



UPPSALA
UNIVERSITET

Självständigt arbete i informationsteknologi
20 maj 2021

Identifiering av vattenläckor med hjälp av statistisk processkontroll

För en säkrare och pålitligare vattentillförsel

Martin Dannelind
Anton Nyström
Niklas Sundqvist
Filip Åslin



UPPSALA
UNIVERSITET

Institutionen för
informationsteknologi

Besöksadress:
ITC, Polacksbacken
Lägerhyddsvägen 2

Postadress:
Box 337
751 05 Uppsala

Hemsida:
<https://www.it.uu.se>

Abstract

Identifiering av vattenläckor med hjälp av statistisk processkontroll

För en säkrare och pålitligare vattentillförsel

Martin Dannelind
Anton Nyström
Niklas Sundqvist
Filip Åslin

Leaking pipes in your bathroom or kitchen can cause a lot of damage to both your property and wallet. Water leaks in your house or apartment are often easily spotted but how do you know if a pipe is leaking if it is located several meters below ground? This is a big problem for water suppliers around the world and in this project a desktop application was created to help find these water leaks. Deviations from the normal water flow are identified as leaks with the use of water flow data from pipes and statistical process control. The system managed to identify when a pipe was leaking using real world data but also identified deviations in the flow where there were no leaks.

Extern handledare: Karolina Gahne, Stuns Energi
Extern intressent: Uppsala Vatten och Avfall AB
Handledare: Björn Victor
Examinator: Björn Victor

Sammanfattning

Läckande vattenledningar i ditt badrum eller kök kan ha förödande konsekvenser på både egendom och plånbok. Läckor i ditt hus eller lägenhet är ofta lätta att hitta men hur vet du att en vattenledning läcker om det befinner sig flera meter under jord? Det här är ett stort problem för vattenförsörjare världen över och i detta projekt skapades en desktopapplikation som underlättar identifiering av läckor. Applikationen använder sig av data på vattenflödet i ledningar. Med hjälp av statistisk processkontroll så identifieras avvikelser från det normala vattenflödet som då anses vara läckor. Vid tester på riktig data från vattenledningar så lyckades systemet identifiera de läckor som hade skett. Däremot så indikerades läckor även i data där det inte hade skett någon läcka.

Innehåll

1	Introduktion	1
2	Bakgrund	1
3	Syfte, mål och motivation	3
3.1	Intressenter	3
3.1.1	Vattenproducenten	4
3.1.2	Konsumenten	4
3.1.3	Samhället	4
3.1.4	Naturen	5
3.2	Etiska aspekter	6
3.3	Avgränsningar	6
4	Relaterat arbete	7
4.1	Extern akustisk läckagedetektering	7
4.2	Intern detektering med massbalans	8
5	Metoder	9
5.1	Utvecklingsmiljöer	9
5.2	Olika typer av felaktig identifiering	9
5.3	Styrdiagram för identifiering av läckor	10
5.3.1	Shewhart-diagram	10
5.3.2	CUSUM-diagram	11
5.3.3	EWMA-diagram	14
5.4	Hantering av oregelbundet vattenflöde	16

6	Systemstruktur	16
6.1	Systemets tre huvuddelar och deras funktion	17
6.1.1	Datahanterare för omformatering av flödesdata	17
6.1.2	Styrdiagram för läcksökning	18
6.1.3	Grafiskt gränssnitt för användarvänlighet	18
7	Krav och utvärderingsmetoder	18
7.1	Krav	19
7.2	Utvärderingsmetoder	19
8	Implementation	20
8.1	Lagring av data	20
8.2	Hantering av flödesdata med hjälp av datahanterare	20
8.2.1	Omformatering av data	20
8.2.2	Uppdelning av data	21
8.3	Implementation av styrdiagrammen	22
8.3.1	Shewhart	22
8.3.2	CUSUM	23
8.3.3	EWMA	23
8.4	Grafiskt gränssnitt	24
9	Utvärderingsresultat	26
9.1	Identifiering av läckor	27
9.1.1	Resultat av dataset 1	27
9.1.2	Resultat av dataset 2	28
9.1.3	Resultat av dataset 3	28

9.1.4	Resultat av dataset 4	29
9.1.5	Resultat av dataset 5	30
9.2	Hantering av felaktigt filformat	30
9.3	Tidskomplexitet	31
10	Resultat och diskussion	31
10.1	Resultat	31
10.2	Diskussion	32
10.2.1	Val av styrdiagram	32
10.2.2	Säsongsvariation	33
10.2.3	Typ I-fel	33
10.2.4	Val av datahanterare	34
11	Slutsatser	35
12	Framtida arbete	35
12.1	Vattenledningar och vattenverk	35
12.2	Maskininlärning med hjälp av fler mätvariabler och dataset	35
12.3	Kontinuerlig läcksökning i realtid	36

1 Introduktion

Städer och samhällen är idag beroende av en stabil och tillförlitlig vattentillförsel. Vattenverk är de industrier som tillverkar kranvatten genom att rena vatten från sjöar, hav eller grundvatten. Vattenverken är stora arbetsgivare och deras vattennätverk med tillhörande faciliteter kräver konstant underhåll. Uppsala Vatten och Avfall AB, som i resten av denna rapporten kommer refereras till som Uppsala Vatten, omsatte 597 miljoner kronor under 2018 [Upp19]. För att stora företag ska vara ekonomiskt hållbara så är det gynnsamt att vara effektiva. Det kranvatten Uppsala Vatten producerar i sina vattenverk måste nå konsumenterna vilket innebär många svåra utmaningar och en av dem är vattenläckor. Uppsala Vatten säljer kranvatten till invånare och företag i kommunen men om vattnet inte når fram till konsumenten kan de inte debitera för det.

Många vattenledningar befinner sig under vägar och annan infrastruktur vilka försvårar arbetet att lokalisera läckor. Vattenledningar kan därför läcka i flera år utan att det upptäcks [Bä19]. Målet för projektet var att hitta de läckande vattenledningarna för att minska den mängd vatten som rinner ut i marken och inte kommer till användning.

Under projektets gång utvecklades en skrivbordsapplikation för att identifiera läckor i vattenledningar med hjälp av matematiska modeller. Dessa modeller appliceras på flödesdata från vattenledningar i Uppsala kommun och visar på abnormiteter i vattenflödet som kan tyda på att det finns läckor i ledningarna. För att visualisera resultatet i applikationen så presenteras diagram över de matematiska modellerna applicerade på flödesdatan samt en redovisning av de datum där programmet indikerat att läckor kan ha uppkommit i vattenledningen.

2 Bakgrund

Läckor i utbredda vattenledningsnät är ett stort problem då det i längden blir stora volymer vatten som förloras. Mängden vatten som går till förlust i kommuners vattennät varierar beroende på vilken kommun det gäller. Svenskt Vatten AB har kartlagt läckor och skriver att i genomsnitt förloras 15% av vattnet i form av läckage [Bä19]. De mäter inte verkliga läckor utan skillnaden mellan den vattenmängd som levereras till nätet och vattenmängden som förbrukas av kunderna. Med utgångspunkt att 100% är levererat vatten så debiteras kunderna i vattennätet för 75%. Av de kvarvarande 25 procenten utgörs ca 5 procentenheter av VA-verksamheter, förbrukning inom andra förvaltningar och brandvatten. Det innebär att hela 20% av vattenproduktionen förloras. Man räknar med en mängd otillåten användning som kan vara stöld av vatten eller förskingring. Fel på mätare kan medföra att små vattenmängder inte registreras och i statistiken ser ut

som förluster. När den otillåtna användningen och mätfelen är borträknade visar Svenskt Vattens rapport att de verkliga läckorna ligger på runt 15%.

Uppsala Vatten har i uppgift att försörja Uppsala kommuns invånare med vatten. De presenterade i sin årsredovisning 2018 att de vid bokslut hade producerat en vattenmängd på 18,2 miljoner kubikmeter [Upp19]. Med de beräknade förlusterna på 15%, som direkt konsekvens av läckor, uppkommer mängden vattenförluster till 2,73 miljoner kubikmeter. I genomsnitt förbrukade en boende i Uppsala 155 liter per dag eller 57 kubikmeter per år under 2018 [Den18]. 2,73 miljoner kubikmeter i vattenförluster skulle alltså ha kunnat använts till att försörja cirka 50 tusen invånare under ett år. Läckor i vattenledningar genererar stora ekonomiska förluster men även stora energiförluster då det produceras mycket mer vatten än vad som egentligen behövs. Vattenproduktionen medför också miljöpåverkan i form av föroreningar och utsläpp vilket blir en onödig påfrestning på miljön om produkten läcker ut i marken utan att komma till nytta. Ett system som kan upptäcka läckor snabbt är därför kritiskt för både vattenleverantörers framtid och för miljön.

Uppsala vatten har AdCon-enheter installerade i sina ledningar och anläggningar. Dessa enheter har sensorer som mäter bland annat vattenflöde, tryck och temperatur [AdC]. I dagsläget använder sig Uppsala Vatten av data som visar vattenflödet när de identifierar läckor. Flödesdatan används för att skapa grafer som tolkas manuellt för att identifiera läckor. Att manuellt tolka och analysera dessa grafer är en tidskrävande process som kan utföras effektivare av ett datoriserat system.

Mindre vattenläckor är svåra att upptäcka och det kan krävas att vattenledningen brister och skapar en stor läcka för att det ska upptäckas. Ledningar riskerar alltså läcka under en lång tid innan det uppmärksammas. Om en läcka uppstår för Uppsala vatten är det vanligt att den upptäcks genom att ledningen till slut brustit och invånare har larmat.

Inom industrier används statistisk processkontroll för att identifiera oregelbundenheter [Ame]. En sådan oregelbundenhet kan vara ökat vattenflöde till följd av en läcka. Statistisk processkontroll används för att effektivisera processer och minska avfall av resurser. En process kommer över tid att vara stabil kring ett medelvärde som anses vara normalt. Statistisk processkontroll använder sig av statistiska modeller och teorier för att larma när en process avviker från det normala, då anses processen vara statistiskt ur kontroll. Styrdiagram används för att visualisera detta.

Statistisk processkontroll och styrdiagram uppfanns under 1920-talet av fysikern Walter A. Shewhart [BN06]. Shewhart arbetade med ingenjörer vars uppgift var att förbättra kvaliteten på hårdvaran i telefoner. Fokuset var att reducera variationer i tillverkningsprocessen för att förbättra kvaliteten. Shewhart valde att klassificera variation som antingen normal variation eller speciell variation. Normal variation tyder inte på något fel

i processen utan kan bero på något tillfälligt och slumpmässigt, som att en mätare inte registrerat ett mätvärde. Speciella anledningar till variation tyder på att någonting är fel i processen. Det kan till exempel vara en läcka i en ledning. Något som kommer orsaka problem över tid och bör lagas. Shewhart skapade och använde styrdiagram som en metod för att skilja på de olika typerna av variation. Målet var att en process endast skulle ha normal variation. I sådant fall ansågs processen vara statistiskt under kontroll och kvaliteten kunde säkras. Det finns olika typer av styrdiagram där man använder sig av olika modeller. Syftet är att med hjälp av kontrollgränser identifiera avvikelser från den normala variationen. Shewhart-diagram, CUSUM-diagram och EWMA-diagram är tre styrdiagram som fungerar på detta sätt. Dessa förklaras i detalj i avsnitt 5.3.

Inom reglertekniken använder man sig av återkoppling i ett system. Detta för att kompensera för avvikelser. Statistisk processkontroll kompenserar inte för avvikelser utan identifierar dem endast. Det gör att statistisk processkontroll lämpar sig i fall där stora och oregelbundna avvikelser sker. Avvikelser som behöver identifieras snabbt och hanteras manuellt. Till exempel går det inte att kompensera för en brusten vattenledning, när det inträffat måste ledningen grävas upp och bytas ut.

3 Syfte, mål och motivation

Projektet strävade efter att minska det vatten som går till spillo i form av läckage i större vattenledningar. Programmet som utvecklats gör det snabbare och lättare att upptäcka läckage än metoder som används av Uppsala Vatten idag. Effektiviseringen uppnås genom ett system som analyserar data som hämtas från mätare i vattenledningar. Datan bearbetas med olika statistiska modeller och ett system som tolkar dessa markerar när det finns en vattenläcka. I huvudsak finns fyra intressenter och/eller aspekter som tjänar på att få detta problem löst: vattenproducenten, konsumenten, samhället och miljön. De beskrivs nedan i 3.1. I 3.2 så beskrivs etiska aspekter av systemet och i 3.3 så beskrivs de avgränsningar som gjorts.

3.1 Intressenter

I denna del beskrivs de huvudsakliga intressenter som fanns samt deras problem och behov.

3.1.1 Vattenproducenten

Om vatten går till spillo på vägen till kund resulterar det i en obalans mellan producerad mängd vatten och konsumerad mängd vatten. Vattenproducenten producerar då mer vatten än vad de kan fakturera kunderna för. Den industribyggnad som producerar vatten kallas vattenverk och produktionen av vatten kallas vattentillverkning. Vattentillverkning innebär att producera dricksvatten genom en reningsprocess. Den processen innebär driftkostnader för vattenverket och eftersom vattenläckage innebär vattentillverkning i onödan har producenten ett ekonomiskt intresse i att minimera vattenläckage. Den direkta åtgärden för att stoppa ett läckage är att reparera vattenledningen som har brustit. En sådan reparation medför flertalet kostnader som exempelvis arbetstimmar, nya vattenledningar och slitage på verktyg och maskiner. Större läckor resulterar i större insatser och mer pengar krävs för att åtgärda dem. För vattenleverantören är det alltså välmotiverat att vilja hitta läckage i deras ledningar så tidigt som möjligt innan läckan blir större.

3.1.2 Konsumenten

När en stor vattenläcka uppstår är det bråttom att laga den, i vissa fall så pass bråttom att vattnet måste stängas av utan förvarning till de hushåll som den skadade ledningen försörjer med vatten. Att som konsument få sitt vatten avstängt helt oplanerat kan ställa till med problem om man planerat in saker som kräver vatten, något som kan leda till missnöje med vattenförsörjaren och/eller kommunen. Om läckan istället upptäcks innan det är alltför bråttom kan kunderna meddelas i god tid för att kunna planera för att leva utan vatten ett tag.

3.1.3 Samhället

Vattenledningar sträcker sig bland annat över eller längs med vägar vilket innebär att vid reparation av ledningar kan vägar tvingas stängas av och trafik ledas om. Beroende på hur hårt trafikerad vägen är blir det olika problematiskt men det kan bidra till att andra vägar blir högre belastade. Långdragna reparationsarbeten kan alltså leda till påfrestningar på trafikplaneringen och en stads infrastruktur. Därför kan hela samhället se en vinning i att lösa problemet som detta projekt fokuserar på. Vattenläckage som skapar stora vattensamlingar kan skapa problem med instabil grund under byggnader eller vägar som förvrängs och förstörs.

I vissa fall tas vatten som ska renas från grundvattenmagasin istället för vattenmassor som sjöar eller hav. Det är främst grundvattnet som är känsligt för vattenbrist eftersom

vattnet där löper högre risk att förbrukas i snabbare takt än det fylls på. Vissa regioner förlitar sig nästan helt på grundvatten som källa. Gotland är ett exempel på en sådan region och de tvingas ibland ha bevattningsförbud för att hushålla på resurserna [Reg20]. Vatten kan alltså vara en bristvara och då är det fördelaktigt att så lite som möjligt slösas genom att rinna ut i marken via en läcka.

3.1.4 Naturen

När vatten försvinner ut i marken tvingas vattenverk tillverka mer vatten än vad som faktiskt konsumeras. Om antalet läckor minskas kan därför mängden vatten som måste produceras minskas. Eftersom vattentillverkningen medför viss miljöpåverkan är det önskvärt att inte producera mer än nödvändigt. Tidig upptäckt av en läcka kan därför medföra en miljömässig vinst i form av minskad belastning på miljön. Eftersom programmet som utvecklats i projektet kan hjälpa till att minska antalet vattenläckor och därmed vattenproduktionen kan projektet kopplas till flera av FN:s utvecklingsprogramms globala mål [UND].

Ett av miljöproblemen som vattenverk medför är kopplat till det slam som uppstår som restprodukt av tillverkningen. Ytvattenverk är vattenverk som ligger placerade i anslutning till vattenområden i vilka restsлам släpps ut. Enligt Naturvårdsverket kan en slambank bildas i vilken det över tid uppstår anaeroba förhållanden. Där inne sker en nedbrytningsprocess i vilken ämnen som metan, koldioxid och svavelväte utvecklas. Utsöndring av dessa ämnen gör att vattnet försuras och kan ge upphov till utfällning av metaller och andra föroreningar. I kraftigt påverkade sjöar kan aluminium utsöndras i sådan grad att fisk kommer till skada [Nat92]. Att minimera vattenproduktionen är därför till nytta för det marina livet. Eftersom programmet som utvecklats kan bidra till minskning av föroreningar och försurning av vattenmassor kan projektet kopplas till mål 14 som handlar om hav och marina resurser [UND20b]. Mer specifikt kan det kopplas till 14.1 *"Minska föroreningarna i havet"*, och 14.3 *"Minska havsförsurningen"*. En del av delmål 14.1 är att förebygga och minska föroreningar såsom tillförsel av näringsämnen i havet med särskilt inriktning på landbaserad verksamhet, vilket vattenverk är.

Obalansen i att det produceras mer vatten än vad som förbrukas knyts till mål 12 *"Hållbar konsumtion och produktion"* [UND20a]. Delmål 12.2 *"Hållbar förvaltning och användning av naturresurser"* är kopplat till detta eftersom vatten är en naturresurs och producenten i och med vattenförlusterna tvingas använda en naturresurs i onödan. Särskilt eftersom den kan vara en bristvara, som beskrivet i 3.1.3. Detta är också länkat till mål 6 *"Rent vatten och sanitet för alla"* och delmål 6.4 *"Effektivisera vattenanvändning och säker vattenförsörjning"* [UND20c]. Projektet förbättrar möjligheterna

till säker vattenförsörjning eftersom det kan minska antalet brådskande reparationer som utan förvarning till kunder bryter vattentillförseln. Dessutom förbrukar vattenverk ström och värme. Om mindre vatten behöver produceras leder det till en minskning i sådan energiförbrukning.

3.2 Etiska aspekter

Programmet har, på grund av tidsbegränsning, behövts utvecklats med vissa kompromisser som kan leda till etiska frågeställningar. Programmet tillhandahåller användaren med information genom ett grafiskt gränssnitt och eftersom inga verktyg för att underlätta för blinda finns är det antagligen omöjligt för en blind person att använda programmet.

För att inkludera så många potentiella användare som möjligt är programmet utvecklat så att det går att köra på de tre populära datoroperativsystemen Linux, Microsoft Windows och macOS.

Programmet analyserar datan och ger svar i form av dels presentation av datan applicerad på de tre styrdiagrammen och dels i form av att datum där läckage är troligt skrivs ut. Det gör att även om användaren inte besitter kunskaper för att kunna tolka styrdiagrammen ska programmet ha någon sorts funktion och vara användbart.

3.3 Avgränsningar

Vattenmätarna som används tillhandahåller fler variabler än endast flödesmängd, som till exempel tryck och temperatur, dessa skulle kunna tas med i beräkningarna för att eventuellt förbättra möjligheterna att upptäcka en läcka men för att passa inom tidsramen så avgränsar sig det här projektet till att bara analysera vattenflödet.

Uppsala Vatten får inte lämna ut sekretessklassad eller hemlig allmän handling till obehörig då denna information kan orsaka stor skada i samhället. Vart vattenledning- ar ligger, vilken funktion de har och annan data kan vara väldigt farligt i händerna på människor med onda intentioner. På grund av detta har tillgång till data varit begränsad under projektets gång.

Ett optimalt program skulle analysera flödesdata kontinuerligt eftersom det då ännu tidigare kan markera att en vattenläcka existerar. Eftersom Uppsala Vatten i dagsläget inte kan tillhandahålla sådan information avgränsar sig detta projekt till att låta en användare mata in en datamängd med historisk data.

4 Relaterat arbete

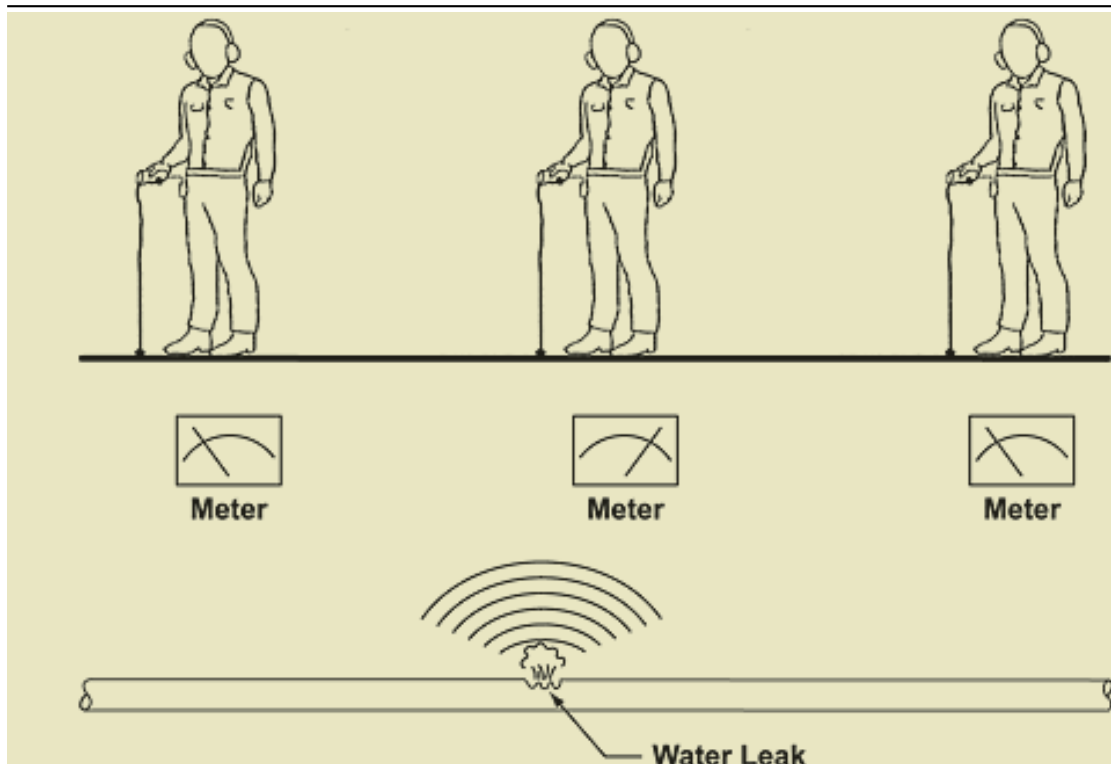
Inom området att detektera läckor i vatten- och gasledningar kategoriseras detekteringsmetoderna in i två olika huvudområden: externa och interna metoder [Ame02]. Interna metoder använder mätare som är placerade i ledningarna för att övervaka parametrar som flöde, tryck och temperatur. Externa metoder använder sig av tekniker utanför ledningssystemet, som användning av infraröda kameror och detektering med hjälp av akustik. Metoden i det program som har utvecklats i detta projekt klassificeras som intern detektering. Detta då data från invändiga mätare i ledningarna används i programmet för att upptäcka läckor.

4.1 Extern akustisk läckagedetektering

En metod som idag finns på marknaden och som relaterar till detta arbete är detektering av vattenläckor med hjälp av akustik [Duf17]. Vatten som läcker ut från ledningar under högt tryck i ett vattendistribueringsystem ger ifrån sig ett tydligt ljud [Duf17]. Ljudvågorna transporteras, likt en signal, längsmed ledningen åt båda riktningarna. Enligt Daniel P. Duffy [Duf17] använder extern akustisk detektering dessa två egenskaper hos läckan för att finna och precisera den i fyra steg:

1. Läckan inträffar ute i en vattenledning och skickar ljudvågor längsmed ledningen.
2. En stationär enhet avlyssnar ljudvågorna och detekterar att en läcka har skett ute i distribueringsystemet.
3. En mobil enhet används för att precisera läckans position genom att lyssna efter när ljudvågornas styrka ökar när avståndet till läckan minskar likt att styrkan minskar när avståndet ökar, se figur 1 nedan.
4. Den mobila enheten placerar användaren vid läckan med en felmarginal på ca 1 meter. Här grävs ledningen upp för att repareras.

Det finns nackdelar med denna typ av detektering då ljudvågorna till den mobila enheten kan hindras alternativt förstöras innan de plockas upp. Detta kan ske på grund av djup lera, tjock asfalt eller att området är särskilt tungt trafikerat [Sub16].



Figur 1 Precisering av en läcka med akustisk läckdetektering. En mobil enhet används för att lyssna efter ljudvågorna från läckan, vilka blir starkare nära läckan och svagare längre bort ifrån läckan [Sub16].

4.2 Intern detektering med massbalans

En intern detekteringsmetod som idag används nyttjar lagen om massans bevarande [SSH11]. I boken *Pipeline Leak Detection Handbook* beskrivs metoden som användande av två eller flera sensorer placerade inne i ledningen [MCE16]. Dessa mäter massflödet som passerar de positionerna där mätarna sitter. För att upptäcka en läcka undersöks massbalansen mellan två av sensorerna. Vid ett tillfälle då ingen läcka existerar ska massbalansen uppnås mellan flödet genom första och andra mätaren. Alltså ska samma mängd vatten passera båda mätarna. Massflödesbalansen mellan mätarna beskrivs med formeln:

$$M_{tot} = M_1 - M_2$$

Där M_1 är massflödet som passerar första mätaren; M_2 är massflödet som passerar andra

mätaren. Från detta kan då slutsatsen dras att:

$$M_{tot} = \begin{cases} \text{Ingen läcka} & \text{om } M_{tot} = 0 \\ \text{Läcka} & \text{om } M_{tot} > 0 \end{cases} \quad (1)$$

Storleken på läckan kan då beskrivas med variabeln M_{tot} , då detta är det massflödet som försvinner. Idealt ska alltså $M_{tot} = 0$, detta är dock inte fallet i verkligheten eftersom mätvärdena inkomna från sensorerna medför en viss osäkerhet. Osäkerheten leder till både typ I- och typ II-fel, se del ?? [MCE16]. En andra negativ aspekt av denna detekteringsmetod är det faktum att det behövs två mätare för att detektera läckor. Detta medför osäkerhet på två ställen i form av mätningar samt även att läckan måste propagera längsmed hela sträckan [MCE16].

5 Metoder

I denna del beskrivs metoder, modeller, koncept och verktyg som använts.

5.1 Utvecklingsmiljöer

Ett antal verktyg valdes under projektets gång för att smidigt och effektivt föra arbetet framåt och få ett bra program som fungerar enligt kravbilden. Visual Studio Code [Mic] och Atom [Ope] är två utvecklingsmiljöer som användes under projektet. De är väl använda inom programutveckling enligt Stack Overflows årliga undersökning för 2019 [Sta] och stöds på flera olika datoroperativsystem som Windows, macOS och Linux. Programmeringsspråk som Java [Ora] och C [Wik20a] skulle troligen kunna användas men eftersom det finns mycket information på nätet om datahantering och matematiska modeller i Python [Pyt] valdes det under projektet. Python har en mängd bibliotek för matematiska modeller och hantering av data [Des19]. Ett exempel är Pandas [Pan] som är ett bibliotek vilket användes i projektet för att hantera och analysera data. Biblioteket Tkinter [Pyt20a] användes för att skapa ett grafiskt gränssnitt till användaren av programmet.

5.2 Olika typer av felaktig identifiering

Inom statistikens hypotesprövning så klassificerar man fel i en hypotes i två kategorier. Typ I-fel och typ II-fel [DKLM05a]. I detta projekt var hypotesen huruvida det fanns

en läcka eller ej. Typ I-fel är de fel där hypotesen säger att det är en läcka men det inte finns någon. Om en invånare till exempel bestämmer sig för att fylla en stor pool under natten så kommer vattenflödet att vara onormalt högt i ledningarna och det skulle kunna tolkas som en läcka. Det skulle då ske ett typ I-fel. Den typen av fel kallas också för "*falsk positiv*". Typ II-fel är de fel där hypotesen säger att det inte är en läcka men det finns en läcka. Om en läcka är så pass liten att den inte upptäcks av modellerna så har ett typ II-fel skett. Den typen av fel kallas också för "*falsk negativ*". Modellen som används för att forma hypotesen avgör risken att ett typ I-fel eller typ II-fel sker [DKLM05b]. En modell som identifierar få avvikelser kommer i regel att ha få typ I-fel. Däremot kommer den att missa fler avvikelser vilket leder till fler typ II-fel. På samma sätt kommer en modell som identifierar många avvikelser i regel att ha få typ II-fel men också identifiera felaktiga avvikelser vilket ger fler typ I-fel.

5.3 Styrdiagram för identifiering av läckor

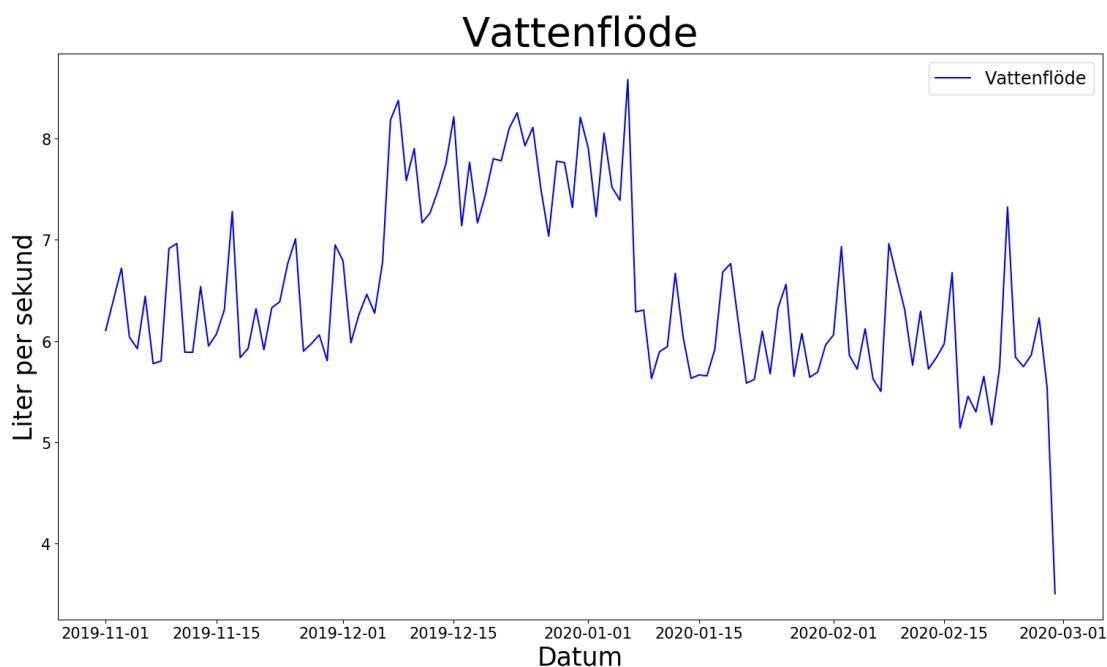
Tre olika typer av styrdiagram rekommenderades av Uppsala Vatten och dessa har använts i projektet. Användning av tre stycken styrdiagram istället för endast ett grundar sig i två olika anledningar. Den första är det faktum att styrdiagrammen är olika känsliga för variationer från medelvärdet. Shewhart-diagram känner endast av förändringar från medelvärdet som är större än $1,5\sigma$ [HAMS13], där σ är standardavvikelsen i flödesdatan. Denna egenskap gör det nödvändigt att också använda sig av styrdiagram som reagerar på mindre förändringar. EWMA- och CUSUM-diagram används för att effektivt uppfatta avvikelser som ligger i intervallet $0 < \sigma \leq 1.5$ [HAMS13]. Den andra anledningen till användningen av tre styrdiagram är lik nödvändigheten med trefaktorautentisering, alltså säkerheten. För att finna en gradering av hur troligt det är att en läcka har skett så vägs det in hur många av styrdiagrammen som reagerat på förändringen.

I kommande beskrivningar används flödesdata i liter/sekund vid en mätare belägen i en vattenledning som indata till styrdiagrammen. Datan som benämns i följande stycken är alltså den originaldata som har registrerats vid denna mätare. Denna data visualiseras i figur 2.

5.3.1 Shewhart-diagram

Shewhart-diagram använder sig av den angivna datans medelvärde och standardavvikelse. Med hjälp av dessa två variabler skapas en övre kontrollgräns (UCL) och en undre kontrollgräns (LCL). Dessa bestäms med formeln:

$$UCL = \bar{x} + 3\sigma \quad (2)$$



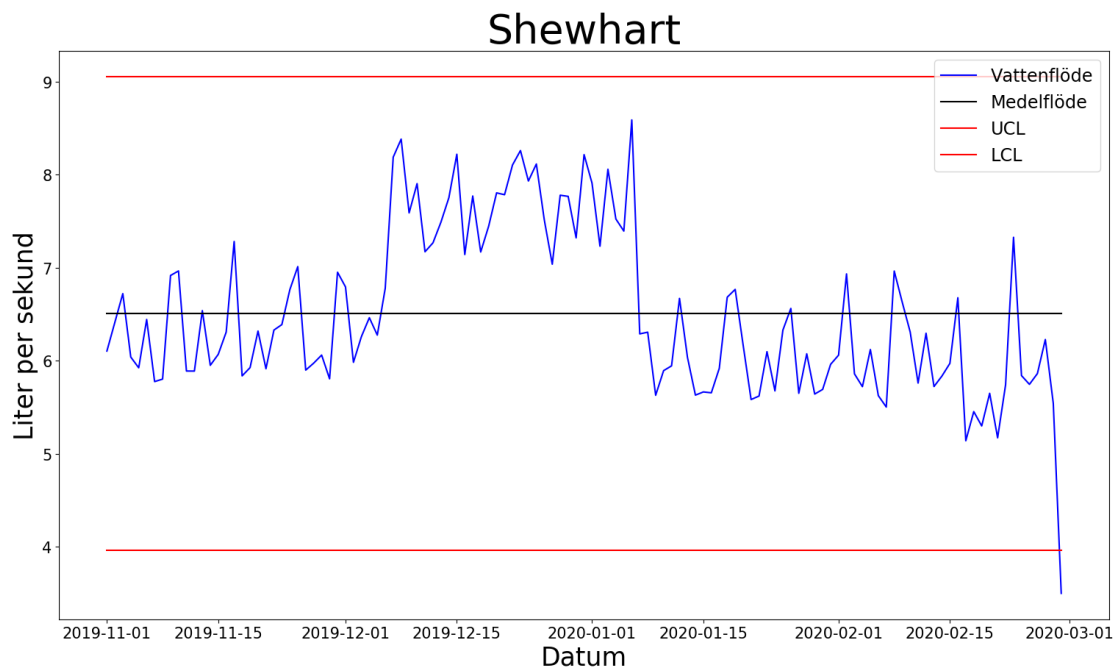
Figur 2 Flödesdata från en mätare i en vattenledning.

$$LCL = \bar{x} - 3\sigma \quad (3)$$

Där \bar{x} är datans medelvärde; σ är datans standardavvikelse. Det skapas då en zon mellan dessa två gränser, se figur 3. Enligt centrala medelvärdessatsen så ska 99.3% av datapunkterna ligga innanför denna zon [Ryd14]. Detta innebär då att av en samling på 370 datapunkter ska endast 1 punkt ligga utanför gränserna. Är då fler punkter utanför dessa gränser anses processen vara statistik ur kontroll, vilket innebär i detta fall vid analys av flödesdata att det sannolikt existerar en läcka. I figur 3 visas ett exempel på en mängd data som anses vara under kontroll, då punkterna befinner sig innanför de angivna gränserna som illustreras med röda linjer.

5.3.2 CUSUM-diagram

CUSUM är en analytisk metod som används för att identifiera förändringar i data. Den användes i projektet för att se små förändringar i mätvärden från vattenledningarna. Metoden är väldigt känslig för små förändringar. Vid en liten vattenläcka så skulle ett Shewhart-diagram kunna visa flera datapunkter i rad precis innanför kontrollgränserna, men inte indikera någon avvikelse. Ett CUSUM-diagram skulle i ett sådant fall indikera



Figur 3 Flödesdata från en mätare i en vattenledning med UCL, LCL och avg för flödets kurva.

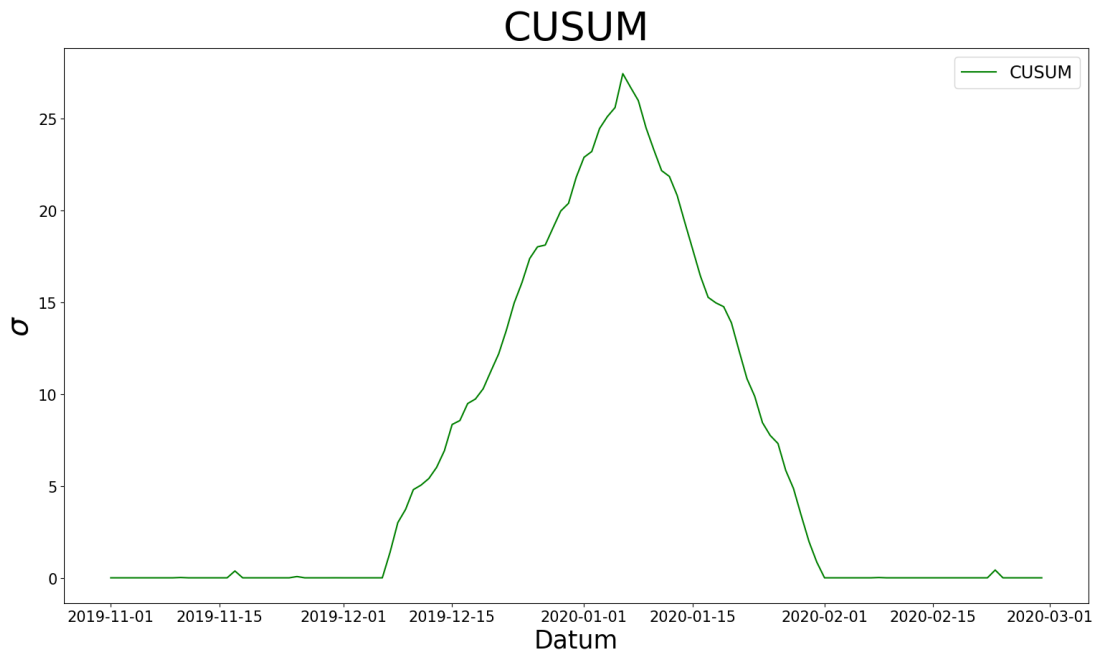
att något har hänt. CUSUM står för ”*cumulative sum*” och är en typ av kumulativ kurva. I CUSUM-kurvan C räknas den i :te punkten $C(i)$ ut som följande:

$$C(i) = \begin{cases} C(i-1) + x_i - \bar{x} - k & \text{om } C(i-1) + x_i - \bar{x} - k > 0 \\ 0 & \text{om } C(i-1) + x_i - \bar{x} - k \leq 0 \end{cases} \quad (4)$$

Där x_i är den i :te datapunkten i originaldatan; \bar{x} är originaldatans medelvärde; k är en parameter som väljs utifrån hur känslig modellen ska vara för förändringar [Nat]. Genom att formeln endast använder värden större än 0 så identifieras endast positiva förändringar.

Figur 4 visar ett CUSUM-diagram. Den gröna kurvan är en CUSUM-kurva C baserat på flödesdatan x från figur 2. När det efter 1:a december sker en positiv förändring i x så ökar C kraftigt. Efter att förändringen skett i x så fortsätter den att ha höga värden fram till cirka 6:e januari. Detta medför att C fortsätter att öka. C indikerar att datapunkterna fortsätter att befinna sig ur kontroll fastän x inte ökar lika mycket. När x den 7:e januari minskar börjar C också minska då x rör sig mot att vara i kontroll.

En positiv förändring indikerar inte nödvändigtvis en avvikelse. En tillräckligt stor po-



Figur 4 Ett CUSUM-diagram baserat på flödesdatan i figur 2.

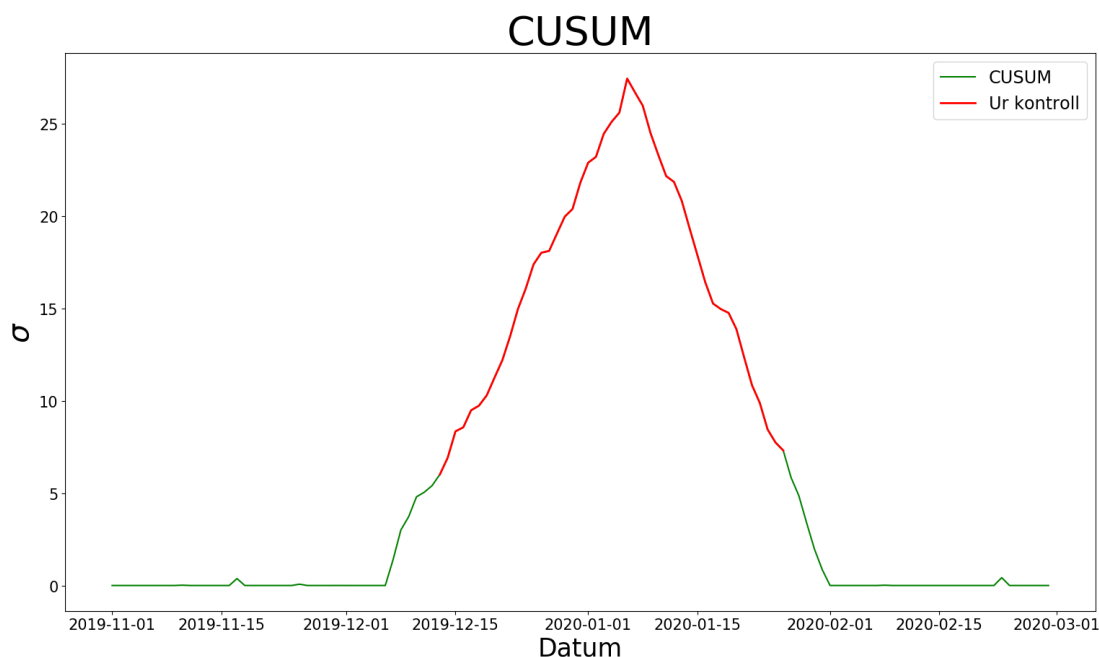
sitiv förändring innebär att en avvikelse har skett. För att identifiera de datapunkter där det skett en tillräckligt stor förändring och datan befinner sig ur kontroll så använder man ett referensvärde h . h -värdet ges av följande formel:

$$h = k * 2/\sigma^2 * \ln(1 - B/a) \quad (5)$$

k är samma parameter som tidigare, se formel 4. σ representerar standardavvikelsen. Variablerna a och B avgör sannolikheten för typ I- fel och typ II- fel, se del ?? [Nat].

h -värdet används som en gräns. Alla värden i C som ligger över h -värdet anses vara ur kontroll. Figur 5 visar samma CUSUM-diagram som i figur 4 men C är rödmarkerad i de datapunkter som är över h -värdet och då anses vara ur kontroll. I figur 5 så är h -värdet cirka 5.9

Modellen som använts i projektet använder 0.0027 som a -värde och 0.01 som B -värde. Ett a -värde på 0.0027 motsvarar samma täckning som tre standardavvikelser gör i ett Shewhart-diagram, se del 5.3.1. B -värdet väljs utifrån den statistiska styrkan i sitt test [Jon09].



Figur 5 Ett CUSUM-diagram baserat på flödesdatan i figur 2. Kurvan är röd när y-värdet överstiger referensvärdet $h = 5.9$ och den anses vara ur kontroll.

5.3.3 EWMA-diagram

EWMA står för "*Exponential Weighted Moving Average*", detta betyder i sin korthet att styrdiagrammet lägger exponentiellt mindre värde på datapunkter längre bak i tiden. Ett flytande medelvärde innebär att ett glidande fönster med storleken n används för att beräkna medelvärdet de n första värdena. Detta blir då den första nya datapunkten. Nästa punkt beräknas då med de $n + 1$ första datapunkterna [Bin]. Den variation av flytande medelvärde som används i detta styrdiagram lägger då mindre vikt i punkter i fönstret som är placerade längre ifrån n [Bin]. Detta gör att styrdiagrammet kan monitorera processens medelvärde [HGK18]. Formeln för att beräkna värdet på EWMA-punkt är följande:

$$EWMA_i = Z_i = \lambda x_i + (1 - \lambda)Z_{i-1} \quad (6)$$

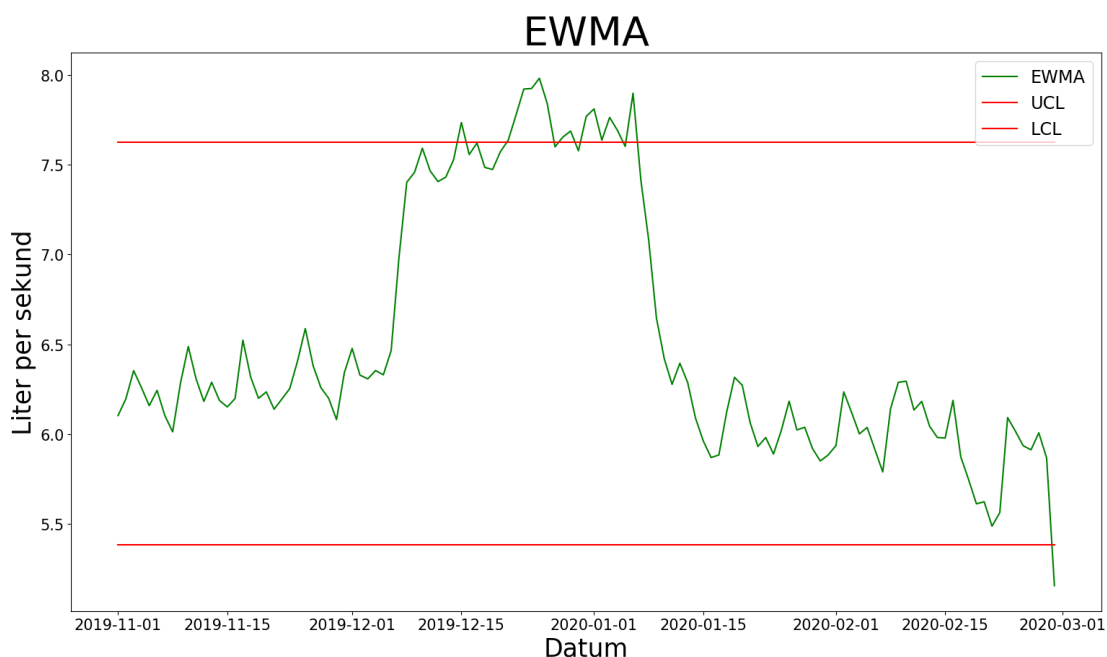
Där Z_i är värdet EWMA-punkt nummer i ; x_i är datapunkt i ; λ är en tyngdfaktor ($0 < \lambda < 1$) [McN14]. Detta räknar ut värden som ersätter datapunkterna så att de nya värdena används för att rita kurvan. Nästa steg är då att beräkna en övre kontrollgräns

(UCL) samt en undre kontrollgräns (LCL). Gränserna beräknas med formeln

$$UCL = \mu_0 + 3\sigma\sqrt{\frac{\lambda}{(2-\lambda)}} \quad (7)$$

$$LCL = \mu_0 - 3\sigma\sqrt{\frac{\lambda}{(2-\lambda)}} \quad (8)$$

Där μ_0 är målvärdet processen ska hålla sig efter, ofta likställt med medelvärdet [McN14]. Likt Shewhart-diagrammet använder sig denna metod av gränserna för att se när processen inte längre är statistiskt under kontroll. Den stora skillnaden är att EWMA-diagram kan användas för att detektera antingen stora eller små skillnader från medelvärdet, detta genom att ändra storleken på tyngdfaktorn [HGK18]. En tyngdfaktor nära 1 ger en nästan identisk kurva och resultat som ett Shewhart-diagrammet ger, vilket gör att endast stora skillnader kommer att detekteras. Om faktorn istället väljs nära 0 kommer styrdiagrammet reagera på mindre skiftet [HGK18]. I figur 6 är kurvan på EWMA visualiserad med en tyngdfaktor på 0.3. Detta gör att, i jämförelse med Shewhart-diagrammet i figur 3, styrdiagrammet signalerar att datan inte är statistiskt under kontroll.



Figur 6 EWMA över flödesdatan från en mätare i en ledning.

5.4 Hantering av oregelbundet vattenflöde

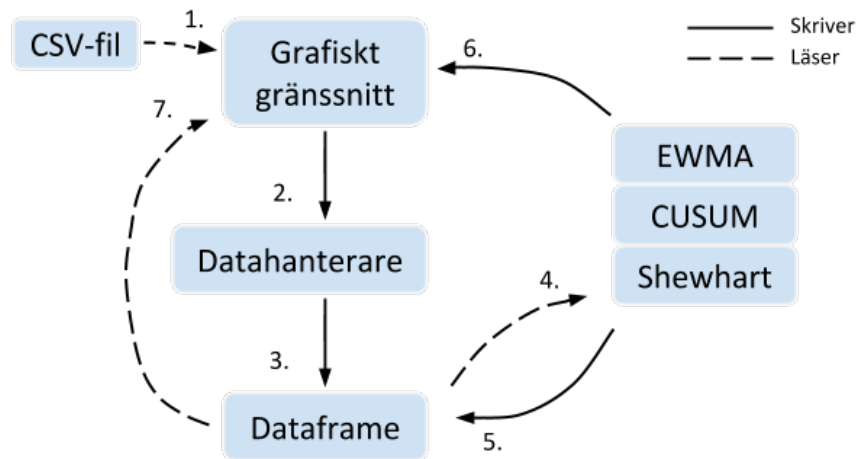
Vattenflödet i en ledning är inte konstant under ett dygn. Under frukost, lunch och middag används i regel mer vatten än under natten till följd av matlagning, dusch och toalettanvändning med mera. Fluktuationer i vattenflödet syns även i de dataset som använts under projektets gång. Detta kallas oregelbundet flöde och det kan också förekomma utan någon direkt anledning då sensorer kan skicka fel värden eller att det har hänt något på andra delar av vattennätverket. I vissa fall kan det leda till att det är samma flöde klockan 2 på natten som klockan 12 på dagen vilket sannolikt inte är en representation av vad som händer i verkligheten. Stor variation mellan datapunkter kan göra det svårt för styrdiagrammen vilket kan resultera i fel datum som pekats ut som läckor. Genom att släta ut flödesdatans kurva minskas dessa variationer vilket förbättrar slutresultatet.

För att hantera flödesdatan så skapades 6 olika datahanterare som omformaterar datan på olika sätt under dygnet för att minska störningar. Funktionerna räknar ut medelflödet för varje dygn, varje dag och varje natt samt medelflödet för varje timme på dygnet, varje timme på dagen och varje timme på natten. Dagar räknas som timmar på dygnet mellan klockan 10 och klockan 16. Nätter räknas som timmar på dygnet mellan klockan 24 och klockan 04. Tiderna på dagen valdes för att få relativt konstanta värden med mycket vattenflöde och tiderna på natten valdes för att få relativt lite vattenflöde. Undersökningar av de givna dataset gav att det är minst skillnad på vattenflödet mellan dessa tider vilket är viktigt för att släta ut flödesdatans kurva så mycket som möjligt. De olika funktionerna användes slutligen för att se om en läcka i en ledning är lättare att peka ut beroende på tiden på dygnet.

I datan kan det även finnas datapunkter som inte har något värde. Det kan bero på att värdet var ogiltigt eller att mätaren på grund av något fel inte registrerade ett värde. Dessa datapunkter tas bort ur datasetet och ignoreras. Då medelvärden över natt eller dag räknas ut så spelar inte en datapunkts värde så stor roll utan den kan ignoreras utan att påverka medelvärdets relevans.

6 Systemstruktur

Flödesdatan som används har bestått av mindre dataset i form av 5 stycken CSV-filer som innehåller information från en ledning på okänd plats med flödeshistoria. Dessa dataset omformateras med hjälp av datahanterare och analyseras sedan av styrdiagrammen Shewhart, CUSUM och EWMA för att avgöra om det är en läcka eller inte.



Figur 7 1. Användaren väljer en CSV-fil som ska analyseras; 2. Användaren väljer en datahanterare som ska omformatera CSV-filen; 3. Datahanteraren returnerar en omformaterad dataframe; 4. Styrdiagrammen läser ifrån den omformaterade dataframen; 5. Styrdiagrammen adderar kolumner till den omformaterade dataframen; 6. Styrdiagrammen returnerar en lista där de anser att datan är ur kontroll; 7. Grafiska gränssnittet läser av dataframen

6.1 Systemets tre huvuddelar och deras funktion

Enligt Figur 7 bygger projektet på flera olika enheter. Först väljer användaren vilken CSV-fil som ska undersökas med vilken datahanterare i det grafiska gränssnittet. Datahanteraren producerar då en omformaterad dataframe av den valda filen. Styrdiagrammen läser sedan av denna dataframe och adderar det resultat som producerats på samma dataframe samtidigt som de returnerar var sin lista med datum då de tror att en läcka kan ha skett. Allt detta styrs alltså av ett grafiskt gränssnitt vilket möjliggör att användaren kan välja CSV-filer med flödesdata från en ledning som ska undersökas för läckor. Användaren kan även välja vilken datahanterare som ska dela upp flödesdatan och resultatet från styrdiagrammen kan sedan ses tydligt med grafer och datum som indikerar när en läcka kan ha skett.

6.1.1 Datahanterare för omformatering av flödesdata

Det finns 6 olika datahanterare som alla delar upp flödesdatan på olika sätt. Vilken datahanterare som ska användas på datan väljer användaren ifrån det grafiska gränssnittet. De olika alternativen visas i en dropdown-lista där de kan markeras för att visualisera vilken datahanterare är vald. Den valda CSV-filen skickas till datahanteraren när

användaren har tryckt på knappen ”Analysera”. Funktionen returnerar då ett omformatat dataset som sedan kan användas i styrdiagrammen.

6.1.2 Styrdiagram för läcksökning

Efter den valda datahanteraren har omformaterat datan så skickas den vidare till styrdiagrammen. EWMA, CUSUM och Shewhart är separata styrdiagram som tar emot samma dataset. Varje funktion applicerar sedan sina matematiska modeller på datan och adderar kolumner på samma dataset. På detta sätt minskas minnesandvändningen av datorn då det inte behöver användas ett dataset per modell. Varje styrdiagram returnerar också en lista som innehåller de datum som styrdiagrammet visat att datan är ur kontroll och anser att en läcka är identifierad.

6.1.3 Grafiskt gränssnitt för användarvänlighet

För att förbättra funktionaliteten och användarvänligheten av projektet togs ett grafiskt gränssnitt fram. För att välja CSV-fil öppnas användarens utforskare där hen kan välja vilken CSV-fil med flödesdata som ska analyseras. Användaren kan därefter välja vilken datahanterare som ska omformatera datan med hjälp utav en dropdown-lista. Resultatet från styrdiagrammen visualiseras därefter med grafer och datum där de anser att en läcka är identifierad.

7 Krav och utvärderingsmetoder

Tre kravområden identifierades. Ett område som kan kopplas tydligt till projektets syfte att underlätta identifiering av läckor och två områden som rör funktionaliteten i systemet. Kraven som rör funktionaliteten i systemet är krav som måste uppnås. Då intressenten i dagsläget inte använder sig av något liknande system så är det inte lika viktigt att kraven som rör projektets syfte är fullt uppnådda. Även om systemet inte identifierar alla läckor eller ger falska alarm så kan det underlätta identifieringen och skapa en grund för framtida utvecklingar.

För varje krav identifierades metoder för att utvärdera huruvida kraven uppnåddes. Resultatet av dessa metoder presenteras i del 9. Resultaten tolkas och värderas sedan i del 10.

7.1 Krav

Intressenten uttryckte en önskan om att etablera kontrollgränser som kunde identifiera läckor. I systemet används Shewhart-diagrammets kontrollgränser, CUSUM-diagrammets h -värde och EWMA-diagrammets kontrollgränser tillsammans för att identifiera läckor. Om en läcka finns i datasetet ska kontrollgränserna identifiera den. Om datasetet inte innehåller någon läcka ska kontrollgränserna inte identifiera någon läcka.

Då användaren själv väljer fil i gränssnittet så behöver systemet kunna hantera att användaren väljer en fil av fel format. Datahanterarna kan endast hantera CSV-filer och användaren ska inte kunna krascha programmet genom att välja en fil av annan typ.

Dataseten som systemet skulle analysera innehöll många datapunkter. De dataset som användes under utvecklingen innehöll kring 300 000 datapunkter och dataseten som intressenten skulle använda var av liknande storlek. Stora dataset innebar att om algoritmer med hög tidskomplexitet användes hade det kunnat göra systemet oanvändbart [Sip06]. Det bestämdes som krav att alla algoritmer i implementationen skulle ha en tidskomplexitet lägre än $\theta(n^2)$. En tidskomplexitet på $\theta(n^2)$ innebär att antalet instruktioner som utförs växer exponentiellt med mängden data.

7.2 Utvärderingsmetoder

Intressenten tillhandagav fem dataset innehållande flödesdata som användes i utvärderingen. Om ett dataset innehöll en läcka fanns det endast specificerat vilket datum som läckan hade lagats. Det antogs att det under en okänd tid innan datumet då läckan lagades hade funnits en läcka. Datumet då lagning hade skett användes för att utvärdera systemets kontrollgränser. Om en kontrollgräns indikerade en läcka i anslutning till datumet då det hade lagats en läcka nådde kontrollgränsen kravet. Om kontrollgränsen inte indikerade någon läcka i anslutning till datumet då det hade lagats en läcka hade ett typ II-fel skett och kontrollgränsen nådde inte kravet. Om kontrollgränsen indikerade en läcka även fast det inte var i anslutning till att någon läcka hade lagats hade ett typ I-fel skett och kontrollgränsen nådde inte kravet.

Systemets förmåga att hantera ett felaktigt val av fil från användaren utvärderades genom att testa att ange filer av fel format. Alla filer som inte var en CSV-fil ansågs vara fel format.

För att utvärdera tidskomplexitet analyserades alla funktioner i koden. Genom att räkna antalet maskininstruktioner funktionen utför för varje iteration och sedan förenkla uttrycket för när antal iterationer går mot oändligheten räknades funktionens tidskom-

plexitet ut.

8 Implementation

I denna del beskriv implementationen av systemets datahanterare, styrdiagram och det grafiska gränssnittet. Det beskrivs hur flödesdatan omformateras och uppdelas i datahanterarna och hur Shewhart, Cusum och EWMA är implementerade med hjälp utav olika bibliotek. Utöver detta beskrivs det även hur en dataframe ser ut och hur det grafiska gränssnittet fungerar och är konstruerat.

8.1 Lagring av data


Datastrukturen som kontinuerligt används för att lagra data i programmet är en Pandas dataframe [Pan]. Denna datastruktur består av en 2-dimensionell lista där den lodräta delen kallas kolumner och den vågräta rader. I en dataframe finns möjlighet att lägga till både nya rader och kolumner, samt ta bort från båda delarna. Dataframen, som används i programmet, har datum och klockslag som index och en kolumn för flödesdata.

8.2 Hantering av flödesdata med hjälp av datahanterare

Alla datahanterare består två delar. En del som omformaterar den fil med flödesdata som har valts av användaren och en del som delar upp denna data på olika sätt. I denna del beskrivs dessa i mer detalj.

8.2.1 Omformatering av data

Vilken fil som ska undersökas välj av användaren av programmet. Filerna har formatet CSV vilket är en förkortning av "*Comma Separated Values*". Detta innebär att datan är lagrad i en kommaseparerad lista i textformat [Wik20b]. För att datahanterarna ska kunna arbeta med filen måste den först omformateras vilket innebär att oönskade mätvärden av flödet måste tas bort. För att göra detta tas rader med asterisk bort då de representerar mätvärden då flödesmätaren inte registrerade ett värde. Kolumnerna "*Date*" och "*Time*" slås också ihop till en kolumn "*Date*" för att minska dataframen. Den tredje kolumnen döps om till "*Flow (l/s)*" då den kan ha olika namn beroende på vilken fil användaren väljer. I figur 8 visualiseras detta då den vänstra dataframen är den fil som

Index	Date	Time	Flöde (l/s)		Index	Date	Flow (l/s)
1	2019-09-10	00:00:00	0.1		1	2019-09-10 00:00:00	0.1
2	2019-09-10	00:00:15	0.1		2	2019-09-10 00:00:15	0.1
3	2019-09-10	00:00:30	*		5	2019-09-10 00:01:00	0.2
4	2019-09-10	00:00:45	*		6	2019-09-10 00:01:15	0.2
...
2300	2019-09-10	23:59:30	0.2		2300	2019-09-10 23:59:30	0.2
2301	2019-09-10	23:59:45	0.2		2301	2019-09-10 23:59:45	0.2
2302	2019-09-11	00:00:00	0.2		2302	2019-09-11 00:00:00	0.2
...
4618	2019-09-11	23:59:45	0.3		4618	2019-09-11 23:59:45	0.3
4619	2019-09-12	00:00:00	0.2		4619	2019-09-12 00:00:00	0.2
...

Figur 8 Till vänster syns en dataframe med två separata kolumner för datum och tid samt asterisker på vissa rader. Till höger syns en dataframe med en kolumn för datum och tid och alla rader med asterisker borttagna.

användaren laddar in till programmet och den högra dataframen är den omformaterade datan.

8.2.2 Uppdelning av data

Uppdelningen av den omformaterade datan gör datahanterarna. Det är 6 funktioner som gör liknande uträkningar men med små variationer. Funktionernas namn är *"dateMean"*, *"dayMean"*, *"nightMean"*, *"dateHours"*, *"dayHours"* och *"nightHours"*.

Alla funktioner börjar med att dela in den omformaterade datan i datum. *"dateMean"* returnerar därefter medelvärdet av alla datapunkter per datum. Resterande funktioner räknar sedan ut medelvärdet per timme för varje datum men tar bort timmar som inte faller inom deras intervall. *"dateHours"* inkluderar alla timmar på dygnet så inga datapunkter tas bort. *"dayHours"* och *"dayMean"* tar bort alla timmar som inte är mellan klockan 10 och klockan 16 medan *"nightHours"* och *"nightMean"* tar bort alla timmar som inte är mellan klockan 24 och klockan 04. *"dateHours"*, *"dayHours"* och *"nightHours"* returnerar därefter sin dataframe som innehåller medelvärdet för varje timme inom deras intervall. *"dayMean"* och *"nightMean"* returnerar istället medelvärdet av timmarna inom intervallet. Figur 9 beskriver funktionen *"dayHours"* då dataframen till vänster beskriver ett exempel på den omformaterade datan och dataframen till höger en

index	Date	Flow (l/s)		index	Flow (l/s)
1	2019-09-10 00:00:00	0.1		2019-09-10 10:00:00	0.145
2	2019-09-10 00:00:15	0.1		2019-09-10 11:00:00	0.167
...
2300	2019-09-10 23:59:30	0.2		2019-09-10 16:00:00	0.302
2301	2019-09-10 23:59:45	0.2		2019-09-11 10:00:00	0.199
2302	2019-09-11 00:00:00	0.2		2019-09-11 11:00:00	0.196
...
4618	2019-09-11 23:59:45	0.3		2019-09-11 16:00:00	0.377
4619	2019-09-12 00:00:00	0.2		2019-09-12 10:00:00	0.354
...

Figur 9 Till vänster syns en dataframe med flödesdata var 15:e sekund och till höger syns en dataframe med medelvärde av flödesdata för varje timme mellan klockan 10 och klockan 16.

uppdelning av samma data där index för varje datapunkt är timmarna mellan klockan 10 och klockan 16 för varje datum.

8.3 Implementation av styrdiagrammen

För varje styrdiagram implementerades ett bibliotek som kan importeras för att använda modellerna. Varje bibliotek har en funktion som kan kallas på för att applicera modellerna. Funktionen tar en dataframe som argument. Då objekt i Python använder sig av referenssemantik så innebär det att funktionerna kan skapa nya kolumner i den existerande dataframen och inte behöver returnera en ny version. Varje funktion skapar nya kolumner i dataframen och returnerar en lista som innehåller de datum då styrdiagrammet visat att datan är ur kontroll och anser att en läcka är identifierad.

Biblioteken kräver att dataframen som används innehåller en kolumn med namn *"Flow (l/s)"*. Den kolumnen anses vara originaldatan som används i modellerna.

8.3.1 Shewhart

Biblioteket som skapades för att praktiskt kunna använda sig av ett Shewhart-diagram består av en funktion. Funktionen lägger till två nya kolumner i den inmatade da-

taframen. De två kolumnerna är benämnda "*UCL*" som är den övre kontrollgränsen och "*LCL*" som är den undre kontrollgränsen. I jämförelse med den generella metoden nämnd i 5.3.1 används en speciell variant för att beräkna gränserna. Varianten används för att reducera antalet typ II-fel. Algoritmen för att reducera felen består av stegen:

1. Beräkna UCL och LCL.
2. Leta efter punkter som ligger utanför gränserna.
3. Om punkter hittas utanför bortses dessa punkter ifrån och återgår till steg 1.
4. De nu beräknade gränserna i UCL och LCL läggs till i dataframen.

Algoritmen är nödvändig för att datapunkter som inte är statistiskt under kontroll inte ska influera kontrollgränserna [Dun20].

8.3.2 CUSUM

Ett bibliotek för att med hjälp av ett CUSUM-diagram identifiera läckor implementerades. Biblioteket innehåller en funktion *cusum()* som kan anropas med en dataframe som argument. Funktionen skapar två nya kolumner med namn "*cusum*" och "*oc_cusum*" i dataframen. Datapunkterna för en CUSUM-kurva placeras i kolumnen "*cusum*". Kolumnen "*oc_cusum*" skapas för att spara de datum där CUSUM-kurvan var ur kontroll.

När CUSUM-kurvan räknas ut så används formeln i ekvation 4. För varje punkt i originaldatan räknas CUSUM-kurvans motsvarande punkt ut och läggs till på samma datum i "*cusum*"-kolumnen. Punktens värde jämförs med referensvärdet för att se om den befinner sig ur kontroll. Om den befinner sig ur kontroll så läggs punktens värde till på samma datum i "*oc_cusum*"-kolumnen. Om den befinner sig i kontroll så läggs ett None-värde istället till på datumet i "*oc_cusum*"-kolumnen. Ett None-värde är Pythons motsvarighet till ett null-värde [Coo13].

Funktionen plockar sedan ut datumen från de punkter i "*oc_cusum*"-kolumnen som inte innehåller ett None-värde och returnerar dem som en lista. Listan innehåller de datum då det anses ha identifierats en läcka.

8.3.3 EWMA

Biblioteket för användande av ett EWMA-diagram består av en funktion. Denna funktion skapar tre nya kolumner i den inmatade dataframen. De tre nya kolumnerna är

"EWMA", "UCL_EWMA" och "LCL_EWMA". Värdena som används för att skapa kurvan för EWMA:n placeras i kolumnen "EWMA", som beräknas med formeln i ekvation 6. Gränserna för den övre och undre kontrollgränsen lokaliseras i "UCL_EWMA" respektive "LCL_EWMA", samt beräknas med formlerna i ekvationerna 7 och 8. För att skapa en lista med alla datum då EWMA:n inte är under kontroll, då det med stor sannolikhet existerar en läcka, jämförs varje rad i kolumnen "EWMA" med "UCL_EWMA" och "LCL_EWMA". Om värdet på EWMA:n är större än den övre kontrollgränsen eller mindre än den lägre kontrollgränsen sparas indexet för raden i en lista, indexet i dataframen är datumet.

8.4 Grafiskt gränssnitt

Figur 10 Startsidan för det grafiska gränssnittet

Namn	Status	Senast ändrad	Typ	Storlek
Radata_vatten_plats1	✓	2020-04-22 10:30	Microsoft Excel-fil med kommaavgränsade värden	8 264 kB
Radata_vatten_plats2	✓	2020-04-01 11:30	Microsoft Excel-fil med kommaavgränsade värden	4 239 kB
Radata_vatten_plats3	✓	2020-04-01 11:29	Microsoft Excel-fil med kommaavgränsade värden	8 362 kB
Radata_vatten_plats4	✓	2020-04-01 11:31	Microsoft Excel-fil med kommaavgränsade värden	8 370 kB
Radata_vatten_plats5	✓	2020-04-01 11:29	Microsoft Excel-fil med kommaavgränsade värden	8 365 kB

Figur 11 Val av CSV-filer som kan analyseras

Det grafiska gränssnittet är byggt med hjälp av biblioteket Tkinter [Pyt20a]. Detta bibliotek innehåller stöd för knappar, dropdown-listor, texttrutor med mera som används för att användaren ska kunna välja CSV-fil och datahanterare samt visualisera resultatet från styrdiagrammen. Det grafiska gränssnittet visualiserar den valda datan applicerad på styrdiagrammen genom att tre grafer skapas med respektive styrdiagram och flödesdata.

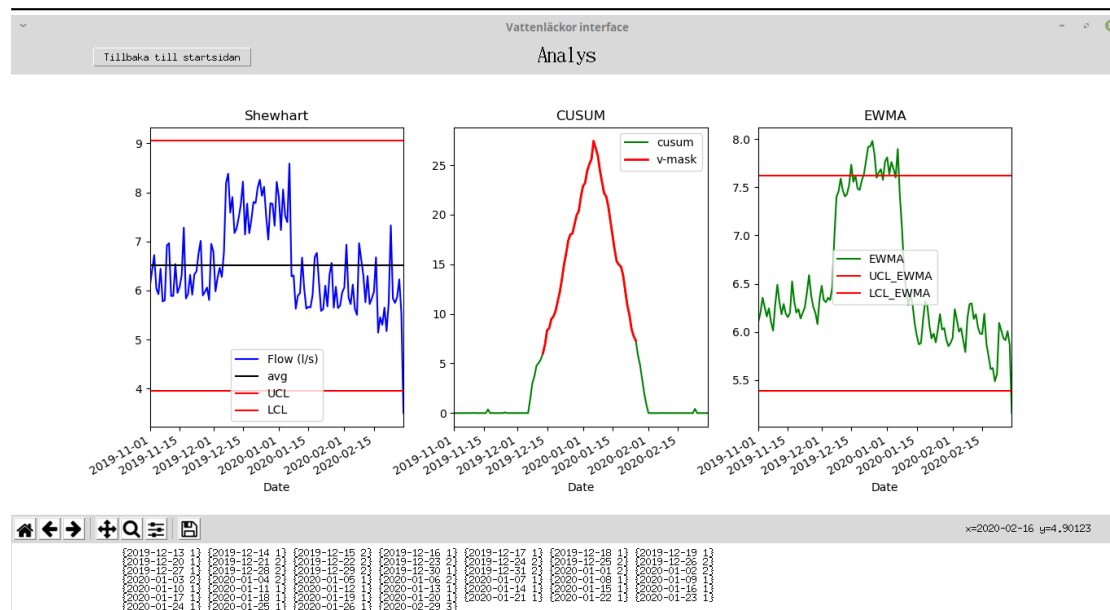


Figur 12 Vald CSV-fil och en dropdown-lista för att välja datahanterare

I graferna syns kontrollgränserna för när en datapunkt anses vara ur kontroll och användaren kan själv se om den valda datan passerar dessa gränser i diagrammen. Eftersom x-axeln består av datum går det också att läsa av när läckan är. Funktionerna för styrdiagrammen returnerar en lista med datum då modellen anser att flödesdatan är ur kontroll. Eftersom de listorna finns går det att undkomma fel med avläsning i graferna genom att det även finns en ruta där programmet skriver ut de datum då styrdiagrammen markerat en läcka. Bredvid varje datum finns en siffra som beskriver hur många av styrdiagrammen som markerar en läcka på det datumet.

Koden är kompilerad till en körbar fil för att användaren enkelt ska kunna starta programmet. När programmet startar visas först en startsida, se figur 10. För startsidan utnyttjades funktioner i Tkinter för att implementera en filutforskare i vilken användaren kan välja vilken fil programmet ska analysera, se figur 11. På startsidan finns också en dropdown-lista som används för att välja vilken datahanterare som ska användas, se figur 12. Slutligen finns en knapp "Analysera" som används för att gå till en ny sida i programmet.

Den andra sidan visar analysen av vald data. För att navigera mellan de två sidorna används en form av stack i Tkinter. På den stacken ligger det två så kallade frames [Tut]. Frames i Tkinter fungerar som en låda i vilken andra saker kan organiseras. Allt som visas på start- och analysidan ligger i sidornas respektive frames. Den frame som ligger högst upp på stacken är den som visas för användaren och programmet visar olika innehåll genom en funktion som flyttar vilken frame som ligger högst upp. När "Analysera"-knappen trycks på körs alltså en funktion som flyttar analys-frame högst upp.



Figur 13 Analys-sidan som visar styrdiagrammen och datum med länkor

Programmet är implementerat så att direkt när analys-sidan visas körs en funktion som anropar alla tre styrdiagrammen med den datahanterare som valts. Listorna med datum som de returnerar sparas. När funktionerna för styrdiagrammen körts så läser gränssnittet från den uppdaterade dataframen och använder värdena i den för att plotta tre figurer som visualiserar den valda indatan applicerad på styrdiagrammen. De tre listorna som returnerats från styrdiagrammen slås ihop till en och programmet räknar hur många gånger samma datum förekommer. Datumen läggs sedan in som en tuple [Pyt20b] i en lista där det andra värdet i tuplen är en siffra som representerar hur många styrdiagram som markerar en länka det datumet. Den listan skrivs sedan ut under de tre diagrammen, se figur 13.

9 Utvärderingsresultat

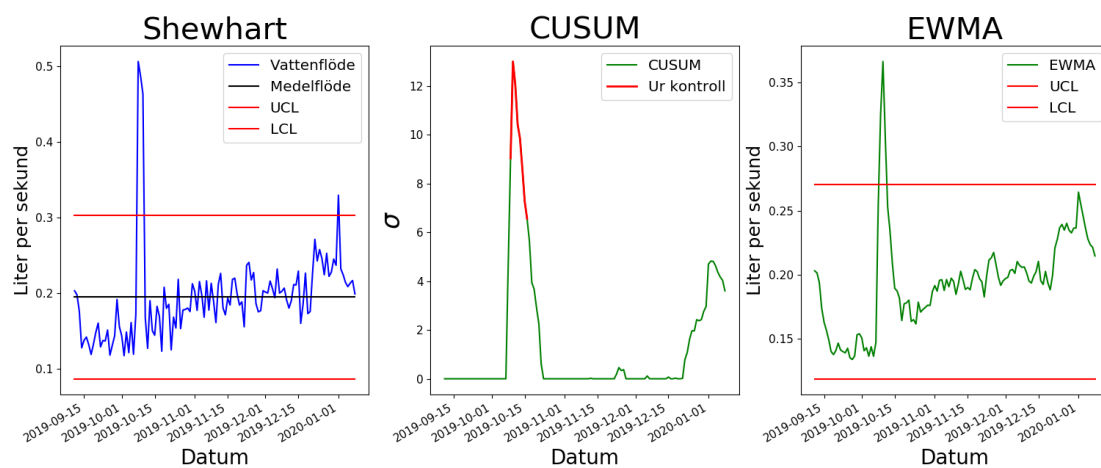
Här presenteras resultaten av utvärderingsmetoderna i del 7.2. Endast resultaten av de enskilda utvärderingsmetoderna presenteras. Värderingar, tolkningar och slutsatser av utvärderingarna återfinns istället i del 10.

9.1 Identifiering av läckor

I figur 14-18 så visas ett Shewhart-diagram, ett CUSUM-diagram och ett EWMA-diagram för varje av de fem dataseten med flödesdata som intressenten tillhandagav. Datanteraren *"nightMean"* har använts vilket innebär att diagrammen baseras på snittflödet av vatten under nätterna.

Två av dataseten innehöll en läcka. I båda de dataseten så identifierades läckan av två av tre modeller. Den tredje modellen gav ett typ II-fel i båda dataseten. I de tre dataseten där det inte fanns läckor så gav alla tre modeller ett typ I-fel.

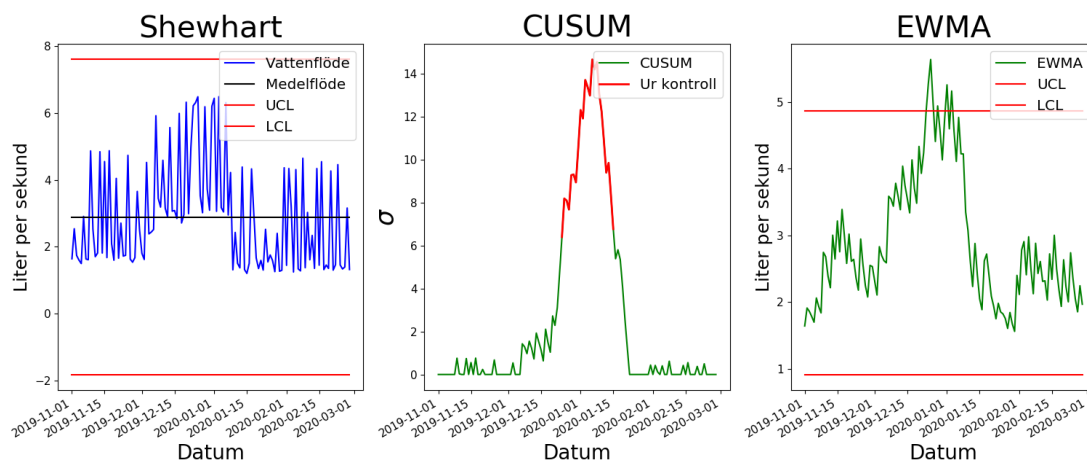
9.1.1 Resultat av dataset 1



Figur 14 Resulterade styrdiagram Shewhart, CUSUM och EWMA för dataset 1

I Shewhart diagrammet i figur 14 så överstiger flödet den övre kontrollgränsen 8-10:e oktober. I CUSUM-diagrammet i figur 14 så överstiger kurvan referensvärdet 9-16:e oktober. I EWMA-diagrammet i figur 14 så överstiger kurvan den övre kontrollgränsen 9-11:e oktober.

Detta dataset innehöll ingen läcka. Alla tre modeller gav ett typ I-fel och kontrollgränserna klarade inte kravet.



Figur 15 Resulterade styrdiagram Shewhart, CUSUM och EWMA för dataset 2

9.1.2 Resultat av dataset 2

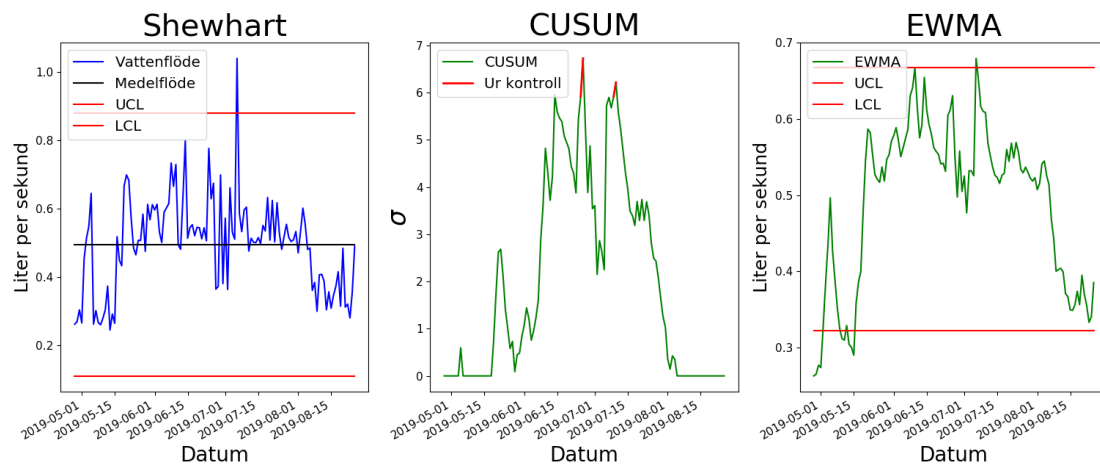
I Shewhart diagrammet i figur 15 så är flödet alltid inom kontrollgränserna. I CUSUM-diagrammet i figur 15 så överstiger kurvan referensvärdet 24:e december-15:e januari. I EWMA-diagrammet i figur 15 så överstiger kurvan den övre kontrollgränsen 23:e december-6:e januari.

Detta dataset innehöll en läcka som lagades den 7:e januari. CUSUM och EWMA indikerade en läcka i anslutning till att det lagades en läcka och deras kontrollgränser klarade kravet. Shewharts kontrollgränser gav ett typ II-fel och klarade inte kravet.

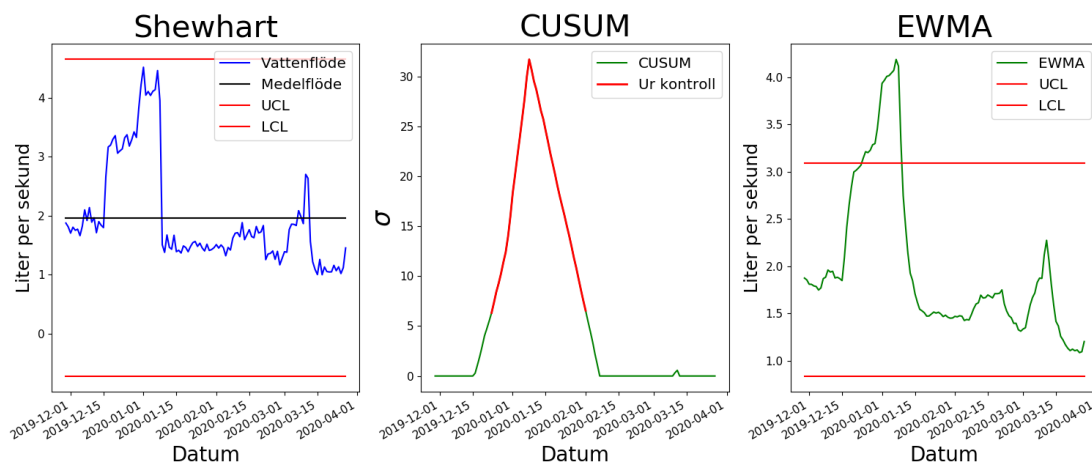
9.1.3 Resultat av dataset 3

I Shewhart diagrammet i figur 16 så överstiger flödet den övre kontrollgränsen 6:e juni. I CUSUM-diagrammet i figur 16 så överstiger kurvan referensvärdet 25-26:e maj och 9-10:e juni. I EWMA-diagrammet i figur 16 så överstiger kurvan den övre kontrollgränsen 10:e juni och 14:e juni.

Detta dataset innehöll ingen läcka. Alla tre modeller gav ett typ I-fel och kontrollgränserna klarade inte kravet.



Figur 16 Resulterade styrdiagram Shewhart, CUSUM och EWMA för dataset 3



Figur 17 Resulterade styrdiagram Shewhart, CUSUM och EWMA för dataset 4

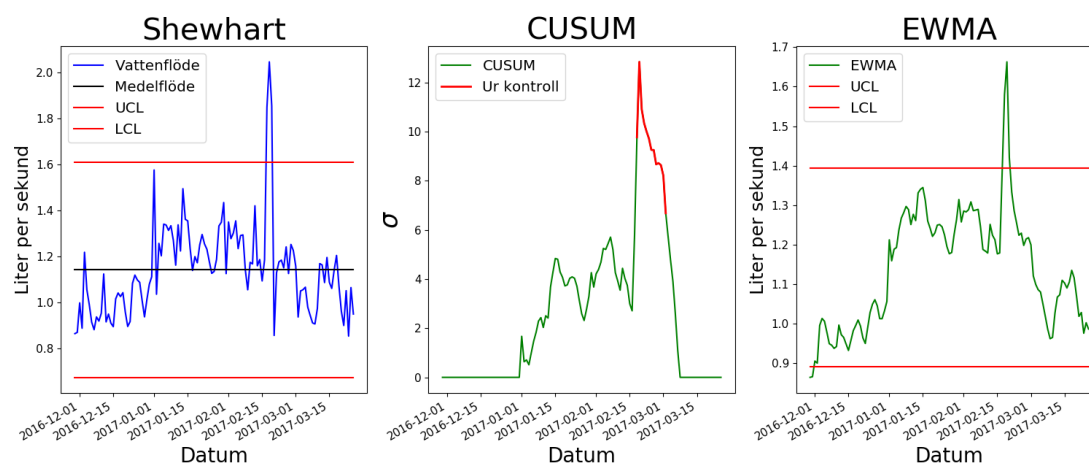
9.1.4 Resultat av dataset 4

I Shewhart diagrammet i figur 17 så är flödet alltid inom kontrollgränserna. I CUSUM-diagrammet i figur 17 så överstiger kurvan referensvärdet 23:e december-1:a februari. I EWMA-diagrammet i figur 17 så överstiger kurvan den övre kontrollgränsen 22:a december-10:e januari.

Detta dataset innehöll en läcka som lagades den 9:e januari. CUSUM och EWMA indikerade en läcka i anslutning till att det lagades en läcka och deras kontrollgränser

klarade kravet. Shewharts kontrollgränser gav ett typ II-fel och klarade inte kravet.

9.1.5 Resultat av dataset 5



Figur 18 Resulterade styrdiagram Shewhart, CUSUM och EWMA för dataset 5

I Shewhart diagrammet i figur 18 så överstiger flödet den övre kontrollgränsen 17-19:e februari. I CUSUM-diagrammet i figur 18 så överstiger kurvan referensvärdet 18:e februari-2:a mars. I EWMA-diagrammet i figur 18 så överstiger kurvan den övre kontrollgränsen 18-21:a februari.

Detta dataset innehöll ingen läcka. Alla tre modeller gav ett typ I-fel och kontrollgränserna klarade inte kravet.

9.2 Hantering av felaktigt filformat

Ett av kraven i avsnitt 7.1 var att användaren ska kunna välja en fil av fel format utan att programmet kraschar. Användaren har alltså möjligheten att välja en fil av annat format än CSV utan att programmet stängs ner eller slutar fungera. Det grafiska gränssnittet kollar därför om filen slutar på `".CSV"` för att avgöra om filen är i CSV format och returnerar ett textmeddelande som uppmanar användaren till att välja en annan fil av rätt format. Kravet kommer därför uppnås då endas CSV-filer kommer att ha ett filnamn som slutar på `".CSV"`.

9.3 Tidskomplexitet

Alla funktioner i implementationen analyserades och tidskomplexitet räknades ut för algoritmerna enligt utvärderingsmetoden i del 7.2. Förutom CUSUM-biblioteket så hade alla algoritmer en loop och i den loopen som mest en if-sats. De algoritmerna hade då en tidskomplexitet på $\theta(n)$ eller $\theta(\log(n))$ vilket är lägre än $\theta(n^2)$ och de uppnådde kravet. $\theta(n)$ innebär att antalet instruktioner som utförs växer proportionellt med mängden data och $\theta(\log(n))$ innebär att de växer logaritmiskt. Funktionen som i CUSUM-biblioteket utförde beräkningarna innehöll en nästlad loop vilket innebar en tidskomplexitet på $\theta(n^2)$. Detta hade inte klarat kravet. En algoritm med samma resultat men endast en loop skapades och tidskomplexiteten sänktes till $\theta(\log(n))$. Då hade alla algoritmer i systemet en tidskomplexitet lägre än $\theta(n^2)$ och kravet var uppnått.

10 Resultat och diskussion

I denna del presenteras det resultat som tagits fram och olika felkällor som kan ha påverkat detta. Styrdiagrammen Shewhart, CUSUM och EWMA:s prestation tas upp och möjliga felkällor diskuteras.

10.1 Resultat

En desktopapplikation som används för att beräkna ifall det finns potentiella läckor skapades. Beräkningen görs med hjälp av dataset bestående av flödesdata vid mätare ute i ledningarna. Den analys som görs av dataseten för att finna avvikande data, som ett resultat av en läcka, görs med hjälp av tre statistiska processkontroll metoder: Shewhart, CUSUM och EWMA. För att testa ifall programmet fungerade efter önskad funktionalitet försågs det med fem dataset där läckor existerade vid givna tidsperioder beskrivna i del 9.1. Applikationen fann läckor i båda två fallen vid de givna datumen. Programmet indikerade dock läckor även i de andra tre dataseten. Dessa avvikelser berodde, enligt intressentens uppgifter, endast på variation i vattenförbrukning alternativt fel vid mätaren. Resultaten presenteras grafiskt i applikationen med diagram samt textrutor där de datumen med avvikande data visas, se figur 13. Denna typ av presentation visas för alla tre styrdiagrammen. Analys av styrdiagrammen visar att EWMA och CUSUM fungerar bäst. EWMA- och CUSUM-diagrammen identifierade läckan i båda två dataseten, medan Shewhart-diagrammet missade båda. Applikationen uppnådde delvis kravbild, beskriven i del 7.1, med undantag för dessa typ I-fel.

10.2 Diskussion

Utgången av projektet är jämfört med de kraven som fanns delvis tillfredsställande. Alla angivna läckor i dataseten identifierades med antingen en eller flera av styrdiagrammen. De styrdiagrammen som påvisade en läcka i alla fallen var CUSUM och EWMA, medan Shewhart missade båda. En stor anledning till detta är att båda dessa styrdiagram kan identifiera mycket mindre variationer i datan, i jämförelse med Shewhart. Då Shewharts kontrollgränser bygger starkt på medelvärdet i datasetet skapar det problematik när läckan är över en längre tid. Ett exempel på när detta inträffar, för nämnt styrdiagram, är i figur 15. Där utgör en stor del av den angivna datan en läcka, vilket resulterar i att medelvärdet höjs och kontrollgränserna skjuts längre ifrån varandra. När detta sker identifierar inte Shewhart-diagrammet en läcka medan båda de andra styrdiagrammen gör det. För att motverka en sådan effekt vid användande av Shewhart-diagrammet skulle en större mängd data behövs användas. En större mängd data kan leda till att läckan och den avvikande datan utgör en mindre del av hela datamängden, vilket skulle skapa bättre och precisare kontrollgränser.

En annan intressant aspekt som skulle kräva en större mängd data är att skapa gränserna i styrdiagrammen på en del av datan och sedan testköra på den andra delen. Detta skulle leda till ett mer verklighetstroget resultat, då det inte egentligen är intressant för intressenten att det har varit en läcka som de lagat utan snarare att en läcka har inträffat just nu.

10.2.1 Val av styrdiagram

Vid första anblick av utvärderingsresultatet så är en enkel slutsats att Shewhart-diagrammet inte fungerar alls medan CUSUM och EWMA beter sig ganska likvärdigt. Modellerna har dock olika fördelar och kan vara användbara i olika syften.

Som tidigare nämnt kan Shewhart-diagrammet kräva ett dataset där läckan inte utgör för stor del för att korrekta gränser skall skapas. Med korrekta gränser kan diagrammet fungera bra för att snabbt indikera en stor läcka. Då det är rå flödesdata som jämförs med kontrollgränserna så kommer en stor ökning i vattenflöde direkt avspeglas på kurvan. I jämförelse med CUSUM och EWMA, som väger in föregående datapunkter, så räcker det i Shewhart med att en datapunkt har tillräckligt högt värde för att indikera en läcka. I teorin skulle korrekta kontrollgränser alltså kunna indikera en läcka så snabbt som mätaren avläses.

CUSUM-diagrammets användande av föregående punkter gör att det kommer att kunna identifiera små läckor bättre än Shewhart och EWMA. Flera efterföljande punkter i

flödesdatan som ligger strax ovanför medelvärdet kommer att göra att CUSUM-kurvan stiger kontinuerligt och till slut överstiger referensvärdet. Med ett korrekt medelvärde och referensvärde så kommer CUSUM-diagrammet alltså över tid att kunna indikera en läcka oavsett storlek.

EWMA-diagrammet kan anpassas genom att ändra på variabeln λ . Genom att sätta λ närmare 1 eller närmare 0 så kommer diagrammet lägga mer eller mindre vikt vid föregående punkter. Högre λ ger mindre vikt vid föregående punkter och gör att den likt Shewhart kommer att kunna upptäcka större läckor snabbare. Lägre λ ger mer vikt vid föregående punkter och gör att den likt CUSUM kommer att kunna upptäcka läckor oavsett storlek men långsammare.

Shewhart och CUSUM har sina tydliga fördelar. Om man vill identifiera stora läckor snabbt är Shewhart det bästa valet, om man vill identifiera små läckor så är CUSUM det bästa valet. Om man vill utnyttja båda modellernas styrkor så kan man använda EWMA och sätta λ efter vilken styrka man värderar högst. Det kan dock krävas många dataset för att hitta det λ som är lämpligast.

10.2.2 Säsongsvariation

En viktig reell aspekt som har varit problematisk och haft tydlig inverkan på resultatet är säsongsvariationen i vattenförbrukning. Med säsongsvariation menas det faktum att mängden vattenförbrukning inte är konstant året om. Mängden kan variera på grund av flera olika anledningar, till exempel tillfällen som att det är sommar och flera personer vill fylla på poolen eller vid en period på året då en stor mängd av invånarna är på ledighet. Då applikationen endast har körts på dataset som innefattar fyra månader är problemet med säsongsvariation svårt att korrigera. Med en längre tidsperiod på exempelvis tre år skulle den varierande vattenförbrukningen eventuellt kunnat kartläggas och en metod för att hantera detta arbetats fram.

10.2.3 Typ I-fel

Antalet typ I-fel som producerats efter analys av dataseten är ett problem. Typ I-felen som registreras beror med stor sannolikhet på fel ute i mätaren. Detta är en slutsats som dras på det faktum att när dessa fel uppstår når flödesdatan en hög topp snabbt för att sedan sjunka drastiskt, se exempel i figur 14. För att minska dessa fel är det möjligt att ändra på känslighetsvariablerna i EWMA- och CUSUM-diagrammen, se formel 6 samt 5. Det skulle leda till att dessa reagerar olika mycket på förändringar i datan. Att minska känsligheten skulle dock få följdproblemet att typ II-felen då ökar, alltså att

läckor kan ske utan att de identifieras. Denna balans mellan de olika typerna av fel är något väldigt komplicerat att balansera. För att kunna göra någon typ av optimal balans mellan de blir mängden dataset att analysera tillsammans med intressentens önskemål relevanta. En stor mängd dataset där flera läckor existerar skulle bidra till att läckors olika karaktäristiska drag skulle kunna analyseras i en större utsträckning. Intressentens önskemål skulle kunna ge en riktlinje angående vad som är viktigast. Antingen så kan programmet ge få falska alarm men läckor kan ske utan att det identifieras eller så kan programmet ge många falska alarm men alla läckor identifieras.

En noterbar detalj i de existerande typ I-felen är skalorna på y-axlarna. När en läcka faktiskt har inträffat ökar vattenflödet med en noterbart större mängd än vid ett falskt alarm. Anledningen till att detta falsklarm sker beror på att när ingen läcka existerar skapas de olika kontrollgränserna för styrdiagrammen betydligt tätare kring målvärdet än när en läcka finns. Det leder till att toppar där identifieras som läckor medan de inte hade setts som läckor med de andra kontrollgränserna. Alltså skulle falsklarmen kunna undvikas i dagsläget genom att fysiskt avläsa de skapade kurvorna.

10.2.4 Val av datahanterare

Alla datahanterare testades under projektet för att se vilka som upptäckte läckor. Slutsatsen var att de datahanterare som räknade ut medelvärden för flera timmar var både snabbare och gav bättre resultat. Alltså funktionerna som räknade ut medelvärdet för varje datum (*"dateMean"*), dag (*"dayMean"*) och natt (*"nightMean"*). Funktionernas producerade dataset som var mindre än från funktionerna som räknade ut medelvärdet för varje timme vilket gjorde att styrdiagrammen hade lägre exekveringstid. Av dessa datahanterare producerade medelvärdet varje natt (*"nightMean"*) det dataset som hade mest relativt konstanta värden.

Uppsala Vatten använder i dagsläget medelvärde per dygn för att analysera vattenflödet i sina ledningar. Detta inkluderar möjliga toppar av flöde på dagen då mycket vatten behövs för industrier eller liknande. En möjlig förbättring skulle vara att istället använda medelvärdet varje natt då dessa toppar inte blir lika markanta på grund av den låga vattenförbrukningen.

En utvecklingsmöjlighet är att dynamiskt ändra tider då flödesdatan sparas från originaldatan. Alltså att ta fram de datapunkterna med minimalt vattenflöde per natt och sedan räkna medelvärdet av punkterna. Eventuella toppar på natten skulle då minimeras och det omformaterade dataset skulle troligen ha mer konstanta flödesvärden. I nuläget finns det möjlighet att toppar av flödet finns med i uträkningen vilket kan markant höja medelvärdet för vattenflödet vissa nätter.

11 Slutsatser

Projektet kan sammanfattas med att en desktopapplikation har skapats som med hjälp av flödesdata från en mätare ute i en vattenledning kan identifiera när en läcka har inträffat. Applikationen bygger på teorin om statistisk processkontroll och identifierar läckorna med styrdiagrammen: Shewhart, CUSUM och EWMA. Detta arbete har skapat applikationen som en viktig grund för att minska Uppsala vattens vattenförlust och då påverkan på miljön. Det visar även att det är möjligt att endast med hjälp av dessa matematiska modeller finna läckor. En viktig lärdom att ta vidare till vidare utveckling är betydelsen av stora mängder data för att kunna skapa representativa övre och undre kontrollgränser som skapar en önskad balans mellan typ I- och II-felen.

12 Framtida arbete

Under projektets gång har frågeställningar och problem diskuteras vilket har lett till olika idéer för att hitta läckor på bättre sätt. Förbättringspotentialen är stor och i denna del kommer vissa idéer om potentiella framtida utvecklingar för programmet diskuteras.

12.1 Vattenledningar och vattenverk

För att förbättra det program som tagits fram skulle mer kunskap om hur Uppsala Vatten sköter sina vattenledningar och hur vattenförsörjningen fungerar vara till stor hjälp. På grund av att denna typ av information klassificeras som känslig och inte får spridas till allmänheten så har det varit svårt att få svar på frågor under projektets gång. Varför vattenflödet ökar och inte minskar när det skett en läcka och att vattenflödet ökar utan direkt anledning mitt på natten är exempel på frågeställningar som kan utforskas vidare.

12.2 Maskininlärning med hjälp av fler mätvariabler och dataset

En integration av maskininlärning kan vara intressant att undersöka hur det kan gynna läcksökningen i programmet. De mätare som använts under projektet registrerar, utöver flöde, vattnets temperatur och tryck i ledningen. Med hjälp av maskininlärning finns möjligheten att hitta ett samband mellan dessa variabler som kan tyda på läckage. Fler

och större dataset skulle då krävas för att lära programmet om vad en läcka innebär och hur de ser ut i datan.

12.3 Kontinuerlig läcksökning i realtid

En möjlig implementation för att automatisera läcksökningen ytterligare är att dataflödet skulle läsas kontinuerligt vilket innebär att programmet läser av flödesmätarna i realtid. Användare skulle i detta fall inte behöva välja en vattenledning med flödeshistoria som ska analyseras av programmet utan kan välja en vattenledning som ska undersökas för läckor under en längre tid framöver. Programmet skulle då läsa data kontinuerligt från de mätare som sitter i den valda vattenledningen med jämna intervaller. Möjligheten att kommunicera med mätarna var inte möjligt på grund av att Uppsala Vatten inte får dela sekretessklassad eller hemlig allmän handling till obehörig men det är en punkt som de själva kan arbeta vidare på.

Referenser

- [AdC] AdCon. Adcon, sensors. [Online]. Tillgänglig: <https://www.adcon.com/products/sensors-284/>
- [Ame] What is statistical process controll? American Society for Quality. Hämtad 2020-04-20. [Online]. Tillgänglig: <https://asq.org/quality-resources/statistical-process-control>
- [Ame02] American Petroleum Institute, "Computational pipeline monitoring for liquid pipelines," 2002, hämtad 2020-04-20. [Online]. Tillgänglig: <https://law.resource.org/pub/us/cfr/ibr/002/api.1130.2002.pdf>
- [Bin] Moving average explained. Binance Academy. Hämtad 2020-04-29. [Online]. Tillgänglig: <https://www.binance.vision/economics/moving-averages-explained>
- [BN06] M. Best and D. Neuhauser, "Walter a shewhart, 1924, and the hawthorne factory," Apr. 2006, hämtad 2020-04-27. [Online]. Tillgänglig: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2464836/>
- [Bä19] M. Bäckström, "Fakta om utläckage av dricksvatten i ledningsnätet," Sep. 2019, hämtad 2020-04-20. [Online]. Tillgänglig: <https://www.svensktvatten.se/vattentjanster/rornat-och-klimat/fakta-om-utlackage/?fbclid=IwAR01IKBIC6dK-MP2O-m3U7xXYFCpbs1NweApOjxLTquvxxkAWfQaxwzpRl20>
- [Coo13] J. Cooper, "Python's null equivalent: None," Jun. 2013, hämtad 2020-05-07. [Online]. Tillgänglig: <https://www.pythoncentral.io/python-null-equivalent-none/>
- [Den18] P. Denbaum, "Bättre koll på vattenförbrukning," Mar. 2018, hämtad 2020-04-20. [Online]. Tillgänglig: <https://www.unt.se/ekonomi/battre-koll-pa-vattenforbrukning-4927294.aspx>
- [Des19] R. Desai, "Top 10 python libraries for data science," *Medium*, 2019, hämtad 2020-05-06. [Online]. Tillgänglig: <https://towardsdatascience.com/top-10-python-libraries-for-data-science-cd82294ec266>
- [DKLM05a] F. M. Dekking, C. Kraaikamp, H. P. Lopuhaä, and L. E. Meester, *A Modern Introduction to Probability and Statistics*. Springer-Verlag, 2005, hämtad 2020-04-29. [Online]. Tillgänglig: https://books.google.se/books?id=odn7_ausAnEC&

printsec=frontcover&dq=isbn:9781852338961&hl=sv&sa=X&ved=0ahUKEwjn953RxL3pAhXk_CoKHTUEBmgQ6AEIKDAA#v=onepage&q&f=false

- [DKLM05b] F. M. Dekking, C. Kraaikamp, H. P. Lopuhaä, and L. E. Meester, *A Modern Introduction to Probability and Statistics*. Springer-Verlag, 2005, hämtad 2020-04-29. [Online]. Tillgänglig: https://books.google.se/books?id=odn7_auSAnEC&printsec=frontcover&dq=isbn:9781852338961&hl=sv&sa=X&ved=0ahUKEwjn953RxL3pAhXk_CoKHTUEBmgQ6AEIKDAA#v=onepage&q&f=false
- [Duf17] D. Duffy. (2017, Dec.) Successful water leak detection and audit methods. Hämtad 2020-04-17. [Online]. Tillgänglig: <https://www.waterworld.com/home/article/14070706/successful-water-leak-detection-and-audit-methods>
- [Dun20] K. Dunn, “Shewhart charts,” May 2020, hämtad 2020-05-07. [Online]. Tillgänglig: <https://learnche.org/pid/process-monitoring/shewhart-charts>
- [HAMS13] M. Hussein, A. Al-Morshedi, and H. Shomran, “The comparison between shewhart control chart ,cusum and ewma,” Dec. 2013. [Online]. Tillgänglig: https://www.researchgate.net/publication/320616755_The_Comparison_between_Shewhart_Control_Chart_Cusum_and_EWMA
- [HGK18] A. Haq, R. Gulzar, and M. B. C. Khoo, “An efficient adaptive ewma control chart for monitoring the process mean,” *Quality and Reliability Engineering International*, 2018. [Online]. Tillgänglig: <https://onlinelibrary.wiley.com/doi/abs/10.1002/qre.2272>
- [Jon09] J. Jonasson, “Hypotesprövning - styrka,” Feb. 2009, hämtad 2020-05-08. [Online]. Tillgänglig: http://www.math.chalmers.se/Stat/Grundutb/CTH/lma135/0809/hypotes_styrka.pdf
- [MCE16] H. Morgan, P. Carpenter, and N. R. Edward, *Pipeline Leak Detection Handbook*. Elsevier, 2016. [Online]. Tillgänglig: https://app.knovel.com/web/toc.v/cid:kpPLDH000C/viewerType:toc/root_slug:pipeline-leak-detection?kpromoter=federation
- [McN14] D. B. McNeese. (2014, Sep.) Keeping the process on target: Ewma chart. Hämtad 2020-04. [Online]. Tillgänglig: <https://www.spcforexcel.com/knowledge/variable-control-charts/keeping-process-target-ewma-chart#introduction>

- [Mic] Microsoft. Visual studio code. Hämtad 2020-05-06. [Online]. Tillgänglig: <https://code.visualstudio.com/>
- [Nat] National Institute of Standards and Technology. Cusum control charts. Hämtad 2020-04-23. [Online]. Tillgänglig: <https://www.itl.nist.gov/div898/handbook/pmc/section3/pmc323.htm>
- [Nat92] Naturvårdsverket. (1992, Mar.) Vattenverk. Hämtad 2020-04-17. [Online]. Tillgänglig: <https://www.naturvardsverket.se/Documents/publikationer/91-620-9378-9.pdf?pid=4315>
- [Ope] Open source. Atom. Hämtad 2020-05-06. [Online]. Tillgänglig: <https://atom.io/>
- [Ora] Java programming language. Oracle. Hämtad 2020-05-06. [Online]. Tillgänglig: <https://www.java.com/sv/>
- [Pan] Pandas. Pandas - python data analysis library. Hämtad 2020-05-06. [Online]. Tillgänglig: <https://pandas.pydata.org/>
- [Pyt] Python Software Foundation. Python programming language. Hämtad 2020-05-06. [Online]. Tillgänglig: <https://www.python.org/>
- [Pyt20a] Python Software Foundation. (2020, May) Graphical user interface with tk. Hämtad 2020-05-06. [Online]. Tillgänglig: <https://docs.python.org/3/library/tk.html>
- [Pyt20b] Python Software Foundation. (2020, May) Tuple objects. Hämtad 2020-05-11. [Online]. Tillgänglig: <https://docs.python.org/3/c-api/tuple.html>
- [Reg20] Region Gotland. (2020, Feb.) Spara vatten. Hämtad 2020-05-18. [Online]. Tillgänglig: <https://www.gotland.se/sparavatten>
- [Ryd14] J. Rydén, *Stokastik för ingenjörer*. Studentlitteratur AB, 2014, hämtad 2020-04-28.
- [Sip06] M. Sipser, *Introduction to the theory of computation*, 2nd ed. Boston, USA: Thomson Course Technology, 2006.
- [SSH11] R. W. Sterner, G. E. Small, and j. M. Hood, "The conservation of mass," 2011, hämtad 2020-04-20. [Online]. Tillgänglig: <https://www.nature.com/scitable/knowledge/library/the-conservation-of-mass-17395478/>
- [Sta] Stack overflow developer survey. Most popular development environments. Hämtad 2020-05-06. [Online]. Tillgänglig: <https://insights.stackoverflow.com/survey/2019#technology--most-popular-development-environments>

- [Sub16] SubSurface Leak Detection, Inc, "How to find leaks," 2016, hämtad 2020-04-17. [Online]. Tillgänglig: http://www.subsurfaceleak.com/find_leaks.html
- [Tut] Python - tkinter frame. Tutorialspoint. Hämtad 2020-05-11. [Online]. Tillgänglig: https://www.tutorialspoint.com/python/tk_frame.htm
- [UND] UNDP (FN:s utvecklingsprogram). Globala målen. Hämtad 2020-04-20. [Online]. Tillgänglig: <https://www.globalamalen.se/>
- [UND20a] UNDP (FN:s utvecklingsprogram), "Globala målen, mål 12," Apr. 2020, hämtad 2020-04-20. [Online]. Tillgänglig: <https://www.globalamalen.se/om-globala-malen/mal-12-hallbar-konsumtion-och-produktion/>
- [UND20b] UNDP (FN:s utvecklingsprogram), "Globala målen, mål 14," Apr. 2020, hämtad 2020-04-20. [Online]. Tillgänglig: <https://www.globalamalen.se/om-globala-malen/mal-14-hav-och-marina-resurser/>
- [UND20c] UNDP (FN:s utvecklingsprogram), "Globala målen, mål 6," Jan. 2020, hämtad 2020-04-20. [Online]. Tillgänglig: <https://www.globalamalen.se/om-globala-malen/mal-6-rent-vatten-och-sanitet/>
- [Upp19] Uppsala Vatten och Avfall AB. (2019, Mar.) Årsredovisning 2018. Hämtad 2020-04-20. [Online]. Tillgänglig: https://www.uppsalavatten.se/globalassets/dokument/om-oss/arsredovisning/uv_arsredovisning_2018_web.pdf
- [Wik20a] Wikipedia contributors. (2020) C (programming language) — Wikipedia, the free encyclopedia. Hämtad 2020-05-06. [Online]. Tillgänglig: [https://en.wikipedia.org/wiki/C_\(programming_language\)](https://en.wikipedia.org/wiki/C_(programming_language))
- [Wik20b] Wikipedia Contributors. (2020) Comma-separated values — Wikipedia, the free encyclopedia. Hämtad 2020-05-08. [Online]. Tillgänglig: https://en.wikipedia.org/wiki/Comma-separated_values