rilevamento del cambiamento si riferisce alle procedure per identificare i cambiamenti improvvisi in un fenomeno[7]. Per cambiamento improvviso si intende qualsiasi cambiamento, più veloce di quanto previsto, di alcune caratteristiche dei dati come ampiezza, media, varianza e frequenza, rispetto ai precedenti dati noti.

Come suggerisce il nome, l'algoritmo CUSUM prevede il calcolo della somma cumulativa (che è ciò che lo rende "sequenziale") delle variazioni positive e negative ($g^+[t]$, $g^-[t]$) nei dati (x) e il confronto con un valore di soglia (o threshold, anche indicato con h). Quando questa soglia viene superata da $g[t]$, viene rilevata un'anomalia al tempo t_a (oppure t_{alarm}). Per evitare il rilevamento di un'anomalia senza che essa sia effettiva, viene aggiunto all'algoritmo un parametro di deriva (o drift, anche indicato con ν). Il parametro di deriva ν riduce l'effetto dei dati passati riducendo il numero di falsi allarmi rilevati. In generale la scelta di h e ν determina le prestazioni dell'algoritmo. Il funzionamento dell'algoritmo è mostrato in Figura 2.9, dove H(t) è la variabile su cui si effettua l'analisi, t_s rappresenta il tempo in cui è iniziata l'anomalia, t_f rappresenta il tempo in cui si conclude l'anomalia. t_s e t_f corrispondono ai tempi in cui la pendenza di dG/dt diventa rispettivamente positiva e negativa o al massimo uguale a zero.

La scelta dei parametri CUSUM, come mostrato nelle Figure 2.10, 2.11 e 2.12, influenzerà le prestazioni dell'algoritmo. Infatti, la soglia h impone il limite inferiore per la dimensione dell'anomalia che verrà rilevata e il valore del parametro di deriva ν può influenzare il tempo di rilevamento. Sebbene la diminuzione di ν e h espanda il range di anomalie rilevabili, sia la deriva che la soglia devono essere sufficientemente grandi da evitare situazioni di falso allarme. Pertanto, per prestazioni ottimali, i parametri devono essere sintonizzati in modo specifico per una particolare rete.

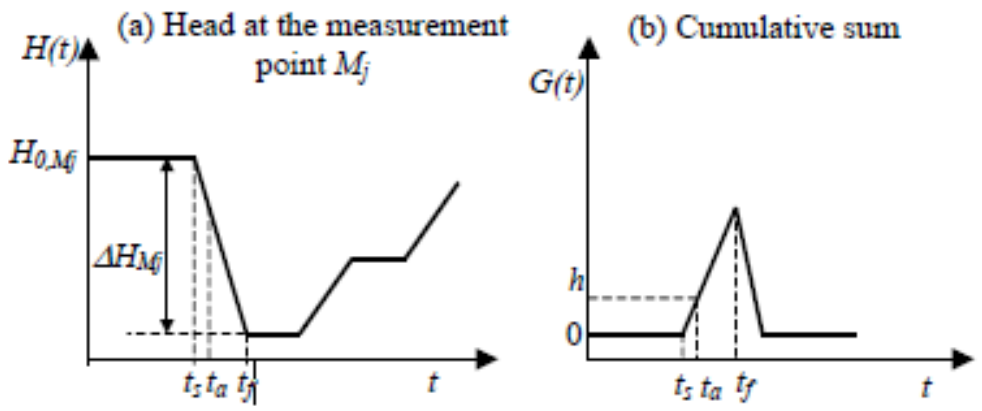


Figura 2.9: Principio di funzionamento dell'algoritmo CUSUM [7].

Per applicare tale algoritmo in Python si è fatto riferimento allo script [8], opportunamente modificato, riportato in appendice A.1. Matematicamente, il test CUSUM è formulato come mostrato nelle equazioni 2.1 e 2.2.

$$\begin{cases} s[t] = x[t] - x[t-1] \\ g^+[t] = \max(g^+[t-1] + s[t] - drift, 0) \\ g^-[t] = \max(g^-[t-1] - s[t] - drift, 0) \end{cases} \tag{2.1}$$

$$\text{Se} \quad g^+[t] > h \quad \text{o se} \quad g^-[t] > h\text{:}$$

$$\begin{cases} t_{alarm} = t \\ g^+[t] = 0 \\ g^-[t] = 0 \end{cases} \tag{2.2}$$

Si noti che a differenza dalla Figura 2.9(b), nell'algoritmo utilizzato, il valore di soglia h non viene mai superato, in quanto, appena raggiunto tale valore, le due funzioni cumulate vengono immediatamente poste uguali a zero in modo da poter individuare la successiva anomalia.

Per poter comprendere appieno l'algoritmo utilizzato si riportano alcuni esempi d'uso: si pone di voler analizzare attraverso l'algoritmo CUSUM la funzione 2.3

$$x = \sin k\pi \qquad \text{con } 0.5 \leq k \leq 6.5 \quad \text{con step} = 0.01 \tag{2.3}$$

A seconda dei valori attribuiti ai parametri si hanno diversi output. In Figura 2.10, avendo scelto un valore di h troppo piccolo, l'algoritmo individua alcune

anomalie anche se i dati anomali risultano avere un comportamento "normale". In Figura 2.11 si è attribuito un valore di ν troppo alto, ed essendo lo scarto tra ogni dato s[t] minore di ν le due sommatorie cumulate non vengono mai incrementate ma rimangono sempre uguali a zero. In Figura 2.12 si è posto un valore di h troppo alto e di conseguenza non viene riscontrata alcuna anomalia.

Si precisa che nei grafici ottenuti dall'applicazione dell'algoritmo CUSUM, oltre al dato anomalo evidenziato in rosso, saranno riportate anche due frecce verdi le cui ascisse sono indicative dei tempi t_s e t_f. Tali tempi sono utili per stabilire la durata e l'esatta collocazione temporale di un evento anomalo.

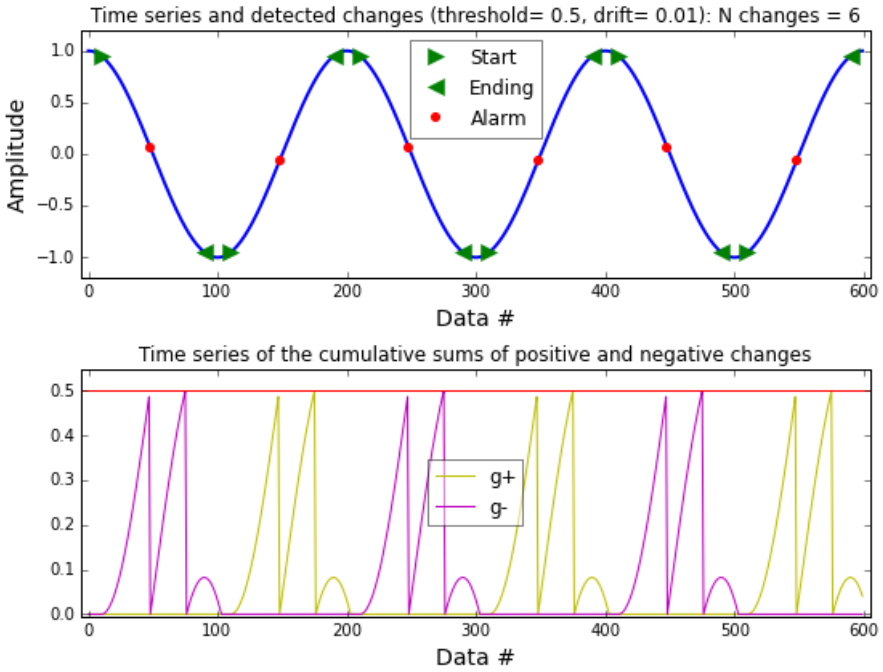


Figura 2.10: Esempio CUSUM test con h troppo piccolo.

Risulta dunque evidente, per ottenere prestazioni efficienti, l'opportuna calibrazione dei parametri h e ν per la specifica serie analizzata. Secondo Gustafsson [7], questa sintonizzazione può essere eseguita ripetendo i passaggi di seguito riportati:

1. Si pone un valore di h molto ampio.
2. Si pone ν in modo tale che le $g[t]$ siano uguali a zero per più del 50% delle volte.
3. Si reimposta il valore di h in modo da ottenere il numero richiesto di falsi allarmi.

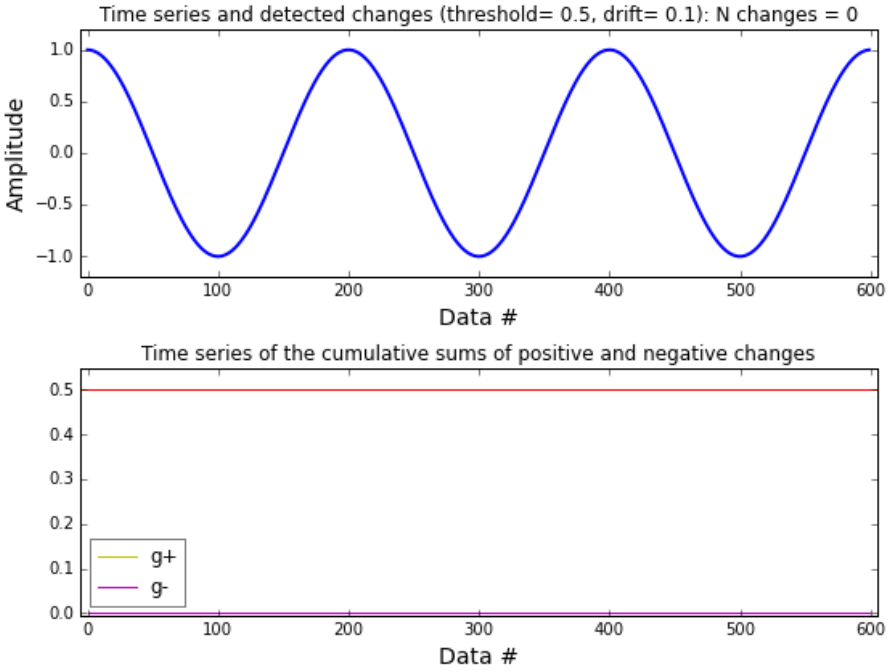


Figura 2.11: Esempio CUSUM test con ν troppo grande.

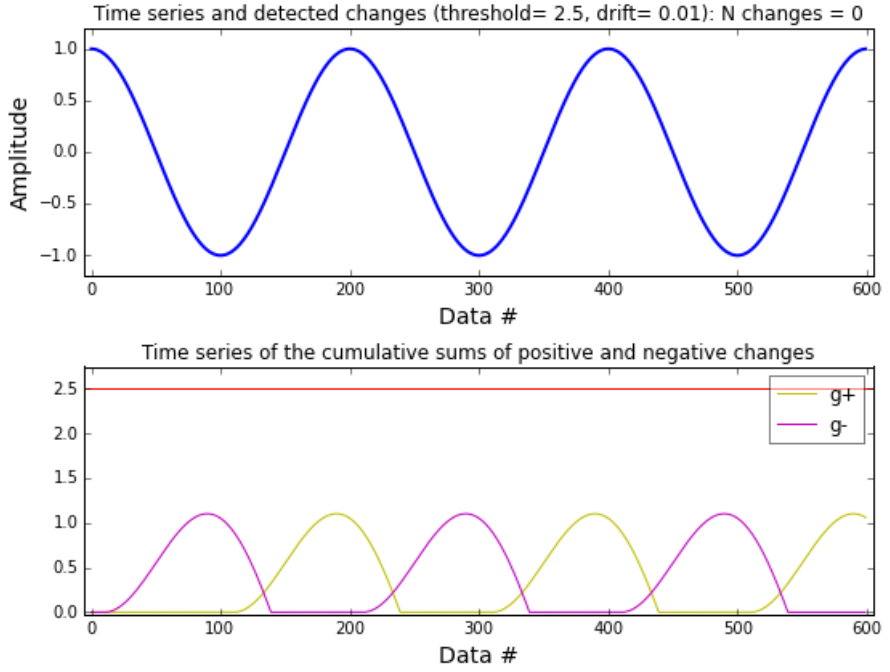


Figura 2.12: Esempio CUSUM test con h troppo grande.

4. Se viene richiesto un numero inferiore di falsi allarmi si provi ad aumentare ν.

Un'alternativa, sempre proposta da Gustafsson [7], è quella di porre $\nu = \sigma$ e $h = 5\sigma$, dove σ è la deviazione standard della serie sottoposta ad analisi. Applicando tale metodologia alla serie costruita dall'equazione 2.3 si ottiene $\nu = 0.7071$ e $h = 3.5355$, essendo il valore di ν relativamente alto, tale caso ricade in quello mostrato in Figura 2.11. Come esposto nel paragrafo 4.3, si è ottenuto un metodo migliore, per la corretta individuazione del valore di tali parametri, lavorando sui percentili delle serie analizzate.

2.3.1 Implementazione CAD

Esistono diverse implementazioni della CUSUM Anomali Detection, tra cui le più importanti sono la normalizzazione e l'eliminazione del rumore dei dati.

Eliminazione del rumore. Il test CUSUM è stato ampiamente applicato per il rilevamento del cambiamento in diversi problemi di analisi delle serie temporali [9]. Se i dati di misurazione sono sporcati a causa del rumore, viene applicato il filtro dei minimi quadrati ricorsivi (Recursive Least Squares - RLS). Il filtro stima il segnale θ_t dalla variabile misurata H_t (contenente il rumore) come mostrato nell'equazione 2.4:

$$\theta_t = \lambda\theta_{t-1} + (1-\lambda)H_t \tag{2.4}$$

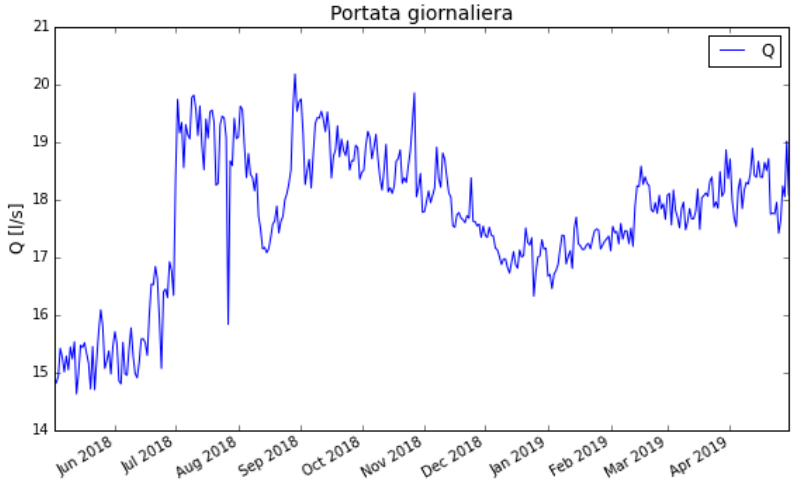
dove il parametro $\lambda \in [0,1)$ è il fattore che limita l'effetto levigante del filtro. A seconda del livello di rumore nei dati misurati, il fattore di correzione viene regolato esponenzialmente in tempo reale tra i valori minimo e massimo selezionati. In questo caso lo scarto tra i dati si calcola come $s[t] = \theta_t - \theta_{t-1}$

Normalizzazione dei dati. Alcuni database, specialmente quelli di grandi dimensioni, presentano dati che subiscono variazioni naturali nel corso del tempo. Un esempio può essere l'andamento medio giornaliero delle portate in uscita da una centrale idrica nel corso dell'anno 2.13(a). I valori di portata riscontrati nei mesi estivi a causa delle alte temperature saranno molto maggiori rispetto a quelli rilevati nei mesi invernali, ma, nonostante ciò, essi saranno comunque definiti normali. Per tener conto di tale pattern si procede con la normalizzazione dei dati attraverso l'equazione 2.5:

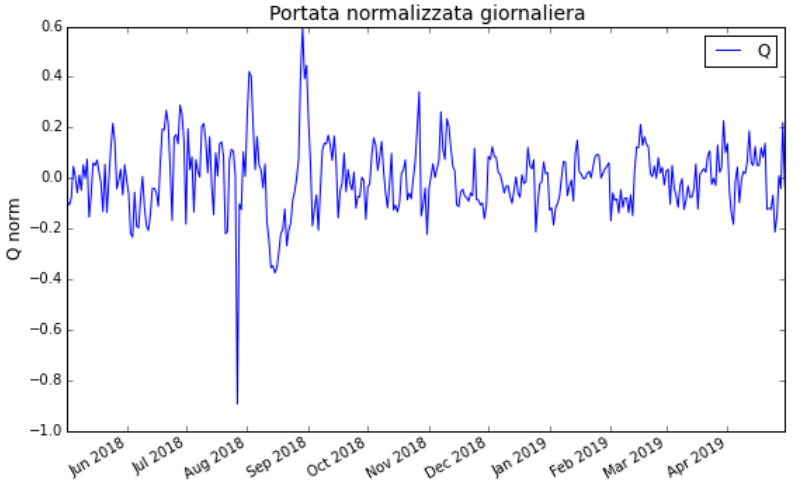
$$z_i = \frac{x_i - \overline{x}_i}{\sigma_i} \tag{2.5}$$

Dove z_i è il valore della variabile normalizzata, x_i è la variabile misurata al tempo $i-esimo$, $\overline{x}_i$ e σ_i sono la media e la deviazione standard nel rispettivo periodo temporale.

Come evidente dal confronto in Figura 2.13, grazie alla normalizzazione (in questo caso eseguita rispetto ai valori mensili) i dati non variano più in funzione del periodo dell'anno e dunque utilizzando l'algoritmo CUSUM si avrà una corretta rilevazione delle anomalie.



(a) *Andamento standard.*



(b) *Andamento normalizzato.*

Figura 2.13: Andamento medio delle portate giornaliere nel corso dell'anno.

Capitolo 3

Individuazione di anomalie in acquedotto

Le aziende che si occupano della gestione e distribuzione delle acque stanno gradualmente trasformando le loro operazioni da locali e manuali ad operazioni centralizzate e non presidiate [19]. Gli operatori che controllano fisicamente e continuamente un singolo luogo sono sostituiti da supervisori che sorvegliano un numero di località in una regione solo durante l'orario di ufficio. Questo implica che la distanza tra l'operatore umano o il supervisore e i processi di produzione e distribuzione dell'acqua sta gradualmente aumentando. Tale crescente distanza comporta un rischio non di poco conto, soprattutto per quanto riguarda i guasti nel sistema, che potrebbero rimanere inosservati nei momenti in cui nessun supervisore umano monitora i processi. Negli acquedotti le apparecchiature (pompe, valvole, sfioratori ecc.), tramite il sistema informatico SCADA (Supervisory Control And Data Acquisition), inviano, in caso di guasto, all'operatore designato la segnalazione dello specifico problema. Ciò nonostante l'operatore non è avvisato in caso di variazione dei valori standard di funzionamento. Tali valori sono spesso correlati a perdite idriche o malfunzionamenti e pertanto molte delle rotture che si verificano lungo le tubazioni, da cui scaturiscono delle perdite, rimangono inosservate nella rete e i gestori idrici entrano in azione solo dopo i reclami da parte dei clienti di una bassa pressione o di flussi d'acqua che si riversano sulle strade.

Nasce dunque l'esigenza di un sistema che, funzionando in real-time, possa, in maniera automatizzata, segnalare la presenza di un evento anomalo. Per tale scopo, durante l'elaborazione della presente tesi, è stata creata, mediante il software Python, la funzione *Anna*. Tale funzione, che può essere anche utilizzata in tempo-reale, richiede come input una serie temporale della variabile che si vuole analizzare

e restituisce come output i rispettivi valori anomali.

Oltre all' individuazione di nuove perdite idriche, la funzione generata può essere anche una comoda seconda linea di difesa in caso di guasto dei normali protocolli di sicurezza della rete. Infatti, se da una parte i sistemi SCADA offrono il vantaggio di essere comodamente monitorati e controllati da remoto, dall'altro questa maggiore praticità comporta un aumento della vulnerabilità, in quanto persone ostili, attraverso cyber-attacks, potrebbero compromettere la rete aziendale e addirittura prendere il controllo del sistema SCADA.

Dunque, il rilevamento delle anomalie fornisce un secondo livello di sicurezza in caso di fallimento delle normali politiche di sicurezza della rete. Questa difesa secondaria è fondamentale in queste applicazioni perché limiterebbe il danno risultante da un sistema compromesso. Un esempio, può essere visto dall'attacco al sistema di trattamento delle acque di Maroochy. Nel marzo 2000, il sistema di trattamento delle acque di Maroochy nel Queensland in Australia è stato compromesso da un aggressore scontento perché gli era stato negato un lavoro presso la struttura [17]. L'aggressore, da remoto, ha fatto cessare il controllo di 150 stazioni di pompaggio e nei tre mesi necessari per rilevare la violazione è stato in grado di liberare 150 milioni di litri di acque reflue nella rete di distribuzione locale. Il sistema aveva meccanismi di sicurezza in atto ma l'aggressore era coinvolto nell'installazione degli aggiornamenti del sistema ed ha usato queste conoscenze per aggirarli. Il sistema ha mostrato un comportamento inspiegabile durante questo periodo che però è stato notato dagli ingegneri solo dopo tre mesi. Successivamente il fautore dell'attacco è stato rintracciato ed arrestato, ma, a questo punto, erano già stati causati ingenti danni. Questo esempio dimostra le potenzialità intrinseche ad un sistema di anomaly detection. Infatti, grazie ad esso, visto il comportamento anomalo del sistema, ci sarebbe stata una segnalazione automatica tempestiva, evitando la gran parte dei danni causati dal cyber-attack [18].

3.1 Perdite idriche

Si tratta di una delle più importanti criticità da affrontare nell'obiettivo di una gestione degli acquedotti improntato a correttezza economica e sostenibilità ambientale. Le perdite che si verificano lungo le tubazioni determinano la differenza fra la quantità d'acqua immessa nella rete e l'acqua oggetto di fatturazione. Le perdite idriche comportano problemi ambientali (spreco della risorsa), finanziari (acqua non fatturata e quindi danni economici), energetici (aumento dei consumi di energia per le attività di pompaggio), infrastrutturali (apertura di cantieri temporanei per il risanamento) e latamente sociali (possibili danni a cose e persone).

Di tutta l'acqua potabile immessa nei 500 mila km di rete di distribuzione italiana, il 41,4% non viene recapitato alle utenze. In un solo anno si sprecano 3,45 miliardi di metri cubi d'acqua. In Figura 3.1 sono riportate in percentuali le perdite idriche per provincia. Stimando un consumo annuo medio per abitante di $80m^3$, tale volume è in grado di soddisfare le esigenze idriche per un anno di circa 40 milioni di persone. Le 2,6 milioni di famiglie che lamentano abituali irregolarità nell'erogazione idrica sono i primi a pagarne le conseguenze. Si parla di un danno economico di circa 4 miliardi di euro, in quanto l'acqua immessa in rete ha già subito un costoso processo di depurazione.(*Dati 2015 ISTAT*)

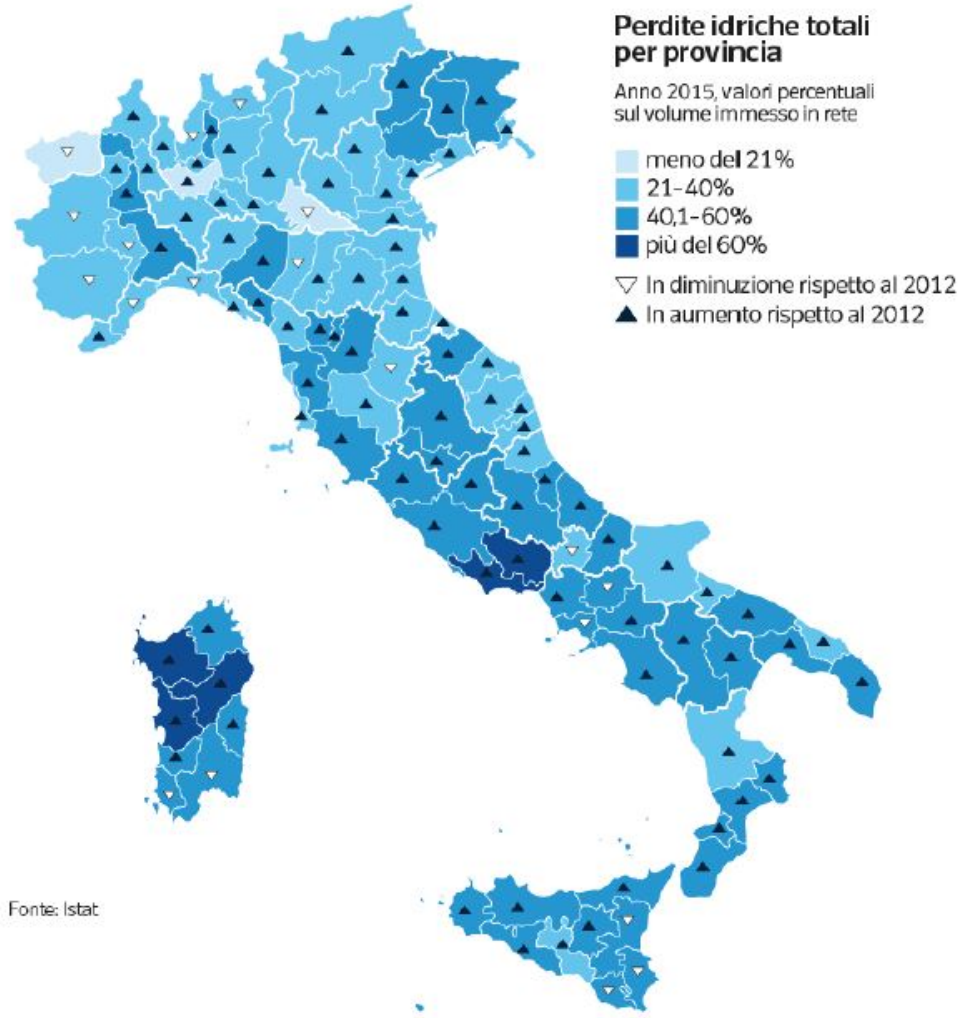


Figura 3.1: Perdite idriche per provincia [10].

Questo gigantesco spreco è dovuto alle pessime condizioni delle tubature, specialmente nella rete di distribuzione, che porta il servizio dalle condotte adduttrici alle utenze. Sono gli acciacchi dell'età: fra il 60/70% della rete idrica ha più di 30 anni, il 25% supera i 50. Per questa ragione sempre più spesso qualche tubo si rompe, provocando improvvisi allagamenti e di conseguenza le strade cittadine vengono chiuse al traffico. Ma mettere mano agli acquedotti italiani costa tempo e denaro. Secondo la Federazione che riunisce le Aziende che operano nei servizi pubblici dell'acqua ci vogliono 3 miliardi per le opere di manutenzione.[10] Attualmente il rinnovo della rete idrica procede a un ritmo di 3,8 km l'anno. Di questo passo *Utilitalia* stima che ci vorranno 250 anni prima di aver ristrutturato le migliaia di km di tubi, a quel punto, le "nuove" condotte sarà già usurate.

Le perdite nel sistema di approvvigionamento idrico urbano possono essere suddivise in due parti principali: perdite apparenti e perdite reali. Le *perdite apparenti* (o amministrative) sono dovute a volumi sottratti senza autorizzazione (allacciamenti abusivi) e a volumi consegnati, ma non misurati, a causa dell'imprecisione o del malfunzionamento dei contatori. Le *perdite reali* (o fisiche), oltre al volume sfiorato dai serbatoi, comprendono l'acqua che fuoriesce dal sistema distributivo disperdendosi nel sottosuolo [11]. Esse sono ottenute come differenza tra le perdite totali e quelle apparenti. Una classificazione più dettagliata è fornita dall'International Water Association (IWA) ed è riportata in Figura 3.2

<table>
<tr><td rowspan="8">Volume immesso nel sistema</td><td rowspan="3">Consumo autorizzato</td><td rowspan="2">Consumo autorizzato fatturato</td><td>Consumo fatturato e misurato, incluse le acque esportate
(consumo fatturato basato sulla misura)</td><td rowspan="2">Acqua fatturata</td></tr>
<tr><td>Consumo fatturato non misurato
(consumo fatturato non basato sulla misura, es.idranti e bocche anti-incendio)</td></tr>
<tr><td>Consumo autorizzato non oggetto di fatturazione
(volumi tecnici di servizio)</td><td>Consumo non fatturato misurato</td><td rowspan="5">Acqua non fatturata</td></tr>
<tr><td></td><td>Consumo non fatturato non misurato</td></tr>
<tr><td rowspan="5">Perdite idriche</td><td rowspan="2">Perdite apparenti
(o "perdite commerciali", volumi consumati ma non fatturati)</td><td>Consumo non autorizzato
(allacci abusivi, furti)</td></tr>
<tr><td>Imprecisione misuratori utenze
(errore misura contatori: normalmente sottoconteggio)</td></tr>
<tr><td rowspan="3">Perdite reali
(o "perdite fisiche")</td><td>Perdite dalle condotte di trasporto e distribuzione
(grandi condotte di adduzione e reti urbane)</td></tr>
<tr><td>Perdite e sfiori dai serbatoi</td></tr>
<tr><td>Perdite dagli allacciamenti</td></tr>
</table>

Figura 3.2: Classificazione perdite proposta dall'IWA [27].

La grande parte delle perdite reali è causata da fenomeni di corrosione, deterioramento o rotture nelle tubazioni e giunti difettosi. In Figura 3.3 sono mostrate

le diverse tipologie di perdite e la rispettiva relazione tra tempo di riparazione t e portata defluita Q [12],[13].

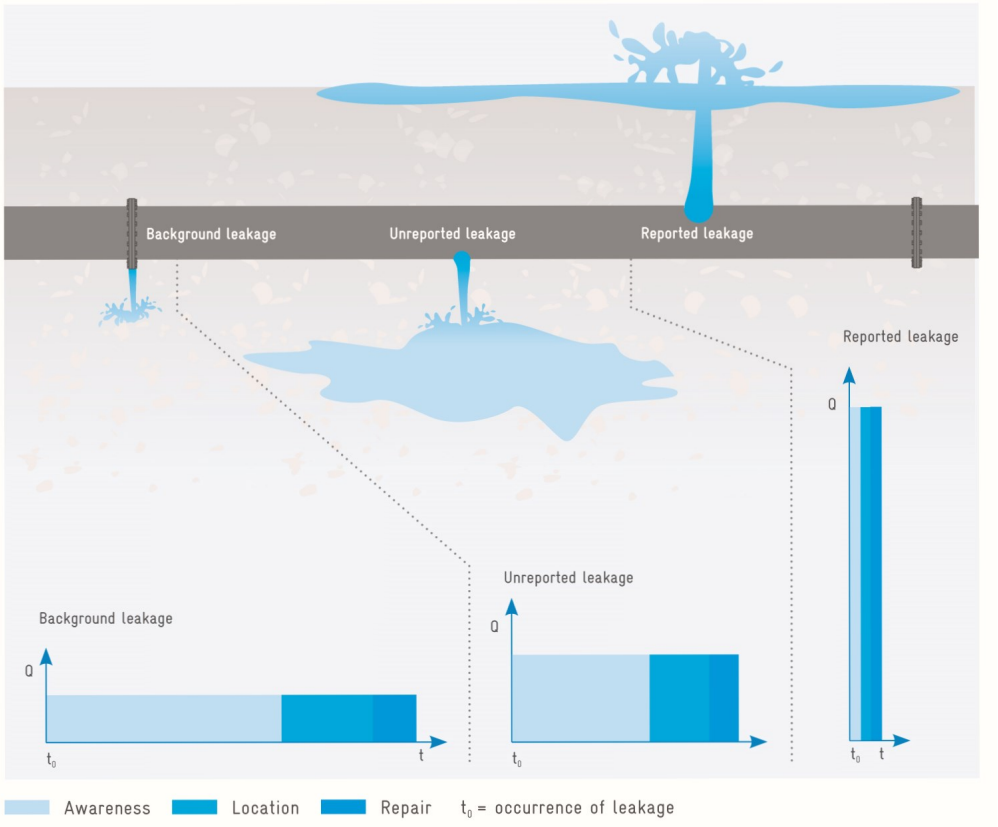


Figura 3.3: Tipologie di perdite [28].

Queste rotture sono relativamente frequenti nei sistemi di distribuzione dell'acqua. Dato che molti sistemi di approvvigionamento idrico sono vecchi e in cattive condizioni, è praticamente impossibile prevenire i guasti dei tubi ma le perdite possono essere ridotte riducendo al minimo il tempo di rilevamento e localizzazione. Sebbene la maggior parte delle rotture provochi di norma la comparsa d'acqua sulla superficie del terreno, acqua che viene segnalata dai clienti o dal personale della compagnia idrica (rilevamento passivo), il tempo di localizzazione medio può essere piuttosto lungo. Morrison [14] stima il tempo di consapevolezza e localizzazione di una perdita di $4 m^3/h$ in 5 giorni. Secondo Obradovic [15], i tempi di localizzazione sono di circa 18 ore. Questo ritardo nella localizzazione delle perdite causa un aumento dei costi complessivi associati alle rotture dei tubi, i quali, oltre al costo dell'acqua persa e la messa a punto della condotta, includono anche la riparazione dell'infrastruttura circostante danneggiata e i danni di immagine del gestore idrico

causati dei reclami dei clienti relativi alla fornitura d'acqua interrotta.

Questo lavoro di tesi si è posto anche l'obiettivo di fornire alla Società Metropolitana Acque Torino (SMAT s.p.a.), i mezzi per l'individuazione di perdite idriche, o comunque quello di ridurre il periodo di tempo che va dalla nascita alla conoscenza di una potenziale rottura lungo la tubazione. Questo lasso temporale, anche chiamato (*Unawareness period* rappresenta, come mostrato in Figura 3.4, uno degli step costituenti il ciclo vitale di una perdita idrica.

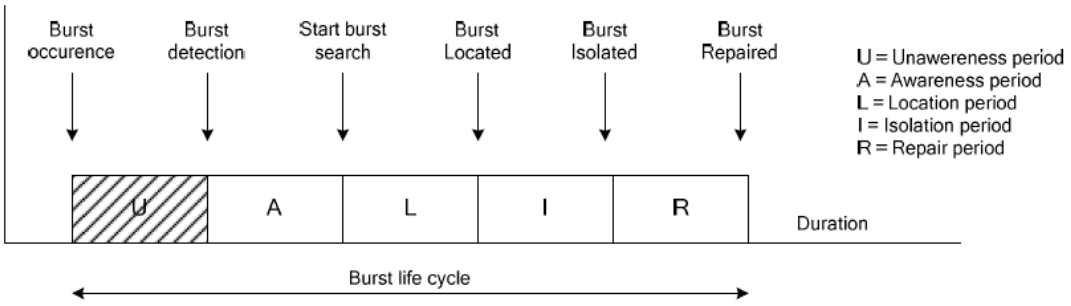


Figura 3.4: Ciclo vitale di una perdita idrica [16].

Al fine di avere una conoscenza quasi immediata del verificarsi di una perdita, nel corso del lavoro di tesi, è stata formulata sul software Python la funzione *Anna*. Tale funzione, che può essere anche utilizzata in Real-Time, richiede come input una serie temporale della variabile che si vuole analizzare e restituisce in output i rispettivi valori anomali. Questi ultimi sono delle indicazioni di una variazione delle condizioni standard di esercizio spesso causate da perdite idriche. Dunque, grazie alla funzione *Anna*, è possibile classificare in tempo reale il valore della variabile in "normale" oppure "anomalo". In quest'ultimo caso si indicherebbe la formazione di una perdita, eliminando quindi l'unawereness period. La scomparsa di questo lasso temporale, componente predominante nel ciclo di una perdita in sottosuolo oppure di giunzione [16], vedi Figura 3.3, comporta, per quanto sopra esposto, anche un significativo risparmio economico.

3.2 Sistemi SCADA

Nell'ambito dei controlli automatici, l'acronimo SCADA (Supervisory Control And Data Acquisition) indica un sistema informatico distribuito per il monitoraggio e la supervisione di sistemi fisici. Si tratta di una tecnologia in essere da oltre 30 anni e che si è costantemente evoluta grazie al progresso dell'elettronica, dell'informatica e delle reti di telecomunicazioni, principalmente utilizzata in ambito industriale e infrastrutturale.

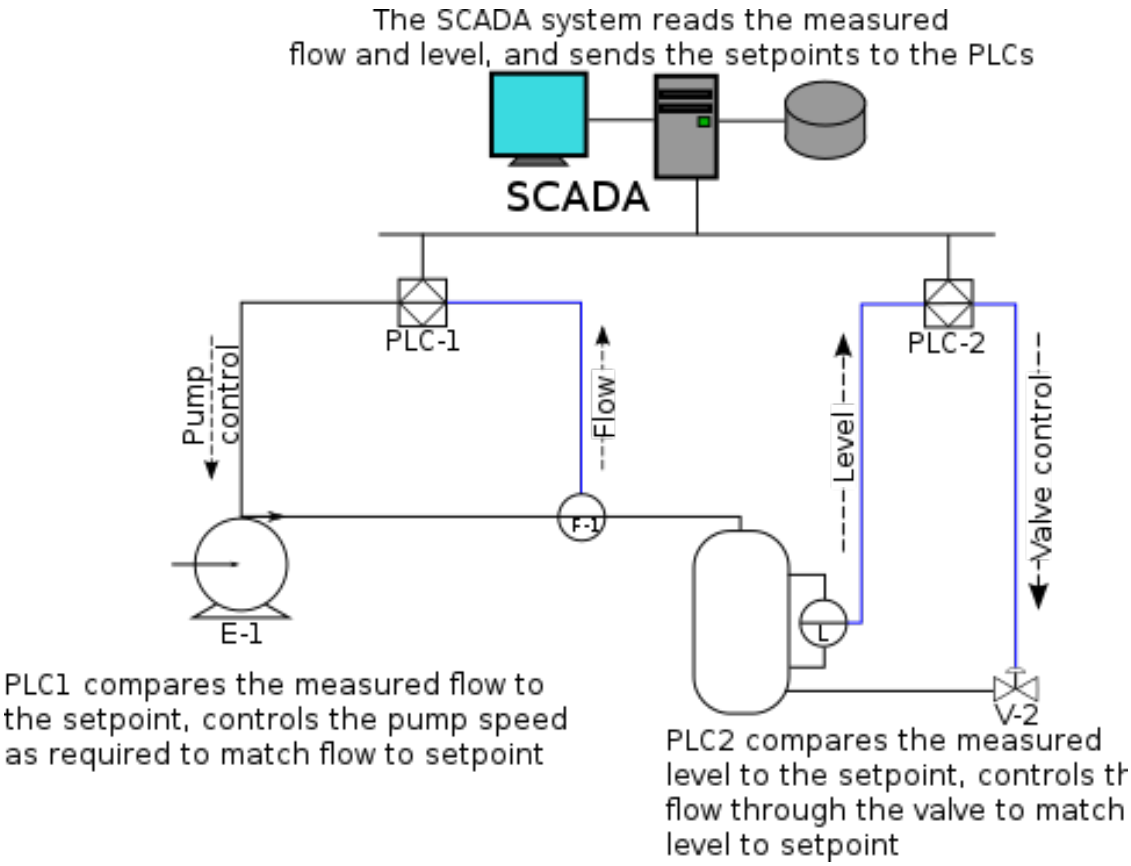


Figura 3.5: Schema esempio SCADA [20].

Allo stato attuale il termine SCADA può identificare un software, installato su personal computer o server, che permette la realizzazione e il funzionamento di sistemi di supervisione, controllo e telecontrollo senza dover necessariamente scrivere codici attraverso complessi linguaggi di programmazione. Quest'ultimo punto è particolarmente rilevante in quanto coloro che realizzano e utilizzano i sistemi SCADA sono spesso tecnici con background nel controllo di processo piuttosto che informatici o programmatori, un esempio di funzionamento è riportato in Figura

3.5, in cui i PLC sono i controlli logici programmabili. Normalmente i sistemi SCADA vengono impiegati all'interno delle control-room delle fabbriche, delle stazioni ferroviarie, degli aeroporti, degli acquedotti o dei grandi complessi di edifici piuttosto che per sistemi più piccoli, in prossimità del processo da controllare.

3.2.1 Telecontrollo Idrico

Il telecontrollo delle reti idriche, che avviene grazie ai sistemi SCADA, è il complesso insieme di sensori, convertitori, interfacce, periferiche, reti di trasmissioni e software che permettono il controllo del processo di gestione di un acquedotto, tramite misure, comandi, report, allarmi e data management. A titolo esemplificativo, in Figura 3.6 è riportata la schermata del telecontrollo dei sistema del centro abitato di Potenza.

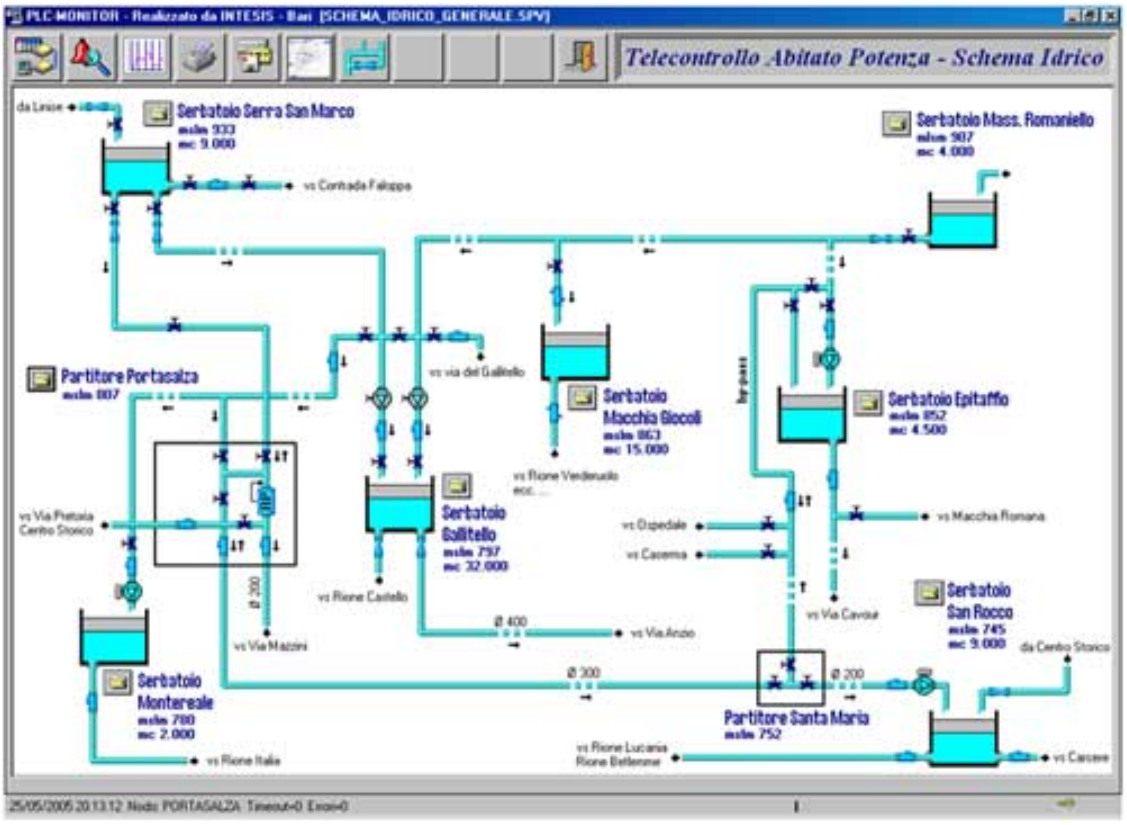


Figura 3.6: Telecontrollo del centro abitato di Potenza [29].

La parola telecontrollo, è in realtà un termine consolidato ma in parte fuorviante rispetto alle effettive potenzialità di questi sistemi che, pur definendosi tali, hanno prestazioni che vanno ben oltre il semplice "controllo a distanza". L'insieme delle

informazioni e delle azioni gestibili da un sistema di telecontrollo, adeguatamente trattate dai moderni software gestionali, realizza di fatto un efficace sistema di Business Management, applicabile nella gestione degli acquedotti come già da anni avviene nei processi industriali.

La gestione di impianti geograficamente distribuiti di assoluta rilevanza sociale, asserviti a leggi e norme, non può essere effettuata in maniera efficace senza l'uso di un sistema di telecontrollo evoluto. I vantaggi di una tale evoluzione, applicata alla gestione degli acquedotti, sono intuitivi e vanno dalla riduzione delle perdite all'ottimizzazione dei consumi energetici, al mantenimento degli asset, ecc.. In sintesi, una sicura riduzione dei costi ed una più efficace gestione.

Di seguito sono elencate le principali funzionalità che rendono indispensabile l'uso di sistemi SCADA per una corretta gestione dell'acquedotto:

- *Acquisizione dati.* Tramite controllori logici programmabili (Programmable Logic Controller - PLC) a loro volta connessi ai sensori o agli attuatori vengono acquisiti dati in real-time da tutti i componenti connessi al sistema (pompe, serbatoi, valvole di pressione, ecc..). I dati scambiati sono normalmente grandezze digitali, analogiche oppure stringhe di testo.

- *Rappresentazione grafica.* I dati vengono rappresentati all'interno di un web-browser o sullo schermo di uno smartphone. Essi possono essere rappresentati sia come grandezza che sotto forma di animazione grafica (esempio un serbatoio con il liquido colorato che sale e scende in base alla lettura del livello). Dall'interfaccia grafica è anche possibile inviare comandi al sistema di automazione.

- *Storicizzazione del dato.* I dati di interesse possono essere storicizzati su archivi locali o distribuiti, in varie metodologie: da file di tipo binario a database relazionali, a seconda del tipo di piattaforma impiegata e delle esigenze del progetto. Tali dati poi possono essere visualizzati dall'operatore direttamente dalla piattaforma anche sotto forma di grafici oppure esportati e gestiti su sistemi terzi che consentano l'analisi di situazioni critiche che si siano verificate.

- *Gestione degli allarmi.* L'allarme è una particolare condizione del processo che viene modellizzata dal progettista della piattaforma SCADA e che richiede l'interazione da parte di un operatore. In caso ad esempio di un blocco di una pompa, si potranno avere icone lampeggianti, l'emissione di un suono oppure nei sistemi più complessi l'invio di E-Mail, SMS, ovvero una chiamata telefonica in sintesi vocale all'operatore reperibile, il tutto al fine di informare lo

stesso della necessità di un'azione umana per risolvere un'anomalia sul sistema non gestibile in autonomia.

3.2.2 Early Warning

In generale, un Early Warning System (EWS) è un sistema di allerta precoce che, attraverso un avviso preventivo consente di prepararsi ad un evento, al fine di limitarne l'impatto. Esso solitamente si divide in 4 fasi: conoscenza del rischio, monitoraggio e segnalazione dell'inusuale cambiamento, comunicazione e diffusione dell'allerta, capacità e modalità di risposta all'evento. In Figura 3.7 sono rappresentate le fasi di un ESW in caso di alluvione.

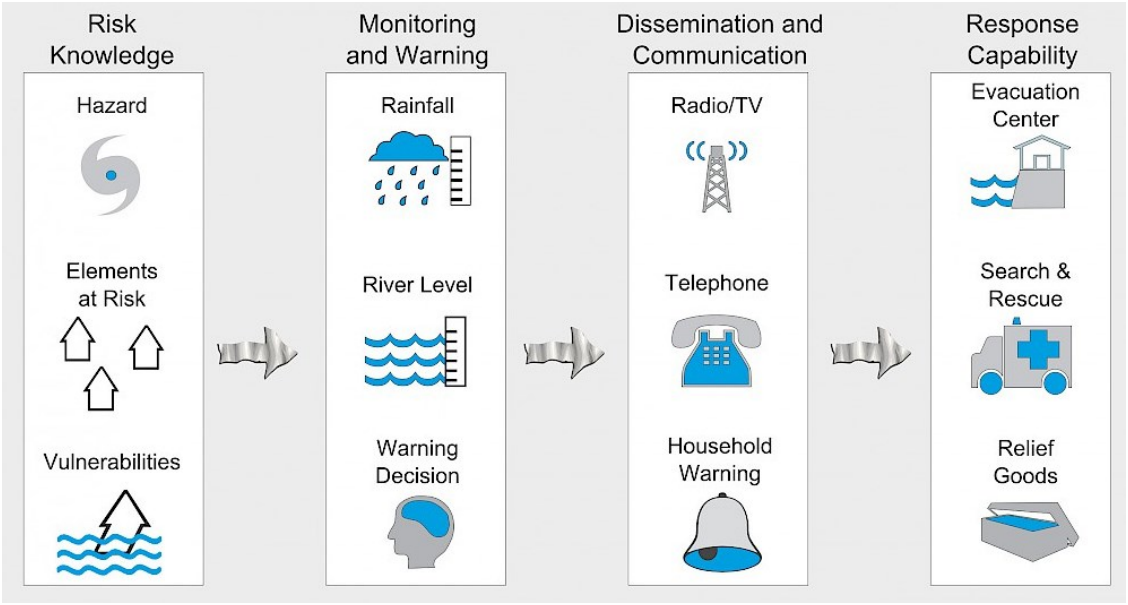


Figura 3.7: Step di un EWS catastrofico in caso di alluvione [23].

Un EWS può essere implementato e può comprendere sensori, sistemi di rilevamento e sottosistemi decisionali. Le varie componenti lavorano insieme per prevedere e segnalare i disturbi che influenzano negativamente la stabilità dell'ambito di riferimento [22].

Il sistema può essere applicato ai più svariati settori: da quello economico in cui può di aiutare le aziende a cogliere le opportunità del mercato e a reagire alle minacce in modo tempestivo a quello catastrofico dove l'EWS previene la perdita di vite umane e riduce l'impatto economico e materiale delle catastrofi [23].

Per quanto detto in questo capitolo, nei sistemi acquedottistici, perdite idriche e attacchi informatici causano danni via via crescenti con lo scorrere del tempo. Per

tal motivo, per i gestori idrici, la tempestività di intervento è un obiettivo primario da perseguire. Proprio sulla base di questo obiettivo, durante l' elaborazione di questa tesi, si è costruita la funzione *Anna*. Infatti, integrando tale funzione al sistema SCADA, in grado di acquisire in real-time i dati di tutti i componenti connessi al sistema, è possibile ottenere messaggi di warning indicanti un'anomalia connessa al relativo componente. Quindi, una volta riconosciuta l'anomalia, si può agire di conseguenza, limitandone l'impatto e provvedendo a risanare il malfunzionamento, in modo da tornare alle corrette condizioni di esercizio.

Capitolo 4

Casi di studio

Con l'obiettivo di rilevare in real-time eventuali comportamenti anomali dei componenti connessi al sistema di telecontrollo della SMAT s.p.a., sono stai forniti dati inerenti la centrale di Cavoretto e di Avigliana, già monitorate attraverso il sistema SCADA. Nel corso del capitolo, dopo aver analizzato i dati si è cercato di automatizzare i parametri del CUSUM test. Infine è stato formulato l'algoritmo *Anna* in grado di classificare in real-time il valore della variabile in "normale" oppure "anomalo".

4.1 Centrale di Avigliana

La Centrale di Avigliana si trova ad ovest rispetto al comune di Torino e in Figura 4.1 è riportata l'area geografica rifornita da tale impianto.

Figura 4.1: Area rifornita dalla centrale di Avigliana.

Nella centrale è presente un impianto di sollevamento tramite cui è pompata una portata media di 7.48 l/s. La rete di distribuzione associata è lunga 26.65 km e rifornisce circa 3200 abitanti rappresentanti l'utenza. Tale rete è caratterizzata da un rapporto di $Fughe/km/anno = 0.25$. Essendo un valore relativamente basso si deduce che lungo la rete si rilevano poche rotture in tubazione. Più realisticamente tale valore è dovuto ad un bassa densità abitativa ed alla presenza di un terreno permeabile che rendono difficilmente individuabile una perdita idrica, comportando dunque un spreco sempre maggiore della risorsa.

La Centrale di Avigliana funziona come mostrato in Figura 4.2, dove QR1 è la portata sollevata dalla centrale di pompaggio e diretta verso l'area di riferimento.

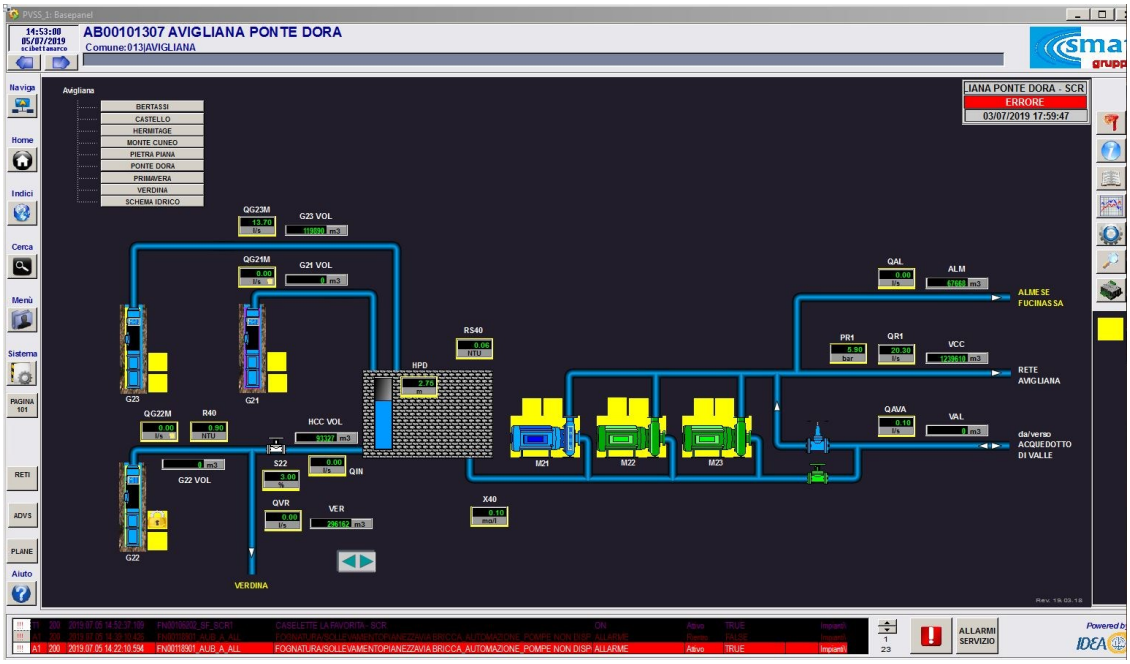


Figura 4.2: Schema della centrale di Avigliana.

Sono stati forniti i dati riguardanti la QR1. Tali dati individuano una finestra temporale che va dal 02-05-2018 00:00 al 30-04-2019 23:45 con frequenza ogni 15 minuti. In Figura 4.3 si illustra l'andamento della portata media giornaliera. Osservando la figura si può notare che quasi l'intero database è influenzato da una perdita verificatasi 02-07-2018. Tale perdita, non essendo mai stata riparata, ha traslato verticalmente l'intero diagramma da quel momento in poi. Inoltre in figura è visibile un picco anomalo riscontrato il 27-08-2018. Esso è causato da un errore nella trascrizione dei dati, ma tale errore verrà discusso successivamente.

L'andamento riportato, oltre ad essere influenzato da eventi anomali, si relaziona alle variazioni di richiesta idrica da parte dell'utenza dovute al diverso consumo in

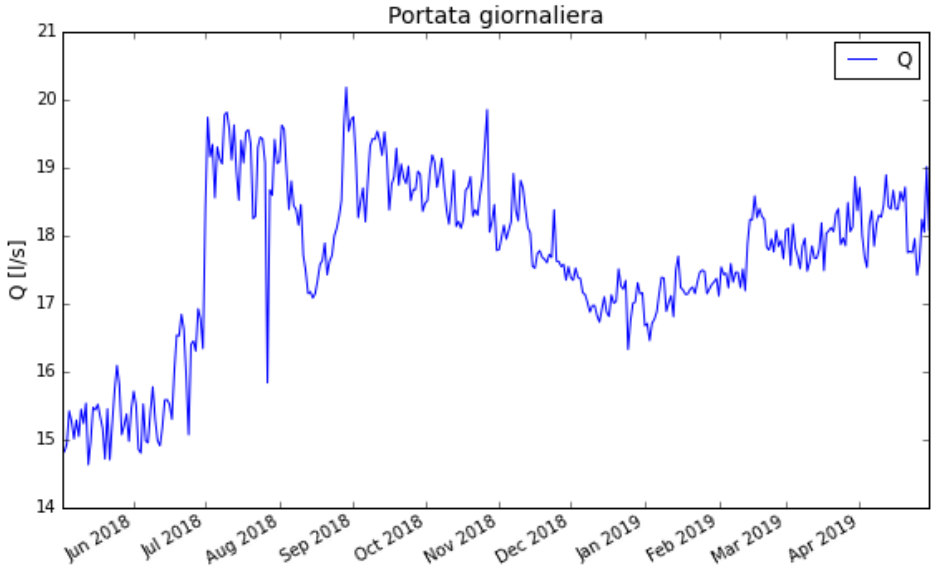


Figura 4.3: Andamento della portata media giornaliera.

scala giornaliera e alle differenti condizioni climatiche in scala annuale.

Al fine di illustrare l'andamento giornaliero standard della portata in uscita, in Figura 4.4 viene rappresenta la portata al variare dell'ora in tre giorni contigui (andamenti corrispondenti dal 03 al 05 Gennaio 2019).

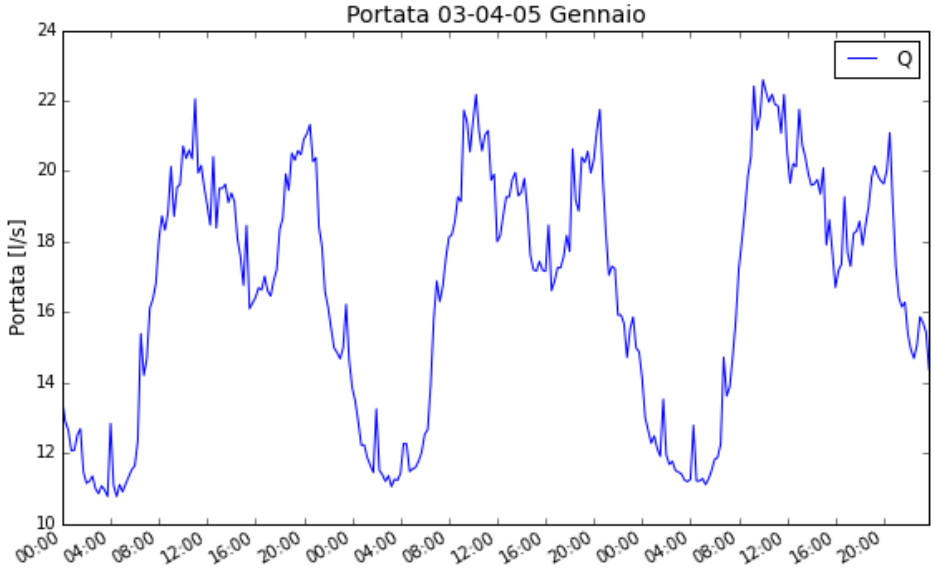


Figura 4.4: Andamento standard della portata nel corso della giornata.

Come mostrato in figura, il sistema, essendo programmato in modo da soddisfare le richieste idriche dell'utenza, assume un andamento ciclico. Infatti da mezzanotte alle 7 del mattino si ha un consumo minimo, successivamente si ha una crescita esponenziale fino a raggiungere un picco intorno alla 10:00. In seguito, a parte una fluttuazione negativa nel primo pomeriggio, la portata rimane stabile fino alle 20:00. Infine decresce rapidamente tornando al valore già riscontrato alla precedente mezzanotte.

4.1.1 CUSUM test Avigliana

Si è provato ad effettuare il rilevamento di anomalie sulle portate in uscita dalla centrale di Avigliana utilizzando l'algoritmo CUSUM. Per applicare tale algoritmo in Python si è fatto riferimento allo script [8], opportunamente modificato, riportato in appendice A.1.

Sfruttando la conoscenza, da parte dei tecnici SMAT, di due anomalie note, verificatesi il 06-02-2019 alle ore 14:15 e il 13-02-2019 alle ore 14:45, a cui sono associate rispettivamente le portate 14.93 l/s e 13.62 l/s, si è sottoposto il Dataframe al CUSUM test, in modo da verificare se tale algoritmo riuscisse ad individuare le anomalie note. Al fine di comprenderne la proporzione, si riportano in Figura 4.5 e 4.6 i grafici rappresentanti l'andamento delle portate nei giorni 5-6-7 Febbraio e 12-13-14 Febbraio 2019, nei quali si indicano in rosso i due eventi anomali.

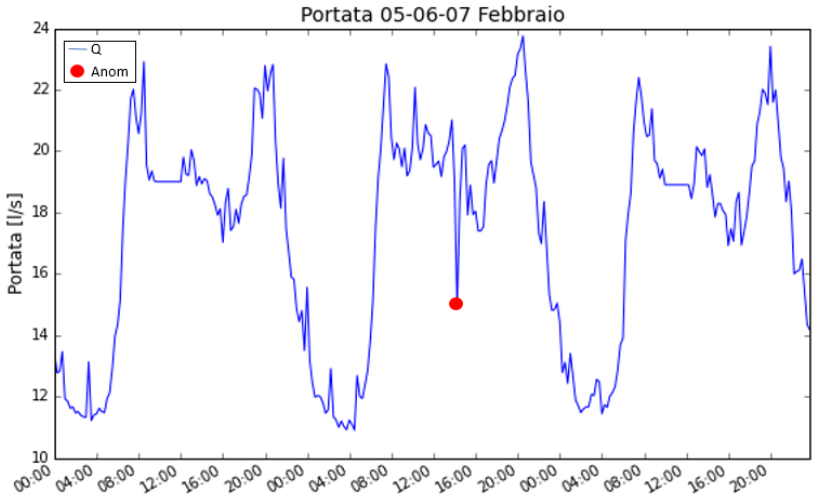


Figura 4.5: Anomalia del 06/02/2019 alle ore 14:15.

Per cercare di comprendere gli effetti che le variazioni climatiche annuali comportano sulla domanda idrica, sono state acquisite, per l'intero periodo in esame,

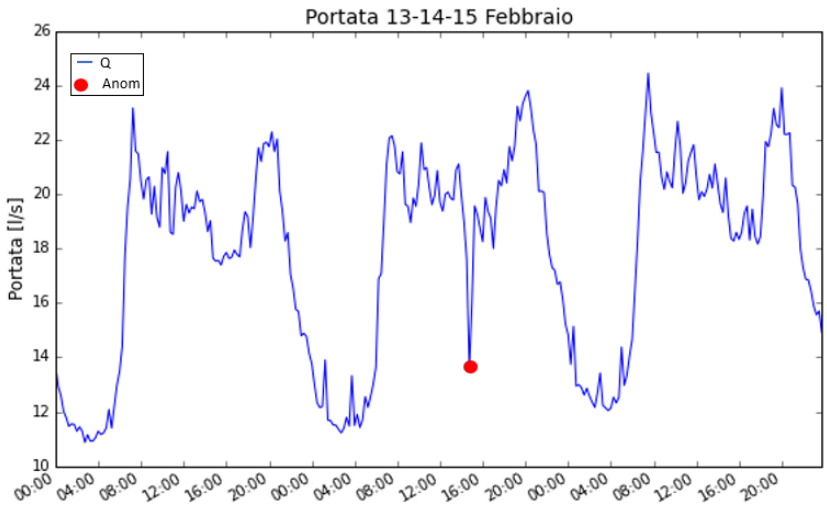


Figura 4.6: Anomalia del 13/02/2019 alle ore 14:45.

le temperature medie giornaliere verificatesi nel comune di Avigliana [24]. Al fine di sottolineare la correlazione tra richiesta idrica e temperatura ambientale, viene riportato in Figura 4.7 l'andamento mensile di portata e temperatura.

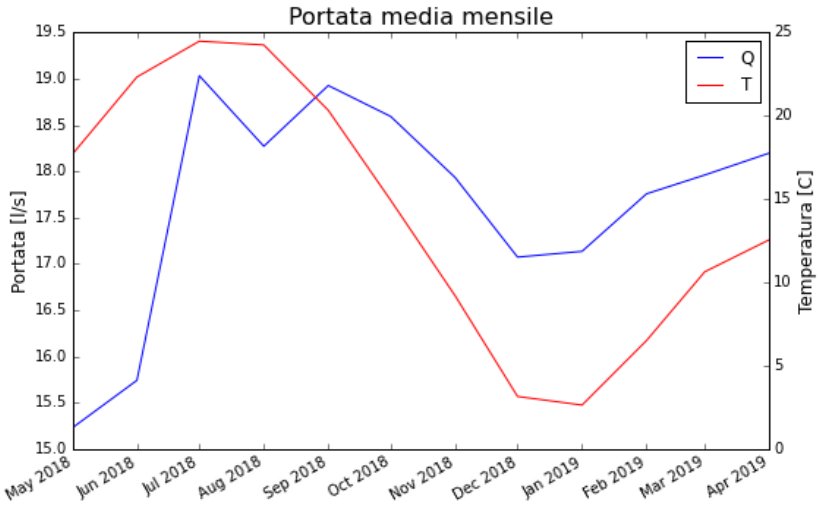


Figura 4.7: Andamento mensile di portata e temperatura.

Essendo evidente la correlazione tra le due grandezze, si è deciso di effettuare il CUSUM test, oltre che sui dati già a disposizione, anche sui dati normalizzati. Si

sono dunque, per ogni mese, normalizzate le portate attraverso l'equazione 4.1:

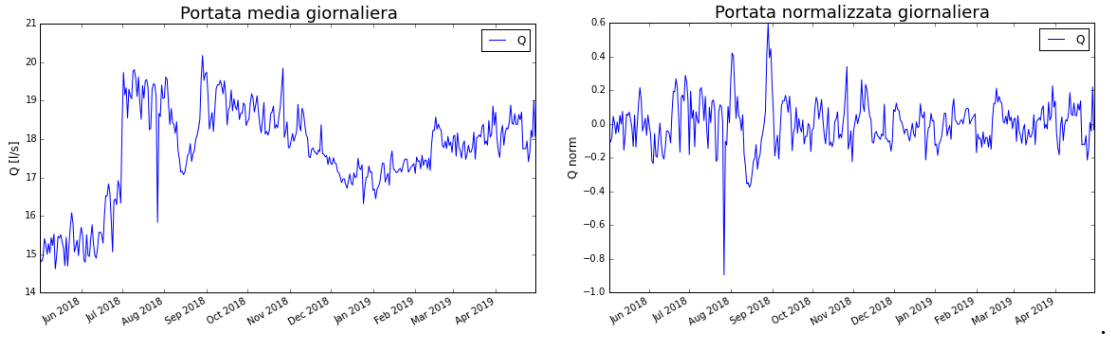
$$z_i = \frac{x_i - \overline{x}_i}{\sigma_i} \tag{4.1}$$

Dove z_i è il valore della portata normalizzata, x_i è la variabile misurata al tempo i-esimo, $\overline{x}_i$ e σ_i, riportati in Tabella 4.1, sono le medie e le deviazioni standard mensili.

	Mesi											
	Mag	Giu	Lug	Ago	Set	Ott	Nov	Dic	Gen	Feb	Mar	Apr
$\overline{x}_i\ [l/s]$	15.2	15.7	19.1	18.3	18.9	18.6	17.9	17.1	17.1	17.8	17.9	18.2
$\sigma_i\ [l/s]$	3.93	4.05	3.58	3.18	3.53	3.67	3.70	3.54	3.70	3.85	3.95	3.67

Tabella 4.1: Medie e deviazione standard mensili.

Come riportato in Figura 4.8, comparando l'andamento della portata media giornaliera, normalizzata e non, si evince l'effetto della normalizzazione. Infatti, è evidente che tale operazione stabilizza le portate nel corso dell'anno, annullando le fluttuazioni dovute alle variazioni di temperatura mensili.



(a) *Portate giornaliere standard.* (b) *Portate giornaliere normalizzate.*

Figura 4.8: Comparazione tra andamento portate normalizzate e non.

Inizialmente si è sottoposta al test l'intera serie di dati, normalizzati e non, con frequenza giornaliera. Come prime tentativo per l'individuazione dei parametri h e ν si sono seguite le indicazioni fornite da Gustafsson [7], il quale pone $\nu = \sigma$ e $h = 5\sigma$, dove σ è la deviazione standard della serie sottoposta ad analisi, vedi Figura 4.9. Per la serie standard si ottiene $\nu = 1.24\ l/s$ e $h = 6.18\ l/s$, per quella normalizzata $\nu = 0.139$ e $h = 0.69$. Tutti i valori di soglia e di drift risultano essere troppo elevati. Infatti, anche se effettivamente presenti, non viene riscontrata alcuna anomalia, eccetto il 27-07-2018 in cui si verifica una portata media giornaliera di

15.83 l/s che è quindi estremamente bassa rispetto alla media annuale di 17.66 l/s. Si precisa inoltre che tale metodo per l'individuazione dei parametri, anche se non riportato successivamente nel corso dell'elaborato, è stato provato per differenti serie analizzate ma ha sempre riscontrato i medesimi problemi. Per tal motivo, nelle successive serie analizzate non sarà più riportato.

Assodato che il metodo di Gustafsson non comporta un buon riscontro, per la calibrazione dei parametri si è partiti scegliendo un valore di drift molto piccolo, pari a 0.1, ma comunque maggiore di zero in modo da ridurre l'effetto dei dati passati e di conseguenza anche il numero di falsi allarmi rilevati. Volendo classificare come anomali i picchi più alti e più bassi della serie, rappresentanti i valori minimi e massimi, si sono provati più valori di h in modo da trovare quelli che si adattassero meglio alla serie. Individuati come valori di soglia ottimali $h = 1\ l/s$ e 0.25 (rispettivamente per la serie standard e normalizzata), si è effettuato il CUSUM test riportato in Figura 4.10. Le anomalie riscontrate vengono riportate in Tabella 4.2.

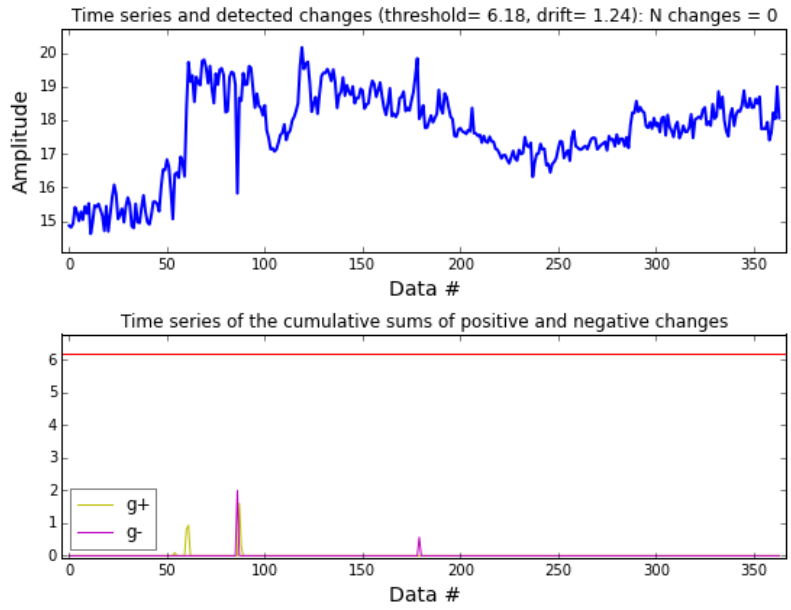
$Q_{giorn\ norm}$ $h = 0.25$ $\nu = 0.1$			$Q_{giorn\ stand}$ $h = 1\ l/s$ $\nu = 0.1\ l/s$	
Anomalie			Anomalie	
Data	z_i	Q_i [l/s]	Data	Q_i [l/s]
02-07-2018	0.195	19.73	2018-05-25	16.08
27-07-2018	-0.895	15.83	2018-06-19	16.52
28-07-2018	-0.103	18.67	2018-06-24	15.06
29-08-2018	0.596	20.17	2018-06-25	16.39
03-09-2018	-0.190	18.26	2018-07-01	18.37
28-10-2018	-0.150	18.04	2018-07-21	18.24
			2018-07-27	15.82
			2018-07-28	18.66
			2018-08-11	17.70
			2018-08-28	19.58
			2018-09-03	18.25
			2018-10-27	19.84
			2018-10-28	18.04
			2019-04-29	19.00

Tabella 4.2: Anomalie riscontrate nelle portate giornaliere normalizzate e non.

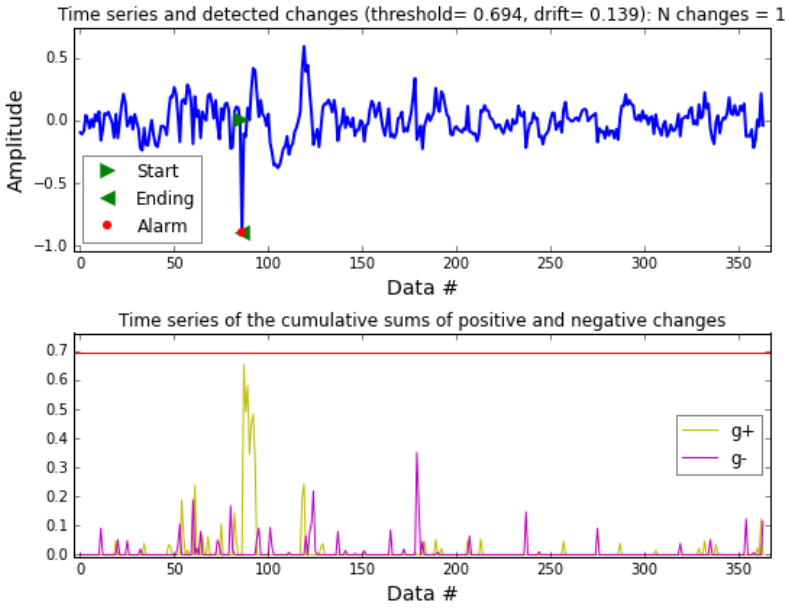
Si noti che in questo caso, le anomalie note non sono state riscontrate. Ciò è dovuto al fatto che tali anomalie sono durate pochi minuti e, non essendo prolungate nel tempo, hanno determinato un peso minimo nella portata media giornaliera

risultante.

Individuato il problema, si è quindi deciso di sottoporre al CUSUM test l'intera



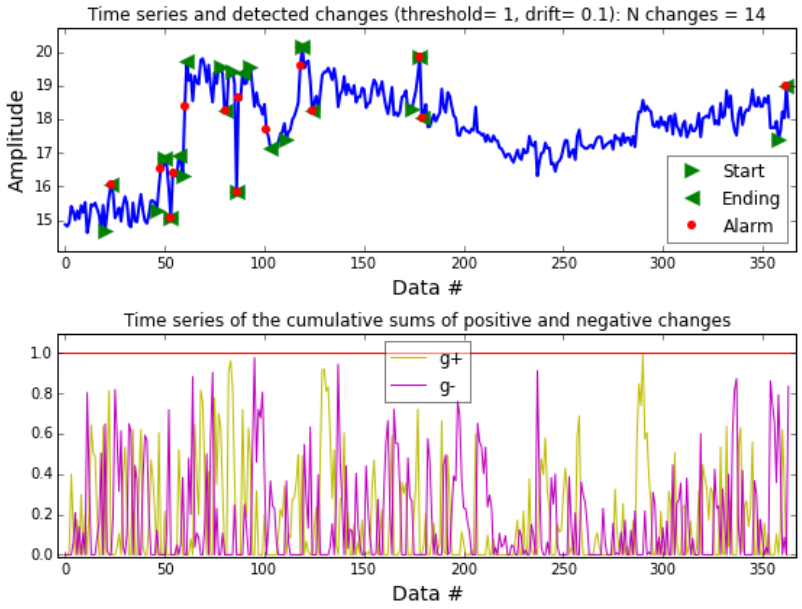
(a) *Portate giornaliere standard.*



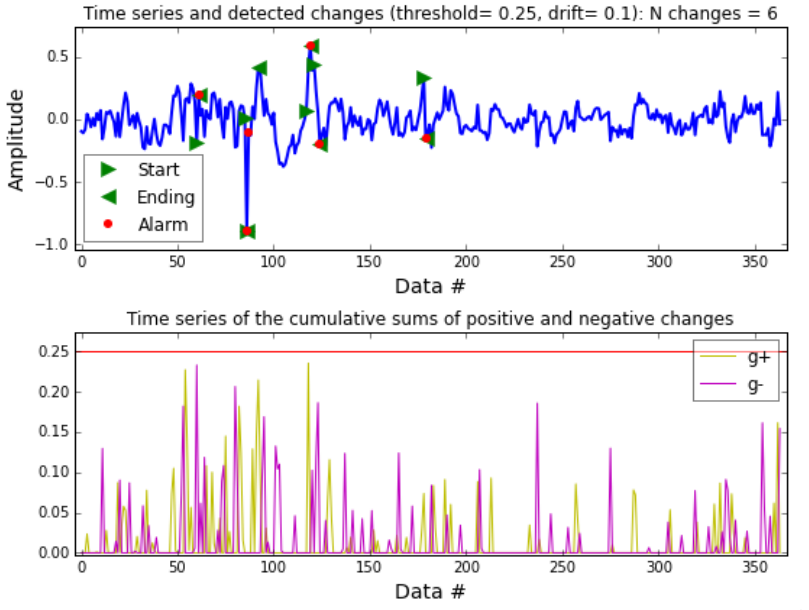
(b) *Portate giornaliere normalizzate.*

Figura 4.9: Comparazione tra portate giornaliere normalizzate e non secondo il metodo di Gustafsson.

serie di dati con frequenza di 15 minuti. Come fatto in precedenza si è partiti ponendo un valore di drift molto piccolo, $\nu = 0.1$, successivamente si sono provati più valori di h in modo da trovare quelli che si adattassero meglio alla serie, ottenendo



(a) *Portate giornaliere standard.*

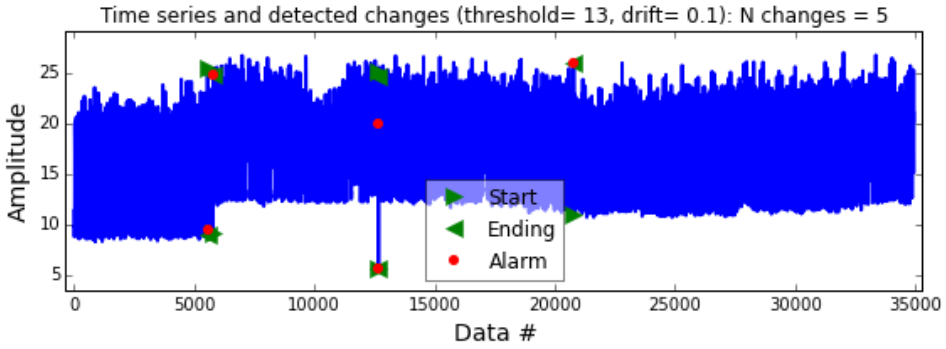


(b) *Portate giornaliere normalizzate.*

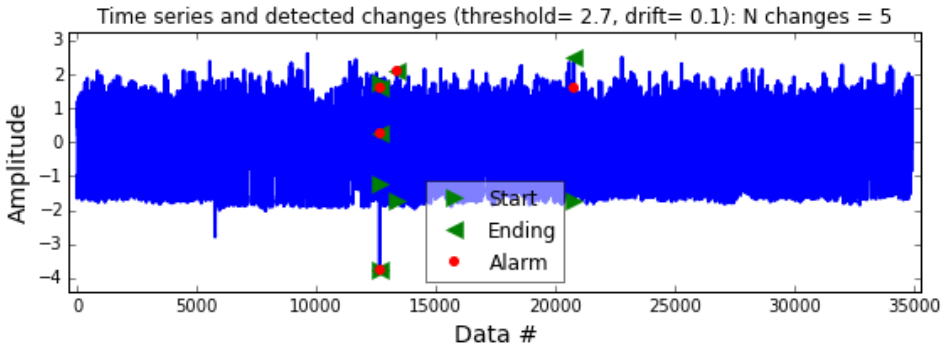
Figura 4.10: Comparazione tra andamento portate giornaliere normalizzate e non.

$h = 13\,l/s$ e 2.7. Gli output sono riportati in Figura 4.11 e le anomalie in Tabella 4.3.

Si precisa che, a partire da questa figura, visto i numerosi CUSUM test effettuati, nelle immagini rappresentanti i rispettivi output, (a meno di casi in cui risultino esplicativi) non verranno più riportati gli andamenti delle somme cumulate $g^+[t]$ *e* $g^-[t]$, evitando in questo modo di aumentare troppo il peso grafico della tesi.



(a) *Portate standard.*



(b) *Portate normalizzate.*

Figura 4.11: Comparazione tra andamento portate normalizzate e non.

Si noti che, anche in questo caso, le anomalie note non sono state ritrovate. Ciò è dovuto al fatto che tali anomalie di portata determinano con i dati adiacenti, uno scarto $s[t]$ minore rispetto alle variazioni che si verificano normalmente nell'arco della giornata (specialmente tra le 07:00 e le 08:00). Dunque lavorando su questa serie di dati, risulta impossibile riscontrare tali anomalie, senza produrre un altissimo tasso di falsi allarmi.

Per esplicitare quanto appena detto, si sono analizzati i dati rappresentanti le serie di portate dal giorno 5 al 7 e dal 12 al 14 Febbraio 2019, in quanto contenenti

$Q_{15min\,norm}$ $h = 2.7$ $\nu = 0.1$			$Q_{15min\,stand}$ $h = 13\,l/s$ $\nu = 0.1\,l/s$	
Anomalie			Anomalie	
Data	z_i	Q_i [l/s]	Data	Q_i [l/s]
2018-09-10 21:45	-3.782	5.54	2018-06-29 01:30	9.38
2018-09-10 22:00	0.283	19.93	2018-07-01 09:30	24.83
2018-09-10 22:30	1.588	24.55	2018-09-10 21:45	5.54
2018-09-18 07:15	2.099	26.36	2018-09-10 22:00	19.93
2018-12-04 11:30	1.605	19.22	2018-12-04 11:45	25.93

Tabella 4.3: Anomalie riscontrate nelle portate normalizzate e non.

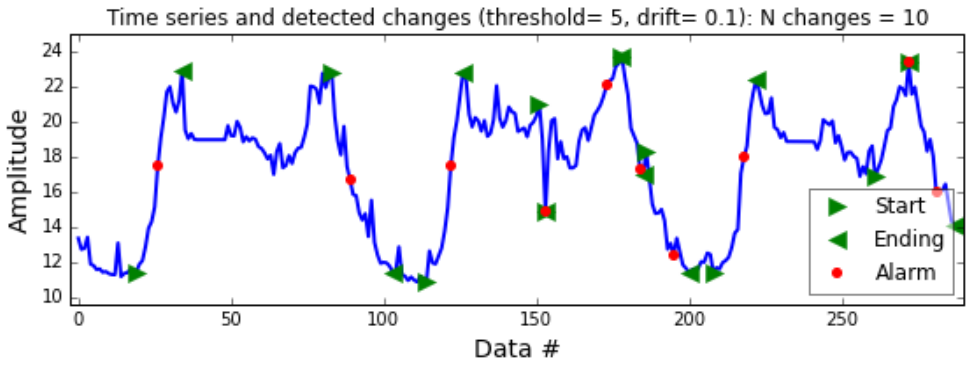
le anomalie note. Per entrambe le serie, posto $\nu = 0.1$, si è iterato h al fine di trovare il valore tramite cui fosse possibile riscontrare le anomalie note. Si è dedotto che i valori di soglia massima, che individuano l'evento anomalo del 06-02-2018 alle 14:45, rispettivamente della serie normalizzata e non, siano $h = 0.25$ e $1.3\,l/s$. L'output di tali serie sono riportati in Figura 4.12 e le anomalie in Tabella 4.4 in cui si evidenzia quella nota in grassetto.

Anomalie dal 05 al 07 Febbraio 2019				
$Q_{15min\,stand}$ $h = 5\,l/s$ $\nu = 0.1\,l/s$		$Q_{15min\,norm}$ $h = 1.3$ $\nu = 0.1$		
Data	Q_i [l/s]	Data	z_i	Q_i [l/s]
2019-02-05 06:30	17.48	2019-02-05 07:00	0.654	20.28
2019-02-05 22:15	16.72	2019-02-06 07:00	0.605	20.09
2019-02-06 06:30	17.51	**2019-02-06 14:15**	**-0.733**	**14.93**
2019-02-06 14:15	**14.93**	2019-02-07 00:15	-1.291	12.78
2019-02-06 19:15	22.08	2019-02-07 07:00	0.740	20.61
2019-02-06 22:00	17.33			
2019-02-07 00:45	12.43			
2019-02-07 06:30	17.98			
2019-02-07 20:00	23.41			

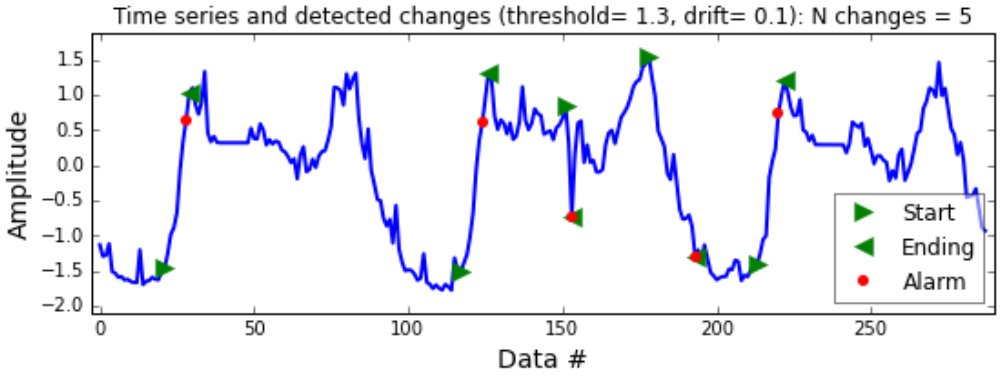
Tabella 4.4: Anomalie dal 05 al 07 Febbraio 2019 con portate normalizzate e non.

Per l'evento anomalo verificatosi in data 13-02-2018 i valori di soglia massima trovati sono $h = 0.25$ e $1.3\,l/s$. L'output di tale serie è mostrato in Figura 4.13 e le anomalie sono riportate in Tabella 4.5. Come per il caso precedente l'anomalia nota è evidenziata in grassetto.

Tramite questa analisi, le anomalie ricercate sono state riscontrate ma si sono



(a) *Cusum test su portate standard.*

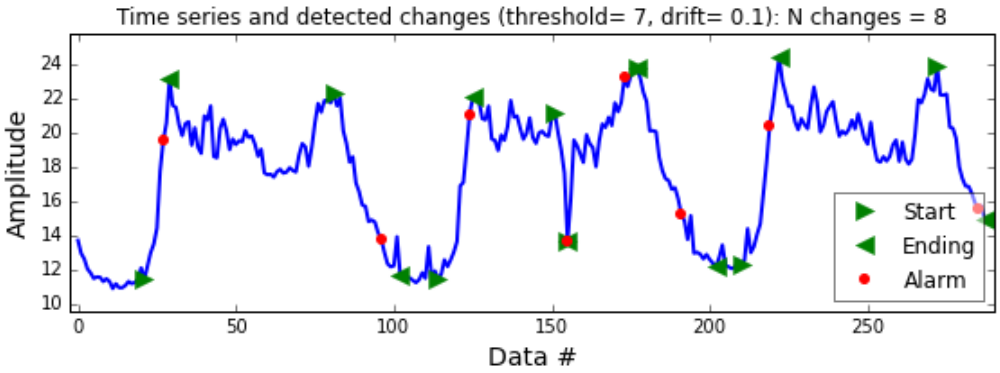


(b) *Cusum test su portate normalizzate.*

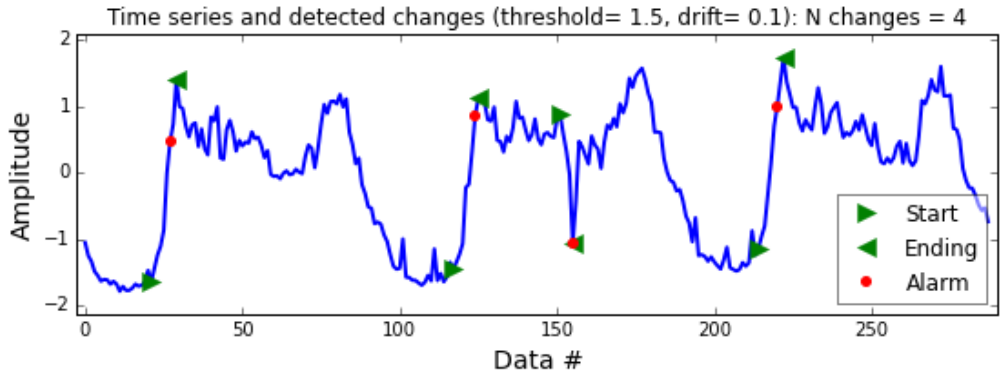
Figura 4.12: Cusum test dal 05 al 07 Febbraio 2019 su portate normalizzate e non.

Anomalie dal 12 al 14 Febbraio 2019					
$Q_{15min\ stand}$ $h = 7\ l/s$ $\nu = 0.1\ l/s$			$Q_{15min\ norm}$ $h = 1.3$ $\nu = 0.1$		
Data	Q_i [l/s]		Data	z_i	Q_i [l/s]
2019-02-12 06:45	19.54		2019-02-12 06:45	0.462	19.54
2019-02-13 00:00	13.76		2019-02-13 07:00	0.864	21.09
2019-02-13 07:00	21.09		**2019-02-13 14:45**	**-1.073**	**13.62**
2019-02-13 14:45	**13.62**		2019-02-14 07:00	0.996	21.60
2019-02-13 19:15	23.22				
2019-02-13 23:45	15.22				
2019-02-14 06:45	20.44				
2019-02-14 23:15	15.56				

Tabella 4.5: Anomalie dal 12 al 14 Febbraio 2019 con portate normalizzate e non.



(a) *Cusum test su portate standard.*



(b) *Cusum test su portate normalizzate.*

Figura 4.13: Cusum test dal 12 al 14 Febbraio 2019 su portate normalizzate e non.

rilevati anche dei falsi allarmi, rispettivamente 9 e 4 per la prima serie, 7 e 3 per la seconda. Essi sono dovuti alle variazioni che si verificano normalmente nell'arco della giornata (specialmente tra le 07:00 e le 08:00 e tra le 21:00 e le 22:00) le quali determinano uno scarto tra i dati s[t] anche maggiore di quello causato dalle anomalie. Un rapporto così elevato tra falsi allarmi ed effettive anomalie non è ovviamente accettabile. Tale problematica deriva dal fatto che siamo in presenza di anomalie contestuali (si veda il paragrafo 2.1.2), infatti, tali valori risultano anomali in relazione al loro attributo di contesto, che in questo caso è l'orario in cui si verificano. Le portate anomale di 14.93 l/s e 13.62 l/s, rispettivamente del 06-02-2019 alle 14:15 e del 13-02-2019 alle 14:45, risultano anomali per quegli orari, in quanto la portata media è circa di 19 l/s ma risulterebbero normali se avvenissero intorno alle ore 07:00 o alle 23:00 dove la portata media in uscita è di circa 14 l/s.

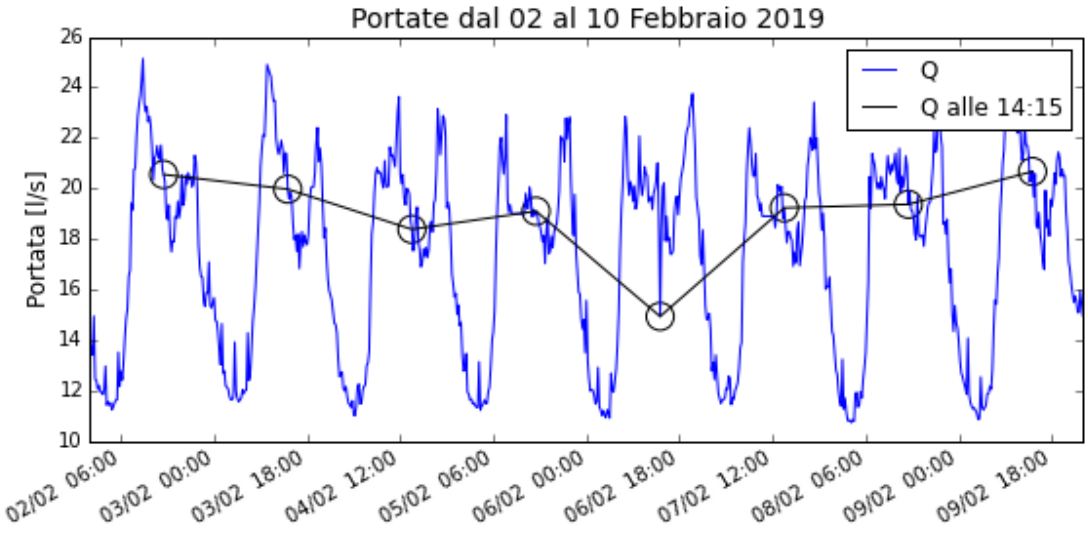
4.1.2 Da anomalia contestuale a puntuale

Rispetto alla ricca letteratura sulle tecniche di rilevamento delle anomalie puntuali, la ricerca sulla rilevazione delle anomalie contestuali è ancora limitata. In generale, per sopperire a tali problematiche si riduce un problema di rilevamento di anomalie contestuali a un problema di rilevamento di anomalie puntuali.

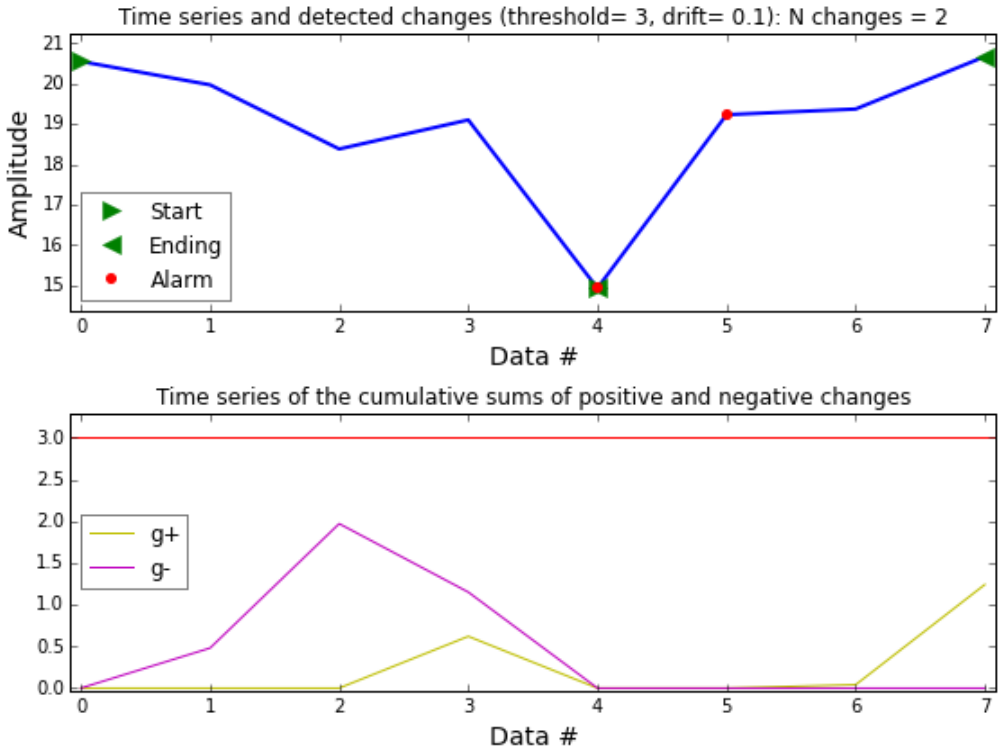
Dal momento che le anomalie contestuali sono istanze di dati individuali (come le anomalie puntuali), ma sono anomale solo rispetto ad un contesto, un approccio consiste nell'applicare una tecnica di rilevamento di anomalie in punti noti all'interno di un contesto. Innanzitutto, si identifica un contesto per ciascuna istanza di test utilizzando gli attributi contestuali. Successivamente, si definisce un'istanza come anomala oppure no all'interno del contesto utilizzando una tecnica di rilevamento per le anomalie puntuali.

Si riporta tale ragionamento al caso in esame, in cui il contesto è l'orario definito ogni 15 minuti. Dunque, a partire dalla serie di dati iniziali, si ottengono 24 (ore in un giorno) $\times$ 4 (15 minuti in 1 ora) = 56 serie, ognuna avente la portata in uno specifico orario al variare dei giorni componenti il database principale. In questo modo si generano 56 serie formate da istanze non più contestuali, ed applicando a questo punto il CUSUM test è possibile ottenere le anomalie ricercate.

A titolo esemplificativo in Figura 4.14(a) si riporta quanto detto sopra per la serie formata dalle portate alle 14:15. Dato il database iniziale, in questo caso ridotto dal 2 al 10 Febbraio (riportato in blu), vengono localizzate le portate allo stesso orario al variare dei giorni (cerchiate in nero in figura) e, a partire da queste, si forma la nuova serie (riportata in nero) in cui le istanze non sono più contestuali. Si può dunque, come mostrato in Figura 4.14(b) effettuare il CUSUM test, il quale rileva una portata anomala di 14.93 l/s il 06-02-2019 alle 14:15. Si è dunque riscontrata l'anomalia nota. Si noti che, nonostante non sia anomala, è riportata come anomalia anche l'istanza relativa al giorno successivo e questo in quanto tra le due istanze è presente uno scarto $s[t]$ maggiore della soglia h. Tale riscontro va a sottolineare l'evento anomalo precedente e non è considerato come falso allarme in quanto l'operatore designato è già consapevole dell'anomalia precedentemente verificatasi.



(a) *Da serie con istanze contestuali a puntuali.*



(b) *Cusum test su serie ridotta.*

Figura 4.14: Cusum test per serie contestuali.

4.1.3 CUSUM test su serie orarie

Avendo a disposizione le portate discretizzate ogni 15 minuti, ed essendo le anomalie note verificatesi alle ore 14:15 e 14:45, si sono, come appenda descritto, costruite ed analizzate le serie corrispondenti a tali orari.

Poiché dalla serie iniziale, in totale, saranno estratte 56 serie differenti, impostare manualmente il valore dei parametri per ognuna risulta un lavoro troppo oneroso e difficilmente replicabile ad altre centrali od ad altre variabili da analizzare. Perciò, posto un valore di drift piccolo, $\nu = 0.1$ si sono iterati più valori di h in modo da trovare quello che si adattasse meglio alla serie sottoposte ad analisi. Si è riscontrato una valore ottimale di $h = 1$ per la serie normalizzata e $h = 4\,l/s$ per quella standard.

Serie delle 14:15

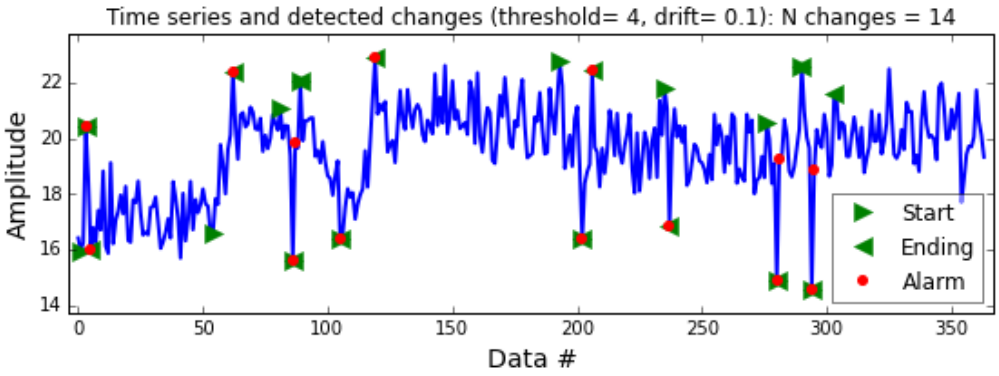
A partire dalla serie iniziale si è formata la serie costituita da tutte le portate verificatesi alle ore 14:15 al variare dei giorni. In Figura 4.15 si riporta il CUSUM test effettuato sulle due serie, una delle quali costituita da portate normalizzate. Le anomalie riscontrate sono riportate in Tabella 4.6.

Anomalie della serie oraria 14:15					
$Q_{15min\ stand}$	$h = 4\,l/s$	$\nu = 0.1\,l/s$	$Q_{15min\ norm}$	$h = 1$	$\nu = 0.1$
Data		Q_i [l/s]	Data	z_i	Q_i [l/s]
2018-05-05 14:15		20.41	2018-07-27 14:15	-0.958	15.60
2018-05-07 14:15		15.99	2018-07-28 14:15	0.222	19.83
2018-07-03 14:15		22.41	2018-08-29 14:15	1.457	22.92
2018-07-27 14:15		15.60	2018-11-20 14:15	-0.413	16.40
2018-07-28 14:15		19.83	2018-11-24 14:15	1.224	22.47
2018-08-15 14:15		16.37	2018-12-25 14:15	-0.057	16.87
2018-08-29 14:15		22.92	**2019-02-06 14:15**	**-0.733**	**14.93**
2018-11-20 14:15		16.40	2019-02-07 14:15	0.381	19.23
2018-11-24 14:15		22.47	2019-02-20 14:15	-0.832	14.55
2018-12-25 14:15		16.87	2019-02-21 14:15	0.285	18.86
2019-02-06 14:15		**14.93**			
2019-02-07 14:15		19.23			
2019-02-20 14:15		14.55			
2019-02-21 14:15		18.86			

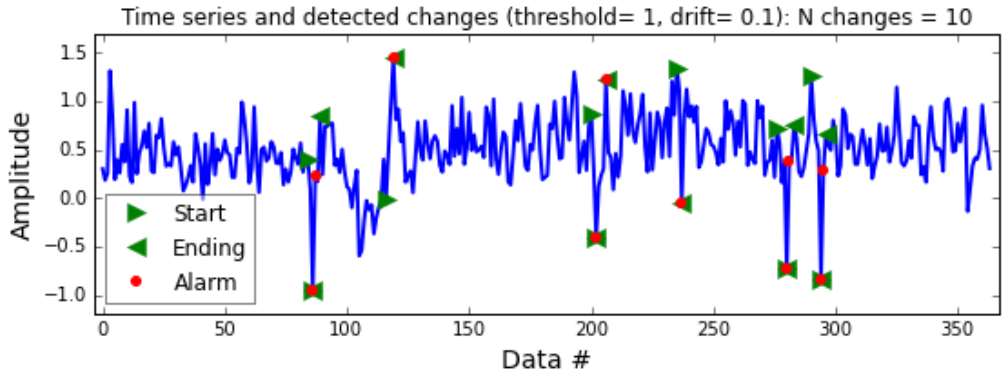
Tabella 4.6: Anomalie della serie oraria delle 14:15 normalizzata e non.

Come visibile in tabella, l'anomalia nota del 06-02-2019 è stata riscontrata, inoltre si sono riconosciuti come anomali gli eventi verificatisi alle seguenti date:

05-05-2018 Registrazione di un picco particolarmente alto.



(a) *Cusum test serie standard.*



(b) *Cusum test su normalizzata.*

Figura 4.15: Cusum test sulle serie oraria delle 14:15.

03-07-2018 e 15/08/2018 Falsi allarmi riscontrati a causa di valori precedenti relativamente alti, vedi Figura 4.16(d).

27-07-2018 e 28-07-2018 Allarmi dovuti ad errori di trascrizione presenti nei dati forniti. Si fa presente che l'analisi dei dati si è svolta sulla serie storica di dati registrati dalla società SMAT. A volte, come in questo caso, si verificano errori di trascrizione, nel passaggio da dati Real Time a serie storiche. Avendo come obiettivo quello di utilizzare tale algoritmo direttamente sui dati Real Time, tali falsi allarmi non saranno riscontrati dai supervisori, vedi Figura 4.16(c).

29-08-2018 In cui si riscontra un picco particolarmente alto, accettabile in quanto si tratta di fine agosto.

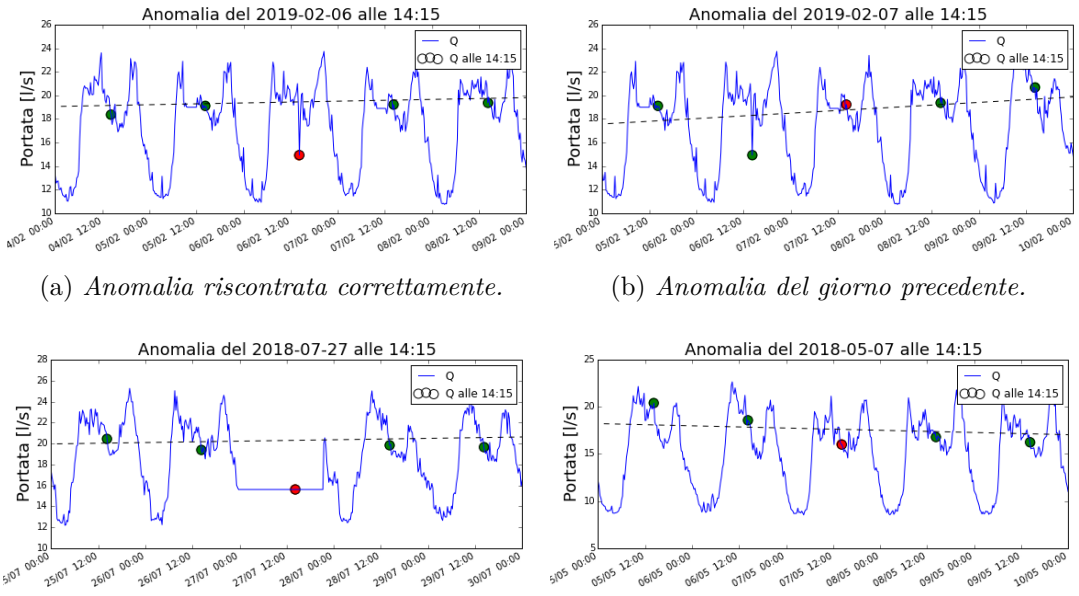
20-11-2018, 24-11-2018, 06-02-2019 e 20/02/2019 In tali date si verificano eventi anomali. Si è successivamente riscontrato che tali eventi sono conseguenze

di interventi effettuati sulla rete idrica di Avigliana, vedi Figura 4.16(a)

25-12-2018 Portata anomala dovuta ad un giorno festivo (Natale)

07-02-2019 e 21-02-2019 Anche se queste due date non sono effettivamente anomalie, l'algoritmo le riconosce tali, in quanto, per come è stata costruita la serie, essendo adiacenti ai valori anomali effettivi, determinano uno scarto $s[t]$ maggiore o uguale al valore fissato di h, vedi Figura 4.16(b).

In Figura 4.16 sono riportate alcune delle anomalie riscontrate in modo da esplicitarne le diverse tipologie. Nel grafico è rappresentato l'andamento delle portate dai due giorni precedenti ai due successivi rispetto all'evento anomalo individuato. Sul grafico sono cerchiati i valori corrispondenti alla serie oraria considerata, l'anomalia è segnalata in rosso mentre i medesimi orari nei giorni restanti sono cerchiati in verde. Si precisa inoltre che la linea nera tratteggiata rappresenta la retta di regressione lineare dedotta dai tre valori subito precedenti e subito successivi all'anomalia riscontrata e pertanto deve essere interpretata solo come un "aiuto" grafico e non come la reale posizione del valore considerato normale.



(a) *Anomalia riscontrata correttamente.* (b) *Anomalia del giorno precedente.*

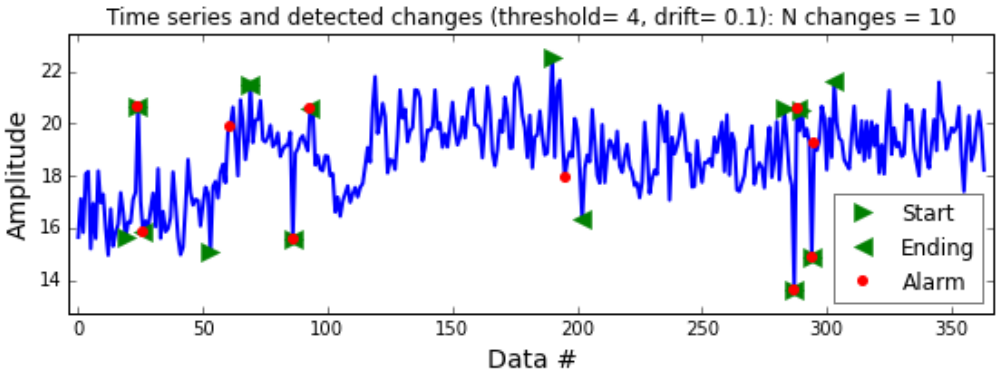
(c) *Errore di trascrizione.* (d) *Falso allarme.*

Figura 4.16: Tipologie di anomalie riscontrate.

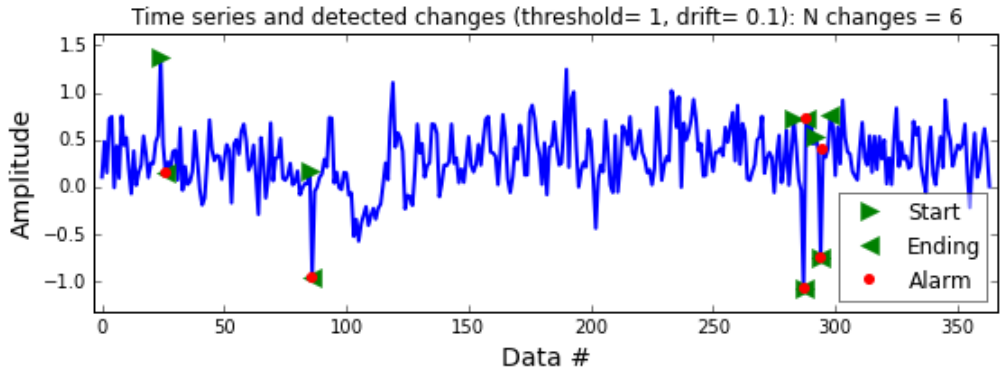
Serie delle 14:45

Analogamente a quanto visto per la serie delle 14:15, si è formata la serie costituita

da tutte le portate verificatesi alle ore 14:45 al variare dei giorni. In Figura 4.17 si riporta il CUSUM test effettuato sulle due serie, una delle quali costituita da portate normalizzate. Le anomalie riscontrate sono riportate in Tabella 4.7.



(a) *Cusum test serie standard.*



(b) *Cusum test su normalizzata.*

Figura 4.17: Cusum test sulle serie oraria delle 14:45.

Come visibile in tabella, l'anomalia nota del 13-02-2019 è stata riscontrata, inoltre, si sono riconosciuti come anomali gli eventi verificatisi alle seguenti date:

26-05-2018 Registrazione di un picco particolarmente alto.

28-05-2018, 03-08-2018 e 13-11-2018 Falsi allarme riscontrati a causa di valori precedenti relativamente alti o ad eventi anomali precedenti.

27-07-2018 Errore di trascrizione.

13-02-2019 e 20-02-2019 In tali date si verificano eventi anomali. Si è successivamente riscontrato che tali eventi sono conseguenze di interventi effettuati sulla rete idrica di Avigliana.

Anomalie della serie oraria 14:45					
$Q_{15min\ stand}$	$h = 4\ l/s$	$\nu = 0.1\ l/s$	$Q_{15min\ norm}$	$h = 1$	$\nu = 0.$
Data		Q_i [l/s]	Data	z_i	Q_i [l/s]
2018-05-26 14:45		20.66	2018-05-28 14:45	0.154	15.84
2018-05-28 14:45		15.84	2018-07-27 14:45	-0.958	15.60
2018-07-02 14:45		19.92	**2019-02-13 14:45**	**-1.073**	**13.62**
2018-07-27 14:45		15.60	2019-02-14 14:45	0.734	20.59
2018-08-03 14:45		20.62	2019-02-20 14:45	-0.741	14.90
2018-11-13 14:45		17.95	2019-02-21 14:45	0.397	19.29
2019-02-13 14:45		**13.62**			
2019-02-14 14:45		20.59			
2019-02-20 14:45		14.90			
2019-02-21 14:45		19.29			

Tabella 4.7: Anomalie della serie oraria delle 14:45 normalizzata e non.

14-02-2019 e 21-02-2019 Anche se in queste due date non ci sono effettivamente anomalie, l'algoritmo le riconosce tali, in quanto, per come è stata costruita la serie, essendo adiacenti ai valori anomali effettivi, determinano uno scarto $s[t]$ maggiore o uguale del valore fissato per h.

Osservazioni

Dalle analisi effettuate si sono dedotte le seguenti osservazioni:

- CUSUM test sui dati discretizzati giornalmente: le anomalie note non sono state riscontrate. Ciò è dovuto al fatto che tali anomalie durando pochi minuti e, non essendo prolungate nel tempo, determinano un peso minimo nella portata media giornaliera, non provocando di conseguenza un evento anomalo.

- CUSUM test sull'intera serie di dati discretizzati ogni 15 minuti: anche in questo caso le anomalie note non sono state ritrovate. Ciò è dovuto al fatto che, tali anomalie di portata determinano con i dati adiacenti una differenza minore rispetto alle variazioni che si verificano normalmente nell'arco della giornata. Dunque lavorando su questa serie di dati, risulta impossibile riscontrare tali anomalie senza produrre un altissimo tasso di falsi allarmi.

- CUSUM test sui giorni in cui le anomalie erano note: le anomalie ricercate sono state riscontrate ma sono stati riscontrati anche dei falsi allarmi dovuti alle variazioni che si verificano normalmente nell'arco della giornata (specialmente tra le 07:00 e le 08:00 e tra le 21:00 e le 22:00)

- CUSUM test su serie di dati aventi le portate al medesimo orario dell'anomalia nota: sono state riscontrate correttamente le anomalie note e se ne sono indicate di nuove.

Dai risultati ottenuti si è dedotto che l'algoritmo funziona correttamente se usato su serie opportune. Per riscontrare anomalie di breve durata è necessario sottoporgli serie formate dalla stessa grandezza allo stesso orario ma al variare dei giorni. Per riscontrare anomalie di grandi dimensioni o durature nel tempo, si possono sottoporre al test anche serie caratterizzate da un valore medio giornaliero della variabile di interesse.

In generale, dal confronto tra le serie standard (a) e le serie normalizzate (b), visibili nelle Figure 4.2, 4.3, 4.15, 4.17, si evince che la normalizzazione mensile effettuata rimuove dai dati le fluttuazioni stagionali derivanti dall'andamento climatico. Inoltre, studiando i risultati riportati nelle Tabelle 4.6 e 4.7 si riscontrano nelle seria standard 5 falsi allarmi non rinvenuti nelle serie normalizzate. Se ne deduce che tramite la normalizzazione si ottengono serie di dati più stabili nel corso dell'anno e di conseguenza risultati più affidabili. Si precisa che ulteriori miglioramenti per l'individuazione dei parametri verranno fatti nel paragrafo 4.3.

4.2 Centrale di Cavoretto

La Centrale di Cavoretto si trova appena oltre il confine sud del comune di Torino. In Figura 4.18 è riportata l'area geografica rifornita da tale impianto.

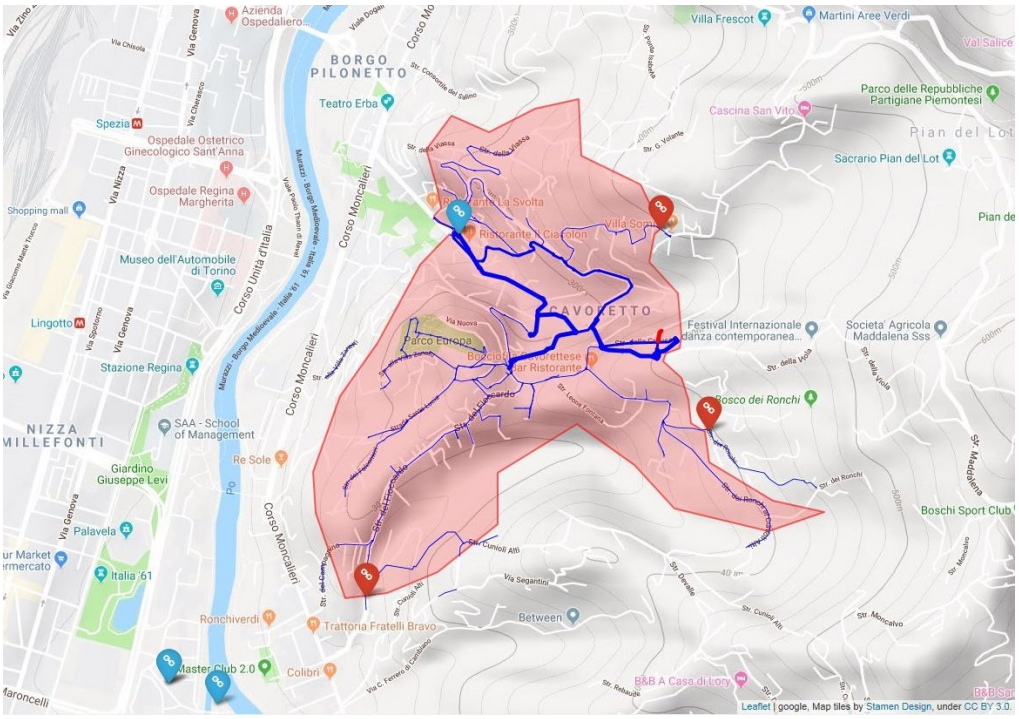


Figura 4.18: Area rifornita dalla centrale di Cavoretto.

Nella centrale è presente un impianto di sollevamento tramite cui è pompata una portata media di $8.80\,l/s$. La rete di distribuzione associata è lunga $13.90\,km$ e rifornisce circa 3800 abitanti rappresentanti l'utenza. Tale rete è caratterizzata da un rapporto di $Fughe/km/anno = 1.60$ ciò descrive un tasso di sostituzione delle tubazioni molto maggiore rispetto a quello riportato per il comune di Avigliana.

La Centrale di Cavoretto funziona come mostrato in Figura 4.19, dove QCA è la portata sollevata dalla centrale di pompaggio verso il serbatoio e HCA è il livello in serbatoio. Tale serbatoio è un serbatoio di estremità, situato dopo l'abitato che viene attraversato dalla condotta maestra, la quale è adibita al servizio di distribuzione lungo il percorso nel tratto di attraversamento del centro urbano. Esso oltre a contribuire ciclicamente all'approvvigionamento idrico dell'abitato assume una funzione di riserva nel caso in cui la richiesta idrica superasse la portata massima sollevata dalle pompe a monte.

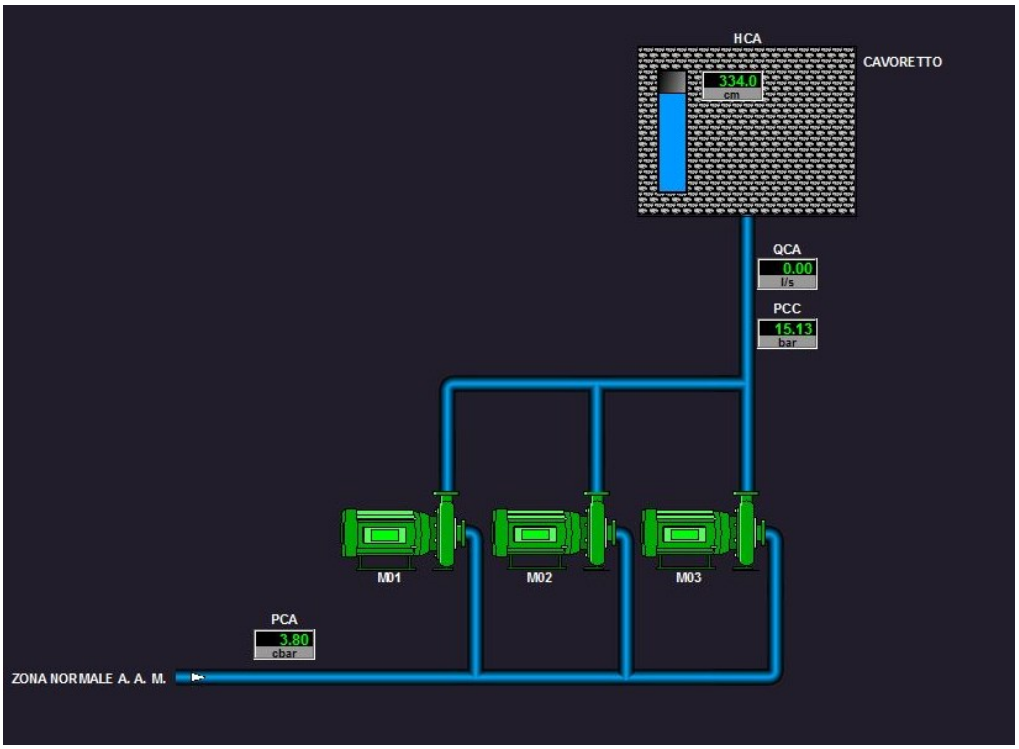
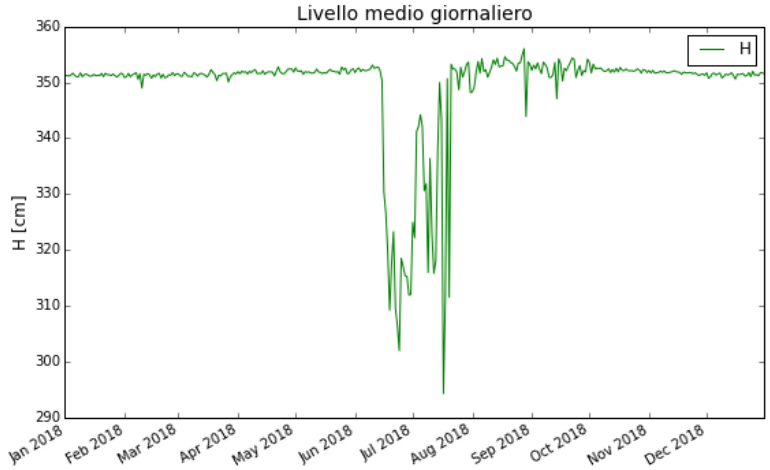


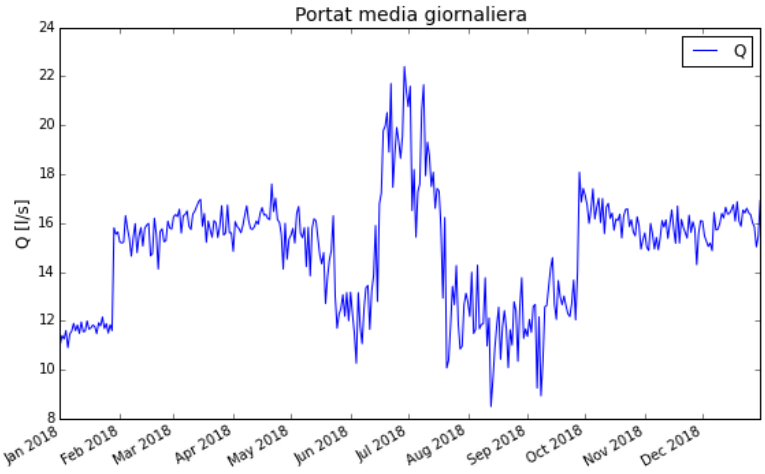
Figura 4.19: Schema della centrale di Cavoretto.

Sono stati forniti i dati riguardanti la QCA e i rispettivi HCA. Tali dati individuano una finestra temporale ca va dal 01-01-2018 00:00:00 fino al 01-01-2019 00:00:00 con frequenza tra 40 e 60 secondi. Si sono poi discretizzati i dati con una frequenza di 15 minuti associandone la variabile media corrispondente. Si riportano in Figura 4.20 i rispettivi i valori medi giornalieri di portata e livello nel corso dell'anno.

Focalizzando l'attenzione sull'andamento giornaliero medio del livello in serbatoio (Figura 4.20(a)), assodato che l'intenzione del gestore idrico sia quella di mantenere il livello medio giornaliero in serbatoio pressoché costante (a 351cm), è evidente la formazione di un evento anomalo. Tale evento è stata la conseguenza di una perdita idrica verificatasi il 15-06-2018 che, sommandosi alle domanda dell'utenza, ha scaturito una richiesta maggiore di quella massima offerta dall'impianto di sollevamento, innescando quindi l'intervento del serbatoio. Con il risanamento in data 20-07-2018 della condotta si sono poi ripristinate le normali condizioni di esercizio. Per quanto riguarda l'andamento giornaliero medio delle portate Figura 4.20(b)), esso assume forma più complessa, in quanto, oltre ad essere influenzato da eventi anomali, si relaziona alle variazioni di richiesta idrica da parte dell'utenza principalmente dovuti alle variazioni climatiche annuali, al fine di mantenere il



(a) *Livello medio giornaliero.*



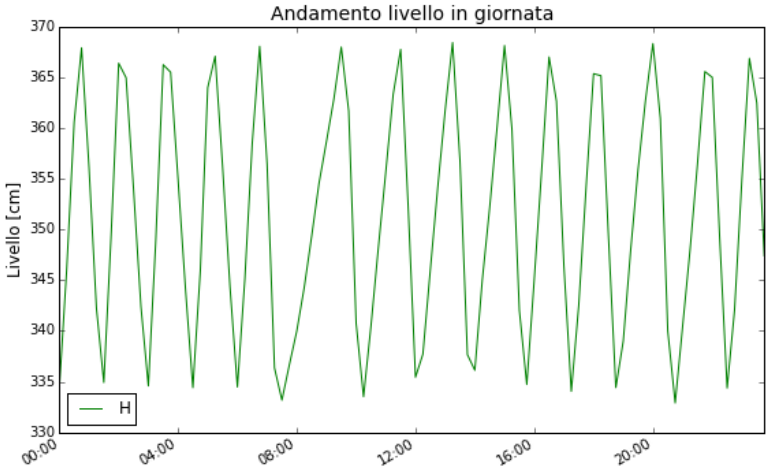
(b) *Portata media giornaliera.*

Figura 4.20: Andamento di livello e portata medi giornalieri nel corso dell'anno.

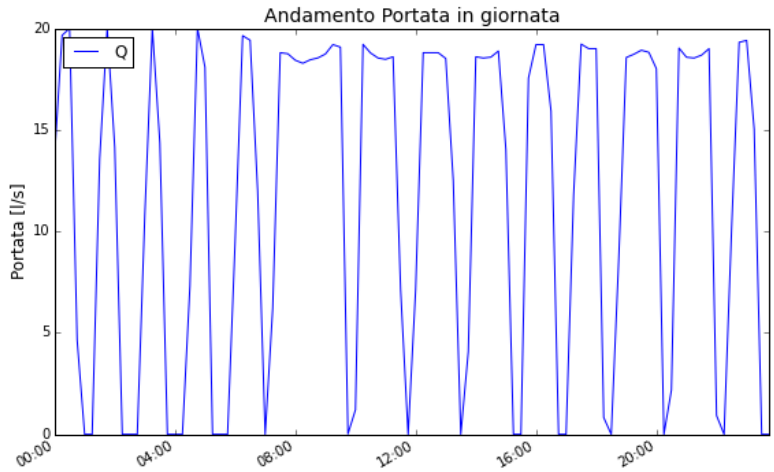
livello in serbatoio costante.

Viene inoltre illustrato in Figura 4.21 l'andamento giornaliero standard di livello in serbatoio e portata (andamenti corrispondenti al giorno 01-17-2018). Il sistema è programmato in modo tale che il serbatoio ciclicamente si riempia fino ad una quota di circa 370 cm, e che, una volta raggiunta quella quota, si svuoti fino a circa 335 cm. Per far si che ciò sia possibile, ci dovrà essere una portata in entrata, vedi Figura 4.21(b), che rispecchi tale andamento. Si noti che la non uniforme periodicità è dovuta al fatto che le utenze, specialmente al mattino, richiedono un maggior apporto d'acqua, il che determina un maggior tempo per il raggiungimento

della quota desiderata in serbatoio.



(a) *Livello durante la giornata.*



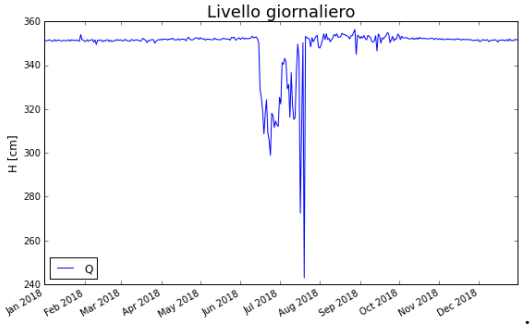
(b) *Portata durante la giornata.*

Figura 4.21: Andamento standard di livello e portata nel corso della giornata.

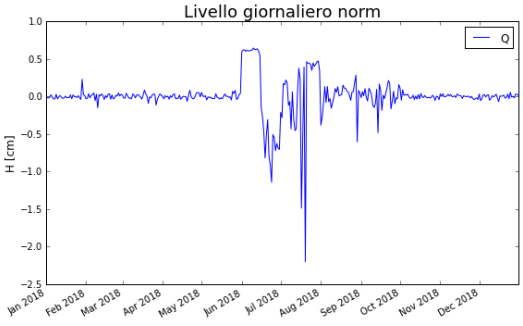
4.2.1 CUSUM test Cavoretto

Si è provato ad effettuare il rilevamento di anomalie sulle altezze in serbatoio della centrale di Cavoretto utilizzando l'algoritmo CUSUM. Per applicare tale algoritmo in Python si è fatto riferimento allo script [8], opportunamente modificato, riportato in appendice A.1.

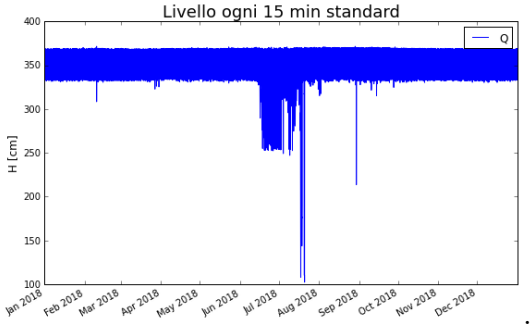
Si è sottoposto ad analisi il livello in serbatoio della centrale di Cavoretto dal 01-01-2018 al 31-12-2018. A differenza del precedente caso, esso non ha un andamento dipendente dall'orario, ma si riempie e si svuota una volta raggiunto un determinato valore di soglia. Seguendo la medesima procedura procedura utilizza nell'analisi della Centrale di Avigliana e facendo riferimento all'equazione 4.1, si sono normalizzati i dati. In Figura 4.22 si riportano, per confronto gli andamenti rappresentanti il livello medio giornaliero normalizzato e non ed il livello avente frequenza di 15 minuti normalizzato e non.



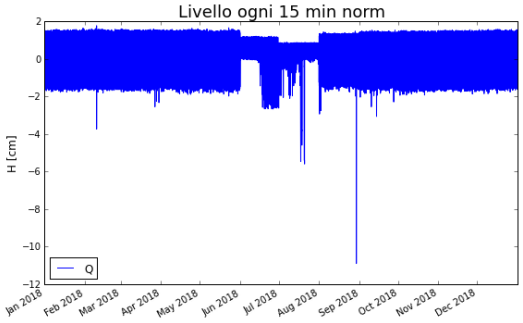
(a) *Andamento del livello medio giornaliero.*



(b) *Andamento del livello medio giornaliero normalizzato.*



(c) *Andamento del livello con frequenza di 15 minuti.*

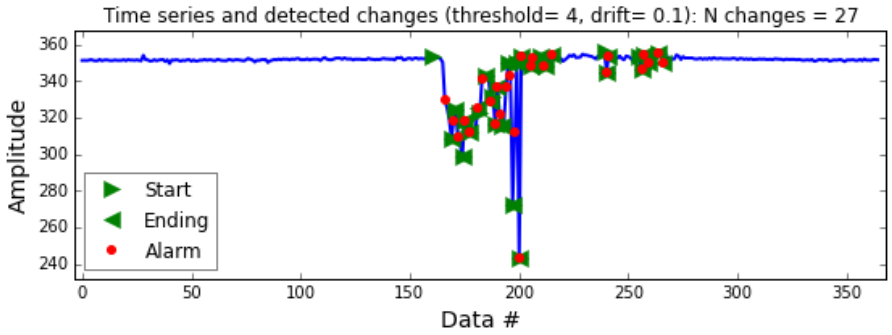


(d) *Andamento del livello con frequenza di 15 minuti normalizzato.*

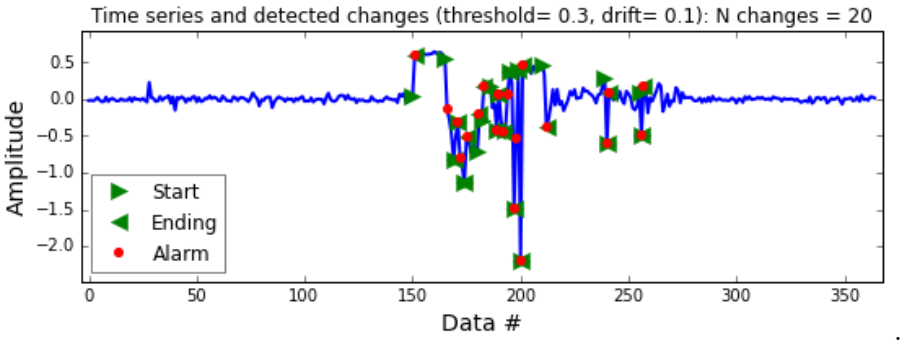
Figura 4.22: Confronto tra andamenti standard e normalizzati.

Facendo riferimento alla Figura 4.20(a) si nota subito un grande evento anomalo che va dal 15-06-2018 al 20-07-2018. A causa di tale evento i dati dei mesi di Giugno e Luglio risultano corrotti e di conseguenza, come visibile in Figura 4.22(b) e (d), effettuare un processo di normalizzazione porta alla costruzione di serie che non rispecchiano il reale comportamento del serbatoio in questione. Per ulteriore conferma si sottopongono le serie ottenute al CUSUM test. Per questa tipologia di serie, in cui in condizione di esercizio l'andamento della variabile d'analisi rimane

sempre costante, individuare il valore dei parametri risulta molto semplice. Posto un valore di drift molto piccolo, $\nu = 0.1$, in modo da tener conto delle piccole variazioni causate da una differente velocità di riempimento/svuotamento del serbatoio, il valore di soglia h sarà posto leggermente maggiore rispetto al regolare scarto tra i dati $s[t]$ dovuto alle oscillazioni del livello in serbatoio. Come visibile in Figura 4.21(a), in condizioni di esercizio il serbatoio inizia a riempirsi a 335 cm per poi svuotarsi una volta raggiunti circa i 370 cm producendo quindi tra i dati uno scarto regolare di 35cm. Per limitare l'individuazione di falsi allarmi, facendo riferimento alla serie standard avente frequenza ogni 15 minuti, si è posto un valore di soglia $h = 40\ cm$. Seguendo lo stesso ragionamento per la serie normalizzata si è dedotto un $h = 4$. Le serie giornaliere sono caratterizzate da un comportamento ancora più semplice, essendo la variazione di livello sempre costante, la media giornaliera risulta fissa 352 cm, ne risulta che per l'individuazione di anomalie basta porre un valore di soglia molto basso, si è scelto $h = 4\ cm$ per la serie standard e $h = 0.3$ per la serie normalizzata. Si riportano in Figura 4.23 e 4.24 gli output ottenuti.



(a) *Livelli giornalieri standard.*



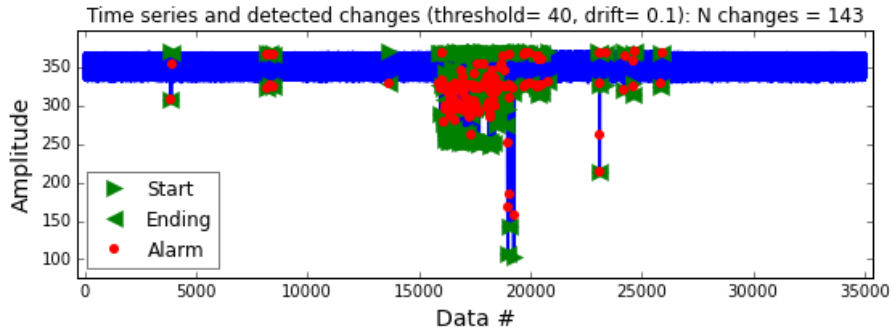
(b) *Livelli giornalieri normalizzati.*

Figura 4.23: Comparazione tra andamento livelli giornalieri normalizzati e non.

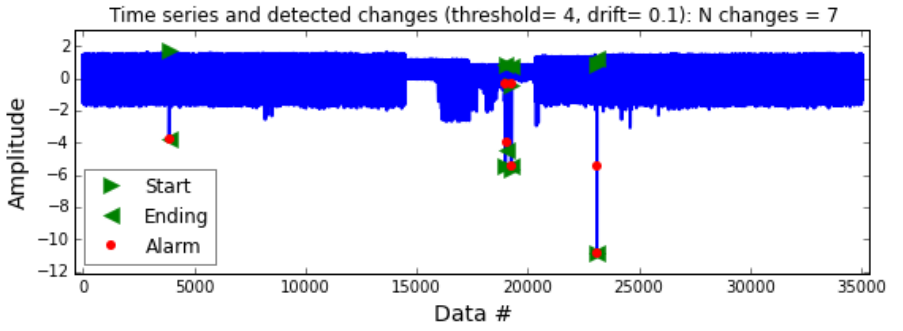
Dalle figure si nota come, in questo caso, effettuare il processo di normalizzazione

sia stato controproducente. Infatti entrambe le serie standard (a) e (c) riscontrano anomalie sia il 15-06-2018 sia il 20-07-2018, corrispondenti ad inizio e fine del grande evento anomalo. Mentre, le serie normalizzate o non segnalano tale evento (come in Figura 4.24(b) in cui la normalizzazione produce un restringimento dell'ampiezza dei dati a partire dall'ascissa *Data* # 1500 corrispondente al primo Luglio) oppure lo riscontrano ma in date differenti (come Figura 4.23(b) in cui la prima anomalia avviene in *Data* # 150, sempre corrispondente al primo giorno di Luglio).

Si precise inoltre che, anche se in Figura 4.24(a) si riscontrano 143 anomalie, alcune di esse sono falsi allarmi dovuti alla non perfetta calibrazione dei parametri di soglia e di drift (su di essi si discuterà nel paragrafo 4.3.2) ma per la maggior parte si tratta di singole anomalie costituenti il grande evento anomalo già citato e pertanto l'effettivo numero di eventi anomali deve valutarsi come significativamente ridotto.



(a) *Livelli standard.*

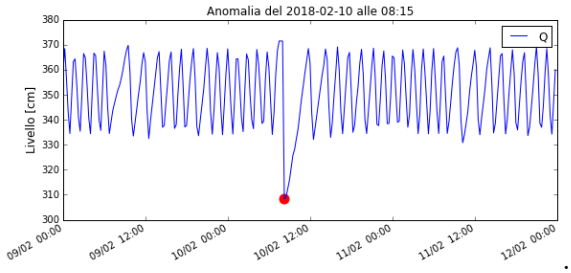


(b) *Livelli normalizzati.*

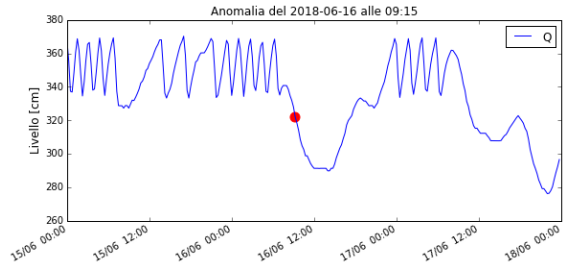
Figura 4.24: Comparazione tra andamento portate normalizzate e non.

Visti i risultati, escludendo le istanze verificatesi nel corso del grande evento anomalo (in quanto risulterebbero tutte classificate come anomale), si è deciso di concentrarsi sulle serie non normalizzate ed in particolare su quella avente frequenza

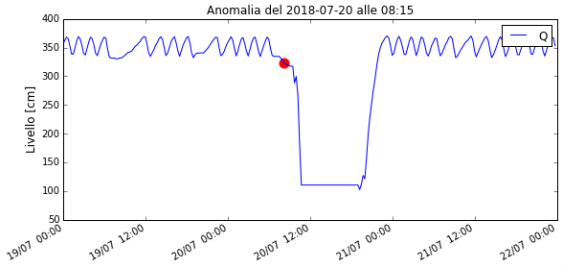
di 15 minuti perché le anomalie di breve durata, avendo poco peso sul livello medio giornaliero, non verrebbero riscontrate. Da tale serie sono estratte e riportate in Figura 4.25 le anomalie corrispondenti all'inizio e alla fine del grande evento oltre ad una precedente ed una successiva.



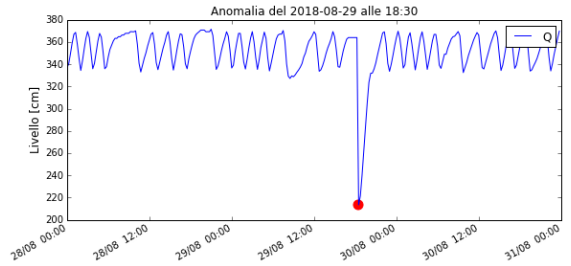
(a) *Anomalie pre-evento.*



(b) *Andamento inizio evento.*



(c) *Andamento fine evento.*



(d) *Andamento post-evento.*

Figura 4.25: Esempi anomalie del livello in serbatoio di Cavoretto.

Osservazioni

Dalle analisi effettuate si sono dedotte le seguenti osservazioni:

- CUSUM test sui dati discretizzati giornalmente: il grande evento anomalo è stato riscontrato. Infatti, essendo prolungato nel tempo, determina un cambiamento anche nell'andamento del livello medio giornaliero. Al contrario le anomalie di breve durata, determinando un piccolo peso, non vengono riscontrate.
- CUSUM test sui dati normalizzati discretizzati giornalmente: essendoci un grande evento anomalo che sporca il normale andamento della serie, la normalizzazione, per come è definita, vedi paragrafo 4.1.1, porta alla formazione di una serie che, almeno per i mesi in cui la maggior parte dei dati risultano corrotti (Giugno e Luglio), non rispecchia il reale comportamento del livello in serbatoio. Al di fuori di questa finestra temporale, le anomalie sono ritrovate correttamente con l'esclusione di quelle che avendo un piccolo peso non comportano un cambiamento del livello medio giornaliero
- CUSUM test sull'intera serie di dati discretizzati ogni 15 minuti: anche in questo caso l'evento anomalo è stato riconosciuto. Inoltre anche piccoli scostamenti dal normale andamento in esercizio sono riconosciuti come anomali. Per questa tipologia di dati tale serie risulta essere la più adatta al riconoscimento di anomalie.
- CUSUM test sull'intera serie di dati normalizzati discretizzati ogni 15 minuti: come nel secondo caso, la normalizzazione porta alla formazione di una serie avente i dati nei mesi di Giugno e Luglio non conformi alla realtà. Da ciò scaturisce anche un mancato riconoscimento del grande evento anomalo. Al di fuori di tale finestra temporale l'analisi funziona correttamente.

Dal caso in esame si evince che in presenza di anomalie prolungate nel tempo, i dati risultano corrotti e dunque la normalizzazione risulta essere una tecnica controproducente. Per ovviare a tale inconveniente, noto l'evento anomalo, si potrebbe studiare la restante parte del database ottenendo ottimi risultati. Alternativamente, per questa tipologia di serie, in cui in condizione di esercizio l'andamento della variabile d'analisi rimane sempre costante, l'analisi può essere svolta direttamente sulle serie standard. Per riscontrare anomalie di breve durata è necessario sottoporre al CUSUM test serie con frequenza di 15 minuti, per anomalie molto grandi o durature nel tempo, sono anche adatte serie caratterizzate da un valore medio giornaliero della variabile di interesse.

4.3 Automatizzazione CUSUM

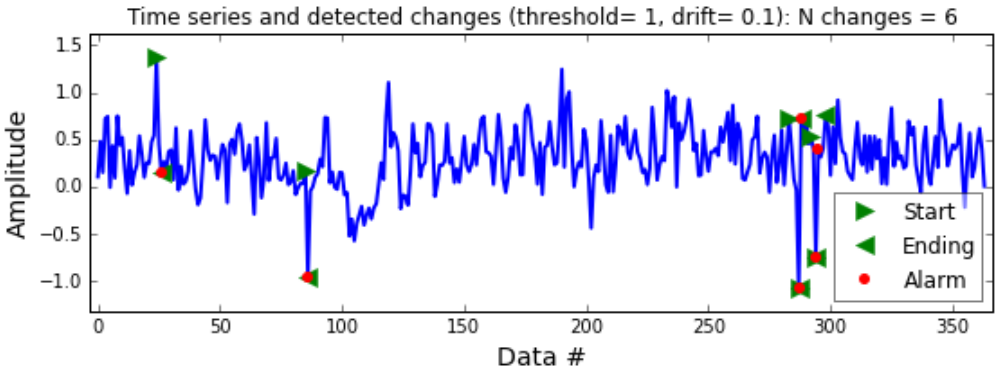
Con il termine automatizzazione si intende l'insieme degli accorgimenti e/o operazioni che permettono la riduzione o eliminazione dell'intervento di operatori nell'effettuazione di un singolo movimento o operazione.

Facendo riferimento al CUSUM test, la automatizzazione risulta di fondamentale importanza, perché, come evidenziato negli scorsi paragrafi, i parametri di riferimento sono intrinseci al sistema e dunque, l'individuazione del loro valore ottimale non solo varia a seconda dell'impianto a cui si fa riferimento ma anche a seconda della variabile d'analisi. Essendo l'obiettivo di questa tesi la formazione di un algoritmo in grado di rilevare comportamenti anomali dei componenti connessi al sistema di telecontrollo della SMAT s.p.a. e, facendo presente che al telecontrollo sono connessi un sostanzioso numero di impianti idrici, ognuno dei quali presenta più variabili da sottoporre ad analisi, un'operazione manuale di calibrazione dei parametri sarebbe un compito più che gravoso. Inoltre, si deve tener conto che le variabili in questione hanno comportamenti dinamici, quindi le condizioni di esercizio possono subire variazioni nel corso del tempo comportando dunque una nuova calibrazione. Infine, si precisa che una variabile sottoposta ad analisi può avere comportamenti differenti in funzione sia del periodo dell'anno, sia dell'ora nel giorno e dunque il valore del parametro che si adatta al meglio alla serie in questione potrebbe non essere costante.

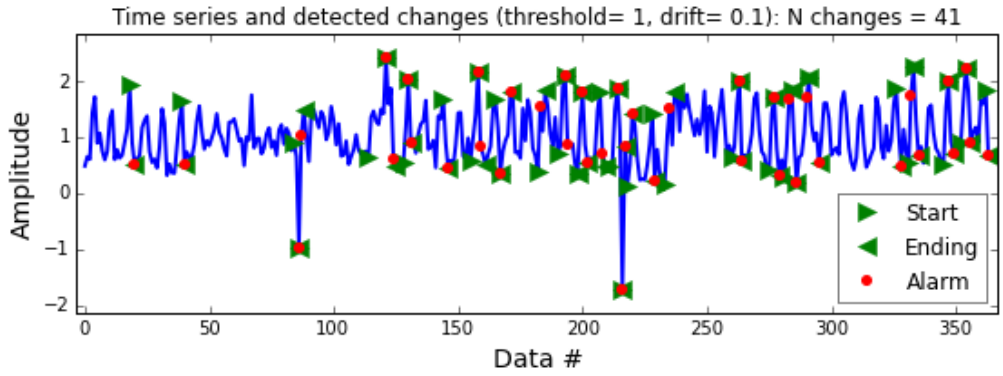
Si vuole dunque determinare un processo automatico che, nota la serie d'analisi, individui il valore ottimale dei parametri, per poi utilizzarli durante l'applicazione del CUSUM test. Per automatizzare i parametri si è deciso di partire facendo riferimento ai dati inerenti la centrale di Avigliana e, una volta noto il processo di automatizzazione, di verificare se esso sia adatto anche per la centrale di Cavoretto.

4.3.1 Automatizzazione Avigliana

Visti gli ottimi risultati ottenuti dall'analisi delle serie delle portate in uscita dalla centrale di Avigliana, rispettivamente negli orari 14:15 e 14:45, riassunti nelle sezione normalizzata delle Tabelle 4.6 e 4.7 a pag.50 e 54, in cui le date trovate rappresentano effettivamente giorni e orari in cui si è registrato il passaggio di una portata anomala, si è deciso di estendere tale analisi a tutti gli orari della giornata, con una frequenza di 15 minuti. Si è notato che i valori di soglia h=1 e drift $\nu = 0.1$ ritenuti ottimali per le serie analizzate risultano inaffidabili in altri intervalli orari. A titolo esplicativo, in Figura 4.26, si riporta l'output del CUSUM test delle serie temporali formate dalle portate normalizzate alle 14:45 ed alle 11:00.



(a) *Cusum test serie ore 14:45*



(b) *Cusum test serie ore 11:00.*

Figura 4.26: Utilizzo degli stessi parametri in due serie differenti.

Come mostrato in figura, se tali valori di h e v si adattano bene alla serie temporale delle 14:45, riscontrando 6 anomalie, rappresentate da pallini rossi che si collocano sui picchi della serie, per la serie delle ore 11:00 (orario in cui si verifica una richiesta idrica da parte dell'utenza con maggior varianza rispetto alla precedente), si riscontrano 41 anomalie ma la maggior parte di esse risultano falsi allarmi. Risulta quindi necessario individuare un metodo che automatizzi il valore di h e v, in modo da adattarlo alla specifica serie. Al fine di evidenziare le differenze tra le due serie in esame si fa riferimento ai box plot.

Il box plot è una rappresentazione grafica utilizzata per descrivere la distribuzione di un campione tramite semplici indici di dispersione e di posizione. Tali indici, mostrati in Figura 4.27 sono descritti in seguito [25]:

La linea interna alla scatola rappresenta la mediana della distribuzione.

Le linee estreme della scatola rappresentano il primo ed il terzo quartile.

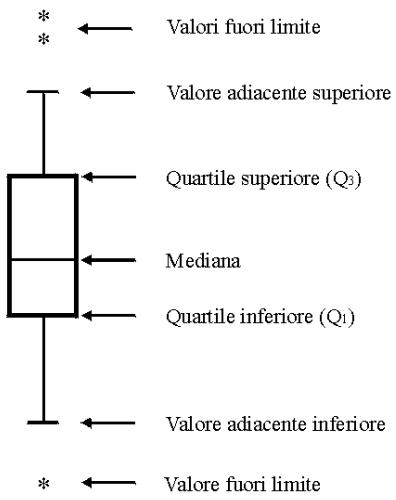


Figura 4.27: Schema indici boxplot [25].

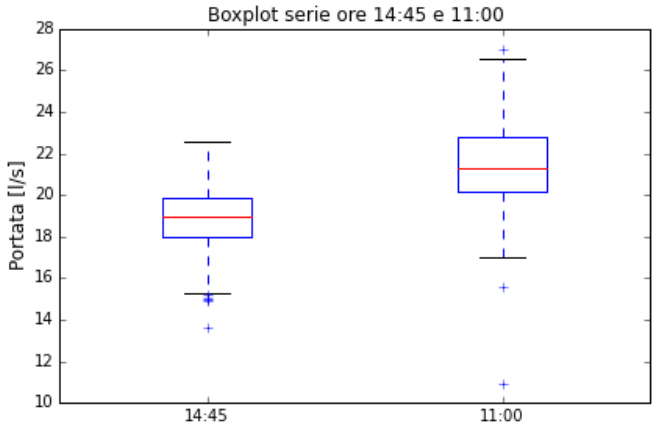
La distanza tra il terzo ed il primo quartile, *distanza interquartilica*, è una misura della dispersione della distribuzione. Il 50% delle osservazioni si trovano comprese tra questi due valori. Se l'intervallo interquartilico è piccolo, tale metà delle osservazioni si trova fortemente concentrata intorno alla mediana; all'aumentare della distanza interquartilica aumenta la dispersione del 50% delle osservazioni centrali intorno alla mediana.

Le distanze tra ciascun quartile e la mediana forniscono informazioni relativamente alla forma della distribuzione. Se una distanza è diversa dall'altra allora la distribuzione è asimmetrica.

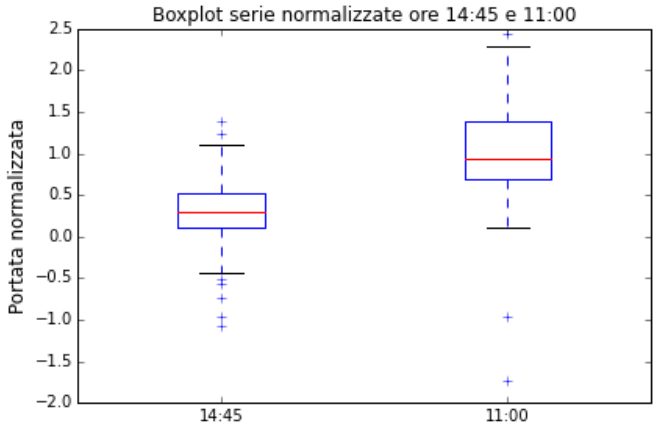
Le linee che si allungano dai bordi della scatola (baffi o whiskers) individuano gli intervalli in cui sono posizionati i valori rispettivamente minori di Q1 e maggiori di Q3; i punti estremi dei baffi evidenziano i valori adiacenti. Se si indica con $r = Q_3 - Q_1$ la differenza interquartilica, il valore adiacente inferiore (VAI) è il valore più piccolo tra le osservazioni che risulta maggiore o uguale a $Q_1 - 1{,}5r$. Il valore adiacente superiore (VAS), invece, è il valore più grande tra le osservazioni che risulta minore o uguale a $Q_3 + 1{,}5r$. Pertanto se gli estremi della distribuzione sono contenuti tra VAI e VAS essi coincideranno con gli estremi dei baffi, altrimenti come estremi verranno usati i valori $Q_1 - 1{,}5r$ e $Q_3 + 1{,}5r$. I valori adiacenti inferiore e superiore forniscono informazioni sulla dispersione e sulla forma della distribuzione ed anche sulle code della distribuzione

I valori esterni a questi limiti (in genere valori anomali), vengono segnalati individualmente nel box-plot per meglio evidenziarne la presenza e la posizione. Questi valori infatti costituiscono una "anomalia" rispetto alla maggior parte dei valori osservati e pertanto è necessario identificarli per poterne analizzare le caratteristiche e le eventuali cause che li hanno determinati. Essi forniscono informazioni ulteriori sulla dispersione e sulla forma della distribuzione. Quando i valori adiacenti, superiore e inferiore, coincidono con gli estremi della distribuzione non comparirà alcun valore fuori limite.

Dunque per avere una visione di insieme sulla distribuzione delle serie, si riportano in Figura 4.28 , i box plot delle serie orarie delle 14:45 e delle 11:00 normalizzate e non.



(a) *Boxplot serie ore 14:45 e 11:00.*



(b) *Boxplot serie normalizzate ore 14:15 e 11:00*

Figura 4.28: Confronto tra box-plot.

Si può notare che nonostante la normalizzazione porti ad assumere differentemente qualche valore come punto interno o esterno, in generale non modifica la distribuzione della serie. Dal confronto tra i due orari, si evince che, nonostante le due serie siano estrapolate dallo stesso dataframe, esse presentano valori statistici differenti. Tali valori sono riportati in Tabella 4.8.

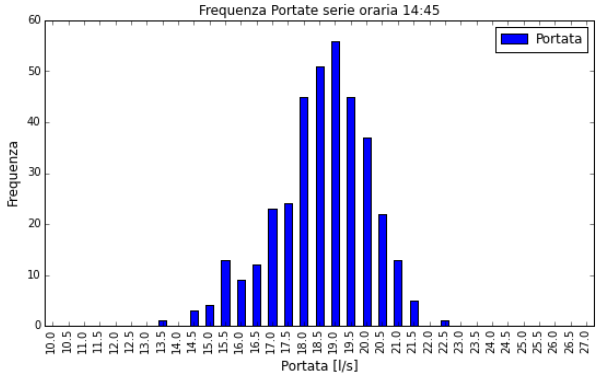
Indici stat.	Serie ore 14:45		Serie ore 11:00	
	Portata [l/s]	Portata norm.	Portata [l/s]	Portata norm
mean	18.81	0.309	21.43	1.022
std	1.480	0.316	2.134	0.491
min	13.62	-1.073	10.94	-1.732
25%	18.01	0.108	20.17	0.682
50%	18.97	0.299	21.31	0.935
75%	19.85	0.521	22.83	1.384
max	22.55	1.379	27.00	2.441
VAI	15.28	0.433	17.02	0.109
VAS	22.55	1.106	26.55	2.286

Tabella 4.8: Indici statistici delle quattro serie considerate

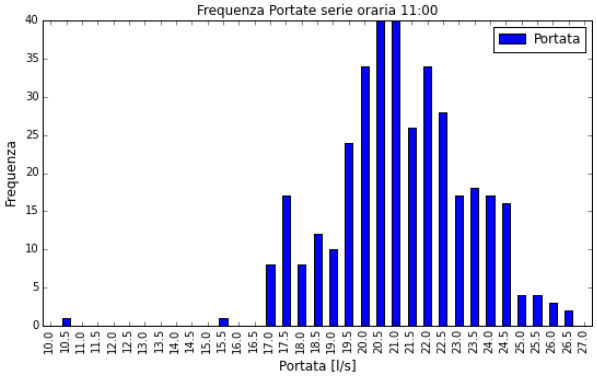
Essendo il CUSUM test, per come è definito, vedi par. 2.3, funzione della variabile di interesse, diviene evidente che il valore di soglia h non possa essere il medesimo per ogni serie ma debba dipendere dalla distribuzione che la variabile stessa assume. Per tale motivazione si riporta in Figura 4.29 la distribuzione in frequenza delle due serie. Osservando la figura, si nota che entrambe le serie assumono distribuzione a campana in cui le anomalie si dispongono agli estremi.

Assodato ciò, si è deciso di ricavare il parametro di soglia h come differenza tra due percentili indicati rispettivamente con q_1 e q_3 (vedi Figura 4.30). Il percentile o quantile indica un valore sotto al quale ricade una percentuale di elementi appartenenti alla serie sotto osservazione, dunque esso assume valori differenti a seconda della serie sottoposta ad analisi. Ponendo quindi $h = q_3 - q_1$ si determina un valore di soglia dipendente dalla distribuzione della serie stessa.

Posto un valore di drift molto piccolo ($\nu = 0.1$), ma comunque maggiore di zero in modo da ridurre l'effetto dei dati passati e di conseguenza anche il numero di falsi allarmi rilevati, devono essere ricercate le percentuali dei quantili q_1 e q_3 determinanti un valore di soglia ottimale. Assunto che porre $q_1 = 0\%$ e $q_3 = 100\%$ non avrebbe alcun senso in quanto si avrebbe un h ricavata come lo scarto tra il valore più grande e il più piccolo della serie, ed essendo tale valore sempre maggiore della differenza tra due dati successivi $s[t]$, non sarebbe riscontrata alcuna anomalia. Si è dunque effettuato un CUSUM test di prova utilizzando percentili determinanti



(a) *Distribuzione in frequenza della serie delle 14:45*



(b) *Distribuzione in frequenza della serie delle 11:00*

Figura 4.29: Confronto delle due distribuzioni in frequenza.

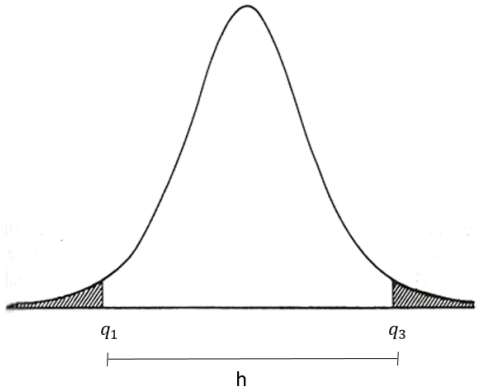


Figura 4.30: Scelta del valore di soglia.

un valore di soglia minore. Facendo riferimento alla serie oraria normalizzata delle ore 11:00, si è posto $q_1 = 10\%$ e $q_3 = 90\%$ ottenendo h = 1.19. Come visibile in Figura 4.31, il valore di h trovato risulta troppo piccolo e infatti, anche se sono state riscontrate 21 anomalie, la maggior parte di esse rappresenta dei falsi allarmi.

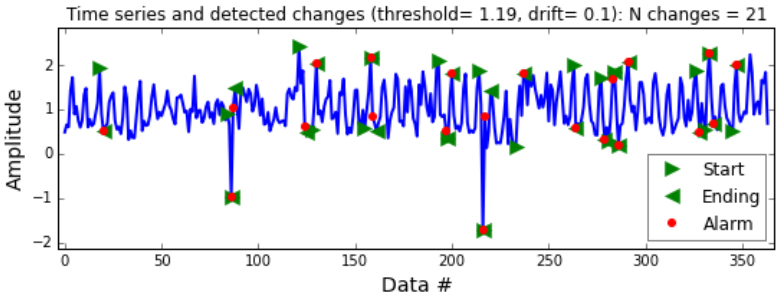


Figura 4.31: Serie ore 11:00 con h troppo basso.

Partendo dai valori di q_1 e q_3 stabiliti in precedenza, al fine di trovare quelli ottimali, si è deciso di iterarli diminuendone il primo e aumentandone il secondo. Dopo diversi tentativi, è stato riscontrato che i valori dei quantili che determinano un valore di soglia che meglio si adatta alle serie analizzate è fornito da $q_1 = 5\%$ e $q_3 = 95\%$. Si riporta in Figura 4.32 la precedente serie, con il nuovo valore di soglia h. Per quanto riguarda la serie delle ore 14:45 si ottiene h=0.95 e di conseguenza l'output sarà quasi identico a quello presente in Figura 4.17(b) a pag. 53.

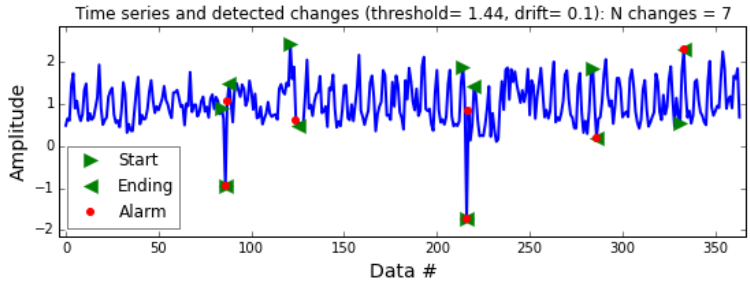
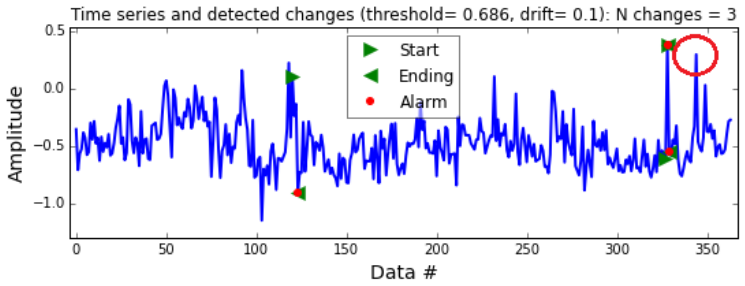


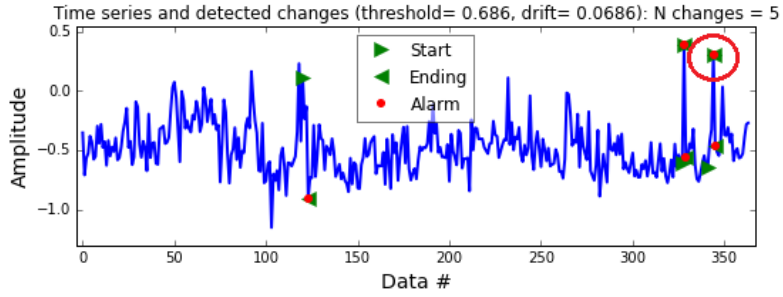
Figura 4.32: Serie ore 11:00 con valore h automatizzato.

Grazie a tale automatizzazione, si ottengono valori di soglia conformi alla serie analizzate, producendo senza dubbio risultati migliori rispetto ad una calibrazione manuale. In alcuni casi, come per la serie delle 23:00, avere un valore di drift fisso, posto uguale a 0.1, risulta troppo cautelativo, specialmente nel caso in cui h è molto piccolo, portando dunque alla non rilevazione di alcune piccole anomalie. Per tale motivo si è deciso di automatizzare anche questo valore ponendolo uguale al 10% di h. In Figura 4.33 sono riportati i CUSUM test della serie delle 23:00 con h =

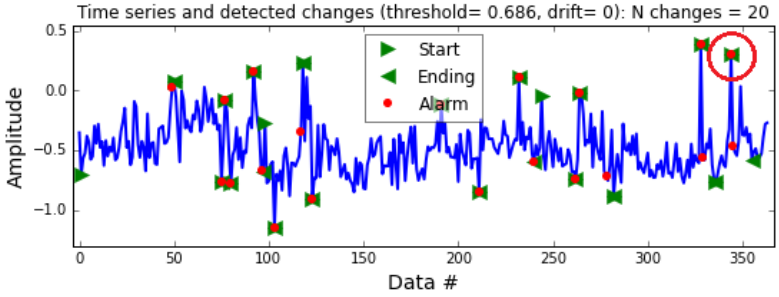
0.069 e il valore di drift posto rispettivamente prima $\nu = 0.1$, poi $\nu = 10\%h$ ed infine uguale a zero. Dalla figura si evince come una netta anomalia, riportata in Figura 4.34, precedentemente non riscontrata, vedi Figura 4.33(a), sia ora rilevata grazie al nuovo valore di drift, vedi Figura 4.33(b) e (c). Si noti infine che porre un valore drift uguale a zero, Figura 4.33(c), risulterebbe troppo cautelativo e causerebbe un forte aumento dei falsi allarmi causati dall'effetto non più smorzato dei dati passati.



(a) *Serie ore 23:00 con valore h automatizzato.*



(b) *Serie ore 23:00 con valore h e ν automatizzati.*



(c) *Serie ore 23:00 con* $\nu = 0$.

Figura 4.33: Confronto tra ν fisso ed automatizzato.

Estendendo l'automatizzazione dei parametri alle $4 \times 24 = 96$ serie, sono stati trovati 710 dati anomali, cioè una media di $710 \div 24 \div 4 = 7.4$ dati anomali per ogni serie. Considerata poi una finestra temporale a partire dal 1 Gennaio fino

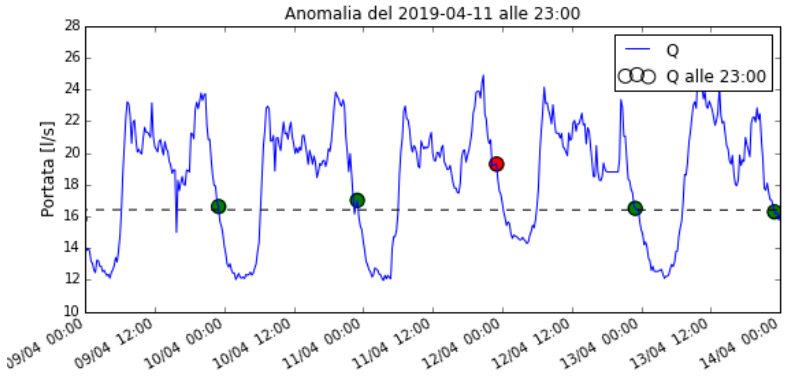


Figura 4.34: Anomalia del 11/04/2019 alle 23:00.

al 30 Aprile 2019 (ultimo giorno del database fornito) si sono valutate singolarmente le 155 anomalie riscontrate per attribuire una certa affidabilità al metodo proposto. Nonostante da un prima impressione degli output del CUSUM test i parametri automatizzati sembrassero adattarsi molto bene alle serie sottoposte ad analisi, estendendo tale automatizzazione a tutte le serie orarie e controllando singolarmente le 155 anomalie trovate nella finestra temporale sottoposta a controllo, si è notato che non tutte le segnalazioni rappresentano degli effettivi eventi anomali. In particolare durante le ore notturne, vengono segnalati come anomali dei piccoli picchi di portata. Nei sistemi di sollevamento, dove sono presenti più pompe idrauliche, per mantenerne un corretto funzionamento, si può verificare, durante le ore notturne, l'attivazione di una pompa e lo spegnimento di un'altra. I picchi rilevati sono quindi causati dalla messa in funzione di una nuova pompa prima dell'arresto di quella già attiva. Questi casi sono comunque segnalati dal CUSUM test come anomali, vedi Figura 4.35.

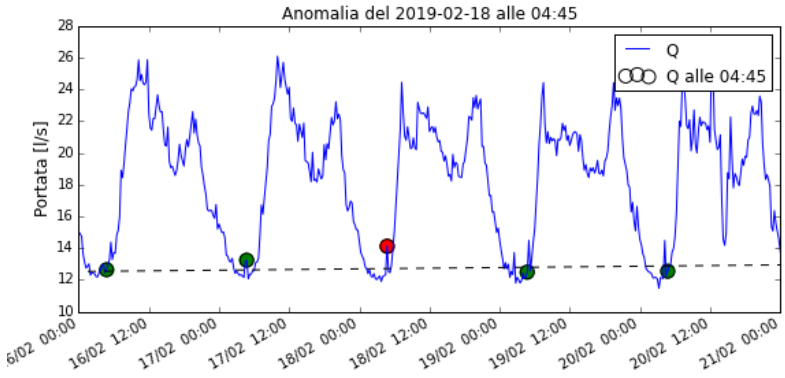


Figura 4.35: Riscontro di una falsa anomalia notturna.

Per evitare il riscontro della maggior parte di questi eventi inusuali ma non considerati effettivamente anomali e per ridurre i numeri dei falsi allarmi riscontrati, si è continuato a manipolare i parametri q_1, q_3 e la dipendenza lineare tra h e ν in modo da aumentare l'affidabilità dell'algoritmo proposto. Dopo numerosi tentativi e rispettivi controlli si è valutato che il valore migliore dei parametri di soglia e drift si ottenessero ponendo $q_1 = 4\%$, $q_3 = 97\%$ e $\nu = 5\%$h. Con tali parametri, sono stati trovati 550 dati anomali, cioè una media di $550 \div 24 \div 4 = 5.73$ dati anomali per ogni serie. Tale dato che rispetto al totale dei dati ($365 \times 24 \times 4 = 35040$) ne rappresenta 1.57%, non risulta dunque eccessivo anche perché la medesima anomalia spesso dura più di 15 minuti, dunque gran parte dei dati anomali sono raggruppati in un numero di anomalie di certo minore. Inoltre per come è definito il CUSUM test, se è presente una grossa anomalia, il test riporta come anomalo anche il dato immediatamente successivo nonostante esso possa essere definito normale, vedi Figura 4.14(b) a pag.49. Infatti, tale riscontro va a sottolineare l'evento anomalo precedente e non è considerato come falso allarme in quanto l'operatore designato è già consapevole dell'anomalia precedentemente verificatasi.

Facendo riferimento ai 4 mesi costituenti la finestra temporale indicata in precedenza, sono stati riscontrati 103 eventi anomali. Raggruppando insieme i dati costituenti un'anomalia maggiore e non tenendo conto dei dati indicanti un'anomalia già riscontrata, il numero di anomalie riscontrate si è ridotto a 34. Tali eventi anomali sono riportati in Tabella 4.9, nella quale oltre ad essere indicato il giorno e l'orario in cui si verifica, viene anche riportato il tipo di anomalia riconosciuta. In tabella è indicata con "v" un'anomalia riconosciuta immediatamente, con "i" una riconosciuta il giorno successivo, infine con t se l'anomalia è dovuta ad un errore di trascrizione. Le anomalie presenti in tabella sono inoltre rappresentate in Allegato B.1 sui grafici riportanti l'andamento delle portate dai due giorni precedenti ai due successivi rispetto l'evento anomalo individuato. Sul grafico, sono cerchiati i valori corrispondenti alla serie oraria considerata, l'anomalia è segnalata in rosso mentre i medesimi orari nei giorni restanti sono cerchiati in verde. Si precisa inoltre che la linea nera tratteggiata rappresenta la retta di regressione lineare dedotta dai tre valori subito precedenti e subito successivi all'anomalia riscontrata e pertanto deve essere interpretata solo come un "aiuto" grafico e non come la reale posizione del valore considerato normale.

Non avendo a disposizione un database contenente tutti gli eventi anomali occorsi nella centrale di Avigliana non è possibile sapere se esistono eventi anomali non riscontrati con tale metodo, di conseguenza risulta difficile stabilirne un indice di affidabilità. D'altro canto, si è riscontrato che tutti gli interventi noti, effettuati sulla centrale di Avigliana, modificanti le condizioni d'esercizio di impianto, sono

Parametri: $q_1 = 4\%$		$q_3 = 97\%$		$h = q_3 - q_1$	$\nu = 6\%h$
Anno	Mese	Giorno	Ora in.	Ora fin.	Tipo
2019	Gen	1	01:30	08:45	v
		12	12:30	12:45	v
		14	17:15	17:30	v
		15	04:30	04:45	v
		15	07:00	07:15	v
		17	04:45	05:00	v
		20	10:00	10:15	v
	Feb	1	04:30	04:45	i
		5	08:30	08:45	v
		6	14:15	14:30	v
		9	14:15	15:30	v
		13	14:45	15:00	v
		20	12:15	12:45	v
		20	14:15	15:00	v
		21	04:15	04:30	i
		24	11:30	11:45	v
		24	14:45	15:00	v
	Mar	10	16:30	16:45	v
		26	22:45	01:30	t
		30	04:15	04:45	v
		30	08:45	09:45	v
		31	11:00	11:15	v
	Apr	9	15:45	16:00	v
		11	23:00	0:00	v
		12	01:15	05:00	v
		12	08:45	09:00	v
		12	19:15	20:15	v
		14	10:30	10:45	v
		14	23:30	23:45	v
		21	13:00	13:45	v
		23	08:45	09:15	v
		24	13:00	13:15	v
		25	09:15	09:30	v
		29	09:15	09:30	v

Tabella 4.9: Anomalie Centrale Avigliana nel 2019

contenuti tra quelli riportati in tabella. Inoltre, non avendo riscontrato nessun falso allarme si potrebbe pensare a tale metodo come infallibile. In realtà per 3 volte un'anomalia già riconosciuta, ha portato alla rilevazione di un falso allarme 2 giorni dopo l'effettivo riscontro, vedi Figura 4.36, mentre per 3 volte ciò è successo 3 giorni dopo. Dunque se l'operatore non è a conoscenza della precedente anomalia o se non ne individua la correlazione, tali segnalazioni comportano dei falsi allarmi. Facendo tale assunzione si può calcolare la precisione del metodo come:

$$Precisione = \frac{Allarmi\ \ effettivi}{Allarimi\ \ effettivi + Falsi\ \ allarmi} = \frac{34}{34+6} \cdot 100 = 85\%$$

Si precisa che calcolando tale parametro per le anomalie non ancora raggruppate si otterrebbe $68 \div (68+10) \cdot 100 = 87\%$

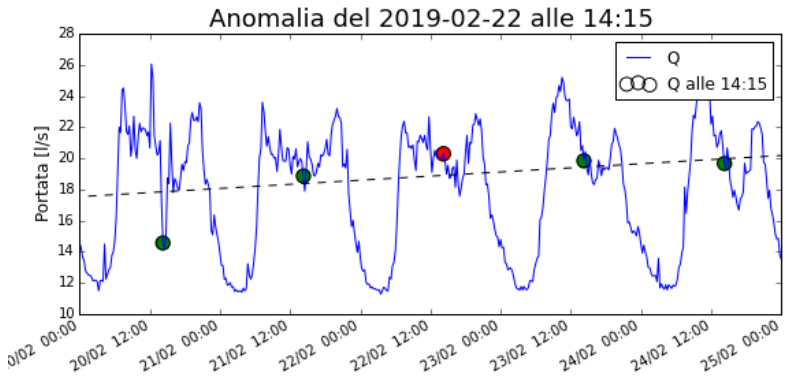


Figura 4.36: Falso allarme causato da un'anomalia precedente.

4.3.2 Automatizzazione Cavoretto

Definito il processo di automatizzazione per i parametri del CUSUM test relativi alla portata in uscita dalla Centrale di Avigliana, si vuole verificare l'adattabilità di tale metodo ad altri componenti idrici. Per tal motivo si è sottoposto ad analisi il livello in serbatoio della centrale di Cavoretto dal 01-01-2018 al 31-12-2018. A differenza del precedente caso, esso non ha un andamento dipendente dall'orario, ma si riempie e si svuota una volta raggiunto un determinato valore di soglia.

Come già descritto nel paragrafo 4.2.1, in questo caso, a causa di un grande evento anomalo che va dal 15-06-2018 al 20-07-2018, vedi Figura 4.20(a) a pag. 58, i dati dei mesi di Giugno e Luglio risultano corrotti e di conseguenza effettuare un processo di normalizzazione porta alla costruzione di serie che non rispecchiano il reale comportamento del serbatoio, vedi Figura 4.22 (b) e (d) a pag. 60.

Dal caso in esame si evince che in presenza di anomalie prolungate nel tempo, i dati risultano corrotti e dunque la normalizzazione risulta essere una tecnica controproducente. Per ovviare a tale inconveniente, noto l'evento anomalo, si potrebbe studiare la restante parte del database ottenendo ottimi risultati. Si ricercherà inizialmente la grande anomalia nota e successivamente si cercherà di studiare la restante parte del database con il metodo automatizzato.

Per riscontrare anomalie di grandi dimensioni o durature nel tempo, occorre somministrare al CUSUM test serie caratterizzate da un valore medio giornaliero della variabile di interesse. Si prova ad effettuare il CUSUM test ed utilizzando i parametri automatizzati $q_1 = 4\%$, $q_3 = 97\%$, $h = q_3 - q_1$ e $\nu = 5\%h$ si ottiene $h = 36.1\,cm$ e $\nu = 1.8\,cm$. L'output è riportato in Figura 4.37.

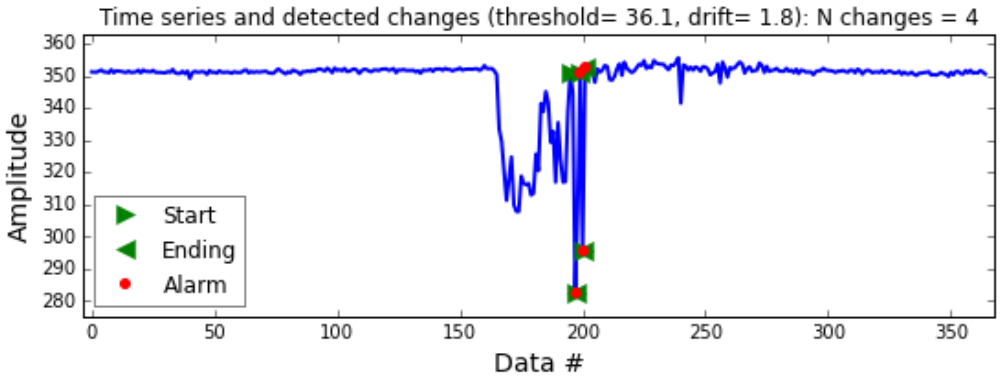


Figura 4.37: Parametri errati CUSUM test Cavoretto.

Il caso in esame evidenzia che porre il parametro ν come funzione di h risulta adatto per serie già normalizzate, dove h assume valori contenuti (tipicamente da 0.5 a 5) ma, per serie non normalizzate, dove h può assumere valori molto maggiori, il valore di ν risultante sarebbe troppo grande, portando soltanto ad un parziale riscontro di anomalie; si è dunque preferito porlo uguale a zero. Inoltre dopo alcuni tentativi, sono stati riscontrati come ottimali, i parametri $q_1 = 5\%$, $q_3 = 95\%$, $h = q_3 - q_1$ e $\nu = 0$ da cui si sono ottenuti i valori $h = 32.1\,cm$ e $\nu = 0\,cm$. Grazie a tali parametri e all'output risultante, vedi Figura 4.38, sono state individuate le date di inizio e fine del grande evento anomalo. I dati di Giugno e Luglio, corrotti da tale evento, saranno dunque esclusi dal database si cui si vogliono ricercare le anomalie.

Per trovare i singoli eventi anomali si è infine diviso il database in 2 parti: la prima da Gennaio a Maggio e la seconda da Agosto a Dicembre. Essendo dunque ora possibile effettuare il processo di normalizzazione, si vuole effettuare il CUSUM

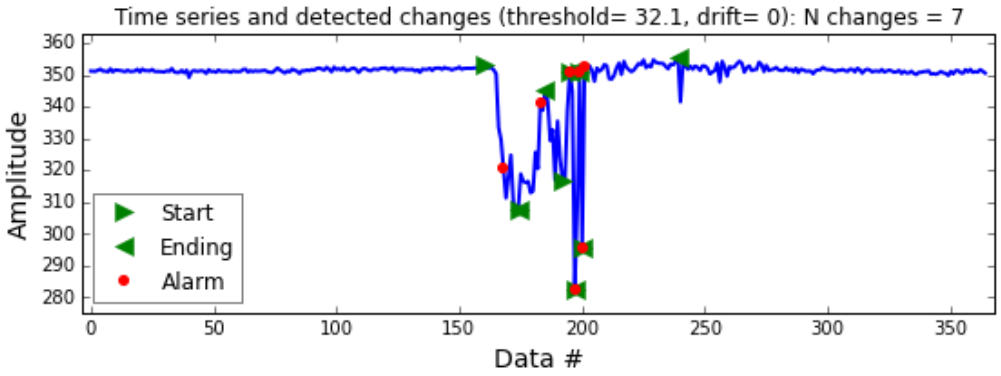


Figura 4.38: Parametri ottimali CUSUM test Cavoretto.

test con i parametri automatizzati per vedere se tale metodo risulta efficace in condizioni diverse rispetto a quella riscontrate per Avigliana.

Al fine di effettuare una valutazione sulle singole anomalie rilevate, si è considerata la seconda parte del database, riportando il CUSUM test in Figura 4.39 e le anomalie in Tabella 4.10. Le anomalie presenti in tabella sono inoltre rappresentate in Allegato B.2 sui grafici riportanti l'andamento del livello in serbatoio dal giorno precedente al successivo rispetto l'evento anomalo individuato.

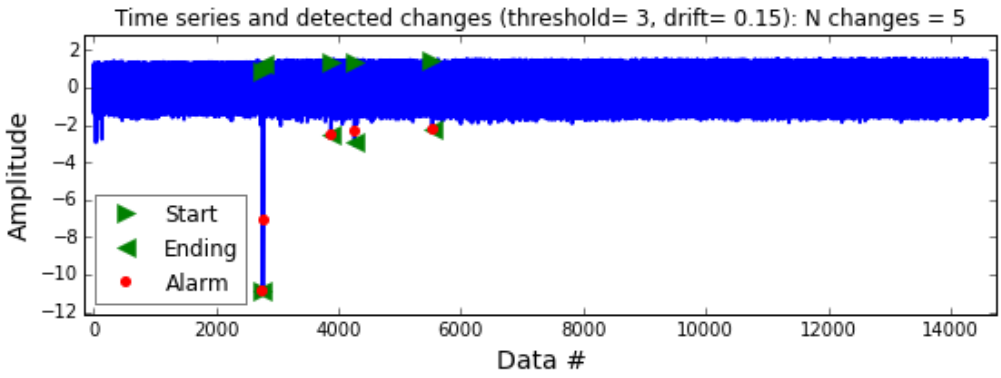


Figura 4.39: CUSUM test Cavoretto Agosto-Dicembre 2018.

A differenza delle portate in uscita dalla centrale di Avigliana, in questo caso, il livello in serbatoio non ha un andamento dipendente dall'orario, ma si riempie e si svuota una volta raggiunto un determinato valore di soglia. Questo comportamento più semplice non ha prodotto nessun falso allarme nel CUSUM test, ne consegue una *precisione* del metodo del 100%.

Parametri: $q_1 = 4\%$		$q_3 = 97\%$	$h = q_3 - q_1$	$\nu = 6\%h$
Anno	Mese	Giorno	Ora in.	Ora fin.
	Ago	29	18:30	19:30
2018		10	8:00	8:15
	Sett	14	7:15	8:00
		27	14:15	14:30

Tabella 4.10: Anomalie Cavoretto Agosto-Dicembre 2018

4.4 CUSUM in Real-Time

Tramite questo lavoro di tesi si è voluto anche fornire alla Società Metropolitana Acque Torino (SMAT s.p.a.), uno strumento utile per l'individuazione di perdite idriche, o comunque la possibilità di ridurre il periodo di tempo che va dalla nascita alla conoscenza di una potenziale rottura lungo la tubazione. Questo lasso temporale, anche chiamato *Unawareness period* rappresenta, come mostrato in Figura 3.4 a pag. 28, uno degli step costituenti il ciclo vitale di una perdita idrica. Sebbene la maggior parte delle rotture provochi di norma la comparsa d'acqua sulla superficie del terreno, acqua che viene segnalata dai clienti o dal personale della compagnia idrica (*rilevamento passivo*), il tempo di localizzazione medio può essere piuttosto lungo. Morrison [14] stima il tempo di consapevolezza e localizzazione di una perdita di $4\,m^3/h$ in 5 giorni. Questo ritardo nella localizzazione delle perdite causa un aumento dei costi complessivi associati alle rotture dei tubi, i quali, oltre al costo dell'acqua persa ed alla messa a punto della condotta, includono anche la riparazione dell'infrastruttura circostante danneggiata e i danni di immagine del gestore idrico causati dei reclami dei clienti, relativi alla fornitura d'acqua interrotta.

Dunque, le perdite idriche causano danni via via crescenti con lo scorrere del tempo. Per tal motivo, per i gestori idrici, la tempestività di intervento è un obiettivo primario da perseguire. Al fine di avere una conoscenza quasi immediata del verificarsi di una perdita, nel corso del lavoro di tesi, è stata formulata sul software Python la funzione *Anna*, riportata in Appendice A.2. Tale funzione, che può essere anche utilizzata in Real-Time, richiede come input una serie temporale della variabile che si vuole analizzare e restituisce in output i rispettivi valori anomali. Questi ultimi sono delle indicazioni di una variazione delle condizioni standard di esercizio spesso causate da perdite idriche. Dunque, integrando tale funzione al sistema SCADA della SMAT s.p.a., essa è in grado di acquisire in real-time i dati di tutti i componenti connessi al sistema e di restituire come output messaggi di

warning indicanti un'anomalia connessa al relativo componente.

Tale comportamento, denominato *rilevamento attivo*, rende possibile classificare in tempo reale il valore della variabile in "normale" oppure "anomalo". In quest'ultimo caso si indicherebbe la formazione di una perdita, e agendo di conseguenza, eliminando quindi l'unawereness period, si può limitare l'impatto e risanare il malfunzionamento, in modo da tornare alle corrette condizioni di esercizio. Si può dunque affermare che l'algoritmo *Anna* esercita anche una funzione di Early Warning nei sistemi distribuzione delle acque.

4.4.1 Definizione della funzione Anna

Dovendo tale metodo essere applicato in real-time, si è costruita una funzione in cui l'input è la serie da analizzare e l'output è la classificazione dell'ultimo dato della serie in "normale" o "anomalo". La funzione *Anna* è così definita:

```
def Anna(file_name, formato=0, delta_t_in=365, delta_t_fn=14,
                                    tipologia=0, freq_camp='15T'):
```

Si riporta di seguito la descrizione dei vari campi costituenti la funzione:

file_name: serve per sottoporre ad *Anna* il database desiderato. Si riporta il nome del file contente il database con i dati da analizzare.

formato: esplica la tipologia del file contenente il database. A seconde dell'estensione, *Anna* seguirà differenti processi di caricamento del database. Si pone $formato = 0$ se il file contenente il database ha un'estensione .h5. Essi sono file Hierarchical Data Format HDF, sono il formato di file standard per l'archiviazione di dati scientifici, utilizzati perchè in grado di contenere database di grandi dimensioni. Si pone $formato = 1$ se il database ha un'estensione .csv. CSV in inglese sta per Comma Separated Values, cioè dati separati da virgole. Un file con estensione .csv è un documento di testo che al suo interno contiene una serie di dati, organizzati in modo da simulare una tabella.

delta_t_in: serve per segnalare un punto di partenza d'analisi diverso da quello predefinito. Si indica il numero di giorni antecedente all'ultimo dato fornito sui cui la funzione effettuerà l'analisi.

delta_t_fn: serve per segnalare un punto di fine analisi diverso da quello predefinito. Si indica il numero di minuti da aggiungere all'ultimo dato fornito. Si precisa che per tener conto dell'ultimo dato, $delta\_t\_fn$ deve essere

uguagliato ad un numero positivo in minuti (di default sono 14 perchè la discretizzazione è impostata ogni quarto d'ora). Inoltre, inserendo un tempo negativo, sempre in minuti, è possibile assumere come punto finale d'analisi un dato antecedente quello predefinito.

tipologia: la funzione seguirà tre differenti processi d'analisi a seconda della tipologia scelta. Si pone $tipologia = 0$ in presenza di anomalie contestuali (dove il contesto è rappresentato dall'orario, ad esempio l'andamento delle portate in giornata). Si pone $tipologia = 1$ in presenza di anomalie puntuali (in cui il database ha sempre lo stesso andamento indipendentemente dal tempo, ad esempio un serbatoio che si svuota e si riempie). Si pone $tipologia = 2$ nel caso di grosse anomalie. In questo caso sono utilizzati dati non normalizzati. Si precisa che essendo i dati non normalizzati e determinando quindi valori di h considerevoli, data la dipendenza lineare tra valore di soglia è di drift, i parametri specificati nel CUSUM test si sono posti: $q_1 = 5\%$, $q_3 = 95\%$, $h = q_3 - q_1$ e $\nu = 0$.

freq_camp: serve per indicare la frequenza di campionamento che si vuole attribuire ai dati contenuti nel database. Ad esempio, se si vuole porre una discretizzazione di 15 minuti si dovrà porre $freq\_camp$ ='15T', se si volesse una discretizzazione giornaliera allora si dovrà porre $freq\_camp$ ='1D'. Tramite questo campo, la variabile assumerà il valore medio dei dati contenuti all' interno dell'intervallo di campionamento.

Sono di seguito mostrati gli output di alcune esempi:

Portata in uscita Avigliana

In questo caso siamo in presenza di anomalie contestuali dove il contesto è rappresentato dall'orario. Per tale motivo nella funzione Anna si riporterà $tipologia = 0$. Si riporta in seguito la funzione *Anna* applicata al database:

```
Anna('QR1',1, delta_t_in=365, delta_t_fn=14, tipologia=0,freq_camp=
                                        '15T')
```

In questo caso la funzione lavora costruendo una serie oraria avente come variabile la portata normalizzata allo stesso orario dell'ultimo dato contenuto nel database (30-01-2019 23:45) al variare dei giorni, un procedimento simile è stato già riportato nel paragrafo 4.1.2. A questo punto effettua il CUSUM test alla serie oraria costruita, utilizzando i parametri automatizzati, vedi Figura 4.40.

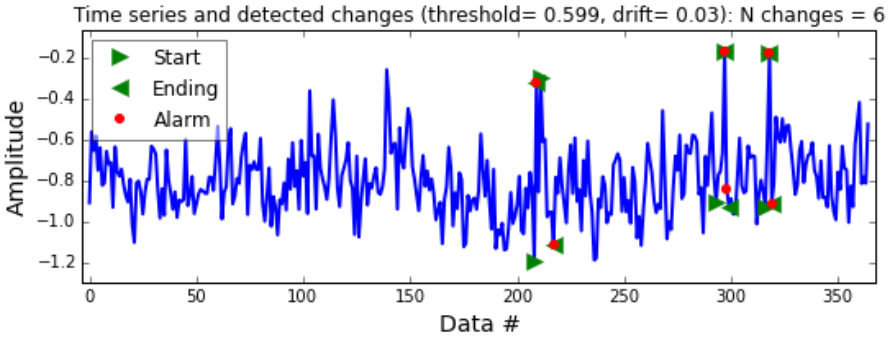


Figura 4.40: CUSUM test Avigliana, serie oraria delle 23:45.

La funzione riporta come output, vedi Tabella 4.11, data e ora dell'ultimo dato del database (tale dato rappresenta la variabile al tempo presente in un funzionamento real-time), media e deviazione standard del mese contenente il dato ed infine il valore normalizzato della variabile. In realtà il test si effettua sui valori normalizzati e grazie a media e deviazione standard si deduce l'effettivo valore della variabile in questione.

Data	Variab.	med. men.	std men.	var. nor.	Output
2019-01-30 23:45	15.18	17.12	3.7057	-0.525	Dato NORMALE

Tabella 4.11: Output database Avigliana

Essendo lo scopo dell'algoritmo *Anna* il funzionamento real-time, in output è riportata solo la classificazione dell'ultimo valore del database. Si precisa però che, se richiesto, tale funzione è anche in grado di riportare le anomalie occorse nel database d'analisi.

Portata in uscita Avigliana modificata

Al fine di verificare se la funzione fosse in grado di classificare un valore come anomalo, si è modificato il database contenente le portate in uscita dalla centrale di Avigliana. In particolare il database è stato tagliato il 23-04-2019 alle 09:15 in modo che l'ultimo dato del database coincidesse con un'anomalia nota. Tale anomalia è riportata in Appendice B.1. Esattamente come nel caso precedente, al nuovo database è stata applicata la funzione *Anna*.

```
Anna('QR2',1, delta_t_in=365, delta_t_fn=14, tipologia=0,freq_camp=
                                        '15T')
```

In questo caso la funzione formerà e successivamente sottoporrà ad analisi la serie oraria delle ore 09:15, vedi Figura 4.41. L'output è riportato in Tabella 4.12.

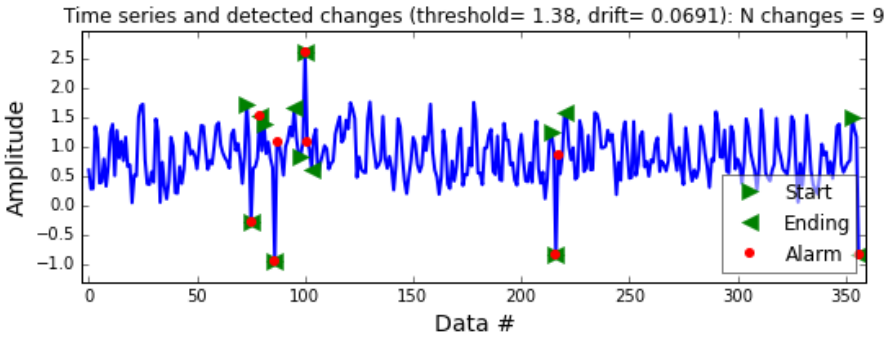


Figura 4.41: CUSUM test Avigliana, serie oraria delle 9:15.

Data	Variab.	med. men.	std men.	var. nor.	Output
2019-04-23 09:15	15.14	18.21	3.721	-0.824	Dato ANOMALO

Tabella 4.12: Output database Avigliana modificato

Lavorando in real-time, a differenza del caso precedente in cui si era riscontrato un valore classificato come normale, in questo caso la funzione registrerà l'evento anomalo e nel momento in cui si verificasse una nuova anomalia, la funzione restituirà come output entrambi gli eventi anomali riscontrati. Così facendo la funzione genererà automaticamente una seria formata da tutti gli eventi anomali a partire dall'attivazione in real-time di tale funzione.

Livello serbatoio centrale Cavoretto

Il livello in serbatoio della centrale di Cavoretto non ha andamento dipendente dall'orario, ma si riempie e si svuota una volta raggiunto un determinato valore di soglia, vedi Figura 4.21(a) a pag.59. Considerato tale funzionamento si è posto *tipologia* = 1 nel rispettivo campo della funzione *Anna*. In questo caso viene utilizzato l'intero database, normalizzandolo prima e sottoponendolo poi al CUSUM test. In questa *tipologia* bisogna controllare che nel database non siano presenti grosse anomalie, tali da influenzare la restante parte dei dati definiti come normali. Si riporta di seguito l'applicazione di tale funzione:

```
Anna('AB00127227_HCA_C_CAL',0, delta_t_in=(31+30+31+30+30),
                                delta_t_fn=14, tipologia=1,
                                freq_camp= '15T')
```

Si precisa che essendo noto un grande evento anomalo che influenza i dati di Giugno e Luglio nel campo *delta_t_in* saranno riportati la sommatoria dei giorni che vanno da Agosto a Dicembre (fine del database) in modo che nella nuova serie creata non siano presenti dati corrotti. Il CUSUM test della serie parziale creatasi è riportato in Figura 4.42. L'output è riportato in Tabella 4.13.

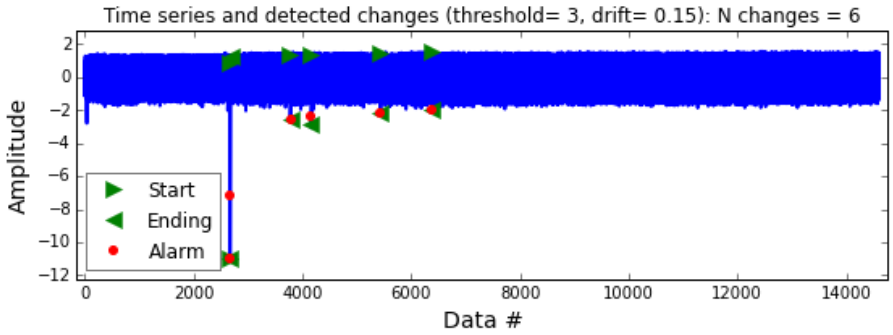


Figura 4.42: CUSUM test, Cavoretto Agosto-Settembre.

Data	Variab.	med. men.	std men.	var. norm.	Output
2018-12-31 23:45	356.25	351.35	11.392	0.430	Dato NORMALE

Tabella 4.13: Output database Cavoretto modificato

Portata centrale Cavoretto

Si vuole effettuare l'analisi sulla portata uscente dalla centrale di Cavoretto. Per questa analisi è stato fornito un database contenente tale variabile, più esteso nel tempo rispetto a quello mostrato in Figura 4.20(b) a pag 58. In particolare tale database fornisce una finestra temporale a partire dal al 2017-01-06 fino al 2019-05-30. L'andamento giornaliero risultante è riportato in Figura 4.43.

In questo caso, attraverso la funzione *Anna* si vogliono ricercare anomalie di grossa entità, si pone dunque $tipologia = 2$ nel rispettivo campo della funzione. La procedura costituente questa tipologia applica il CUSUM test ai dati non normalizzati e per tali serie di dati, come descritto nel paragrafo 4.3.2, il calcolo dei parametri di soglia e di drift è diverso rispetto ai casi precedenti, nello specifico sono utilizzati $q_1 = 5\%$, $q_3 = 95\%$, $h = q_3 - q_1$ e $\nu = 0$. Si precisa che ricercando anomalie di grosse entità, il test dovrà essere effettuato sui dati rappresentanti le medie giornaliere, per tale motivo nella funzione *Anna* verrà posto $freq\_camp$ ='1D'. Si riporta di seguito l'applicazione di tale funzione:

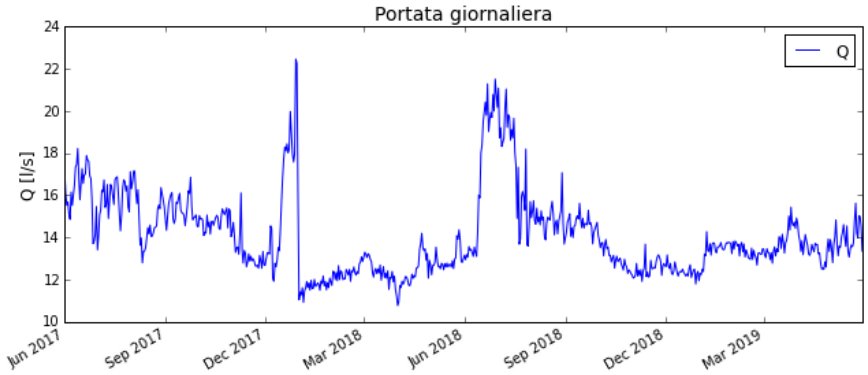


Figura 4.43: Andamento della portata media giornaliera.

```
Anna('QCA',1, delta_t_in=730, delta_t_fn =14, tipologia=2,
                                    freq_camp='D')
```

Il CUSUM test della serie è riportato in Figura 4.44. L'output è riportato in Tabella 4.14.

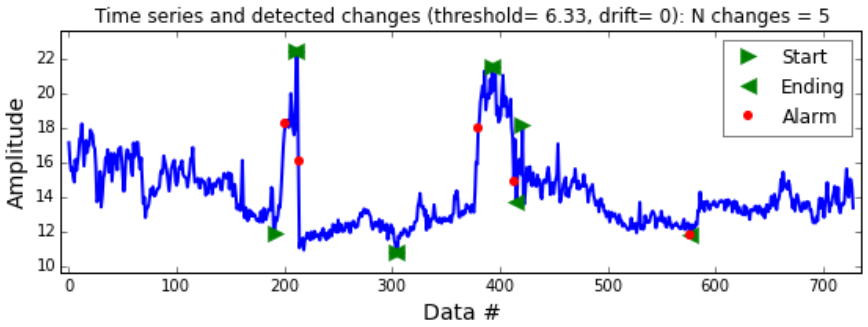


Figura 4.44: CUSUM test, Cavoretto portate giornaliere.

Data	Variabile	Output
2019-05-30	13.34	Dato NORMALE

Tabella 4.14: Output database portate Cavoretto

4.4.2 Implementazione di Anna al sistema SCADA

In data 05-07-2019 la funzione *Anna* è stata implementata al sistema SCADA della SMAT s.p.a. In particolare, in modo da poterne studiare possibili futuri pregi e difetti, essa è stata connessa al sistema di monitoraggio del distretto Belgio.

Attualmente è in atto il processo di distrettualizzazione della città di Torino, ossia di parzializzazione del sistema idrico, definendo aree di distribuzione fra loro disconnesse (District Meter Area - DMAs), alimentate in genere attraverso un numero limitato di punti di immissione muniti di misuratore di portata. La distrettualizzazione si pone in antitesi alla tradizionale prassi progettuale tesa a privilegiare i sistemi distributivi a maglia estesa e ad assicurare un elevato grado di interconnessione. Questi ultimi, da un lato, offrono indubbi vantaggi in termini di elasticità di funzionamento del sistema idrico nelle diverse condizioni di esercizio, dall'altro, comportano un minore grado di controllo e maggiori difficoltà nel monitoraggio delle perdite. La ricerca delle perdite attraverso la distrettualizzazione permette di effettuare controlli sistematici tra la fornitura e il consumo per ogni zona e, quindi, di stimare la quantità d'acqua dispersa. Successivamente, sulla base dei risultati ottenuti, si può procedere all'esatta localizzazione dei punti di perdita mediante sistemi di ricerca in genere elettroacustici e correlativi.

Tramite tale processo si è dunque delimitato il distretto Belgio. Esso è localizzato nella zona nord-est del comune di Torino, in Figura 4.45 è riportata l'area geografica di riferimento.

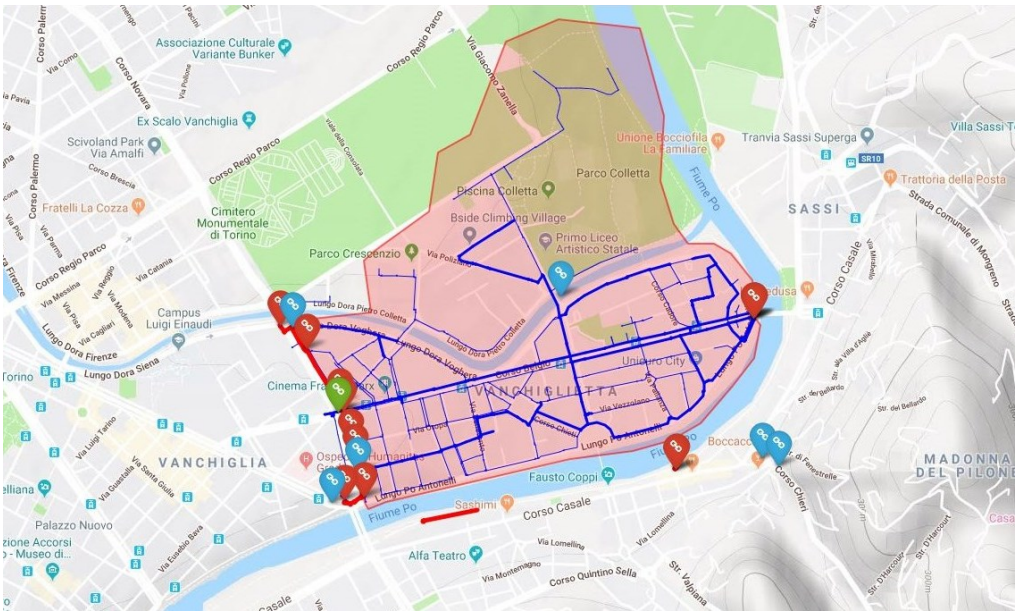


Figura 4.45: Distretto Belgio.

Il distretto è caratterizzato da una media di 42.09 l/s. Si riscontra un valore così elevato in quanto non è assicurata la completa chiusura di alcune saracinesche situate lungo il confine del distretto, da cui potrebbe transitare un contributo idrico non registrato.

La rete di distribuzione associata è lunga 21.65 km . Tale rete è caratterizzata da un rapporto di $Fughe/km/anno = 2.06$ indicante un rinnovo continuo di tubazioni a causa di frequenti rotture associate alle elevate pressioni in rete.

Il distretto è rifornito d'acqua tramite due ingressi muniti di misuratori di portata e pressione, che in real-time comunicano i valori delle due variabili al sistema SCADA. Dalla somma delle due entrate si deduce il classico andamento della portata in grado di soddisfare l'utenza, riportato in alto a destra in Figura 4.46.

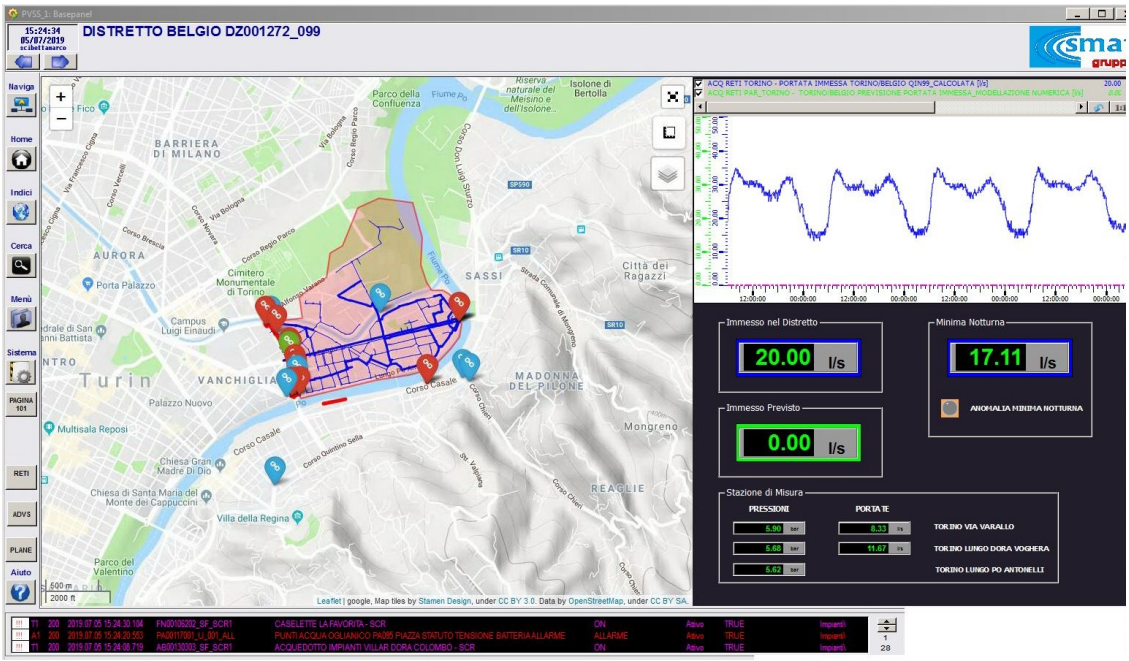


Figura 4.46: Funzione Anna associata al distretto Belgio.

Per l'individuazione di perdite idriche, assume particolare importanza il concetto di portata minima notturna (MNF - Minimum Night Flow). Tale valore che si verifica solitamente tra le 03:00 e 04:00 di notte è il valore minimo di portata giornaliero. Non essendo influenzato dalle utenze, in quanto la maggior parte degli usufruenti non ha bisogno di acqua durante tali ore, non è soggetto ad un comportamento aleatorio. Inoltre non essendoci richiesta idrica da parte dell'utenza, nel parametro assume componente predominante la perdita idrica, che, a differenza della richiesta, rimane costante durante tutte le ore del giorno. Si è quindi deciso di applicare la funzione *Anna* alla MNF del distretto Belgio. In realtà essendo ancora

in atto il processo di distrettualizzazione della città di Torino, ed essendo il distretto Belgio appena formatosi, non si hanno a disposizione i dati storici necessari per definire automaticamente i parametri h e ν e per effettuarne la normalizzazione. Per tal motivo si sono utilizzati i dati standard e si è effettuata una calibrazione manuale dei parametri.

In particolare l'algoritmo assume come MNF la media delle portate verificatesi tra le 03:00 e le 04:00 in modo da avere un valore più stabile possibile nel tempo. Registrando quindi un valore ogni giorno, col passare del tempo, si realizza una serie formata dal MNF al variare dei giorni, sottoponendola poi al CUSUM test. Nel momento in cui si verificasse una perdita, essa causerebbe l'innalzamento del successivo valore di MNF e tale variazione, individuabile dall'algoritmo, sarebbe poi segnalata attraverso l'accensione di una spia, riportata spenta in Figura 4.46. Si precisa che somministrando una serie falsata, contenente un'anomalia si è riscontrato quanto appena esposto.

La procedura descritta è al momento 10-07-2019 in atto e si attende il primo riscontro di un evento anomalo.

Capitolo 5

Risultati

Nei precedenti paragrafi, attraverso la funzione *Anna* si è effettuato un rilevamento di anomalie su due scale differenti. Sottoponendo all'algoritmo serie discretizzate ogni 15 minuti si sono individuate microanomalie, sottoponendogli invece serie con frequenza giornaliera si sono riscontrate macroanomalie.

5.1 Microanomalie

Le microanomalie sono eventi difficilmente individuabili manualmente da un operatore. Esse sono causati da malfunzionamenti o variazioni delle condizioni standard d'esercizio. La loro conoscenza è utile ad avere il pieno controllo dell'impianto e ad segnalare piccoli eventi anomali che usualmente non si ripercuotono sulla successiva attività dell'impianto. Le microanomalie sono state analizzate sia per la serie formata dalle portate in uscita dalla centrale di Avigliana, sia per il livello in serbatoio nella centrale di Cavoretto.

Centrale Avigliana
Durante l'analisi delle portate, facendo riferimento ad una finestra temporale di 4 mesi sono state riscontrate 103 anomalie, successivamente raggruppate in 34 eventi anomali di maggior durata. Tali eventi sono riportati in Tabella 4.9 a pag.75 e in Appendice B.1. Si deve però precisare che per 3 volte un'anomalia già riconosciuta ha portato alla rilevazione di un falso allarme 2 giorni dopo l'effettivo riscontro e per altre 3 ciò è avvenuto dopo 3 giorni. Dunque se l'operatore non è a conoscenza della precedente anomalia o se non ne individua la correlazione, tali segnalazioni comportano dei falsi allarmi. Facendo tale assunzione si è calcolata la precisione del metodo come:

$$Precisione = \frac{Allarmi\ \ effettivi}{Allarimi\ \ effettivi + Falsi\ \ allarmi} = \frac{34}{34+6} \cdot 100 = 85\%$$

Si precisa che calcolando tale parametro per le anomalie non ancora raggruppate si otterrebbe $68 \div (68+10) \cdot 100 = 87\%$.

Centrale Cavoretto
Durante l'analisi del livello, facendo riferimento ad una finestra temporale di 5 mesi sono state riscontrate 9 anomalie successivamente raggruppate in 4 eventi anomali di maggior durata. Tali eventi sono riportati in Tabella 4.10 a pag. 79 e in Appendice B.2. In questo caso, a differenza delle portate in uscita dalla centrale di Avigliana in cui la variabile assume un andamento ciclico in funzione dell'orario, il livello in serbatoio si innalza e decresce una volta raggiunti determinati valori di soglia preimpostati. Questo comportamento più semplice non ha prodotto alcun falso allarme nel CUSUM test, il che fa conseguire una precisione del metodo pari al 100%.

5.2 Macroanomalie

Le macroanomalie sono eventi causati principalmente da perdite idriche dovute a rotture lungo le tubazioni della rete di distribuzione. Fino ad ora le perdite sono state individuate solo grazie a segnalazioni da parte delle utenze o dal personale della compagnia idrica (*rilevamento passivo*). Questo ha comportato tempi di rilevazione e riparazione della rottura piuttosto lunghi provocando gravi perdite in termini di risorsa idrica. Grazie all'algoritmo *Anna* tali tempi si possono ridurre drasticamente. Avendo a disposizione questo mezzo, si vogliono stimare, sia in termini economici sia in termini di risorsa idrica salvata, i possibili vantaggi conseguibili dall'applicazione della funzione prodotta alle centrali analizzate.

Per capire a quanto corrisponde economicamente il volume d'acqua perso si deve introdurre il concetto di *costo marginale*. In economia e finanza il costo marginale unitario corrisponde al costo di un'unità aggiuntiva prodotta, cioè alla variazione nei costi totali di produzione che si verifica quando si varia di un'unità la quantità prodotta: è la derivata del costo totale (C) rispetto alla quantità prodotta (q).

$$C' = \frac{dC}{dq}$$

In questo caso, l'azienda di riferimento è la SMAT s.p.a. e l'unità aggiuntiva prodotta è il volume dell'acqua a m^3. Noto il costo marginale di 0.2€ al m^3,

moltiplicando il volume di acqua persa per il costo marginale si ottiene la perdita economica che tale evento anomalo comporta all'azienda. Si noti che la tariffa media oraria per l'utenza è di 1.37€ al m^3 dunque, se tale acqua fosse sprecata, i costi risulterebbero di gran lunga maggiorati. Si precisa inoltre che, oltre al costo del volume d'acqua perso, in caso di perdita idrica concorrono i costi associati alla messa a punto della condotta, alla riparazione dell'infrastruttura circostante danneggiata e ai danni di immagine del gestore idrico causati dei reclami dei clienti relativi alla fornitura d'acqua interrotta. In questa sede questi costi associati non saranno presi in conto e ci limiteremo a calcolare il valore economico corrispondente all'acqua persa.

Le macroanomalie sono state analizzate sia per la serie formata dalle portate medie giornaliere in uscita dalla centrale di Avigliana, sia per quelle che riforniscono il serbatoio della centrale di Cavoretto.

Portata centrale Cavoretto

Si vuole effettuare l'analisi sulla portata uscente dalla centrale di Cavoretto. In questo caso, attraverso la funzione *Anna* si vogliono ricercare anomalie di grossa entità. Si pone dunque $tipologia = 2$ nel rispettivo campo della funzione. Si precisa che ricercando anomalie di grosse entità, il test dovrà essere effettuato sui dati rappresentanti le medie giornaliere e per tale motivo nella funzione *Anna* verrà posto $freq\_camp =$'1D'. Si riporta di seguito l'applicazione di tale funzione:

```
Anna('QCA',1, delta_t_in=730, delta_t_fn =14, tipologia=2,
                                    freq_camp='D')
```

Il CUSUM test della serie è riportato in Figura 5.1.

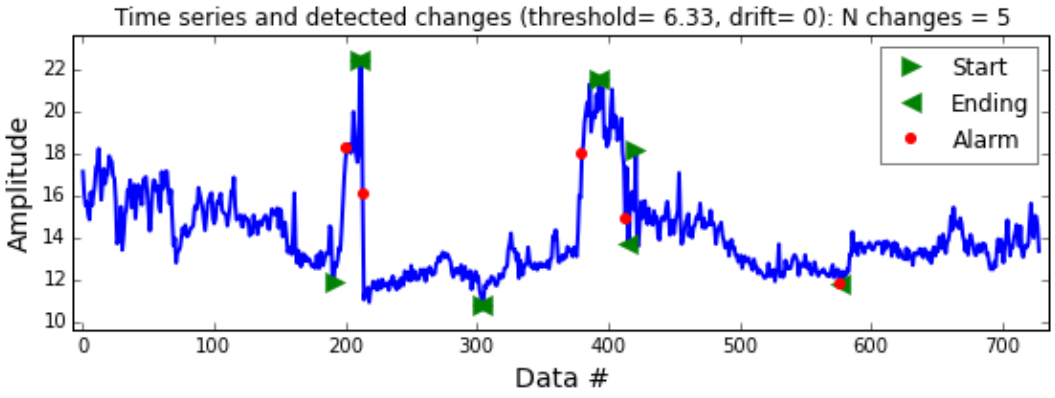


Figura 5.1: CUSUM test, Cavoretto portate giornaliere.

Dall'output sono evidenziate 5 anomalie che si dispongono all'inizio ed alla fine di ogni eventi anomalo. Si riportano in Tabella 5.1 gli eventi così riscontrati

Macroanomalie	Data inizio	Data fine
1	02-12-2017	05-01-2018
2	01-06-2018	28-07-2018
3	20-12-2018	In corso

Tabella 5.1: Macroanomalie Cavoretto

Sono quindi studiate singolarmente le macroanomalie riscontrate.

Macroanomalia 1 Si riporta il primo evento anomalo in Figura 5.2, in cui, sulla curva individuata dalle portata in uscita dalla centrale di Cavoretto, sono cerchiate le portate relative alle date di inizio e fine evento. Congiungendo tali valori tramite una linea tratteggiata si vuole mostrare il comportamento medio teorico che la portata assumerebbe in condizioni standard di esercizio. I grafici contenenti gli eventi anomali riportano sull'asse delle ascisse la portata in l/s e sull'asse delle ordinate il tempo in giorni. Dunque, l'area sottesa dalle due curve, moltiplicata per il numero di secondi in un giorno ($60 \times 60 \times 24 = 86400$), rappresenta il volume in litri di acqua persa. Dividendo infine il valore trovato per mille, si ottiene la conversione in m^3

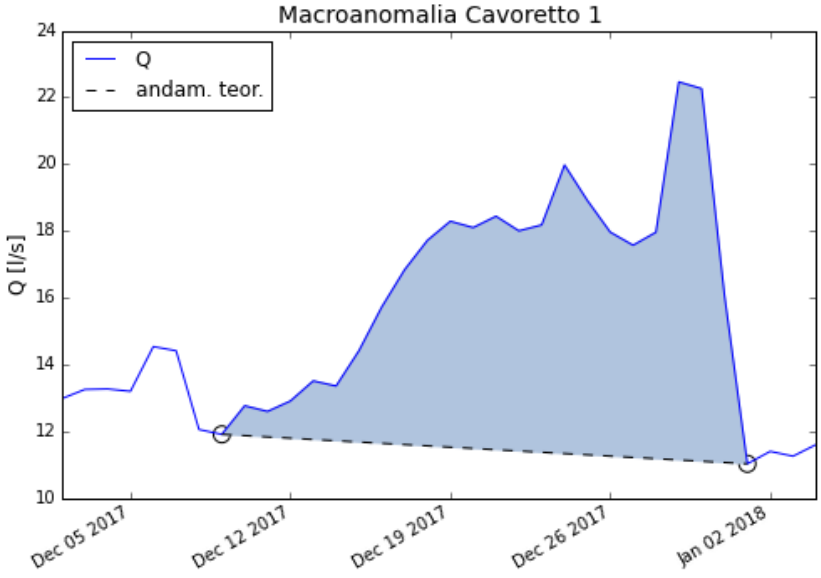


Figura 5.2: Prima macroanomalia Cavoretto.

Macroanomalia 2 Si riporta il secondo evento anomalo in Figura 5.3. Esattamente come nel caso precedente, sulla curva individuata dalle portata in uscita, sono cerchiate le portate relative alle date di inizio e fine evento. Congiungendo tali valori tramite una linea tratteggiata si vuole mostrare il comportamento medio teorico che la portata assumerebbe in condizioni standard di esercizio. L'area sottesa tra le due curve rappresenta la risorsa idrica persa a causa di tale evento.

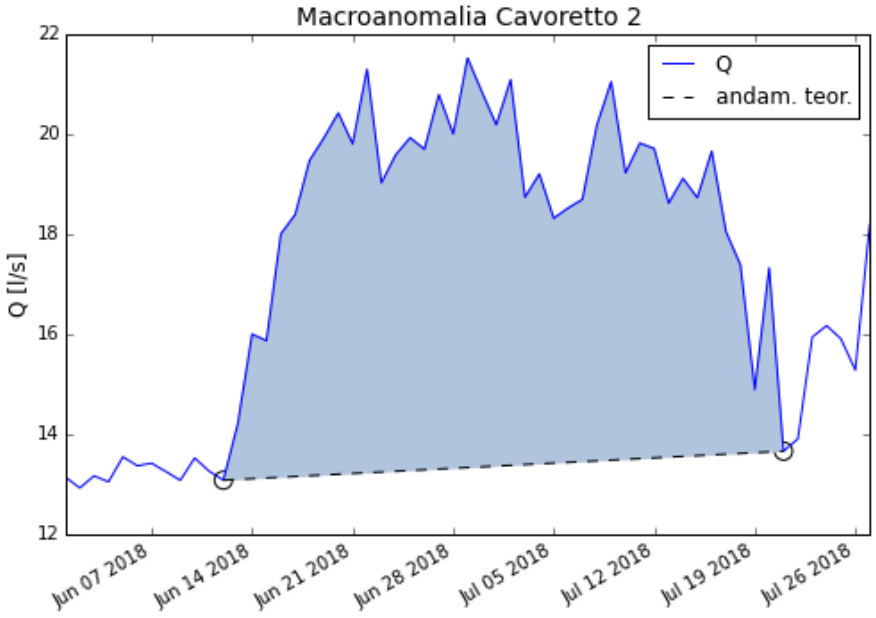


Figura 5.3: Seconda macroanomalia Cavoretto.

Macroanomalia 3 Si riporta il terzo evento anomalo in Figura 5.4. Il primo estremo, analogamente ai precedenti casi, è rappresentato dalla portata relativa alla data di inizio evento. Essendo tale evento ancora in corso si è dovuto stabilire un valore di portata in data 31-05-2019 (data di fine database) in modo da associarlo al secondo estremo. Supponendo che la media mensile delle portata, in assenza di anomalie, non variasse troppo tra inizio anomalia e fine database, si è posto come secondo estremo la portata media mensile del mese di Dicembre ($12.41 l/s$), ultimo mese non influenzato dall'evento anomalo in esame. Congiungendo i due estremi tramite una linea tratteggiata si vuole mostrare il comportamento medio teorico che la portata assumerebbe in condizioni standard di esercizio. L'area sottesa tra le due curve rappresenta l'acqua persa a causa di tale evento.

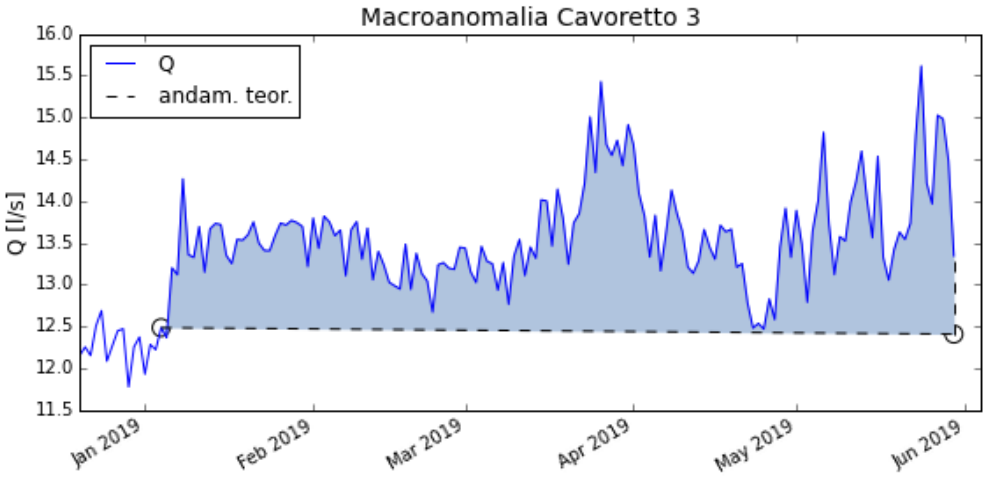


Figura 5.4: Terza macroanomalia Cavoretto.

I risultato ottenuti sono riassunti in Tabella 5.2, dove per ogni macroanomalia viene riportata la data di inizio e fine evento anomalo, il volume d'acqua perso in m^3, il costo per la SMAT s.p.a. equivalente al volume idrico perso e il costo che quel volume assumerebbe se fosse utilizzato dall'utenza.

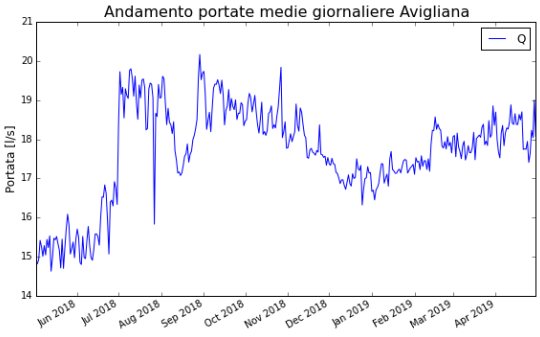
Macroanom.	Data inizio	Data fine	Vol perso $[m^3]$	SMAT €	Utenza €
1	02-12-2017	05-01-2018	10 511	2 102	14 400
2	01-06-2018	28-07-2018	18 546	3 709	25 408
3	20-12-2018	In corso	14 528	2 906	19 903
Totale			43 585	8 717	59 711

Tabella 5.2: Riassunto macroanomalie Cavoretto

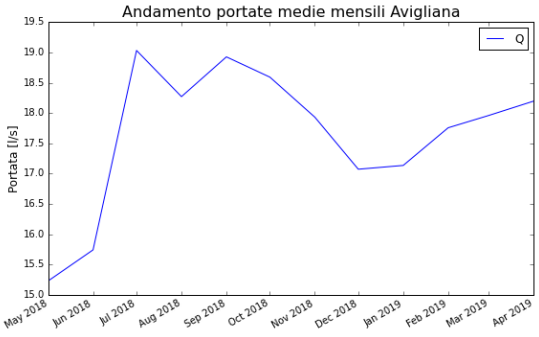
Portata centrale Avigliana

Si vuole effettuare l'analisi sulla portata uscente dalla centrale di Avigliana. Riportando in Figura 5.5 l'andamento giornaliero e mensile, si nota subito l'innalzamento della portata media a partire dall'inizio del mese di Luglio 2018. Tale innalzamento è causato da una perdita idrica non rilevata e attualmente in corso. La conferma è data dalla differenza tra il valore medio mensile riscontrato a Maggio 2018 ($14.23\,l/s$) e quello riscontrato ad Aprile 2019 ($18.20\,l/s$). Infatti, anche se la richiesta d'acqua varia durante l'anno a causa delle diverse condizioni climatiche, passati dodici mesi, il valore medio mensile dovrebbe riposizionarsi nell'intorno del valore

individuato il precedente anno. In questo caso, la variazione di $4\,l/s$ rappresenta la portata che attualmente defluisce dalla perdita idrica non rilevata.



(a) *Andamento portata media giornaliera.*



(b) *Andamento portata media mensile.*

Figura 5.5: Andamento medio giornaliero e mensile della portata di Avigliana

Sottoponendo l'intero database alla funzione *Anna*, essa non è in grado di restituire la data di inizio dell'evento anomalo, in quanto quasi la totalità dei dati ne risulta influenzato. Indicando però alla funzione come data di inizio serie quella di inizio database e come fine quella di un giorno non troppo distante dall'evento anomalo, in modo che la maggior parte dei dati non ne risulti influenzata, ad esempio quella del 31 Luglio, *Anna* restituisce come output la data e il valore di portata di inizio evento anomalo. Si noti che volendo ricercare anomalie di grossa entità, si pone $tipologia = 2$ e $freq\_camp$ ='1D' nei rispettivi campi della funzione. Si riporta di seguito l'applicazione:

```
Anna('QR1', 1, delta_t_in=365, delta_t_fn=-(60*24*30.5*9),
                                    tipologia=2, freq_camp='D')
```

Tramite il CUSUM test, riportato in Figura 5.6, si è riscontrata come inizio evento anomalo la data del 02-07-2018.

Come già fatto per la centrale di Cavoretto, si vogliono stimare, sia in termini economici sia in termini di risorsa idrica salvata, i possibili vantaggi conseguibili dall'applicazione della funzione prodotta alla centrale di Avigliana. A differenza del caso precedente, qui si riscontra una sola grande anomalia. In Figura 5.7 si riporta l'intero database in cui, sulla curva individuata dalle portata in uscita dalla centrale di Avigliana, sono cerchiate le portate relative alle date di inizio e fine evento. In questo caso, il primo estremo è rappresentato dalla portata relativa alla data di inizio evento. Come secondo estremo si è posta, alla data di fine database (30-04-2019), la portata media mensile di Maggio 2018 ($15.23\,l/s$), in quanto, in assenza

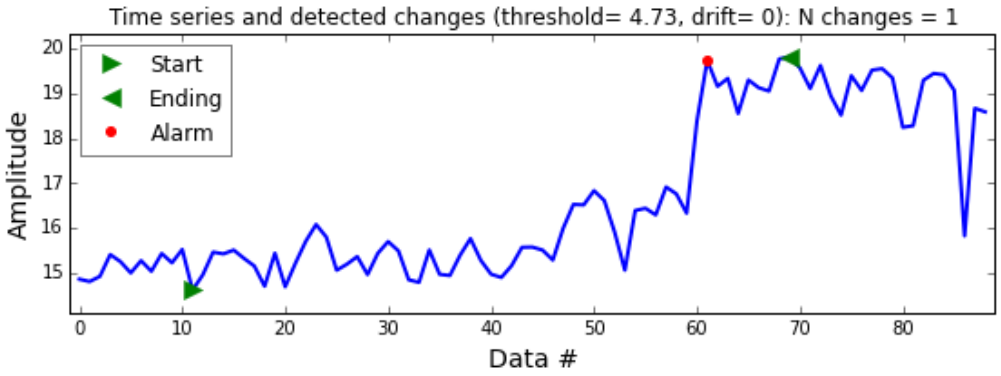


Figura 5.6: Cusum test Avigliana Maggio-Lugio 2018.

di anomalie e passati dodici mesi, il valore medio mensile dovrebbe riposizionarsi nell'intorno del valore individuato il precedente anno. Congiungendo i due estremi tramite una linea tratteggiata si vuole mostrare il comportamento medio teorico che la portata assumerebbe in condizioni standard di esercizio. L'area sottesa tra le due curve rappresenta l'acqua persa a causa di tale evento.

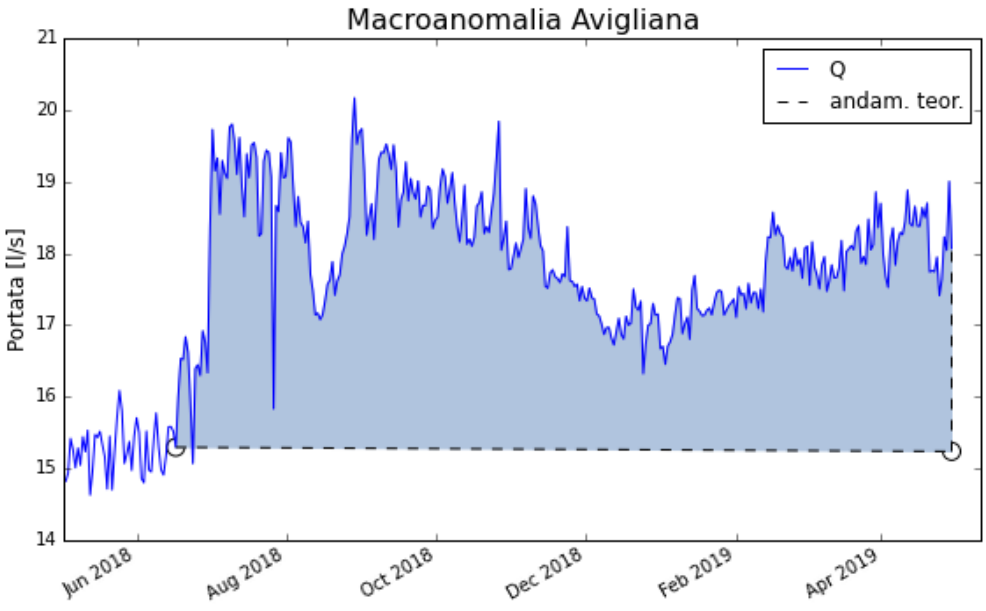


Figura 5.7: Macroanomalia Avigliana.

I risultati ottenuti relativi all'evento anomalo appena descritto insieme a quelli riscontrati nel database contenente le portate della centrale di Cavoretto sono riassunti in Tabella 5.3. In essa per ogni macroanomalia viene riportata la data di

inizio e fine evento anomalo, il volume d'acqua perso in m^3, il costo per la SMAT s.p.a. equivalente al volume idrico perso e il costo che quel volume assumerebbe se fosse utilizzato dall'utenza.

Macroanom.	Data inizio	Data fine	Vol perso $[m^3]$	SMAT €	Utenza €
Cavoretto 1	02-12-2017	05-01-2018	10 511	2 102	14 400
Cavoretto 2	01-06-2018	28-07-2018	18 546	3 709	25 408
Cavoretto 3	20-12-2018	In corso	14 528	2 906	19 903
Avigliana	02-07-2018	In corso	75 440	15 088	103 353
Totale			119 025	23 805	163 065

Tabella 5.3: Riassunto macroanomalie

Per comprendere l'entità del volume d'acqua perso si fa notare che la capacità di una piscina olimpionica è di $2\,500\,m^3$, ed essendo il volume d'acqua perso dalle due centrali negli ultimi 2 anni di $119\,025\,m^3$ esso sarebbe in grado di riempirne ben 48. Inoltre tale valore, in concomitanza con i rispettivi valori economici, è destinato a salire in quanto due delle quattro macroanomalie riscontrate risultano ancora in corso.

Infine si vuole far notare che i valori riscontrati, seppur significativi, sono minimi rispetto a quelli rilevabili tenendo conto di tutte le centrali idriche sotto il controllo della SMAT, anche perché le due centrali riforniscono un'area geografica relativamente ristretta, vedi Figure 4.1 e 4.18 a pag. 35 e 56. Si precisa però che un'effettiva e completa valutazione non può essere attuata applicando banalmente la funzione prodotta ad ogni centrale, in quanto molte di esse risultano fortemente interconnesse e quindi variazioni di portata in una sola risultano spesso correlate al diverso funzionamento di un'altra più che alla rilevazione di effettive perdite idriche. Per tal motivo è necessario ed auspicabile che studi futuri proseguano nella analisi che si è inteso avviare per renderne i risultati e la lettura sempre più completi ed attendibili.

Capitolo 6

Conclusioni

Come premesso nell'introduzione e più compiutamente illustrato nel corpo del lavoro di tesi, nei sistemi di gestione e distribuzione delle acque, le nuove strumentazioni e le sempre più sviluppate conoscenze informatiche hanno condotto ad una sempre maggior distanza tra l'operatore e i processi di produzione e distribuzione dell'acqua.

La crescita di questa distanza ha fatto altresì crescere il rischio che i guasti e i differenti comportamenti associati a problemi occorsi nel sistema possano rimanere inosservati.

Al fine di eliminare o ridurre al minimo questo rischio, nel corso di questo lavoro di tesi è stata formulata, mediante il software Python, la funzione *Anna* che, una volta integrata con il sistema di telecontrollo idrico aziendale e funzionando in real-time ed in maniera automatizzata, si è dimostrata in grado di produrre come output messaggi di warning indicanti una anomalia connessa al relativo componente del sistema.

Il rilevamento di anomalie è stato effettuato su due scale differenti. In generale, sottoponendo all'algoritmo serie discretizzate ogni 15 minuti si sono individuate microanomalie, sottoponendo invece serie con frequenza giornaliera si sono riscontrate macroanomalie.

L'algoritmo *Anna* ha come principio base di funzionamento una tecnica sequenziale per il controllo statistico (CUSUM o cumulative sum control chart) in grado di rilevare le variazioni anomale tra i dati delle serie su cui è svolge l'analisi. La sua elaborazione è stata implementata inizialmente da un processo di automatizzazione in cui, nota la serie d'analisi, si è ricercata un'individuazione autonoma del valore ottimale dei parametri da utilizzare durante l'applicazione della tecnica CUSUM. Tale automatizzazione si è diramata in due processi differenti, ma interconnessi, a seconda della dimensione dell'evento anomalo ricercato.

Per la rilevazione di piccoli malfunzionamenti e quindi di anomalie di piccola entità, dimostrata la correlazione tra richiesta idrica e temperatura ambientale si sono, per ogni mese, normalizzati i valori della variabile. Tale operazione ha portato ad una stabilizzazione della variabile nel corso dell'intero database, annullando le fluttuazioni dovute alle variazioni di temperatura mensili.

Si è poi osservato come le performance della funzione fossero dipendenti dalla scelta dei parametri di soglia (h) e di drift (ν) specificati nella tecnica CUSUM.

Dimostrato che la frequenza dei valori delle variabili d'analisi si dispone secondo una distribuzione a campana in cui i valori anomali ricadono agli estremi, si sono cercati di manipolare i parametri di soglia e di drift in modo da aumentare al massimo le performance e l'affidabilità dell'algoritmo proposto. Dopo numerosi tentativi e rispettivi controlli si è valutato che il valore migliore dei parametri fosse ottenuto ponendo il parametro di drift come funzione lineare di quello di soglia ($\nu = 5\%$h) e quest'ultimo come differenza tra due percentili q_3 e q_1 posti rispettivamente al 97% e al 4% delle osservazioni. Da tali parametri si è dedotta una precisione del metodo del 85% per serie che presentavano anomalie contestuali e del 100% per serie che presentavano anomale puntuali.

Per riscontrare anomalie di grandi dimensioni o durature nel tempo, occorre somministrare al CUSUM test serie caratterizzate da un valore medio giornaliero della variabile di interesse. In presenza di tali anomalie, i dati risultano corrotti e dunque effettuare un processo di normalizzazione porta alla costruzione di serie che non rispecchiano il reale andamento del relativo database.

Inoltre, se porre il parametro ν come funzione lineare di h è risultato particolarmente conveniente per serie normalizzate (in cui h assume valori contenuti tipicamente da 0.5 a 5), per serie non normalizzate (in cui il valore di soglia può assumere valori significativamente superiori) il legame lineare con il parametro di drif produrrebbe valori di quest'ultimo troppo elevati, portando soltanto ad un parziale riscontro di anomalie. Per tal motivo si è preferito porlo uguale a zero e, dopo diverse iterazioni, sono stati riscontrati come ottimali, i parametri $q_1 = 5\%$, $q_3 = 95\%$, $h = q_3 - q_1$ e $\nu = 0$. Attraverso questi parametri si sono sempre riscontrate le date e i valori della variabile d'esame di inizio e di fine evento anomalo.

Ipotizzando dunque una previa integrazione della funzione *Anna* al sistema di monitoraggio delle due centrali idriche analizzate nel corso della tesi, si sono stimati i vantaggi che si sarebbero potuti ottenere sia in termini economici sia in termini di risorsa idrica salvata, ferma restando la necessità di tener conto, in sede di effettiva valutazione economica, dell'incidenza dei tempi variabili indispensabili per eseguire la localizzazione e le specifiche riparazioni delle condotte danneggiate. Dall'analisi effettuata si è dedotto un volume d'acqua perso di $119025\,m^3$ corrispondente ad

un defict di 23805 euro per l'azienda SMAT s.p.a.. Si precisa che se questa risorsa idrica fosse stata utilizzata dall'utenza, il costo associato sarebbe stato di 163065 euro. Inoltre, essendo due delle quattro anomalie riscontrate ancora in corso, tali valori sarebbero destinati a crescere fino al ripristino e messa a punto della condotta causa della relativa perdita.

Si noti che i valori riscontrati, seppur significativi, sono minimi rispetto a quelli rilevabili tenendo conto di tutte le centrali idriche sotto il controllo della SMAT, anche perché le due centrali riforniscono un'area geografica relativamente ristretta.

Si precisa però che un'effettiva e completa valutazione non può essere attuata applicando banalmente la funzione prodotta ad ogni centrale, in quanto molte di esse risultano fortemente interconnesse e quindi variazioni di portata in una sola risultano spesso correlate al diverso funzionamento di un'altra più che alla rilevazione di effettive perdite idriche. Per tal motivo è necessario ed auspicabile che studi futuri proseguano nella analisi che si è inteso avviare per renderne i risultati e la lettura sempre più completi ed attendibili.

I due differenti metodi descritti per l'individuazione di micro e macro anomalie non sono indipendenti; al contrario solo la interconnessione permette di avere un'ottimizzata gestione dell'impianto ed un suo pieno controllo. Infatti, nel corso dell'elaborato, è stato mostrata l'impossibilità di individuazioni di microanomalie nel caso in cui la presenza di eventi anomali prolungati nel tempo producesse dei dati corrotti nel database. Per ovviare a tale inconveniente è dunque necessario effettuare prima l'individuazione di macroanomalie, e noto l'evento anomalo, si è in grado di studiare la restante parte del database ottenendo i risultati più completi ed attendibili.

La funzione è stata infine implementata in modo che potesse operare in real-time. Questo procedimento ha reso possibile una sua integrazione con il sistema SCADA della Società Metropolitana Acque Torino. In particolare, è stata connessa al sistema di monitoraggio del distretto Belgio. Grazie a questo processo, la funzione è in grado di acquisire i dati connessi al sistema e di produrre, nel momento in cui si verificasse una perdita idrica, messaggi di warning mediante l'accensione di una spia indicante un'anomalia nel sistema.

Il processo appena descritto rende possibile una drastica riduzione del periodo di tempo che va dalla nascita alla conoscenza di una potenziale rottura lungo la tubazione (*Unawareness period*).

Dunque la funzione formulata svolge anche un effetto di *Early Warning* in quanto, una volta riconosciuta l'anomalia, si può agire di conseguenza e con immediatezza, limitando l'impatto dannoso della anomalia stessa e provvedendo a risanare il malfunzionamento, in modo da tornare alle corrette condizioni di esercizio.

Bibliografia

[1] V. Chandola, A. Banerjee e V. Kumar, *Anomaly detection: A Survey*, ACM Computing Surveys, vol. 41, n. 3, 2009.

[2] D. M. Hawkins, *Identification of Outliers*, London: Chapman and Hall, 1980

[3] Barbara, D., Couto, J., Jajodia, S., and Wu, N. 2001b. *Detecting novel network intrusions using Bayes estimators. In Proceedings of the 1st SIAM International Conference on Data Mining.*

[4] Stefano, C., Sansone, C., and Vento, M. 2000. *To reject or not to reject: that is the question: An answer in the case of neural classifiers.*

[5] E. Knorr e R. Ng, *Algorithms for Mining Distance-Based Outliers in Large Datasets*, Proc. of the VLDB Conference, pp. 392-403, 1998.

[6] Grigg, Olivia A., V. T. Farewell, and D. J. Spiegelhalter. *Use of risk-adjusted CUSUM and RSPRTcharts for monitoring in medical contexts.* Statistical methods in medical research 12.2 (2003): 147-170.

[7] Gustafsson, Fredrik, and Fredrik Gustafsson. *Adaptive filtering and change detection.* . Vol. 1. New York: Wiley, 2000.

[8] Marcos Duarte, Laboratory of Biomechanics and Motor Control, Federal University of ABC, Brazil. *Detection of changes using the Cumulative Sum (CUSUM).*

[9] Basseville, Michèle, and Igor V. Nikiforov. *Detection of abrupt changes: theory and application*. Vol. 104. Englewood Cliffs: Prentice Hall, 1993.

[10] Milena Gabanelli. *Acqua potabile, una rete colabrodo: si perdono 274mila litri al minuto*. http://www.utilitalia.it/dms/file/open/?d35efb39-ee9b-4ceb-a42f-c8a2a41eaeb9, Corriere della sera, 15 maggio 2018.

[11] Misiunas, Dalius, et al. *Burst detection and location in water distribution networks.* Water Science and Technology: Water Supply 5.3-4 (2005): 71-80.

[12] Farley, Malcolm, et al. *Leakage management and control: a best practice training manual.* No. WHO/SDE/WSH/01.1. Geneva: World Health Organization, 2001.

[13] Thornton, J., Sturm, R. and Kunkel, G. *Water Loss Control.* McGraw-Hill, 2008.

[14] Morrison, John. *Managing leakage by District Metered Areas: a practical approach.* Water 21 (2004): 44-46.

[15] Obradovic, DuŽan. *Modelling of demand and losses in real-life water distribution systems.* Urban Water 2.2 (2000): 131-139.

[16] Bakker, M., et al. *Reducing customer minutes lost by anomaly detection?* WDSA 2012: 14th Water Distribution Systems Analysis Conference, 24-27 September 2012 in Adelaide, South Australia. Engineers Australia, 2012.

[17] Slay, Jill, and Michael Miller. *Lessons learned from the maroochy water breach.* International Conference on Critical Infrastructure Protection. Springer, Boston, MA, 2007.

[18] Ramotsoela, Daniel, Adnan Abu-Mahfouz, and Gerhard Hancke. *A survey of anomaly detection in industrial wireless sensor networks with critical water system infrastructure as a case study.*Sensors 18.8 (2018): 2491.

[19] Worm, G. I. M., et al. *Integration of models, data management, interfaces and training support in a drinking water treatment plant simulator.*Environmental Modelling e Software 25.5 (2010): 677-683.

[20] it.wikipedia.org/wiki/SCADA. *SCADA.* Wikipedia, l'enciclopedia libera 1/06/2019 .

[21] www.acquacampania.com/i-sistemi-di-telecontrollo/. *Telecontrollo acqua campania.* Telecontrollo acqua Campania .

[22] en.wikipedia.org/wiki/Early-warning-system. *Early warning system.* Wikipedia, l'enciclopedia libera 27/03/2019 .

[23] Waidyanatha, Nuwan. *Towards a typology of integrated functional early warning systems.*Sensors 18.8 (2018): 2491.

[24] . *www.ilmeteo.it/portale/archivio-meteo/Avigliana.* Temperature riscontrate nel comune di Avigliana.

[25] William S. Cleveland, *Visualizing Data.* At e T Bell Laboratories, Murray Hill, New Jersey,1993.

[26] Matteo Zanardi. *implementazione di algoritmi di data mining in architetture ad elevato parallelismo.* Tesi di laurea in ingegneria informatica, università di Bologna

[27] Lambert, A., Hirner, W. (2000). *Losses from Water Supply Systems: Astandard Terminology and Recommended Performance Measures.* IWA, 2000.

[28] Farley, Malcolm, et al. *Leakage management and control A best practice training manual.* No. WHO/SDE/WSH/01.1. Geneva: World Health Organization, 2001.

[29] . */www.it-intesis.it/it/home/soluzioni/risorsa-idrica/distribuzione-idrica-nei-centri-urbani*. Telecontrollo idrico dell'abitato di potenza.

Appendices

Appendice A

Scripts Python

In questa appendice saranno riportati gli scrpts formulati su Python.

A.1 CUSUM Test

```
from __future__ import division, print_function
# -*- coding: utf-8 -*-

"""
Created on Wed Apr 10 11:21:44 2019

@author: Arciuli
"""

name = 'prova' #inizializzo un nome, mi servirà per poter salvare i
                                    vari CUSUM test con nomi diversi

#CUSUM ALGORITMO

# %load ./../functions/detect_cusum.py
"""Cumulative sum algorithm (CUSUM) to detect abrupt changes in
                                    data."""


import numpy as np

__author__ = 'Marcos Duarte, https://github.com/demotu/BMC'
__version__ = "1.0.4"
__license__ = "MIT"
```

```
def detect_cusum(x, threshold=1, drift=0, ending=False, show=True,
                                    ax=None):
    """Cumulative sum algorithm (CUSUM) to detect abrupt changes in
                                        data.

    Parameters
    ----------
    x : 1D array_like
        data.
    threshold : positive number, optional (default = 1)
        amplitude threshold for the change in the data.
    drift : positive number, optional (default = 0)
        drift term that prevents any change in the absence of
                                            change.
    ending : bool, optional (default = False)
        True (1) to estimate when the change ends; False (0)
                                            otherwise.
    show : bool, optional (default = True)
        True (1) plots data in matplotlib figure, False (0) don't
                                            plot.
    ax : a matplotlib.axes.Axes instance, optional (default = None)
                                        .

    Returns
    -------
    ta : 1D array_like [indi, indf], int
        alarm time (index of when the change was detected).
    tai : 1D array_like, int
        index of when the change started.
    taf : 1D array_like, int
        index of when the change ended (if `ending` is True).
    amp : 1D array_like, float
        amplitude of changes (if `ending` is True).


    Note that by default repeated sequential changes, i.e., changes
                                        that have
    the same beginning (`tai`) are not deleted because the changes
                                        were
    detected by the alarm (`ta`) at different instants. This is how
                                        the
    classical CUSUM algorithm operates.

    If you want to delete the repeated sequential changes and keep
                                        only the
    beginning of the first sequential change, set the parameter `
                                        ending` to
```

```
    True. In this case, the index of the ending of the change (‘taf
                                        ‘) and the
    amplitude of the change (or of the total amplitude for a
                                        repeated
    sequential change) are calculated and only the first change of
                                        the repeated
    sequential changes is kept. In this case, it is likely that ‘ta
                                        ‘, ‘tai‘,
    and ‘taf‘ will have less values than when ‘ending‘ was set to
                                        False.

    See this IPython Notebook [2]_.

    References
    ----------
    .. [1] Gustafsson (2000) Adaptive Filtering and Change
                                        Detection.
    .. [2] http://nbviewer.ipython.org/github/demotu/BMC/blob/
                                        master/notebooks/DetectCUSUM.
                                        ipynb
    """


    x = np.atleast_1d(x).astype('float64') #Convert inputs to
                                        arrays with at least one
                                        dimension.
    gp, gn = np.zeros(x.size), np.zeros(x.size)
    ta, tai, taf = np.array([[], [], []], dtype=int) #t allarme, t
                                        inizio cambiamento, t fine
                                        cambiamento
    tap, tan = 0, 0  # rappresenato nel ciclo le t di g+ e g-
    amp = np.array([]) #ampiezza del cambiamento
    # Find changes (online form)
    for i in range(1, x.size):
        s = x[i] - x[i-1]
        gp[i] = gp[i-1] + s - drift  # cumulative sum for + change
                                            (g+)
        gn[i] = gn[i-1] - s - drift  # cumulative sum for - change
                                            (g-)
        if gp[i] < 0:
            gp[i], tap = 0, i
        if gn[i] < 0:
            gn[i], tan = 0, i
        if gp[i] > threshold or gn[i] > threshold:  # change
                                            detected!
            ta = np.append(ta, i)    # alarm index
            tai = np.append(tai, tap if gp[i] > threshold else tan)
                                                # start
```

```
            gp[i], gn[i] = 0, 0      # reset alarm
    # THE CLASSICAL CUSUM ALGORITHM ENDS HERE

    # Estimation of when the change ends (offline form)
    if tai.size and ending:
        _, tai2, _, _ = detect_cusum(x[::-1], threshold, drift,
                                     show=False) # x[::-1]
                                     work to make a copy of
                                     the same list in reverse
                                     order
        #--> tai2 rappresenta il tempo in cui inizia un'anomalia
                                     partedo dalla fine dell x
                                      verso l'inizio
        taf = x.size - tai2[::-1] - 1 # si è quindi trovato quando
                                     finisce l'anomalia

        # Eliminate repeated changes, changes that have the same
                                     beginning
        tai, ind = np.unique(tai, return_index=True)
        ta = ta[ind]
        # taf = np.unique(taf, return_index=False)  # corect later
        if tai.size != taf.size: #se sono diversi
            if tai.size < taf.size:
                taf = taf[[np.argmax(taf >= i) for i in ta]]  #
                                            argmax returns
                                            the position of
                                            the largest value
            else:
                ind = [np.argmax(i >= ta[::-1])-1 for i in taf]
                ta = ta[ind]
                tai = tai[ind]
        # Delete intercalated changes (the ending of the change is
                                     after
        # the beginning of the next change)
        ind = taf[:-1] - tai[1:] > 0
        if ind.any():
            ta = ta[~np.append(False, ind)]
            tai = tai[~np.append(False, ind)]
            taf = taf[~np.append(ind, False)]
        # Amplitude of changes
        amp = x[taf] - x[tai]

    if show:
        _plot(x, threshold, drift, ending, ax, ta, tai, taf, gp, gn
                                     )

    return ta, tai, taf, amp
```

```
def _plot(x, threshold, drift, ending, ax, ta, tai, taf, gp, gn):
    """Plot results of the detect_cusum function, see its help."""

    try:
        import matplotlib.pyplot as plt
    except ImportError:
        print('matplotlib is not available.')
    else:
        if ax is None:
            _, (ax1, ax2) = plt.subplots(2, 1, figsize=(8, 6))

        t = range(x.size)
        ax1.plot(t, x, 'b-', lw=2)
        if len(ta):
            ax1.plot(tai, x[tai], '>', mfc='g', mec='g', ms=10,
                     label='Start')
            if ending:
                ax1.plot(taf, x[taf], '<', mfc='g', mec='g', ms=10,
                         label='Ending')
            ax1.plot(ta, x[ta], 'o', mfc='r', mec='r', mew=1, ms=5,
                     label='Alarm')
            ax1.legend(loc='best', framealpha=.5, numpoints=1)
        ax1.set_xlim(-.01*x.size, x.size*1.01-1)
        ax1.set_xlabel('Data #', fontsize=14)
        ax1.set_ylabel('Amplitude', fontsize=14)
        ymin, ymax = x[np.isfinite(x)].min(), x[np.isfinite(x)].max
                                              ()
        yrange = ymax - ymin if ymax > ymin else 1
        ax1.set_ylim(ymin - 0.1*yrange, ymax + 0.1*yrange)
        ax1.set_title('Time series and detected changes ' +
                      '(threshold= %.3g, drift= %.3g): N changes =
                                                          %d'
                      % (threshold, drift, len(tai)))
        ax2.plot(t, gp, 'y-', label='g+')
        ax2.plot(t, gn, 'm-', label='g-')
        ax2.set_xlim(-.01*x.size, x.size*1.01-1)
        ax2.set_xlabel('Data #', fontsize=14)
        ax2.set_ylim(-0.01*threshold, 1.1*threshold)
        ax2.axhline(threshold, color='r')
        ax1.set_ylabel('Amplitude', fontsize=14)
        ax2.set_title('Time series of the cumulative sums of ' +
                      'positive and negative changes')
        ax2.legend(loc='best', framealpha=.5, numpoints=1)
        plt.tight_layout()
        #plt.savefig('Immagini/CUSUM_Avigliana/' + str(name) + '.
                                              png')
        plt.show()
```

A.2 CUSUM Real Time

Di seguito viene riportata la funzione *Anna*, in grado di lavorare in real-time.

```
def Anna(file_name, formato = 0, delta_t_in = 365, delta_t_fn = 14,
                                    tipologia = 0, freq_camp = '15T'
                                   ):
#N.B per avere risultati attendibili, assicurarsi che nel database
                                   caricato non ci siano anomalie
                                   tali
#da comprometterne medie e deviazioni standard mensili, in caso
                                   contrario escludere tali mega-
                                   anomalie dal database

#file_name = se file .h5 scrivere il nome del file senza il _DB
                                   finale (il file deve essere
                                   inserito nel percorso: Dropbox\
                                   TesiCarlo\LAB\Data\Avigliana)
#           se file .csv scrivere il nome del file (il file deve
                                   essere inserito nel percorso:
                                   Dropbox\TesiCarlo\LAB\Data\
                                   Avigliana)
#formato   = 0 se il file è .h5
#          = 1 se il file è .h5
#delta_t_in= è il numero di giorni da sottrarre all'ultimo dato
                                   fornito, in queso modo si
                                   fornisce un'inizio su cui tale
                                   funzione effettuerà l'analisi
#delta_t_fn= è il numero di minuti da aggiungere all'ultimo dato
                                   fornito, in queso modo si
                                   fornisce una fine diversa dall'
                                   ultimo dato
#           si fa notare che per comprendere l'ultimo dato del
                                   database deve essere inserito un
                                   numero positivo (di default è 14
                                   min perchè la discretizzazione è
                                   impostata ogni 15 min)
#           inoltre inserendo un tempo negativo (in minuti) si può
                                    orendere un dato antecedente all
                                   'ultimo.
#tipologia = 0 se anomalie contestuali (dove il contesto è
                                   rappresentato dall'orario, ad es:
                                    andamento delle portate
                                   giornaliere)
```

```
#          = 1 se anomalie puntuali (in cui il database ha sempre
                                     lo stesso andamento periodico
                                     indipendentemente dal tempo, ad
                                     es:serbatoio che si svuota e si
                                     riempie)
#          = 2 in caso di grosse anomalie, qui vengono utilizzati
                                     dati non normalizzati e non
                                     dipendenti dal tempo
#            N.B in questo caso essendo valori non normalizzati e
                                     quindi h molto grandi, qui il
                                     valore di drift è stato posto v=0
#freq_camp = indicare la frequnza di campionamento dei dati, ad
                                     esempio 15T=15minuti, D=1giorno



    import matplotlib.pyplot as plt
    import pandas as pd
    import numpy as np
    import sys
    import datetime

    #annullo un falso allorma che a perdere tempo per la
                                          compilazione del codice
    pd.options.mode.chained_assignment = None  # default='warn'


    #CARICO FILE CON DATA E VARIABILE


    if formato == 0: #carico il file.h5

        if sys.platform.startswith('win'):
            filepath = 'C:\Users\Arciuli\Dropbox\TesiCarlo\LAB\Data
                                              /Avigliana/' + str(
                                              file_name)+ '_DB.h5'
        else:
            filepath = '/home/Marco/Dropbox/TesiCarlo/LAB/data/
                                              Avigliana/' + str(
                                              file_name)+ '_DB.h5'


        df = pd.read_hdf(filepath, str(file_name)+'RT')#la colonna
                                            sarà il nome del file +
                                            RT
        df = df.resample(str(freq_camp)) #discretizzo ogni 15 min
```

```
        df = pd.Series.to_frame(df)#passo il file da series a
                                        datatimeindex

        df.index.names = ['Data']#cambio i nomi di index e colonna
        df['Variabile'] = df[str(file_name)]
        del df[str(file_name)]


    if formato == 1: #carico il file.csv

        if sys.platform.startswith('win'):
            filepath= 'C:\Users\Arciuli\Dropbox\TesiCarlo\LAB\Data\
                                            Avigliana/'+ str(
                                            file_name) +'.csv'
        else:
            filepath = '/home/Marco\Dropbox\TesiCarlo\LAB\Data\
                                            Avigliana/'+ str(
                                            file_name) +'.csv'

        fields = ['Data/Ora', 'Media']
        df = pd.read_csv(filepath, usecols=fields, delimiter=';' )
        df['Variabile'] = df['Media']
        del df['Media']

        df['Data/Ora'] = pd.to_datetime(df['Data/Ora'], dayfirst=
                                        True)#trasformo in date e
                                         specifo che nel file c'
                                        erano i giorni prima dei
                                        mesi
        df = df.set_index('Data/Ora')

        df.index.names = ['Data']#cambio i nomi di index e colonna
        #df = df.resample('15T') #discretizzo ogni 15 min
        #df = pd.Series.to_frame(df)#passo il file da series a
                                        datatimeindex
        df = df.resample(str(freq_camp)) #discretizzo ogni 15 min


    df.Variabile = df.Variabile.fillna(method='ffill') #riempio il
                                    file in caso di qualche NAN

    #imporgo un inzio ed una fine al database che userò per
                                    eseguire l'analisi
    start = pd.to_datetime(df.index[-1] - datetime.timedelta(days=
                                    int(delta_t_in))) #data di
                                    partenza di analisi della
                                    serie
```

```
end = pd.to_datetime(df.index[-1]) + datetime.timedelta(
                                    minutes=int(delta_t_fn)) #
                                    data di fine di analisi della
                                     serie


if tipologia == 0:


        df_vol = df.loc[(df.index.to_datetime() >= str(start))
                                        & (df.index.
                                        to_datetime() < str(
                                        end))] #selezione la
                                        parte del file su cui
                                         effettuerò l'analisi
        df_vol = df_vol.between_time(str(start),str(end))  #
                                        serie temporale (
                                        orario) che si vuole
                                        analizzare


        #NORMALIZZAZIONE DEI DATI
        #creo una colonna temporanea copia dell'index per poter
                                         successivamente
                                        individuare mese e
                                        anno di ogni dato
        df_vol['temp'] = df_vol.index.copy()
        df_vol.temp = pd.to_datetime(df_vol.temp)

        #inizializzo le colonne di mese ed anno che poi
                                        andranno riempite
        df['Mese'] = np.zeros(len(df))
        df['Anno'] = np.zeros(len(df))

        # N.B  aggiungo mese ed anno a df e non a df_vol perchè
                                         altrimate python non
                                         funziona
        for x in df_vol.index:
            df.Mese[x] = df_vol.temp[x].month
            df.Anno[x] = df_vol.temp[x].year

        # riprendo la serie temporale di interesse, stavolta ci
                                         saranno anche le
                                        colonne di mese e
                                        anno
```

```
start = pd.to_datetime(df.index[-1] - datetime.
                                    timedelta( days= int(
                                    delta_t_in))) #data
                                    di partenza di
                                    analisi della serie
end = pd.to_datetime(df.index[-1]) + datetime.timedelta
                                    ( minutes=int(
                                    delta_t_fn)) #data di
                                     fine di analisi
                                    della serie
df_vol = df.loc[(df.index.to_datetime() >= str(start))
                                    & (df.index.
                                    to_datetime() < str(
                                    end))]#selezione la
                                    parte del file su cui
                                     effettuerò l'analisi
df_vol = df_vol.between_time(str(start),str(end))  #
                                    serie temporale che
                                    si vuole analizzare

#individuo la media e la std di ogni mese del database
df = df.loc[(df.index.to_datetime() >= str(start)) & (
                                    df.index.to_datetime
                                    () < str(end))]
stat = df.groupby([df.index.year, df.index.month]).agg(
                                    ['mean', 'std'])#
                                    trovo media e
                                    deviazione standard
                                    di ogni mese
stat.index.set_names(['Year', 'Month'], inplace = True)
                                    #metto in odrne il
                                    vettore stat


#calcolo media e std di ogni dato
df_vol['media_mens'] = np.zeros(len(df_vol))
for x in df_vol.index:
    df_vol.media_mens[x] = stat.Variabile.loc[([int(
                                        df_vol.Anno[x])],
                                        [int(df_vol.Mese[
                                        x])]),'mean']
df_vol['std_mens'] = np.zeros(len(df_vol))
for i in range(len(df_vol)):
   df_vol.std_mens[i] = stat.Variabile.loc[([int(df_vol
                                       .Anno[i])],[int(
                                       df_vol.Mese[i])]),
                                       'std']
```

```
#calcolo la rispettiva normalizzazione del dato
df_vol['norm'] = np.zeros(len(df_vol))
for x in range(len(df_vol)):
    df_vol.norm[x] = (df_vol.Variabile[x]- df_vol.
                                        media_mens[x])/
                                        df_vol.std_mens[x]


#applico il CUSUM test
q1 = df_vol.norm.quantile(0.04)#quantile inferiore
q3 = df_vol.norm.quantile(0.97)#quantile superiore

#df_vol.norm.std() Gustafsson pone v=5teta e h=5*teta
                                    ma ci sta male

h = q3 - q1 #valore di soglia
x = df_vol.norm #inserisco nel Cusum test il valori
                                    normalizzati dei dati
ta, tai, taf, amp = detect_cusum(x, h, 0.05*h, True,
                                    True)
#print ('ampiezza e Qnorm')
#print (amp, x[ta])
df2 = pd.DataFrame(np.transpose([(x.index[ta]), x[ta]])
                                    , columns=['Data', 'z
                                    '])#creo un database
                                    con tutte le anomalie
                                     riscontrate nella
                                    serie
df2.Data = df2.Data.apply(lambda t: t.strftime('%Y-%m-%
                                    d, %H:%M:%S'))#cambio
                                     il formato delle
                                    date



print('Le seguenti sono tutte le anomalie riscontrate',
                                     df2)


try:   #uso il try perchè se df2 fosse vuoto (quindi
                                    non sono presenti
                                    anomalie in tutto il
                                    database) non potrei
                                    prendere l'ultima
                                    riga
```

```
anom =  pd.to_datetime(df2['Data'].values[-1])#
                                    selezione l'
                                    ultimo dato
                                    anomalo della
                                    serie
Anom = pd.DataFrame( pd.TimeSeries(anom), columns=[
                                    'Data'])#creo un
                                    dataframe con il
                                    dato anomalo
Anom['Mese'] = anom.month#aggiungo il mese
Anom['Anno'] = anom.year#aggiungo l'anno
Anom['norm'] = df2['z'].values[-1]#aggiungo il dato
                                     normalizzato
Anom['Variabile'] = float (df2['z'].values[-1]*stat
                                    .Variabile.loc[[
                                    int(Anom['Anno'].
                                    values[-1])],[int
                                    (Anom['Mese'].
                                    values[-1])]),'
                                    std'] + (stat.
                                    Variabile.loc[[
                                    int(Anom['Anno'].
                                    values[-1])],[int
                                    (Anom['Mese'].
                                    values[-1])]),'
                                    mean']))#
                                    denormalizzo il
                                    dato

del Anom['Mese'], Anom['Anno']#cancello la colonna
                                    del mese e dell'
                                    anno per estetica
del df_vol['Mese'], df_vol['Anno']#cancello la
                                    colonna del mese
                                    e dell'anno per
                                    estetica


global Anom_si #in modo che risulti presente anche
                                    a ciclo concluso
#del Anom_si

#controllo se l'ultima anomali trovata è l'ultimo
                                    dato della serie
```

```
            if pd.to_datetime(Anom.Data.values[-1]) != pd.
                                                to_datetime(df.
                                                index[-1]): #se
                                                sono diversi
                                                allora il dato è
                                                normale
                    print (df_vol.tail(1), 'Tale valore è
                                                    NORMALE')

            if pd.to_datetime(Anom.Data.values[-1]) == pd.
                                                to_datetime(df.
                                                index[-1]): #se
                                                sono uguali
                                                allora il dato è
                                                anomalo


                try:
                    Anom_si = Anom_si.append(Anom)
                    print (df_vol.tail(1) ,'Tale valore è
                                                    ANOMALO')
                    return Anom_si
                    print (Anom_si)
                #except UnboundLocalError:
                except NameError: #se non c'erano altre
                                                anomalie
                                                allora ci
                                                sarà solo l'
                                                ultima
                                                anomalia
                    Anom_si = Anom
                    print (df_vol.tail(1) ,'Tale valore è
                                                    ANOMALO')
                    return Anom_si
        except IndexError: #se df2 è vuoto allora non ci sono
                                            anomalie in tutto il
                                            database
            print ('Nessun dato è anomalo in tutto il
                                                daatbase')


if tipologia == 1:
```

```
df_vol = df.loc[(df.index.to_datetime() >= str(start))
                                    & (df.index.
                                    to_datetime() < str(
                                    end))] #selezione la
                                    parte del file su cui
                                     effettuerò l'analisi

            #NORMALIZZAZIONE DEI DATI
#creo una colonna temporanea copia dell'index per poter
                                     successivamente
                                    individuare mese e
                                    anno di ogni dato
df_vol['temp'] = df_vol.index.copy()
df_vol.temp = pd.to_datetime(df_vol.temp)

#inizializzo le colonne di mese ed anno che poi
                                    andranno riempite
df['Mese'] = np.zeros(len(df))
df['Anno'] = np.zeros(len(df))

# N.B  aggiungo mese ed anno a df e non a df_vol perchè
                                     altrimate python non
                                     funziona
for x in df_vol.index:
    df.Mese[x] = df_vol.temp[x].month
    df.Anno[x] = df_vol.temp[x].year

# riprendo la serie temporale di interesse, stavolta ci
                                     saranno anche le
                                    colonne di mese e
                                    anno
start = pd.to_datetime(df.index[-1] - datetime.
                                    timedelta( days= int(
                                     delta_t_in))) #data
                                    di partenza di
                                    analisi della serie
end = pd.to_datetime(df.index[-1]) + datetime.timedelta
                                    ( minutes= int(
                                    delta_t_fn )) #data
                                    di fine di analisi
                                    della serie
df_vol = df.loc[(df.index.to_datetime() >= str(start))
                                    & (df.index.
                                    to_datetime() < str(
                                    end))]#selezione la
                                    parte del file su cui
                                     effettuerò l'analisi
```

```
#individuo la media e la std di ogni mese del database
df = df.loc[(df.index.to_datetime() >= str(start)) & (
                                    df.index.to_datetime
                                    () < str(end))]
stat = df.groupby([df.index.year, df.index.month]).agg(
                                    ['mean', 'std'])#
                                    trovo media e
                                    deviazione standard
                                    di ogni mese
stat.index.set_names(['Year', 'Month'], inplace = True)
                                    #metto in odrne il
                                    vettore stat


#calcolo media e std di ogni dato
df_vol['media_mens'] = np.zeros(len(df_vol))
for x in df_vol.index:
    df_vol.media_mens[x] = stat.Variabile.loc[([int(
                                        df_vol.Anno[x])],
                                        [int(df_vol.Mese[
                                        x])]),'mean']
df_vol['std_mens'] = np.zeros(len(df_vol))
for i in range(len(df_vol)):
   df_vol.std_mens[i] = stat.Variabile.loc[([int(df_vol
                                    .Anno[i])],[int(
                                    df_vol.Mese[i])]),
                                    'std']

#calcolo la rispettiva normalizzazione del dato
df_vol['norm'] = np.zeros(len(df_vol))
for x in range(len(df_vol)):
   df_vol.norm[x] = (df_vol.Variabile[x]- df_vol.
                                    media_mens[x])/
                                    df_vol.std_mens[x]



#applico il CUSUM test
q1 = df_vol.norm.quantile(0.04)#quantile inferiore
q3 = df_vol.norm.quantile(0.97)#quantile superiore
h = q3 - q1 #valore di soglia
x = df_vol.norm #inserisco nel Cusum test il valori
                                    normalizzati dei dati
ta, tai, taf, amp = detect_cusum(x, h, 0.05*h, True,
                                    True)
#print ('ampiezza e Qnorm')
#print (amp, x[ta])
```

```
df2 = pd.DataFrame(np.transpose([(x.index[ta]), x[ta]])
                                    , columns=['Data', 'z
                                    '])#creo un database
                                    con tutte le anomalie
                                     riscontrate nella
                                    serie
df2.Data = df2.Data.apply(lambda t: t.strftime('%Y-%m-%
                                    d, %H:%M:%S'))#cambio
                                     il formato delle
                                    date



print('Le seguenti sono tutte le anomalie riscontrate',
                                     df2)

try:    #uso il try perchè se df2 fosse vuoto (quindi
                                    non sono presenti
                                    anomalie in tutto il
                                    database) non potrei
                                    prendere l'ultima
                                    riga
    anom =  pd.to_datetime(df2['Data'].values[-1])#
                                        selezione l'
                                        ultimo dato
                                        anomalo della
                                        serie
    Anom = pd.DataFrame( pd.TimeSeries(anom), columns=[
                                        'Data'])#creo un
                                        dataframe con il
                                        dato anomalo
    Anom['Mese'] = anom.month#aggiungo il mese
    Anom['Anno'] = anom.year#aggiungo l'anno
    Anom['norm'] = df2['z'].values[-1]#aggiungo il dato
                                         normalizzato
```

```
            Anom['Variabile'] = float (df2['z'].values[-1]*stat
                                       .Variabile.loc[([
                                       int(Anom['Anno'].
                                       values[-1])],[int
                                       (Anom['Mese'].
                                       values[-1])]),'
                                       std'] + (stat.
                                       Variabile.loc[([
                                       int(Anom['Anno'].
                                       values[-1])],[int
                                       (Anom['Mese'].
                                       values[-1])]),'
                                       mean']))#
                                       denormalizzo il
                                       dato

            del Anom['Mese'], Anom['Anno']#cancello la colonna
                                       del mese e dell'
                                       anno per estetica
            del df_vol['Mese'], df_vol['Anno']#cancello la
                                       colonna del mese
                                       e dell'anno per
                                       estetica


            global Anom_si #in modo che risulti presente anche
                                       a ciclo concluso
            #del Anom_si

            #controllo se l'ultima anomali trovata è l'ultimo
                                       dato della serie
            if pd.to_datetime(Anom.Data.values[-1]) != pd.
                                       to_datetime(df.
                                       index[-1]): #se
                                       sono diversi
                                       allora il dato è
                                       normale
                    print (df_vol.tail(1), 'Tale valore è
                                                  NORMALE')

            if pd.to_datetime(Anom.Data.values[-1]) == pd.
                                       to_datetime(df.
                                       index[-1]): #se
                                       sono uguali
                                       allora il dato è
                                       anomalo
                try:
                    Anom_si = Anom_si.append(Anom)
```

```
                    print (df_vol.tail(1) ,'Tale valore è
                                                    ANOMALO')
                    return Anom_si
                    print (Anom_si)
                #except UnboundLocalError:
                except NameError: #se non c'erano altre
                                                anomalie
                                                allora ci
                                                sarà solo l'
                                                ultima
                                                anomalia
                    Anom_si = Anom
                    print (df_vol.tail(1) ,'Tale valore è
                                                    ANOMALO')
                    return Anom_si

        except IndexError: #se df2 è vuoto allora non ci sono
                                            anomalie in tutto il
                                            database
            print ('Nessun dato è anomalo in tutto il daatbase'
                                                )

if tipologia == 2: # in caso di grosse anomalie, qui vengono
                                        utilizzati dati non
                                        normalizzati e non dipendenti
                                         dal tempo



        df_vol = df.loc[(df.index.to_datetime() >= str(start))
                                            & (df.index.
                                            to_datetime() < str(
                                            end))] #selezione la
                                            parte del file su cui
                                             effettuerò l'analisi
        df_vol = df_vol.between_time(str(start),str(end))  #
                                            serie temporale (
                                            orario) che si vuole
                                            analizzare
        #applico il CUSUM test

        df_vol['norm'] = df_vol.Variabile.copy()

        q1 = df_vol.norm.quantile(0.05)#quantile inferiore
        q3 = df_vol.norm.quantile(0.95)#quantile superiore
        h = q3 - q1 #valore di soglia
        x = df_vol.norm #inserisco nel Cusum test il valori
                                            normalizzati dei dati
```

```
ta, tai, taf, amp = detect_cusum(x, h, 0, True, True)
#print ('ampiezza e Qnorm')
#print (amp, x[ta])
df2 = pd.DataFrame(np.transpose([(x.index[ta]), x[ta]])
                                    , columns=['Data', 'z
                                    '])#creo un database
                                    con tutte le anomalie
                                     riscontrate nella
                                    serie
df2.Data = df2.Data.apply(lambda t: t.strftime('%Y-%m-%
                                    d, %H:%M:%S'))#cambio
                                     il formato delle
                                    date

print('Le seguenti sono tutte le anomalie riscontrate',
                                     df2)


try:    #uso il try perchè se df2 fosse vuoto (quindi
                                    non sono presenti
                                    anomalie in tutto il
                                    database) non potrei
                                    prendere l'ultima
                                    riga
    anom =  pd.to_datetime(df2['Data'].values[-1])#
                                        selezione l'
                                        ultimo dato
                                        anomalo della
                                        serie
    Anom = pd.DataFrame( pd.TimeSeries(anom), columns=[
                                        'Data'])#creo un
                                        dataframe con il
                                        dato anomalo


    global Anom_si #in modo che risulti presente anche
                                        a ciclo concluso
    #del Anom_si

    #controllo se l'ultima anomali trovata è l'ultimo
                                        dato della serie
    if pd.to_datetime(Anom.Data.values[-1]) != pd.
                                        to_datetime(df.
                                        index[-1]): #se
                                        sono diversi
                                        allora il dato è
                                        normale
```

```
                print (df_vol.tail(1), 'Tale valore è
                                                NORMALE')

        if pd.to_datetime(Anom.Data.values[-1]) == pd.
                                        to_datetime(df.
                                        index[-1]): #se
                                        sono uguali
                                        allora il dato è
                                        anomalo
            try:
                Anom_si = Anom_si.append(Anom)
                print (df_vol.tail(1) ,'Tale valore è
                                                ANOMALO')
                return Anom_si
                print (Anom_si)
            #except UnboundLocalError:
            except NameError: #se non c'erano altre
                                            anomalie
                                            allora ci
                                            sarà solo l'
                                            ultima
                                            anomalia
                Anom_si = Anom
                print (df_vol.tail(1) ,'Tale valore è
                                                ANOMALO')
                return Anom_si

    except IndexError: #se df2 è vuoto allora non ci sono
                                        anomalie in tutto il
                                        database
        print ('Nessun dato è anomalo in tutto il daatbase'
                                        )
```

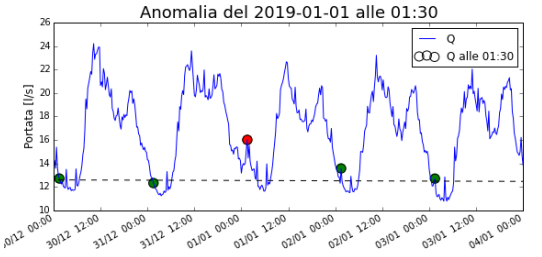
Appendice B

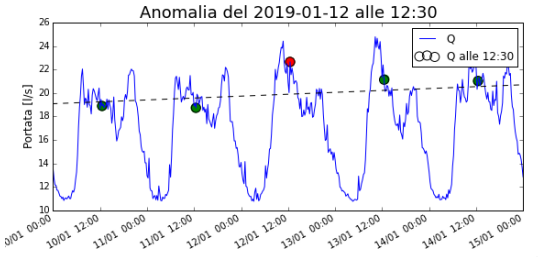
Anomalie controllate

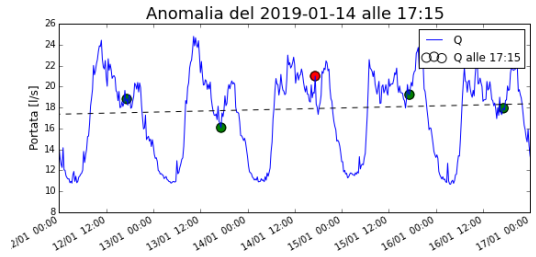
In questa appendice saranno riportate le microanomalie relative alla centrale di Avigliana e di Cavoretto.

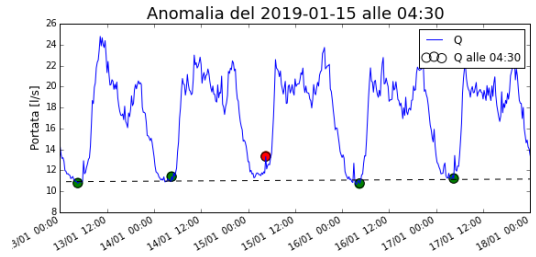
B.1 Anomalie Avigliana

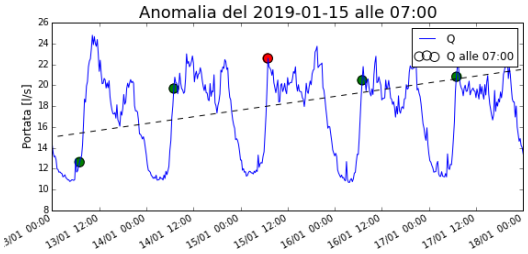
In questa sezione saranno riportate le microanomalie appartenenti la centrale di Avigliana a partire dal 01-01-2019 al 30-04-2019, riscontrate tramite l'utilizzo del CUSUM test automatizzato.

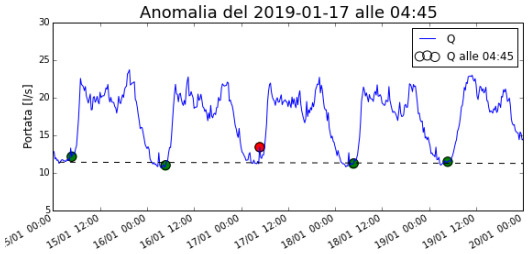


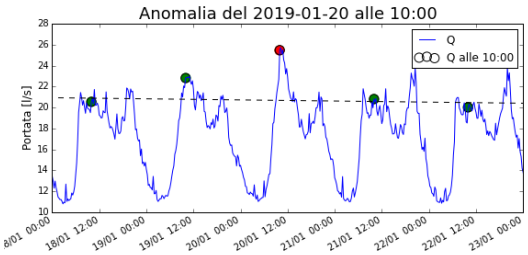


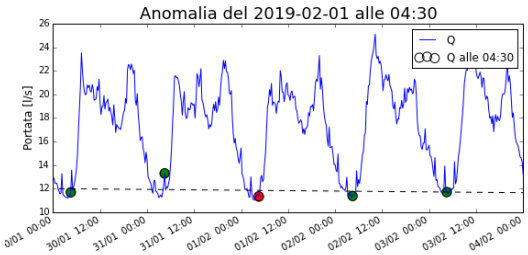


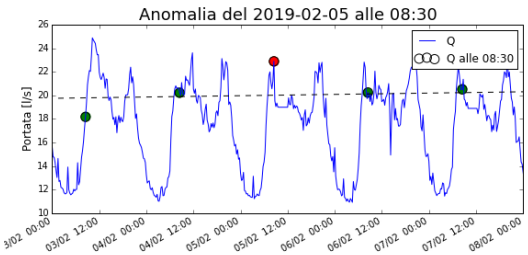


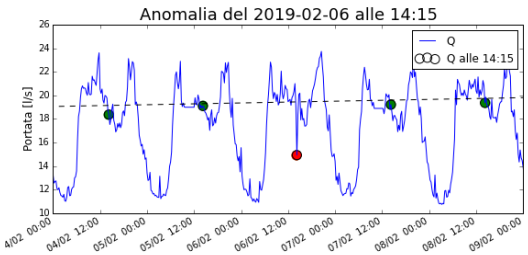

Anomalia del 2019-01-15 alle 07:00
Portata [l/s]
Q
Q alle 07:00

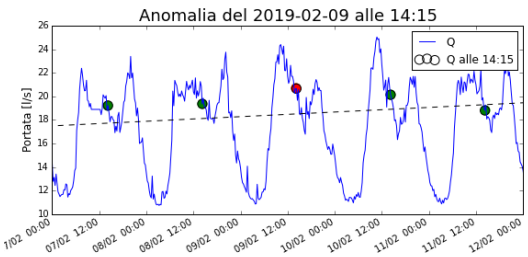

Anomalia del 2019-01-17 alle 04:45
Portata [l/s]
Q
Q alle 04:45

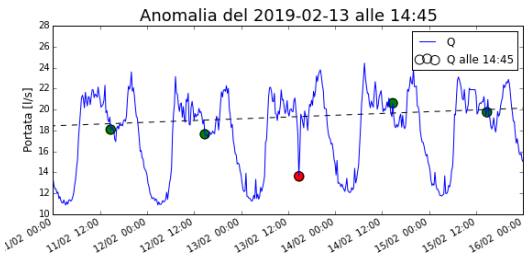

Anomalia del 2019-01-20 alle 10:00
Portata [l/s]
Q
Q alle 10:00

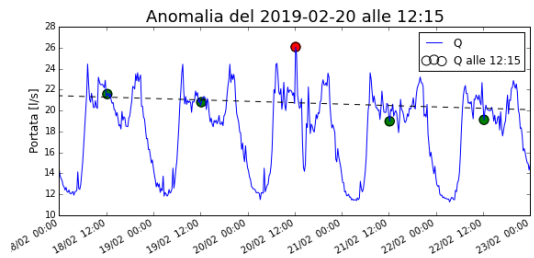

Anomalia del 2019-02-01 alle 04:30
Portata [l/s]
Q
Q alle 04:30

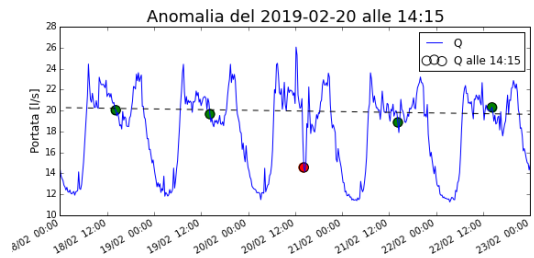

Anomalia del 2019-02-05 alle 08:30
Portata [l/s]
Q
Q alle 08:30

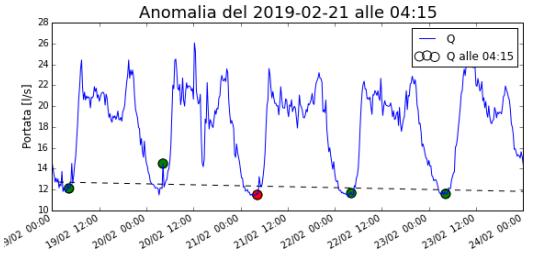

Anomalia del 2019-02-06 alle 14:15
Portata [l/s]
Q
Q alle 14:15

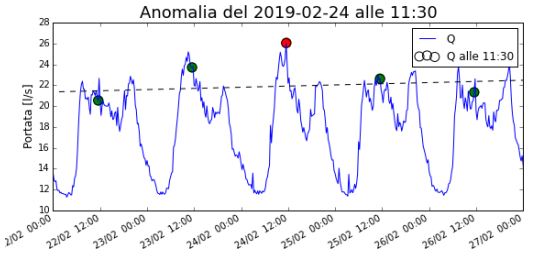

Anomalia del 2019-02-09 alle 14:15
Portata [l/s]
Q
Q alle 14:15

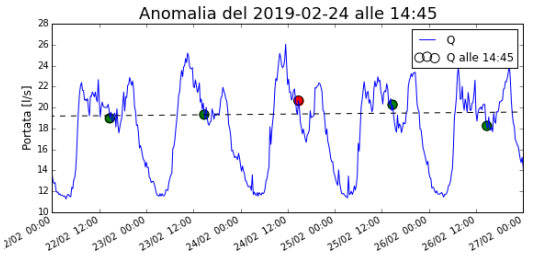

Anomalia del 2019-02-13 alle 14:45
Portata [l/s]
Q
Q alle 14:45

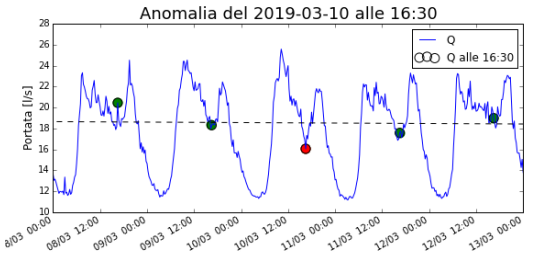

Anomalia del 2019-02-20 alle 12:15
Portata [l/s]
Q
Q alle 12:15

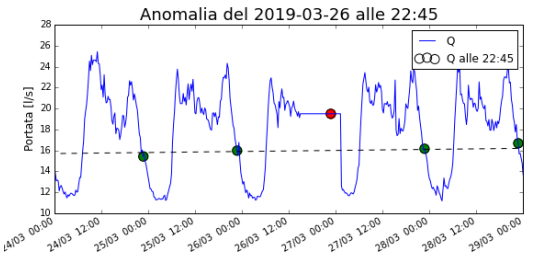

Anomalia del 2019-02-20 alle 14:15
Portata [l/s]
Q
Q alle 14:15

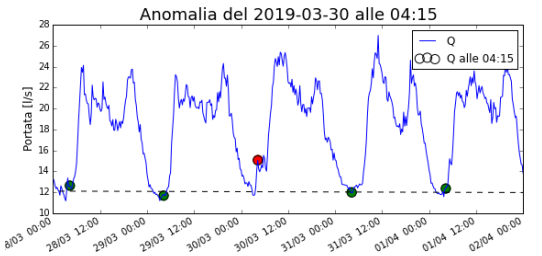

Anomalia del 2019-02-21 alle 04:15
Q
Q alle 04:15
Portata [l/s]

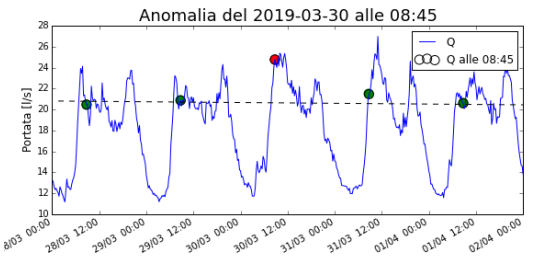

Anomalia del 2019-02-24 alle 11:30
Q
Q alle 11:30
Portata [l/s]

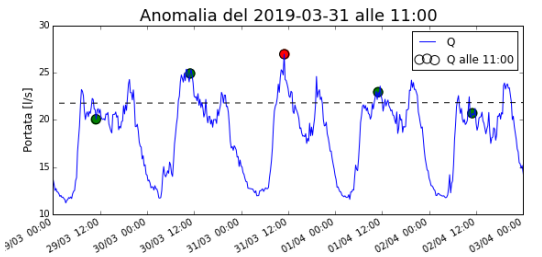

Anomalia del 2019-02-24 alle 14:45
Q
Q alle 14:45
Portata [l/s]

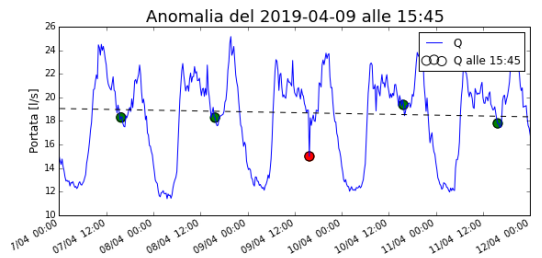

Anomalia del 2019-03-10 alle 16:30
Q
Q alle 16:30
Portata [l/s]

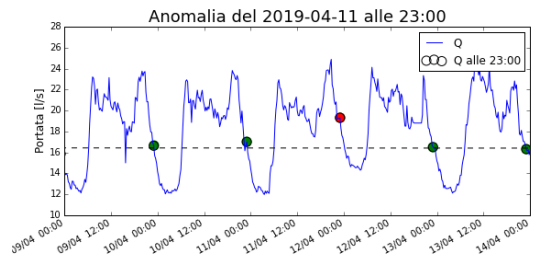

Anomalia del 2019-03-26 alle 22:45
Q
Q alle 22:45
Portata [l/s]

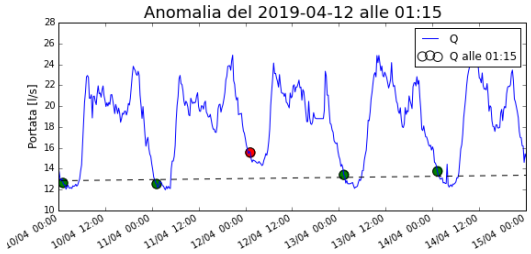

Anomalia del 2019-03-30 alle 04:15
Q
Q alle 04:15
Portata [l/s]

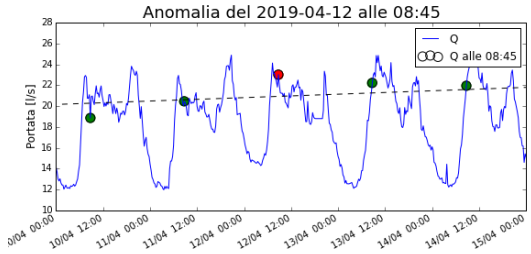

Anomalia del 2019-03-30 alle 08:45
Q
Q alle 08:45
Portata [l/s]

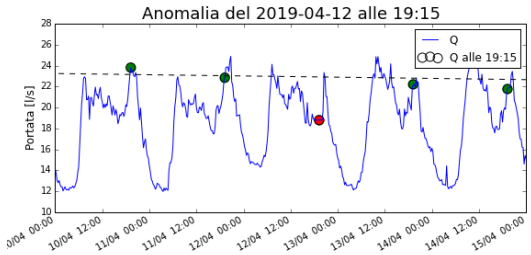

Anomalia del 2019-03-31 alle 11:00
Q
Q alle 11:00
Portata [l/s]

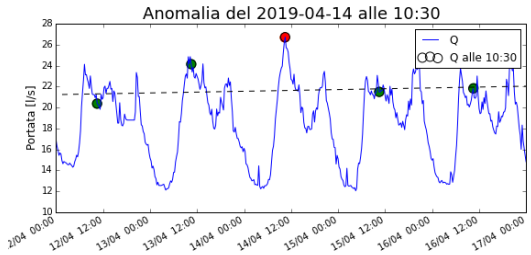

Anomalia del 2019-04-09 alle 15:45
Q
Q alle 15:45
Portata [l/s]

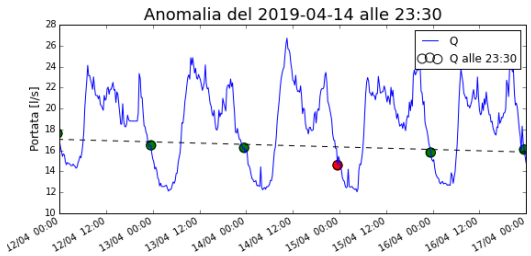

Anomalia del 2019-04-11 alle 23:00
Q
Q alle 23:00
Portata [l/s]

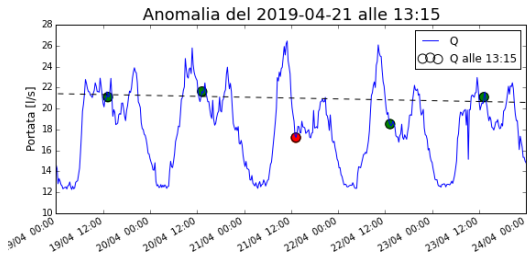

Anomalia del 2019-04-12 alle 01:15
Portata [l/s]
Q
Q alle 01:15

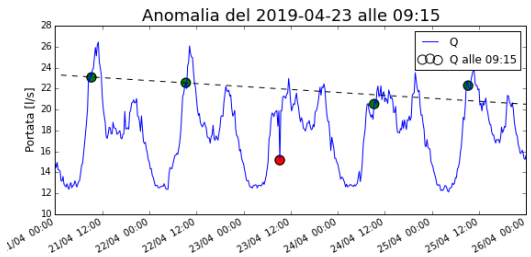

Anomalia del 2019-04-12 alle 08:45
Portata [l/s]
Q
Q alle 08:45

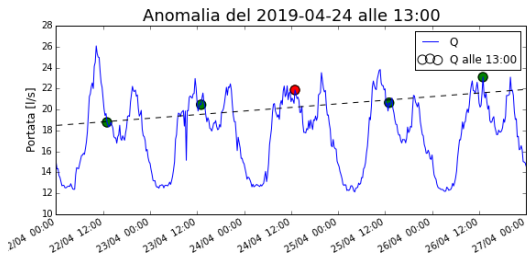

Anomalia del 2019-04-12 alle 19:15
Portata [l/s]
Q
Q alle 19:15

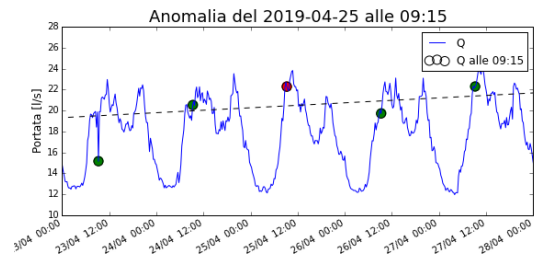

Anomalia del 2019-04-14 alle 10:30
Portata [l/s]
Q
Q alle 10:30

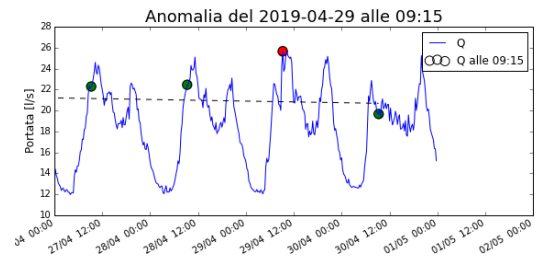

Anomalia del 2019-04-14 alle 23:30
Portata [l/s]
Q
Q alle 23:30


Anomalia del 2019-04-21 alle 13:15
Portata [l/s]
Q
Q alle 13:15


Anomalia del 2019-04-23 alle 09:15
Portata [l/s]
Q
Q alle 09:15

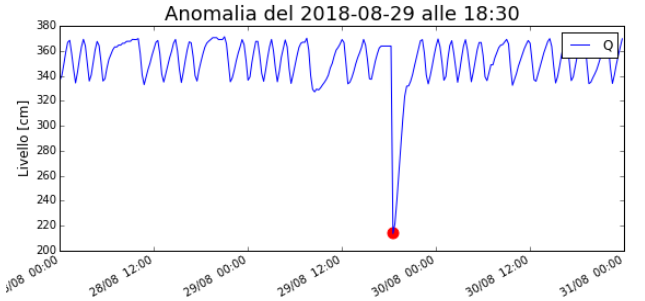

Anomalia del 2019-04-24 alle 13:00
Portata [l/s]
Q
Q alle 13:00

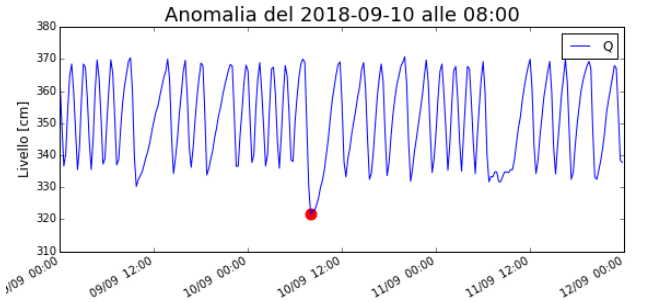

Anomalia del 2019-04-25 alle 09:15
Portata [l/s]
Q
Q alle 09:15

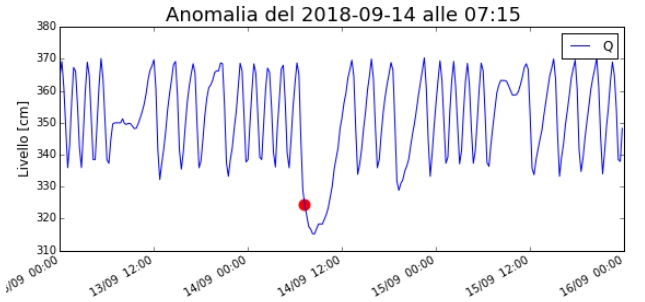

Anomalia del 2019-04-29 alle 09:15
Portata [l/s]
Q
Q alle 09:15

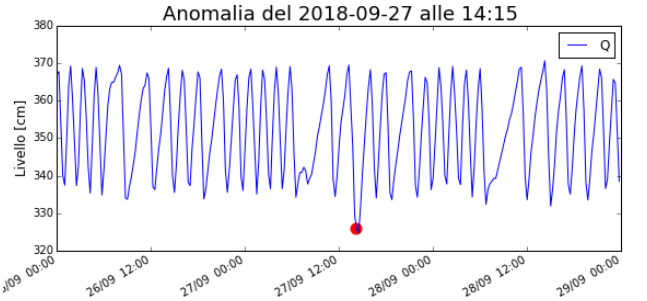
B.2 Anomalie Cavoretto

In questa sezione saranno riportate le microanomalie appartenenti la centrale di Cavoretto a partire dal 01-08-2018 al 31-12-2018, riscontrate tramite l'utilizzo del CUSUM test automatizzato.









Ringraziamenti