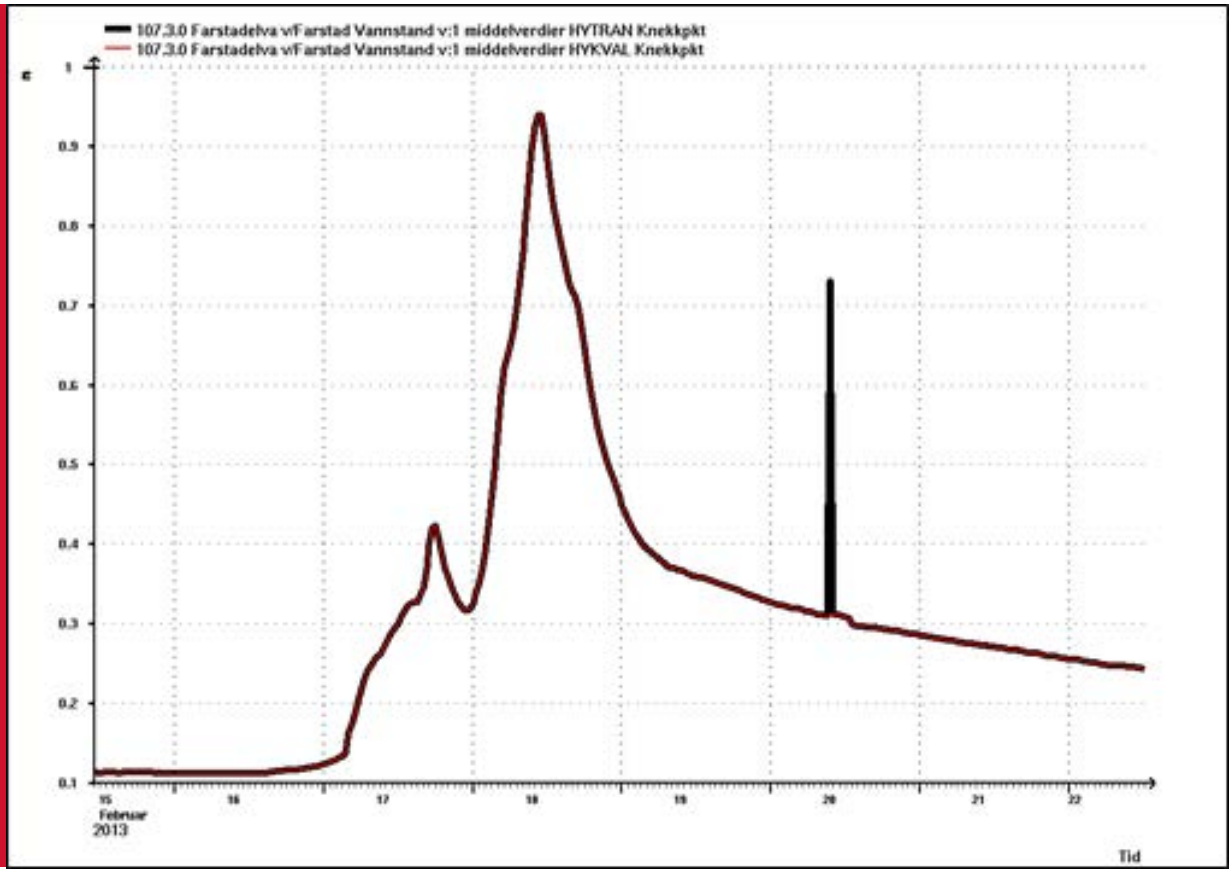
RAPPORT

Nr. 15/2022

Review of methods for automation of quality control on hydrologic time series and considerations for a research approach at NVE

Trond Reitan



NVE Rapport nr. 15/2022
Review of methods for automation of quality control on hydrologic time series and considerations for a research approach at NVE

Published by: Norwegian Water Resources and Energy Directorate

Author: Trond Reitan

Cover photo: Anomaly in data from the discharge station 107.3.0 Farstadelva v/Farstad

ISBN: 978-82-410-2206-7

ISSN: 1501-2832

Print: NVEs hustrykkeri

Number printed: 10

Abstract: This report contains a literature study on the topic of automatic anomaly detection in time series using machine learning. The study has been carried out in order to see what the state of field is, both in academic circles and in real world applications in institutions that gather hydrologic or meteorologic time series. The academic research is far more advanced of what hydrologic and meteorologic institutions are currently implementing, however some initial forays into applying the theory by such institutions have started. The research is however applied to time series across a much wider spectrum of applications, so it is hard to judge how well a method works for NVE purposes. Therefore, testing at NVE is required to evaluate how well various methods work. Performance should be based on how good the end results are as well as their speed and robustness

Key words: Anomaly detection, hydrological time series, machine learning

Norwegian Water Resources and Energy Directorate
Middelthuns gate 29
P.O. Box 5091 Majorstuen
N-0301 Oslo
Norway

Telephone: 22 95 95 95
E-mail: nve@nve.no
Internet: www.nve.no

March 2022

Content

Preface

The demands for providing and handling correct data in real time is increasing. Thus, there is also an incentive for detecting and correcting errors faster than before. This project was started in September 2021 to investigate the possibilities for using machine learning for automatic detection and correction of errors in NVE's hydrological time series. Machine learning is a concept with wide interpretation, spanning from the field of statistics into algorithms with other lines of thinking. The purpose with this report was to check what similar institutions as NVE are doing in terms of testing/using machine learning for correcting hydrological time series, as well as performing a review on available literature.

Error detection and correction includes performing what we at NVE refer to as «primary» and «secondary» control. Primary control can be described as correction of measurement errors in a stage time series period recently transferred from the real time data archive to a more permanent archive. The purpose of the primary control is to achieve correct stage data. Secondary control can be described as adjusting stage values specifically, to make discharge data (calculated through the stage-discharge rating curve) as correct as possible. Adjusting data due to backwater from ice and data reconstruction are the most prevalent secondary control corrections. However, to meet the demands for correct real time data NVE also has the ambition to correct the real time archive directly, possibly both when the data first arrives and again later when it can be put into context.

Anomaly detection in time series, as the field is called in literature, is a large and branching topic with continuous development. Thus, in this report classes of solutions are in focus, rather than specific methods. Strengths and weaknesses are evaluated, rather than simply stating which method is best for which task. It is hoped that this evaluation of methods can help in making decisions for further work in anomaly detection and automatic correction of hydrological data at NVE.

Oslo, March 2022

Hege Hisdal
Director
Hydrology Department

Nils Kristian Orthe
Deputy Section Head
ICT and Information Management – Development and Consulting

This document is sent without signature. The content is approved according to internal routines.

Summary

I have performed a literature search on the topic of automatic anomaly detection in time series using machine learning. This was done in order to see what the state of field was, both in academic circles and in real world applications in institutions that gather hydrologic or meteorologic time series. My impression was that the academic research is far advanced of what hydrologic and meteorologic institutions are currently implementing, but that some initial forays into applying the theory by such institutions have started. The research is however applied to time series across a much wider spectrum of applications, so it is hard to judge how well a method works for NVE purposes, even though it is reported to work well in another context. This is something that NVE simply will have to test, in order to see how well various methods work. Note that performance should not just be based how good the end results are (how well a method catches anomalies), but also on their speed and possibly the robustness of the method (thus the cost in terms of maintenance).

Since the research field of anomaly detection in time series is very wide, I have spent some time making different classifications of the methods. This highlights some of the strengths and weaknesses that many methods inherit and can help in circling in towards a smaller set of methods that needs to be checked.

Introduction

This pilot project consists in examining the possibilities for automatically finding and correcting error in hydrological time series archives. NVE wanted to find out what similar institutions did in the way of automatic error detection in their time series. The field literature was also to be studied. If there was time for doing some initial tests, that should also be done. However, the literature was vast so there was little time for testing.

In the literature, the task of detecting error in data series is called “anomaly detection” (or alternatively “outlier detection”). Even when restricting to time series as data source, there is still much literature to be read. The restriction to time series is important, as it affects statistical and algorithmic methods as well as how the methods are tested. In statistical methods, the autocorrelation of time series needs to be accounted for. Testing is affected in the sense that dividing the dataset into training and test sets needs to be done in larger chunks rather than by random sampling on each data point. Issues such as dealing with missing data and changing time resolution is also specific to the field of time series.

The manual data control routines called “primary control» and «secondary control” here at NVE are certainly two such types of time series control an automatic system could either help with or entirely take over. Primary control is concerned with correcting measurement errors in order to achieve correct stage data. Secondary control is specifically tasked with adjusting data in order to obtain correct discharge data. Corrections due to backwater from ice as well as data reconstruction are the more prevalent type of secondary control here in Norway. Secondary control can require quite a bit of insight and will thus probably the most difficult to automatize. At NVE, these two controls are performed after the time series periods have been transferred from the real time archives to more permanent archives. At NVE, there is one archive for uncontrolled data (called HYTRAN), one archive for primary controlled data (called HYKVAL) and another for secondary controlled data (called HYDAG). We may however also consider doing controls on the real time archive, both at the time a sequence of time series data arrives and also at a later time when the sequences can be seen in context.

It may be that one type of anomaly detection and correction method works well for one type of control but is insufficient or too slow for another. For instance, when handling an incoming sequence of data when it is arriving, the method needs to be fast and robust. This suggests that advanced methods that require large computing resources may not be ideal for that part of the control pipeline. Thus, it may pay off to not commit to just one type of method, but rather find a set of methods, with each method tailored for each part of the control pipeline.

Anomaly detection is wide field of study with lots of literature. Even when restricting oneself to anomaly detection in time series, there is a wide and increasing body of literature. It was thus not possible for me to read every text on the subject, so some methods may have been missed. What I aimed for, was to at least check each type of anomaly detection method, so that I could be able to classify the methods and get a feeling for their strengths and weaknesses. Overview articles gave me a sense of the set of possibilities and pointed me to specific solutions within the different classes of anomaly detection. Thus, my focus was to sample each class of anomaly detection, rather than each specific method. The field is evolving constantly, so that a specific method may soon be outdated in its class of

solutions, sometimes by more sophisticated version from the same authors. However, the classes of methods seem not to expand that fast.

The applications of time series anomaly detection vary greatly, from aviation, electronics and nuclear power plants to meteorology and hydrology. This makes comparisons of the efficiency of methods rather hard. Each new article seems to state that the new method presented is the best in the field, but very often the method is tested on data that other methods were not developed for. Different types of anomalies may be common in different applications. Thus, I do not recommend taking each article's claim of efficiency for granted, but rather that several promising methods are tested. I could have restricted myself to only articles which applies their methods to anomaly detection in hydrology. However, that give a very narrow set of articles (Yu et al. 2014 is the only one referred to here) and it is not given that the few methods tested in the particular field are the best ones possible.

There may however be need for adapting the methods to our particular application, looking at frequently appearing known anomalies. We also often have a secondary set of stage measurements for a hydrological station, which can be an important source when doing anomaly detection. This is not so usual elsewhere, though I did find an article that was specifically concerned with duplicated measurements (Rey&Luck 1991). There is also a need to weight the ability to find known anomalies against the need to find novel anomalies. If one focuses entirely on known anomalies, it may pay of to just make very specific rules targeted towards these anomalies. This is not very robust, as new types of anomalies may appear and addition the nature of the known anomalies may change over time. However, if one focuses only on one general method, this method may turn out not to be able to catch some frequently appearing type of anomaly. Thus, a balance between detecting known and unknown types of anomalies needs to be found.

Then classifying the anomaly detection methods, I will explore several classification schemes, i.e. different ways of dividing the methods into groups. For each classification schemes, each group of methods may have their advantages and disadvantages, which I will try to point out. By looking at the groups deemed advantageous for a particular task in the control pipeline for multiple classification schemes, it may be possible to narrow the set of methods down to a single one or a small set of similar methods. The classification schemes also come with their own perspective, which may not be apparent before delving into this subject matter.

Before delving into these classification schemes, I will give a cursory overview of machine learning methodology. I will then describe what we at NVE have found out about similar efforts at automatic quality control of time series in similar institutions. Then I will go through the different classification schemes for anomaly detection methods. After that, I will describe some technical difficulties that I think will pop up for NVE's systems. The next chapter will be about specific methods or groups of methods. As I have been working on similar topics in a different research setting, I will spend a few moments on some particular set of methods from the field of statistical time series analysis. I will also describe the few tests I have so far been able to perform. Lastly, there is a short discussion on what properties of the methods may be best suited for different tasks in the control pipeline.

1 Technical context and considerations

1.1.1 The NVE timeseries and quality control pipeline

The current time series pipeline of NVE is shown in Fig. 1. Incoming data are (typically) stored in the real time archive (though some go directly to the HYTRAN archive). From there, they are after a while transferred to the more permanent archive called HYTRAN. In the current pipeline, there is no regular control performed on the data until after they have reached HYTRAN, thus the archive reflects the incoming data. After this, a primary control is manually performed. Primary control has as objective to removes or interpolate over measurement where the measurements are obviously flawed. The primary controlled data is stored in the HYKVAL archive. Manual secondary control is then performed in order to create stage values who's inferred discharge (through the stage-discharge rating curve) is correct. Corrections due to icy conditions are the most usual form of secondary control. The secondary controlled data is transferred to the HYDAG archive, which unfortunately has time resolution equal to one day, rather than the time resolution of the incoming data. In order to allow users to fetch some sort of secondary controlled data in the original time resolution, a virtual archive has been made that correct the HYKVAL data according to the daily mean differences in HYKVAL and HYDAG. However, in the future, secondary control will probably be performed with the original time resolution.

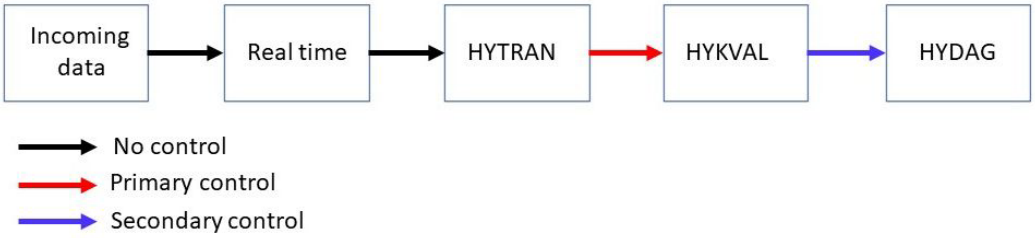


Figure 1: Current pipeline for NVE time series.

The major hope for automatic control systems, is that they should make it possible to perform controls on the first two transfers in this pipeline, alternatively once new data arrives (the first arrow) and during the stay at the real time archive. However, these methods could also be amenable to either helping the hydrological engineers in performing primary and secondary control, or (perhaps later) taking over these tasks. Secondary control is the one that requires most information and reflections, so it may be that this step is never fully automated. However, an automatic system should at least be of assistance in finding possible anomalies and in suggesting replacements.

The first component of such a new quality control pipeline, that of handling newly arrived data, should preferably be very quick. Further control of real time data during its stay or during the transfer to HYTRAN, should also be quick enough that at least one control can be performed per day. It is important that the original data is still stored along the corrected data. It is the corrected data that should be presented to extern users. However, several internal systems, such as flood warnings also rely on the real time archive. Ideally, if the kinks in a new automatic quality control are to be straightened out, then hydrological

engineers would look at the corrections and the internal systems should be able to run with better results on the corrected rather than the original data. If the automatic corrections only appear once manual primary control is performed, the reason for the changes may be obscured and the primary control may in the worst case be performed exclusively on the original data rather than on the corrected data. This will mean the automatic system for the first controls will never be corrected, the automatic control has performed on real time data has no impact on the control pipeline and the primary control may manually correct data that had already been automatically corrected before.

1.2 Programming considerations

NVE has systems running both on Linux and Windows, and there are multiple software platforms available on Windows now. Today's development in NVE is moving towards web application software built on Microsoft technology However, at the present time, NVE also has many systems working on Linux at the moment. My experience is that time series extraction and storing performs well and has a low threshold for introducing new functionality. Since the quality control job consists in fetching time series from the database, analyzing them, create changes and storing those back in the database, this is a job that might just as well go on a Linux system as a Windows system. As far as methods are concerned, most are implemented either in R or in Python, which are available on Windows and Linux alike. Personally, I only have Linux programming competence, so it may be that at least alpha versions of quality control systems can be implemented by me on Linux, and then perhaps transferred to Microsoft technology at a later stage. When it comes to the later stages of the quality control pipeline, where automatic systems in the start probably only will assist the hydrological engineer, it may be that previously stored suggestions for correction can be used. It may however also be that the programs that enables the users to perform this task should have the anomaly detection and interpolation methods implemented locally, or alternatively trigger the automated system to work on the data in question.

Anomaly detection methods usually use one or more components from the fields of statistics/machine learning. These components are often implemented in R or Python, which seem to be the preferred programming languages in academic circles. While the way these components are put together do not come as finished packages, the components themselves are typically implemented and thus do not need to be programmed from scratch. Instead, one can take these R and Python packages into use within a programming framework. This requires that the programs that performs the quality control can communicate with R and/or Python.

2 Machine learning

2.1 What is machine learning?

The field of machine learning cover all types of algorithms that are able to “learn” from the data, where learning means the algorithm has some internal state that is able to adapt to

incoming data before taking an action. This action can for instance be the making of a report, create a prediction, move data into another archive or a take a decision on whether to proceed or stop an industrial process. More specifically for the purpose of this project, the action may be to keep or remove a chunk of a time series, and if remove whether to keep the data sequence of the removed chunk missing or replace it with a prediction. When more data arrives, the algorithms should be able to adapt to it and thus update their way of working.

The phrase “machine learning” can lead the thoughts to artificial intelligence and the concept of deep learning (such as neural networks). However, a method that simply consists of labelling a data point as erroneous when the change from the previous to this data point exceeds a certain threshold, can be called machine learning as long as the threshold is adapted to the data. Bayesian spam filters fall into the class of machine learning, though they just consist of keeping track of the frequency of usage of each word in spam and non-spam emails and calculating a probability based on these frequencies.

The important feature of an anomaly detection method is not how advanced the method is, but how well it performs and how much manual supervision and machine resources it uses. How well it does the job, can be summarized by its false positive and false negative rate, though that requires supervision to calculate. Luckily, NVE has a vast archive of time series data that already has undergone manual primary and secondary control. Thus, performance can easily be tested for time series that has been running for some years. This also gives us the ability to do supervised learning (more on that later), without spending much extra manual effort on this. In addition to performance and the cost in human and machine resources, how simple it is to understand how a method works can also be of importance in circumstances where one has to manually check why the method took a particular decision. The methods for detection errors may themselves contain errors and may thus need debugging. When an anomaly detection method consists of several components (as is often the case), it may not matter if a component is difficult to comprehend, as long as it is robust. But fragile or error-prone components should definitely be easier to examine and study.

2.2 Machine learning and statistics

The field of statistics is concerned with how to model data and then estimate, predict and perform decisions such as rejecting or accepting hypotheses. These methodologies are algorithmic in nature, so there is a strong overlap between the field of statistics and the field of machine learning. Statistical methods such as linear regression, supervised/unsupervised clustering and hypothesis testing can all be described as machine learning methods will frequently appear in machine learning courses.

The overlap is however not perfect, as some topics in statistics are seldom described in machine learning courses (such as the manual process of developing statistical models). Also, several machine learning algorithms were developed without motivation in any statistical model. In such cases, statistical motivations for the method can be found later, but this is not always the case. Other machine learning methods can have components that are statistically motivated, but none the less be put together without a statistical model in mind.

2.3 Training, validation and testing

In order to compare methods and get a final verdict on the efficiency of the best method, datasets in machine learning are often divided into three, the training subset, the validation subset and the test subset. The training subset is used for adapting the machine learning algorithm. In a statistical context, this consists of estimating model parameters. The validation subset is used for fixing so-called hyper-parameters, i.e. variations in the methodology that is not directly adapted to the data but which has consequences for how well the method works. By checking how well the method works for different values of the hyper-parameters, these values can be adapted to the data. Examples of hyper-parameters are the penalty terms in lasso and ridge regression, number of layers and nodes in neural networks and threshold in simple difference-based anomaly detection. In addition, different methods can be compared using the validation subset, in order to reach a decision on which method to use. Lastly, the test subset is used for summarizing how well the best method found works.

The size of these three subsets need not be the same. The complexity of training tends to exceed that of validation and testing; thus, it may be better to let the training subset be larger than the test and validation subsets. No fixed strategy can be made, but 50% training, 25% validation and 25% test may perhaps be suitable, as there is a distinct chance that a particular type of anomaly do not appear in the training subset but rather in the validation or test subset. Thus, perhaps the ratio could be increased to for instance 70% for training and reduced to 15% validation and test. (Test and validation can often be of similar importance as so can often be kept at same size.) If the final test is deemed unnecessary, one could perhaps use 65% training and 35% validation, or perhaps even 80% training and 20% validation.

In a general context, what goes into the training, validation and test subset can be sampled randomly for each data point (with different probabilities so as to give different data sizes for the three subsets). However, autocorrelation and the fact that most time series anomaly detection methods expect data regularly spaced in time means that we will need to do the division into training, validation and test subset according to strict start-and endpoint restrictions for each subset. So, for instance, the training subset for a particular hydrological station could be the data from 2001-01-01 to 2010-12-31, the validation subset could go from 2011-01-01 to 2015-12-31 and the test subset could go from 2016-01-01 to 2020-12-31. Different hydrological stations may have different data size, so it may pay off to specify the ratios rather than the specific start- and endpoints of the time periods of the various subsets. If the method is later recalibrated at the same hydrological station, the size of the entire dataset will have increase, so the size of each subset should also be expanded in order to use the new data.

3 Experiences from other institutions

3.1 Canadian experiences

Our main contact abroad is the Canadian department ECCC (Environment and Climate Change Canada). They used a system called Aquarius developed by the software solutions company Aquatics Informatics for automatic error detection in hydrological time series. From what we learned from Douglas Stiff at ECCC, and also indicated in a video from 2013, this system did not use machine-learning. ECCC sent a manual which seemed to confirm that no machine learning was involved at the time we received the manual.

As with NVE, ECCC divide their quality control into two parts roughly equal to the primary and secondary control at NVE. For primary control, the Aquarius system contained many rules that could be set up. These rules consisted in thresholds which when exceeded would trigger an action. These thresholds could be for absolute values, single changes, change over aggregate values, and thresholds for aggregate changes over longer time periods. The latter type of rule was special in that minimal as well as maximal changes over time could trigger the rule. The minimal aggregate change over time is something I suspect can be problematic for prediction-based anomaly detection (more on that later) and is thus an important addition to the set of anomalies to keep a watch out for. The user of the Aquarius system, i.e. the hydrological engineer, would specify the rules to be used and what the actions were to be performed when the rules triggered an action. This could be simply to warn the user, but it could also be to automatically remove the data until further notification or some more advanced action. (Obviously, the original data must still be stored somewhere). Of the more advanced automatic actions was interpolation over the parts of data the system took away as well as gliding mean, maximum and minimum. As an example, a gliding maximum could be advantageous if the measurements cycles between realistic values and too low values. But this is a decision that the user determines when he/she receives the warning, rather than something that is automatically determined by the situation. An objective of the system is to make the user catch the error as fast as possible, give the relevant information sufficient to make a decision and let him/her make the decision about what the system is to do next and what needs to happen before the system stops correcting the data. The user could also make more advanced rules by combining several of the simpler ones. The Aquarius system seemed geared towards making tools for easily perform manual secondary control corrections, but as far as I could see, had no detection systems specifically made for this task.

3.2 Further thoughts on the Canadian experiences

In a setting with high user (hydrological engineer) interaction, a system with a wide range of actions available could work quite well. However, I think it will be difficult to find the correct action to take in all different circumstances, in particular because novel anomalies may be found. Thus, I think in a context with less user interaction and more focus on automatic quality control, the range of actions ought to be more limited. Some type of warning or logging should always be made. This system would also need a way for the user

to give feedback, if nothing else then a thumbs up or thumbs down to what the automatic quality control did. That way, the control system could update how it works.

However, if the further action consists simply of either removal or replacement based on a single interpolation method, it may already be hard enough to decide when to do which of these two alternatives. The decision may simply be based on hard rules (for instance "only interpolated maximally 6 hours"), but they could also be derived by treating the decision as a hyper-parameter that is estimated from the validation subset. One then needs some measure for deciding what is best between removal of a data point and having an interpolated data point that deviates by a certain amount from what it really should be. Thus, some kind of manually set rule is still needed, though perhaps this balancing threshold can be set globally (for all datasets in the institution rather that for single hydrological stations).

While the ECCC is highly interested in cooperation and an important collaborator for NVE, Aquatic Informatics is a private company and seem to want non-disclosure agreements before sharing anything more about their systems. As for the machine learning part of anomaly detection, everything we've learned points to there not being much to gain from the material shared by Aquatic Informatics.

3.3 Norwegian experiences

Norwegian institutions seem only to be in the starting position when it comes to automatic anomaly detection. MET seems to have come furthest. They seem to already have a simple rule-based anomaly detection up and running, though it seems to not have any machine learning components to it, so far. They are however exploring the possibilities of using machine learning. Whether they have come as far as testing is unclear, but they seem to have started this process before NVE did. NVE and MET have opened the door for more cooperation on this topic in the future.

My impression is that Statkraft is more behind in this process, but they seem interested in the subject and what we at NVE are doing with this. Some cooperation may be possible here. The advantage is that Statkraft have experience with the same kind of data that NVE have while MET gathers different data types (no stage/discharge which is a large portion of NVE's time series).

In conclusion, there is not enough past experience on the topic of automatic anomaly detection using machine learning, but the prospect of future cooperation between Norwegian institutions seems likely.

4 Classification schemes for anomaly detection methods

Since machine learning and anomaly detection in time series are broad and branching subjects, there are several ways of classifying the methods. I will go through the classifications I am able to discern and relate them to NVE's systems. Later, I will make

some tentative class recommendations based on the discussion in this chapter. Before describing the perhaps most important distinction in machine learning, that between supervised and unsupervised, I will describe the distinction between clustering and prediction-based anomaly detection. I found, the discussion of supervised and unsupervised learning benefited from having dealt with the clustering/prediction-based distinction first.

4.1 Clustering vs prediction-based anomaly detection.

The distinction here is between methods that compare the measurements gotten with what one expects (prediction-based) versus methods that simply label the measurements normal or anomalous based on their properties without having a clear definition of what is expected (clustering). The latter type of methods may however have clear definitions of specific ways the data could behave which is not to be expected from correct measurements.

4.1.1 Clustering

Clustering is a statistical concept that deals with how to best identify data with different properties and label each data point accordingly. (I could have used the word “classify” here, but since this chapter also deals classifying methods, I thought using that word for the act of clustering would only cause confusion. One could however also say that this chapter deals with the clustering of anomaly detection methods.) In our case the labels are either “normal measurement” or “anomaly”, though the “anomaly” could be sub-divided into several clusters, such as “spike”, “noisy data stretch”, “gliding measurement error”, “ice conditions”, “bad water communication” and “frozen cable”. However, one could also define anything outside these labels as an anomaly, thus catching novel anomalies. The conceptually simplest clustering algorithms may simply define the cluster of “normal behavior” and let anything outside that definition be an anomaly. However, defining the cluster of “normal behavior” may be a hard task, which blends into what I call prediction-based methods.

Examples of clustering algorithms are classic statistical clustering, rule-based anomaly detection, similarity measures, spectral analysis-based methods, neural networks (with labels as outcomes) and use of the “Ripper” method. These methods will be described later.

4.1.2 Prediction-based

When I use the word “prediction” in “prediction-based methods”, it is meant in the sense of giving estimates for what a sequence of measurements should be in their absence. Thus, it is not meant in the narrow sense of extrapolating forward in time but can also be used in the sense of interpolation within a gap in the time series. When used in the context of anomaly detection, the idea is to take the sequence of measurements to be tested out of the time series and try to predict what sequence values are to be expected in the gap created.

How the algorithm then compares the measured and predicted values, depend on whether the algorithm also estimates the uncertainty of the predicted values, as well as the sophistication of the method. If uncertainties are provided, one can compare the difference between the measured and predicted values against this uncertainty. If uncertainties are not provided, one could assume equal uncertainty (either estimated from quality controlled data

or treated as a hyper-parameter). However, this may be unrealistic, since higher stage values may be associated with larger errors due to more turbulent waters during floods. Perhaps it would be a more realistic assumption on log-transformed discharge values? In extrapolation circumstances, this assumption will also be unrealistic in treating data predicted two weeks forward equally to data predicted one hour forward. In interpolation circumstances, it will be unrealistic to treat data in the middle of the gap with the same uncertainty as those close in time to the start or end points of the gap. One may create and possibly also test different pragmatic solutions to these problems, by testing different ways to make assumptions about how the uncertainty behaves as a function of the different circumstances. However, with methods that realistically estimates uncertainty, one avoids this problem, thus giving such methods an advantage.

It may also be that to flag some residual (residual=measured minus predicted) divided by uncertainty as anomalous when the values exceed a certain threshold is not enough. Fluctuations in the measurements around the prediction may indicate another problem, such as the measurements being more noisy than usual, or the influence of waves or siphon effects. If one in addition looked at the variation in the residuals over the sequence predicted over, one might perhaps catch such anomalies. One might also distinguish between large single residuals and residuals that are somewhat large over the entire span of the sequence predicted over. Thus, predictions may be used for creating more than one measure of anomaly, though some of these measures might be applied directly to measurements (variance over the sequence) rather than to residuals.

Since the gaps can be of different length, this means that ideally all different ways of making a contiguous gap in the time series should be explored. This is however a hard computational task that increases with the square of the number of measurements, $O(n^2)$ where n is the number of measurements. There are various ways of dealing with this problem, which will be described later.

Examples of prediction-based methods are SARIMA (classic time series analysis tool), linear interpolation, spline interpolation, linear regression, hydrological modelling (requires support series), linear stochastic differential equations, hydrological model based non-linear stochastic differential equations, neural networks (with predicted measurements as output), Support Vector Machines (SVM), k nearest neighbor methods (knn) and random forests.

4.1.3 Interpolation for extrapolation-based prediction methods

Some prediction methods are predictions only in terms of extrapolation. For instance, the LSTM method in the *Tensorflow* R package only performs forward prediction. Such methods can still be co-opted into interpolations, by running them both forward and backward, and let interpolation be a weighted mean of the forward and backward prediction where the weight depends on the temporal distance to each of the end points of the gap. A linear dependency is the simplest. Let t_1 be the time of start of the gap, t_2 be the time of end of the gap and $x_F(t)$ and $x_B(t)$ be the forward and backward prediction at time t respectively. Then $x_{FB}(t) = \frac{(t_2-t)x_F(t)+(t-t_1)x_B(t)}{t_2-t_1}$. If the variance as a function of time of the forward and backward predictions, $v_F(t)$ and $v_B(t)$ respectively, are known and it is assumed that the forward and backward prediction errors are independent, then $var(x_{FB}(t)) = \frac{(t_2-t)^2 v_F(t)+(t-t_1)^2 v_B(t)}{(t_2-t_1)^2}$. Assuming that the forward and backward

prediction variance increase proportional at least to the square root of the temporal distance between the prediction time and the start/end point (respectively for forward and backward prediction), this will cause the prediction uncertainty "bubble up" and be largest in the middle, as I think is reasonable to assume, see Fig. 2a and 2b. A naïve handling of uncertainty, where the prediction uncertainty is assumed constant, will however yield an uncertainty which is lowest at the middle, see Fig. 2c.

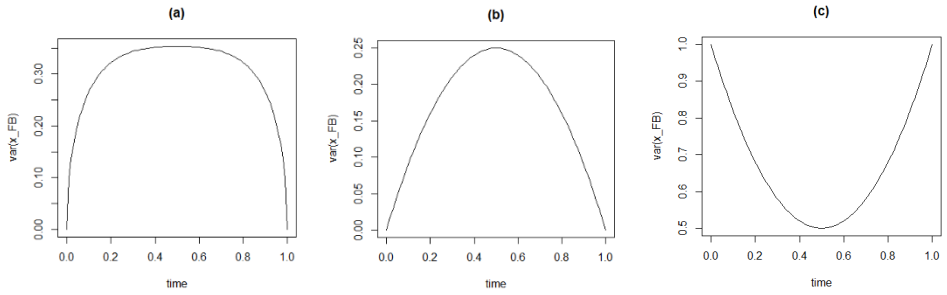


Figure 2: Prediction variance as a function of time. Here, time is scaled so that t=0 is the start of the gap, t=1 is the end of the gap. Figure 2a shows the interpolation variance when the forward and backward prediction variance respectively are $v_F(t) \propto \sqrt{t}$ **and** $v_B(t) \propto \sqrt{1-t}$**.** Figure 2b shows the interpolation variance when the forward and backward prediction variance respectively are $v_F(t) \propto t$ **and** $v_B(t) \propto 1-t$**.** Figure 2c shows the interpolation variance when the forward and backward prediction variance both are constant, so $v_F(t) = v_B(t) = c$**.**

4.1.4 Pattern recognition versus statistical time series modelling prediction

When doing predictions, one can divide between those that are based on statistical time series theory, and those that are treating the sequence of measurements that one is seeking a pattern for. The latter can be thought of as a sort of regression analysis where the previous sequence is the input and the measurements to be predicted are the output. The former gives a theory for the distribution of a new measurement, given the past measurements. Statistical time series models do have some degrees of freedom, both in the sense of their structure and in sense of the parameter values, which are adapted to the data. Thus, they are in a sense also pattern-seeking, but the pattern is not simply in the sense of the estimates, but also the variance and perhaps even the distributional family itself. Pure pattern recognition methods focus on the estimates, and generally do not give a theory for statistical aspects such as auto-correlation, variance or even distribution. Some pattern recognition methods, such as neural networks may however be expanded to give estimates for variance in addition to estimates of the measurements themselves. While there is some freedom in specifying the structure of the time series models, my expectation is that advanced pattern recognition methods have larger degrees of freedom and may potentially find patterns that time series models do not.

When the prediction method is based purely on pattern recognition, the predictions are contingent on the time resolution they were developed for. Note however, that many (but not all) types of statistical time series analyses have the same problem. How this is a problem will be discussed more in a subsection 5.4.

Another problem with some prediction methods, especially pattern-based such, is that the number of time steps they predict forward is fixed. If the gap length deviates from the prediction method's fixed number of time steps just slightly, one might perhaps find a pragmatic solution. Maybe a pragmatic solution is still possible when the gap length is significantly less than the prediction length. However, when the gap length is significantly larger, one needs to think of more robust solutions. An iterative approach to prediction can be employed, but this is difficult to perform while still yielding realistic uncertainty estimates. Running the prediction method for a whole range of different prediction lengths may be necessary in such circumstances. Note that this problem also applies to pattern recognition type of predictors, while time series analysis can be exempt from this problem.

4.1.5 Interpolation/extrapolation, separate task or part of a prediction-based method?

If an anomaly is detected, we may want the automatic quality control system to replace it with corrected values rather than just leave it empty under some circumstances. Thus, the algorithm must create predictions, whether or not it does so when checking for anomalies. Since the prediction job needs to be done anyway, it makes sense to also utilize the prediction algorithm for detecting anomalies. However, there may be a hierarchy of algorithms we may want to use, for different parts of the quality control pipeline. The part of the pipeline that has to do with recently arrived time series sequences, may perhaps only perform operations on small sequences of data, where simple interpolation methods are used. In that context, clustering-based anomaly detection may work as well as prediction-based anomaly detection.

4.1.6 Could and should cluster-based and prediction-based methods be combined?

It may also be good to have some parts of the quality control pipeline that are not based on predictions. Errors such as frozen wire and bad water communication means less variation than what would be expected. However, since any time series prediction model will encode for autocorrelation, i.e. that the future is not too far from the present state, the prediction will be similar to the past value examined. Thus, a system that compares predictions against measurements will find that these two matches quite well when the variation is too low. One could perhaps envision that one also checked for the variation in the residuals. However, since there may be some dynamics in the prediction, the task of comparing measurements to predictions can mask the lack of variation when compared to a method that simply summarized the variation in the measurements themselves.

As stated earlier, it may pay to use more than one anomaly measure on the residuals when determining if an anomaly occurs or not. It may also be that using anomaly measures on residuals can be combined with using anomaly measures on the measurements themselves. Some anomalies might be caught best one way, while other anomalies may better be caught another way. Thus, a hybrid between cluster-based and prediction-based anomaly detection could be optimal, at least for some parts of the quality control pipeline.

4.2 Supervised, unsupervised or semi-supervised learning

Perhaps the most important distinction in machine learning is between supervised and unsupervised learning. Supervised learning trains an algorithm by presenting it with examples of the input it may encounter and the correct output. In the case of clustering-based anomaly detection, this would simply be the label ("normal" or "anomaly"). For prediction-based anomaly detection, this would be already quality controlled data and the difference in residuals between the controlled and uncontrolled data. Unsupervised learning is simply the opposite of this, where the correct output is not provided.

Unsupervised learning requires less before taken into use, since one do not need to find the correct output for any of the input examples. Creating examples of the correct output could in many circumstances be very costly. It should however be noted that for NVE's case, most time series have been manually quality controlled, usually for many years. Thus, the cost of supervised training has already been paid in NVE's case. All things being equal, a method that utilizes more information, in this case, the correct output, should be better at making good decisions than a method where this information is missing. The outputs for prediction-based methods are the correct values, while for clustering-based methods the outputs are just the labels. Thus, if we want to maximize the utility of our manual quality controlled data, this suggests using prediction-based methods. However, there may be other factors involved which may favor cluster-based supervised methods. Also, since supervised learning is not likely to create algorithms better than the supervision it receives, such methods may miss undetected anomalies and novel anomalies. Thus, there may also be motivation for using unsupervised learning in some part of the quality control pipeline.

Semi-supervised learning is a hybrid between the two forms of learning describe above. Here, the method should be able to learn from previously quality controlled data, but then extend what it has learned there to be further able to learn unsupervised from data that has not undergone manual quality control. It is not clear to me how to achieve this in practice, though I think it must be very dependent on the anomaly detection method used. So far, none of the articles I've read on the topic fell into this category. If it is important to learn from new data fast, I'm thinking that some sort of manual check of the automatic system, with an evaluation of "ok" or "false", could give the system the supervision it needs. This may not be a hard task, and since the measurements need quality control anyway and the system should now do most of the work, then maybe such a solution would be satisfactory. If not, we will either have to try to achieve semi-supervised learning somehow or simply accept that there is a significant lag in the supervision and thus training of the methods used. There may be different answers to this problem for different parts of the quality control pipeline.

4.3 Single series vs multiple series control

It is relatively easy to envision anomaly detection on a single time series. The task is basically to compare the sequence of values one wish to control with quality controlled sequences in the same time series. If the examined sequence seems to have the same properties as the quality controlled sequences, the examined sequence can be labeled as "normal". If it has properties that falls outside the normal range, it is deemed anomalous.

From this description one may be misled into thinking this is a simple task, but more sophistication can be necessary in order to define what constitutes normal and anomalous properties of a sequence of time series measurements.

However, when looking only at a single series, one may miss clues about anomalies that may have been found if more information was added. For instance, for many of NVE's stage time series, a secondary stage time series is also measured. When these two stage series diverge, that is a cause for concern when it comes to the quality of either series. Manual control observations/measurements are also sometimes performed, and these can be compared with the time series values. There can also be other relevant data measured at the same station, such as air temperature, water temperature, ground water, water velocity and precipitation. Low ground water and high stage values might be a sign of a potential error, as could high stage values and low water velocity. High stage values without any preceding precipitation may also indicate an error in a small drainage area. Hydrological modelling may be a further source of comparison and can work in parallel as an alternative source of prediction-based anomaly detection. Hydrological models utilize surrounding meteorological stations to give an estimate of the precipitation and temperature in the drainage area. Such surrounding information can also be used directly in anomaly detection. For instance, in prediction-based anomaly detection, the prediction strength may increase when one uses correlations or causal connections between the series checked and similar series from nearby stations.

While multiple series can be a great source of information for prediction-based and perhaps also cluster-based anomaly detection, it can also be a source of confusion. When the set of measurements from various time series do not behave as one would expect, it may be that the time series one wants to examine is not the problem. Instead, there may be an anomaly in one or more of the support series. In a prediction-based anomaly detection, one might perhaps identify the series where the residuals are largest, but that may still not necessarily be the source of the anomaly. For instance, if the precipitation gradually becomes increasingly incorrect, this will cause later larger discrepancies between predicted and measured stage/discharge. Also, if a prediction-based anomaly detection method does not provide uncertainty, it will be hard to compare the residuals of a stage series with the residuals of for instance a temperature series. Also, if only two series are compared and they are gradually getting more inconsistent, it may be hard to idenfity which series is the erroneous one. This may be the case when only primary and secondary stage values are compared, see Fig. 3 and the illustration on the first page. Rey&Luck (1991) had a large discussion of how to deal with the combination of several parallel series, but for only two parallel series, there do not seem to be any way of solving this.

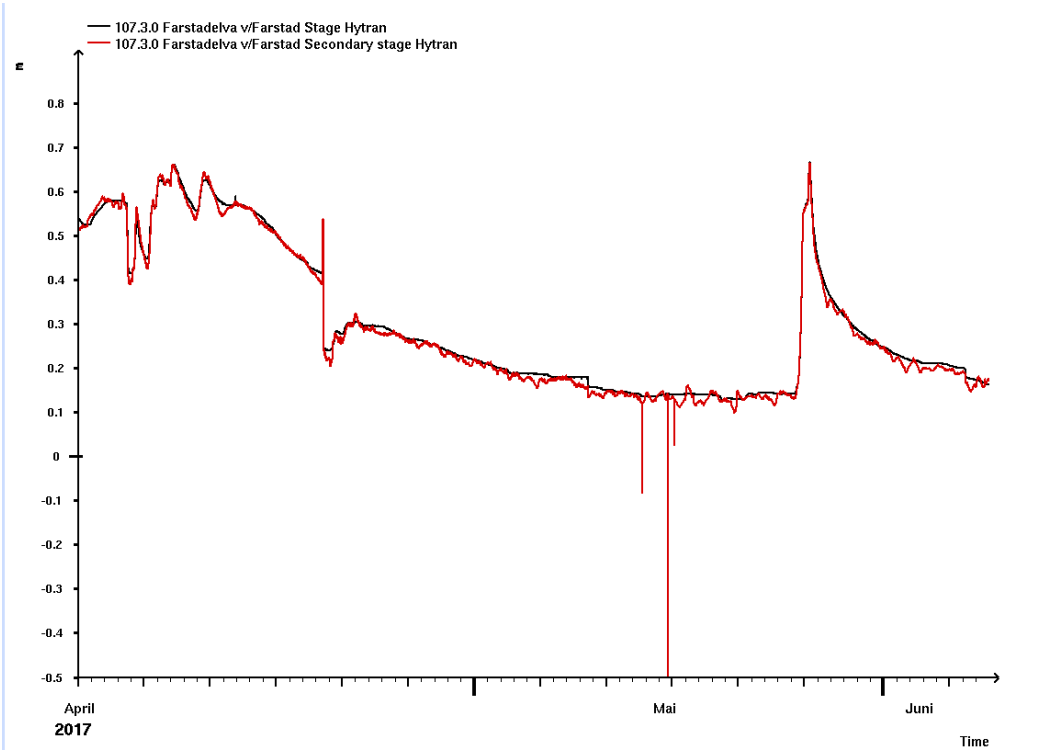


Figure 3: Comparison of primary and secondary stage measurements for Farstadelva (107.3.0) late spring 2017. The secondary stage time series has some serious spikes in May. However, there is also a small spike both for the primary and secondary stage measurements in April.

Perhaps it is best to think in terms of multiple anomaly detection methods in this regard? I am thinking in terms of a hierarchical set of tests.

1) Single series – Here each series is examined without reference to any other series.

2) Duplicate measurements – Here there are duplicate source for the same type of measurement. For instance, for stage, in addition to the primary stage series there may be secondary stage, control measurements and estimates from a hydrological model.

3) Support series from the same station. In additional to duplicate measurements, this may include other types of measurements performed at the same location, which may be correlated (or in a causal relationship) with each other and thus offer mutual informational support for each other.

4) Support series from other nearby stations. Specific discharge values may be similar in nearby stations, thus giving extra information about what the specific discharge values in a particular location ought to be. I think it will be difficult and resource-heavy to just let a machine algorithm examine the whole database in order to find the stations that give support for each other. Thus, I'm thinking that it is a manual job to find the best support stations. Note that the methodology has to be robust to various comparison series situations, see section 5.4.3.

If an error is found at one hierarchical level, one can then go one hierarchical level down in order to find the serie(s) that had the largest indication of having anomalies, even if the indication of anomaly was not sufficient to declare the serie(s) as anomalous on that lower

hierarchical level. In order for this to work though, the output from the anomaly detection method cannot simply be “normal” or “anomaly”, but rather use a gradual score for how anomalous a sequence is, which can then be thresholded into the “normal” or “anomalous” tag. Thus, even if the anomaly score of a sequence belonging to a small set of time series on a lower hierarchical level, it may still be deemed anomalous by a higher hierarchical level if no other series in the higher-level test had a higher anomaly score. This may thus work as a tiebreaker for the problem of deciding which series in a multiple set of series contains the anomaly. If a sequence in a single time series is to be identified though, and the anomaly is found on the highest hierarchical level, the algorithm would have to gradually work through the anomaly scores from the top hierarchical level to the bottom. Note that I haven’t found any literature that does this. So far that I’ve seen, the anomaly detection literature assumes that the set of series to be tested are pre-defined and that if multiple series are examined, the objective is simply to find the time sequence where the anomaly occurred and not necessarily to find which single series to correct. (Though it may perhaps be possible to identify the series through the residuals). The exception is again Rey&Luck (1991), though they seem to focus solely on series of duplicate measurements.

4.4 Fragile vs robust

Methods may depend on certain conditions being met, conditions that in real-world datasets are broken. The conditions can be on what other time series are available at the station or neighboring stations, or the conditions can be on the time series themselves. First, let us focus on conditions on the time series themselves.

4.4.1 Time resolution fragility

The condition I find most glaring when it comes comparing anomaly detection methods vs NVE time series is fixed time resolution. Most cluster-based and very many prediction-based methods perform pattern-recognition, rather than base themselves on statistical time series theory. When the time resolution changes, the patterns will necessarily change also. The typical differences between one measurement and the next will be altered, in that they will increase when the time resolution becomes larger and decrease when the time resolution becomes smaller, and this increase/decrease is not necessarily proportional to the change in time resolution. (For instance, in the Wiener process, the standard deviation in the difference between one measurement and the next, is proportional to the square root of the time resolution). Daily or yearly rhythms will change also, so that where the cyclicity of the measurements had one length before the time resolution change, it has another after the change.

Even when the time resolution is fixed, the problem of time resolution fragility may still arrive in the form of comparison series. A particular comparison series may have changes in time resolution. In other cases, a comparison series may have a fixed time resolution, but that resolution is different from the time series it is compared to.

Statistical time series models are not always exempt from this time resolution fragility. Standard models such as AR, ARIMA and SARIMA (more on these later), all assume a standardized dependency between measurements, independent of the time span between measurements. There are however other models, where that dependency scales with the time span between measurements. Continuous time models are in the category and are thus

able to deal with measurements that are even gatherer irregularly in time (such as instance control measurements and limnigraph measurements at NVE).

I do not recollect having found any method in the anomaly literature that dealt with variations in time resolution. However, it may be that some variant of knn methods (k nearest neighbors) can be adjusted to deal with fixed time intervals rather than fixed number of measurements, and thus be robust to time resolution changes.

If change of time resolution happened very seldom for all hydrological stations, there would be little need in taking changing time resolution into account. I performed an analysis where I examined NVE stage time series belonging to real time stations. When looking at time series from year 2000 to autumn 2021, about 21.8% had no time resolution changes, while 23.6% had only one change within that time period and 54.5% had multiple changes. The overall mean number of changes in this time interval was 3.84, while for only those with multiple changes, the average was 6.6. Thus, for those 54.5% of the stations with multiple changes, there were approximately 3.3 changes per decade. Thus, for any given series, the problem of changing time resolutions will show up much less than once per year. There were however 7 stations (out of 440) with more than one change each second year. One series (35.16.0.1000.1) had 559 changes in those (approximately) 22 years, in other words, 25 time resolution changes per year. Fig. 4 shows a histogram of number of time resolution changes, for stations with multiple such changes and where the extreme case was removed.

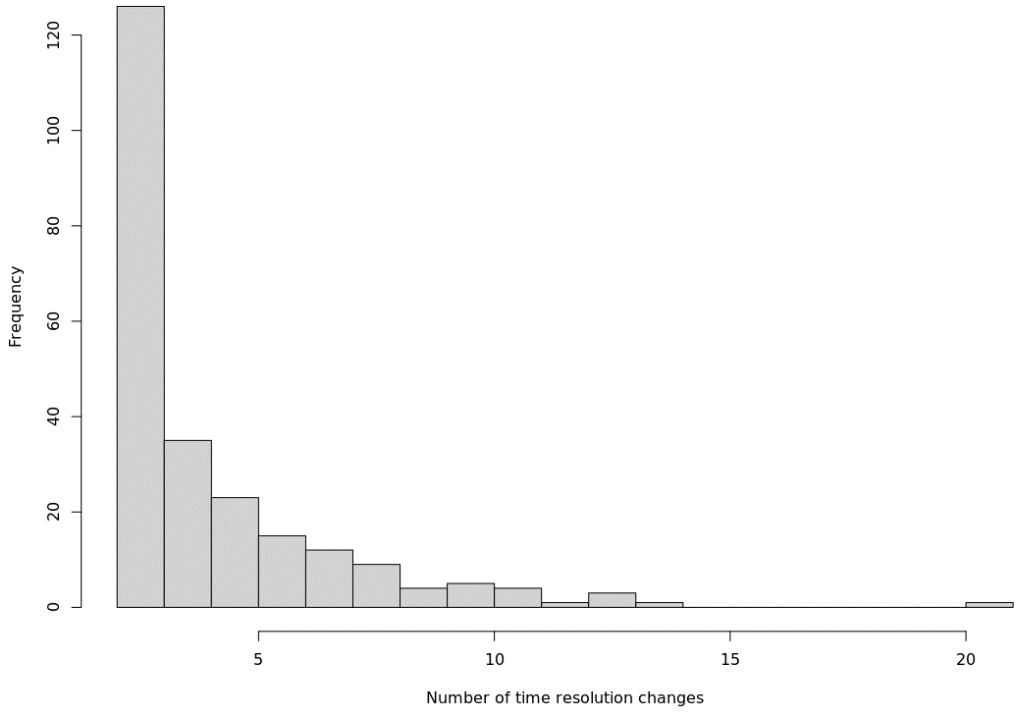


Figure 4: Histogram of number of time resolution changes in the period 2000-2021, for series where the number of changes were larger than 1 and less than 559.

When a method is fragile to time resolutions, there are several ways one can deal with this. One simple solution is to restrict oneself to series where no time resolution changes are

expected to occur. This may however limit the scope of the anomaly detection system severely, since the experience from NVE is that most time series will undergo at least one time resolution change in a 22 year period. One may perhaps surmise that less changes will be required in the future, but such a policy could be a hindrance for improving the measurements in a time series. One might find that a station is measured to infrequently to catch the culmination of a flood. If one is then presented with the choice of either continue to sample too infrequently or take the station out of the automatic anomaly detection system, then the anomaly detection system may be seen as a hindrance rather than a help.

Another way of dealing with changing time resolution in time resolution fragile methods is to perform aggregation into a fixed time resolution before doing the analysis. One must then make sure that the fixed time resolution is large enough that no later measurement sequences are sampled with larger time resolution. At the same time, aggregation removes information, so one wants the smallest aggregated time resolution which does not violate the previous requirement. Since expectation about the future is required, it is hard to make this setting of aggregation time resolution in an automatic way. At best, one can use previous time resolutions. One then will have to make a contingency plan for whenever the time resolution of a new sequence of measurements exceeds the set aggregation time.

It is also possible to perform linear interpolation between the data points belonging to time series periods with coarser time resolution. Thus, when a new, finer resolution is introduced for a time series, one could re-train the anomaly detection method using the complete time series, but then with a lot of interpolated data points for the older parts of the time series. However, if there are any interesting signals in the fine time resolution part of the series, those will be missed by the older parts of the time series and the signals found in the newer parts will be swamped by the older parts. Thus, this is also not a perfect solution.

The last way to deal with time resolution changes that I can think of, is to simply to start over again whenever the time resolution is changed. If the new time resolution matches that of some previous time periods, then the method could be trained on those periods. If such periods are not found, one will simply have to set some reasonable initial parameters for the method that will not render the method too inefficient, and then wait for the training and test data to arrive.

4.4.2 Gap fragility

Most anomaly detection methods are not able to handle gaps, i.e. sequences of missing data, in the records. For such gap fragile methods, one first needs to do interpolation over the gaps before using the method. With a prediction-based anomaly detection method, such a way of interpolating over gaps come with the method itself. However, note that the interpolation needed to fill all gaps will be more extensive than the gap filling done as a response to anomaly detection. Some gaps due to anomaly detection may be larger than what the quality control system is supposed to interpolate over, while other gaps may not be due to an anomaly at all and may for some reason also not be interpolated over by the quality control system. The gap complete interpolation done to circumvent the problem of gap fragility either has to be done on the fly, or the results of such an interpolation has to be stored on an archive different both than the pre- and post-control archives.

Continuous time statistical time series models are able to deal with gaps in that they can simply predict from the start to the end of the gap and predict the steps in between.

However, if the model is also a hidden Markov chain model, where one distinguishes between measurement and process, one need not do even this, as one then simply condition on no data at time points within the gap. Knn methods that deal with neighborhood in terms of time periods rather than the number of measurements, could also conceivably be robust to the gap problem.

4.4.3 Comparison series fragility

Since many of NVE's real time stage series also have secondary stage series, it may be tempting to build one's anomaly detection method on a comparison of primary and secondary stage measurements. However, not all hydrological stations have secondary stage series, so requiring secondary stage would limit the scope of the quality control system. In addition, the station may have secondary stage, but it could be missing for some time periods. In some cases, the source of an anomaly can affect both primary and secondary stage equally, so anomaly detection cannot rest solely on comparison of primary and secondary stage values in any case.

As stated earlier, it may pay off to have a whole hierarchy of tests that utilize different combinations of time series. Thus, with this type of thinking, a single time series test should always be present. However, the system should also be able to deal with the problem of missing comparison series for the higher parts of this hierarchy of tests. This hierarchy must thus be able to check if the conditions for a test are met and circumvent the test if these conditions are not met. For instance, a comparison between primary and secondary stage might be the second level in the hierarchy of tests, so the system must be able to skip test and go directly to higher order tests (involving more time series) whenever the secondary stage is not present.

4.4.4 Meta-information fragility

If the method relies on meta-information, the system may be fragile if the meta-information sometimes is lacking. I'm thinking in particular of the stage-discharge rating curve, which is a type of meta-information available to most stage time series, but not all. Thus, a system that requires specific discharge (which requires a stage-discharge rating curve when stage is measured), will be fragile to lack of stage-discharge rating curves, which will then need special handling. There may also be different types of meta-information that could be useful but may sometimes be lacking.

4.4.5 New measurement type fragility

While parameters are set locally, model structure and the type of series included for comparison series are reasonable to adapt to different measurement types. However, when new types of measurements arrive, the system may be thrown into confusion. A model structure that works well for one type of measurements may not work for another. The type of series that one compares a given measurement type to may also work poorly for a new measurement type. Precipitation may be relevant to stage and vice versa, but is solar radiation relevant for either measurement type? Regional models (more on that later), may be especially vulnerable to the arrival of new measurement types.

4.5 Black box, white box, grey box?

Even when an anomaly detection method functions, it may be hard to understand why it yields a certain result in a certain circumstance. The transparency of anomaly detection methods varies a lot. How well one feels one understands a method and how it related to the task at hand, relates partially on purely subjective factors like how well one is acquainted by the method and the mathematical tool it uses. It may however also depend on more impersonal factors, like the complexity of the method such as how many different steps it involves. How well the estimated inner parameters of the method relate to the subject matter at hand, will also affect the transparency of the method.

A non-transparent method is often called a “black box”, while a transparent method is called a “white box” and methods somewhere in between a “grey box”. Technically, there will be few methods that are completely black or white, but some are found in the lighter and some in the darker regions.

When a method is taken into use, it is important that those doing so has at least a technical understanding of the method. That does however not mean that they will understand why the method yields a certain result at a certain circumstance. If those responsible for integrating an anomaly detection method into their system (typically programmers and statisticians) are not the same that are operating the system on a daily basis (hydrological engineers for example), the latter group may be at an even larger disadvantage. However, it may be debated how important it is to understand why an anomaly detection method yielded a certain result in a given circumstance. In situations where the anomaly detection is clearly faulty, one might be interested in debugging (from a programmer perspective) or manual adjustment of the method (from a hydrological engineer perspective) in or to avoid the same problem in the future. However, if the method consists of several components, such as prediction and comparison between predictions and measurements, transparency might be of more interest to some components than others. For instance, it may be that the prediction method could be black box while the comparison component may be close to a white box. When such an anomaly detection method erroneously labels a sequence as an anomaly, one may find that the mysterious prediction seems reasonable while the transparent comparison component is too sensitive. If the prediction seems unreasonable, and this happens regularly, it may be that in time this black box is replaced with another black box that performs better. Still, one may be fumbling in the blind when trying to find another black box method that works better, while if the prediction method was transparent, it may be easier to adjust it.

As mentioned in the previous sub-section, time resolutions may change. If the method is sufficiently transparent, one may perhaps find a way of re-setting the parameters in the method so that the method works for the new time resolution. With black box models, this is not likely to be possible.

Another problem with black boxes is that since they tend to deal with a large set of parameters that are difficult to interpret, then transferring knowledge from the set of stations so far examined to a new station may not be feasible. Thus, making a method that combines regional and local data may be very hard. (More on this topic later.) The consequence of this is it may be very hard to perform efficient automatic anomaly detection on new time series.

While transparency can be good when one wants to improve the method, performance is the key measure of an anomaly detection method. A transparent method that performs poorly, may perhaps easily be improved, but if these improvements never seem to reach that of a specific black box method, the black box method should probably be preferred. One should however note that performance is not just how well the method works on large unproblematic time series, but also how well it performs on small or new time series and how many problems arise due to the fragility of the method.

4.6 Local, regional or local + regional

The methods described in the anomaly detection literature are trained on a single time series. Thus, when a new hydrological station is started, there will not be anything to train the method on.

It may perhaps be possible to take the parameters of the anomaly detection method adapted to a series considered similar and apply those to the new station. When the station has gathered enough data to train the anomaly detection method on local data, those parameters can then be replaced with those that are locally adapted. However, what is considered a similar station can require manual consideration. In addition, stage values are either in heights above sea-level or height above a certain fixed local point and can thus vary much from station to station even for stations considered similar. If the test was performed on discharge rather than stage, the method might be more transferrable from one place to the next. Specific discharge may be even more comparable between stations. However, not all stage stations have a stage-discharge rating curve, which is necessary for generating (specific) discharge from stage. Most NVE stage stations do however, so the problem of missing rating curves may perhaps be ignorable.

It may perhaps be possible to make a regional model for anomaly detection, by searching for patterns in the adapted local parameters of the anomaly detection method. These patterns may come in the form of a field variables such as drainage area, effective lake percentage, average precipitation or average mean spring temperature. For instance, for a rule-based anomaly detection method, the thresholds used may depend on such field variables in a predictable way. Time resolution may also be tested, in order to find how the locally adapted methods depend on time resolution. Thus, when a new series is started or a new time resolution is used, the method can be applied using regional parameters set according to the field variable values. So, as well as solving the problem of new series, this type of analysis can alleviate the problem of time resolution fragility. It would however require that the method structure do not vary from station to station. Thus, if a prediction-based system uses regression to interpolate and extrapolate, the form of the regression should be the same for all stations. If not, the parameter values will not be transferrable.

The learning itself can be regional, which will be called a true regional model, as opposed to local first and then adapted to form a regional model. The disadvantage of that, is that it requires more computer resources. The advantage is that each time series lend strength to the others when learning. An anomaly that is only found in a small set of time series now, can none the less be found in other time series later. With only local learning, these anomalies will go undetected in time series the first times, until manual control detects this and gives supervision. Anomalies are per definition rare events, so lending strength across

various time series can be the difference between efficient and inefficient learning. However, it may be that local learning and then regional modelling afterwards can gradually learn regional patterns, if the regional model then afterwards updates each local anomaly detection method. Such an interactive approach may perhaps be optimal, since regional learning may be too computer resource hungry to be efficient.

I think a regional model would best be built on specific discharge, when it comes to checking stage time series, as specific discharge has a more transferrable nature than discharge, which again is more transferrable than stage. Note however the statements from the subsection on fragility and robustness; regional models may be vulnerable to the arrival of new measurement types and since they may rely on (specific) discharge, they may also be vulnerable to stage-discharge rating curve problems.

One can simply replace the regional parameters with locally adapted parameters when sufficient time has passed to train the anomaly detection method locally. However, experience from regional flood analysis shows that it is possible to combine regional and local information in such a way as to get more reliable results than when only one of these sources of information is used. As far as I can tell, this combination of regional and local information requires some sort of Bayesian analysis, so that the uncertainty of regional and local information can be weighted. Here, the regional information will form the prior information, while incorporating local information forms the posterior results. This local-regional analysis may however be a quite ambitious thing to do, which as far as I can tell, has not been done before in an anomaly detection context.

It may also be possible to use the regional model always, as it is ultimately made from the collection of local information. However, from experience with regional flood frequency analysis, local time series can deviate quite a bit from what the regional model expects. Thus, for a long time series which thus can be trained well by a local method, a purely regional model will be suboptimal. It will also be suboptimal compared to a local+regional model, if that is possible to make, since that incorporates local information.

Regional anomaly detection methods are ambitious to make but do expand the scope, by making possibly efficient quality control even for new hydrological stations.

5 A list of anomaly detection methods

5.1 Cluster-based methods

5.1.1 Rule-based methods

Rule-based anomaly detection methods are simply methods that classify a time series sequence as an anomaly rather than normal based on a few criteria that are easy to formula and check. Usually, these rules come in the form of some value exceeding (or in some instances go below) a given threshold. Threshold rules can be combined to form more complicated rules, as was seen in the Aquarius system of Aquatics Informatics. Another set of rule-based triggers can be found in the MET report of Vejen et al. (2002).

While I have not found any examples of machine learning combined with threshold rules, rule-based methods should still be amenable to machine learning, in particular supervised learning. For each time series, the thresholds can be automatically adjusted to find the right balance between sensitivity (true negative rate) and specificity (true positive rate), using previously quality controlled periods in the same time series. Making such a machine learning algorithm may be tricky though, since multiple rules are typically used for catching different types of anomalies, and the machine will not know in advance that an anomaly should be found with a specific rule. I do expect machine calibration to be feasible for rule-based systems, though it may be computer resource intensive.

The advantages of rule-based methods are that they are both easy to understand (white box) and not expensive in terms of computer resources. The former means they are easy to check and debug, while the latter also means that they are quick. This is a major advantage for the early stages of the quality control pipeline, in particular when dealing with recently arrived real time data. It may also be that it is possible to make regional rule-based systems, where the individual thresholds are derived from field variables.

Rule-based systems may however not be sophisticated enough to catch subtle anomalies. For instance, it would be very hard to create a simple rule to find a subtle non-consistence between temperature, stage values (possibly transformed into discharge) and precipitation that may be needed in order to catch the need for ice-correction. It may be possible to make rules that compares the values in primary and secondary stage measurements, but higher order parts of multi-series comparisons will at the very least require regression models in order to be efficient. Thus, rule-based systems may be used for rooting out some standard run-of-the-mill anomalies at the start of the quality control pipeline, but I do not recommend using them for more finely tuned quality control such as secondary control and I do not think they are amenable to multi series tests beyond comparing two time series that measure the same thing.

When gaps are found or time resolution is changed, the way one measurement relates to the next will also change. Thus, I do expect rule-based methods to be fragile to time resolution changes and gaps, though some of the rules that have to do with aggregated values will not change.

5.1.2 Summary statistics-based methods

When summary statistics is made in order to create a threshold rule for anomaly detection, I call that a summary statistics-based method. Examples could be threshold on a single value compared to extreme percentiles or mean±k*standard deviation, or similar thresholds for the difference between one value and the next. For larger sequences, it could be a rule for the standard deviation in the sequence compared to the standard deviation of a quality controlled sequence, for instance. Several methods described in Vejen et al. (2002) are summary statistics-based methods.

This may be seen as a sub-genre of rule-based systems. Just as other rule-based systems they are in need of calibration. However, there is also an element of automatic updating with them, since the summary statistics themselves can be re-calculated when new quality controlled data arrives. Thus, summary statistics-based methods are not only learning during calibration but can also learn during normal use. Note however that if the summary statistics is changed much, then a re-calibration is probably necessary.

Another possible advantage to summary statistics-based methods compared to rule-based methods that do not use summary statistics, is that it may be easier to transfer knowledge from one hydrological station to the next. For instance, optimal threshold for measurement differences compared to their standard deviation may be essentially the same for two different stations, even though the standard deviations themselves are different. This may give a direct recipe for treating new hydrological stations. It could also make regional models possible.

Summary statistics could also be made for use in larger multi-series tests than other rule-based method. Linear regression models are based on summary statistics and can be used for comparing values between series. Mahalanobis distance is an alternative way of looking at multi-series values (closely related to Principal Component Analysis, PCA) that can summarize how far from the normal state the system is at a given time. A rule based on Mahalanobis distance could thus catch inconsistencies between time series for a single time point. It could also be used for catching inconsistencies in small sequences in a single series. Summary statistics could thus provide a more powerful version of rule-based methods. However, they would need more statistical intuition to make, and advanced versions could be slightly less transparent than simpler rule-based systems. At the end of the day, they do rely (perhaps indirectly) on statistical theory.

Without a theory behind the change from one measurement to the next, I do expect summary statistics-based methods to be fragile to time resolution changes and gaps.

5.1.3 Decision trees

A decision tree is a set of rules which starts with one branching rule which when applied lead to new rules in a tree graph system. I found a mention of this type of methodology in Chandola et al. (2009), which mentions a specific way of building decision trees, called Ripper (Cohen 1995). As such, such methods constitute an automatic way of generating complex rules, which can perhaps be more sensitive and specific than just using a small set of pre-defined rules. While the way the algorithm works can be complex, the resulting decision tree should be understandable. Thus, the way the classification is performed is a "white box" even if the algorithm can possibly be described as a "grey box" or even a "black box". As the rules of a decision tree can be complicated, the structure will vary from time series to time series, so I do not expect that a regional model can be made from this type of methodology. I also expect the method to be fragile to time resolution changes and gaps. Multi series treatment should however be possible.

5.1.4 Statistical clustering methods

Statistical cluster analysis seeks to predict the class a data points that can be a multi-dimensional value (a vector) belong to. Unsupervised clustering can be tricky, but supervised clustering can simply consist of gathering some summary statistics for data vectors pulled from each pre-defined cluster. In anomaly detection, the cluster could simply be "normal" and "anomalous", but one could also have various anomaly classes. It could also be that one simply describes what belong and does not belong to the "normal" cluster. In time series anomaly detection, the vector could be a sequence of consecutive measures in a single time series, a set of values from different time series or a combination (sequences from multiple time series).

Perhaps the clustering methods that are most closely motivated by statistical theory are the distribution-based method, such as linear discriminants (based on the assumption of multi-normality and equal covariance structure in the different clusters), quadratic classification (which drops the assumption of equal covariance structure) and logistic regression (which assumes a functional form for how the values translates into classes).

Naïve bayes is a classification method that uses a distribution for each part of the vector (each single measurement) given the cluster it belongs to and then combines the knowledge in the various measurements to give a probability for a point belonging to a cluster using Bayes formula and the assumption that all the measurements are independent. The independence assumption is not very reasonable for time series, which is perhaps why I have not seen it used in the time series anomaly detection literature. It should however be noted that classification based on Bayes formula and the actual distribution of data points belonging to each cluster is the optimal classifier according to classification theory. However, having a distributional theory for time series is a harder task than for independent measurements and will be described later in the prediction-based part of this section.

Density-based clustering methods are similar to distribution-based methods except the attempt to estimate how dense data points belonging to a cluster is packed in a region of the vector-space without committing to a distributional family. As such, they are non-parametric methods. One particularly popular method is the DBSCAN method (mentioned both in Phung et al. 2018 and Chandola et al. 2009).

K-means clustering (centroid-based clustering where the number of clusters are given), classifies a vector according to which cluster centroid is nearest to it. As such, it does not rely on a given distribution, but does rely on the assumption that the various elements in the vector are comparable in value. As such, this may not be so easy to make efficient in a multi-series context.

Grid-based clustering classifies vectors according to which grid cell they belong to. This will be difficult for high-dimensional vectors (large sequences and/or many series), but could work for smaller sequences. Fig. 5 shows a so-called “Markov plot” in the FINUT program at NVE for pairs of daily discharge values from the station Gryta (6.10.0.1001.1) and shows gridded frequencies of measurement combinations for quality controlled data. This could conceivable be used for rotting out unrealistic changes from one day to another.

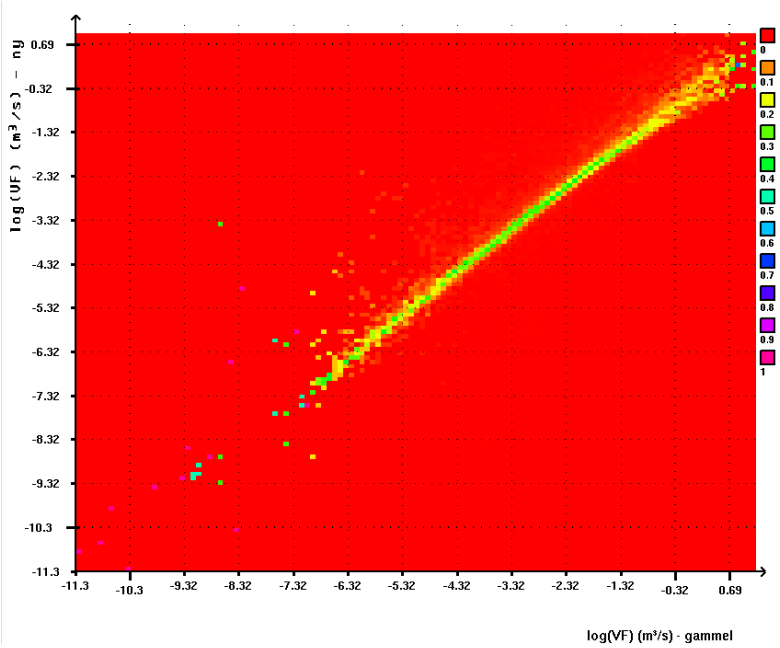


Figure 5: Gridded frequencies of daily discharge pairs at Gryta (6.10.0.1001.1) going from 1980 to 2020.

While clustering methods are mentioned and frequently used in the time series anomaly detection literature, my (perhaps arbitrarily collected) reading list did not include any methods that relied solely on statistical clustering methods (with the exception of a chapter in the master's degree of Tinawi 2021). However, there were examples where clustering was combined with pre-processing and/or post-processing, for instance Sodja (2021) and Laptev et al (2015).

As can be seen, statistical clustering method are varied. It is thus difficult to give a single judgement about transparency, time resolution or gap robustness, multi-series possibilities or possibilities for making regional models.

5.1.5 Nearest neighbor methods

Nearest neighbor methods utilizes a sliding window of values around the data point or sequence to be examined, rather than the whole time series. The advantage of that is the method is adapted to the current season and climatic and possibly meteorological situation rather than to the whole time series, which can consist of multiple seasons and situations. The disadvantage is of course that a lot of information is thrown away. I have found no use of neighbor clustering methods by themselves, but that line of thinking seems to be applied to more complex methods with multiple components, see for instance Qiu et al. (2012).

5.1.6 Similarity score methods

Similarity scores are ways of comparing a time series sequence to other sequences in the time series. Often the methods restrict itself to a temporal region around the sequence examined, so that it has an element of nearest neighbor methodology in it. No single such score is used instead multiple scores are combined using Pareto analysis. In the literature I found, Hsiao et al. (2016) and Wang et al. (2020), similarity scores and Pareto analysis formed a large part of the method, but there were also other components. The overall impression was that these methods are quite complicated and requires insight and substantial programming resources to implement. No source code material was mentioned

in the cited articles, though perhaps some do exist. The methods I found were also unsupervised, which is not optimal for NVE's case, since quality controlled data is available for most stations. The methods seem quite sophisticated though, so it may perhaps be that they perform well even without supervision.

The methodology seemed highly non-transparent due to multiple steps and (for me) unfamiliar theory. Whether these methods can handle multiple series is unknown to me. I would certainly think that no regional model for this method could be made that could handle new time series. It is also hard to see how robust such methods are to time resolution changes and gaps, though by the pattern-seeking framework of the method, I would expect it to be fragile to such things.

5.1.7 Neural network methods

Neural networks are complex regression systems based on an analogy to biological neurons. Simulated neurons are typically stacked in layers in order to create an output that can be a highly non-linear function of the input. As far as the anomaly detection literature I found, there was no examples of using neural networks explicitly only for clustering. Those that I found were prediction-based anomaly detection methods (Song et al. 2020, Tinawi 2021, Shipmond et al. 2017, Buda et al. 2018). However, neural networks are also capable of giving classifications, thus potentially they could be used as a set of clustering method. However, once an anomaly is found, one might want to replace the values, and since neural networks are also capable of giving predictions, it is perhaps best to use them in this capacity. Still, it could be that a clustering method based on neural networks could catch situations where the predictions and measurements never stray far apart but where the behaviour of the measurements suggests an anomaly none the less.

5.1.8 Support Vector Machines

Support Vector Machines (SVM) is a method of non-probabilistic clustering (classification) that can examine datasets with large dimensionality (such as large time series sequences). Dimension reduction is achieved through penalty terms and regularization. Some of the similarity scores in Hsiao et al. (2016) utilized SVM. They were also mentioned in Chandola et al. (2009) and Yu et al. (2014). The methodology is not very transparent, and I would expect robustness problems. However, the method can be linear, which could perhaps make regional models possible.

5.2 Pattern recognition type prediction-based methods

As mentioned earlier, prediction-based anomaly detection methods are methods that predict what the measurements "ought to have been" in a small time series period (a sequence), and then compares the prediction with the actual measurement values. I distinguish here between pattern recognition and time series types of prediction-based methods, as the former is based on predicting new values simply on the patterns found in earlier sequences but without a statistical understanding of the process that created the time series, while the latter also includes a statistical understanding of the process. It should be noted that some pattern recognition methods are amenable to some type of estimation of uncertainty and can thus give some statistical grounding for the anomaly detection. In other cases, one may

perhaps separately construct a model of the standard deviation of the residuals (the difference between predictions and measurements) as a function of the placement in the sequence. If this is done, then one can look at the standardized residuals, which can be given the same treatment for all residuals. If not, one might perhaps use a uniform threshold for the residuals, however I do expect that to be a sub-optimal solution. In many methods, quite a bit of time is spent examining the properties of the residuals.

However, pattern recognition types of prediction-based methods will not yield the probability for a sequence of data points. This also means that they are not amenable to using Bayes optimal clustering, though that may not necessarily be used for time series types of prediction-based methods either.

5.2.1 Linear interpolation

A linear interpolation is simply a line drawn from the start to the end of a gap in the dataset. (Keep in mind that when checking a sequence in a prediction-based method, one pretends the sequence is missing and predicts what the values should have been). As such, the residuals are simply the difference between the measurements and that straight line.

One major problem with this method is that prediction also means extrapolation. With nothing on the other end, the only thing to do is either to extend a flat line from the last measurement (i.e. the prediction is the last measurement) or extend a sloped line from the last measurement which matches the previous estimated derivative.

Since this is a method without any reference to a statistical model, there are no uncertainties attached. One could perhaps construct a model for the standard deviation of the residuals from previous residuals in comparisons between quality controlled and non-controlled data. This needs to be a function of the placement in the gap, however, which can get quite complicated. However, if efforts are spent on the residuals, it might be that some extra effort should be applied also to the interpolation method itself.

Note that the Wiener process (“random walk”) in continuous time series modelling creates interpolations that are linear and extrapolations that are flat lines. Thus, this can create a statistical underpinning for the interpolation method and will provide standard deviations for the residuals as a function of their placement in the gap. That would however move it into the class of time series types of prediction-based methods, and I do not expect that the Wiener process is the optimal model for any kind of hydrological processes.

This method is of course very transparent and robust, but not amenable to multi series treatment.

5.2.2 Other algorithmic interpolation methods

Polynomic regression and splines can make for smoother more natural looking interpolation than linear regression. Other than that, these methods do however come with the same weaknesses, no statistical theory behind and thus no uncertainty given. They may however be slightly better for extrapolation.

5.2.3 Nearest neighbor methods

In a prediction setting, nearest neighbor methods can be simple. For instance, one can simply let the predicted value be the mean or median of the k nearest neighbors. This will behave much like the linear interpolation method, though there will be some differences.

There is also a regression variant of k nearest neighbors, where a regression line is fitted the nearest neighbors. A small test showed that this did not work too well, see Fig. 6. Knn regression do not necessarily fit the end points of the gap it interpolates over. A linear rescaling should take care of that, but even so, the form of the regression is not as one would expect from discharge values.

If the number of measurements used is not predefined but rather defined by a time interval around the measurement to be predicted, then such a method can be robust to gaps and time resolution changes. They are also quite transparent. I do not think they are amenable to multiple series treatment, however.

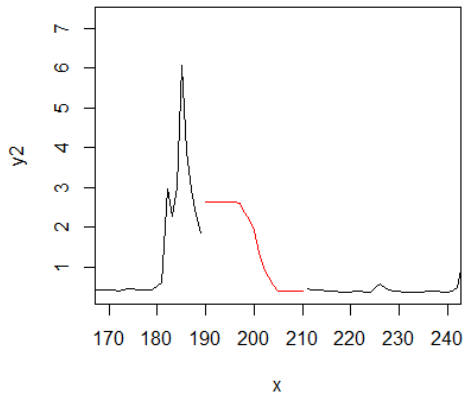


Figure 6: Knn regression test on a gap in the discharge values for the Farstad station (107.3.0.1001.1) for the middle of the summer 2021.

5.2.4 Neural networks

Neural networks are simulated "neurons" that work as non-linear regression formulas for the input they receive. These can be organized into layers, where the original data is fed into the first layer, the first layer then sends its output as input to the second layer and so forth until the final layer sends its output as the prediction. Song et al. (2020), Tinawi (2021), Shipmond et al. (2017), Buda et al. (2018) and Vishwakarma et al. (2021) all use prediction-based (on the time series itself rather than its anomaly status) neural networks for anomaly detection. With lots of degrees of freedom, neural networks can pick up many subtle patterns, but there is a danger of over-learning, meaning that it learns patterns that are due simply to stochasticity and will not be repeated anytime soon.

The usual feed-forward neural networks have what is known as short-term memory, which means they quickly forget previous states. For time series, a so-called long short-term memory version (which has feedback loops in the neural network), LSTM, may be preferrable. As such, most time series anomaly detection methods that use neural networks,s eem to use LSTMs. However, the experience of Shipmon et al (2017) was that so-called recurrent neural networks (RNN) worked better for the examples they looked at, so LSTMs are not universally the best option for time series. Sun et al. (2021) compared several prediction-based anomaly detection methods, where LSTM were as good as other neural networks, but not significantly better. A time series based systems called EGADS (Laptev et al. 2015), came out as better than any of the neural networks in that study, however. Vishwakarma et al. (2021) used only feed-forward neural networks. Still, from

the frequent mention LSTMs get in the literature, I would guess that they are the default type of neural network used for time series prediction.

Neural networks require one to specify the number of neurons and layers, and thus need a validation phase in order to set these hyper-parameters.

As mentioned before, neural networks typically only give predictions, not standard deviations. They can however be expanded to give some sort of estimate of uncertainty. I did not find out how easily uncertainties can be estimated nor how realistic the uncertainty estimates will be. (Are the uncertainties uniform or do the uncertainty bubble up in the middle of a gap as they should?)

Note that neural networks are highly non-linear systems. A set of average parameter values from nearby stations or a regression model for the neural network parameters based on field variables, can thus be expected to perform poorly on new datasets. Thus, regional models will probably not be possible. Also note that as far as I know, neural networks require fixed time resolution. They are however amenable to multi series treatment, making it possible to utilize comparison series.

Neural networks are often treated as the ultimate example of "black boxes". There are ways of getting into the internal states of neural networks, potentially rendering them "grey" instead. This does however require extra analysis and may not be very easy to accomplish.

5.2.5 Random forests

The random forest type of regression is a weighted sampling of decision trees, used for prediction. As they are a combination of many decision trees, they can get quite complex and thus have many degrees of freedom, just as neural networks. These means these methods are quite good at picking up patters, but also that over-learning is a distinct possibility. Random forests do however weight the degrees of freedom automatically, unlike neural networks, thus no hyper-parameters need to be set. I do not know if they are able to give prediction uncertainty or not.

As with neural networks, random forests are black boxes (maybe even to a larger degree) and are fragile to time resolution changes. As neural networks, they are amenable to multi series treatment. I also do not think they are amenable to regional modelling.

5.2.6 Support Vector Regression

This is a method similar to Support Vector Machines (SVM), but which yields predictions rather than just clustering. Support Vector Regression allows for large regressions to be performed with penalty terms for regularization and contains a method for performing this regularization. Ma&Perkins (2003) utilized SVR for anomaly detection in time series.

5.2.7 Bayesian networks

Bayesian networks are hierarchical models, where each component is estimated by Bayesian inference. As such, they can also encompass hidden Markov-chain time series models, where some parameters influence the state of the time series process which again influences the time series measurements. In the literature I collected, there were no examples of the use of Bayesian networks outside of time series models. Thus this is not a method per se, but a handle for a wide variety of methods that often gets mentioned in the anomaly detection literature.

5.3 Statistical time series prediction-based methods

Statistical time series models are models that attempts to give a probabilistic description of the set of measurements and possibly also the underlying process. The focus is not on prediction (and possibly uncertainty as a separate task), but on describing how each measurement depends on the previous ones in a distributional form. That is, a time series model gives the probability for a measurement given the previous ones, which is called autocorrelation. Since the statistical dependency is described, predictions and uncertainties are available through such models. Anomaly detection methods may utilize the entire statistical modelling framework of time series analysis (Fox 1972, Qiu et al. 2012, Yu et al. 2014, Buda et al. 2018, Silva et al. 2018, Battaglia et al. 2020, Battaglia&Cucina 2020, Sun et al. 2021) or simply use some of the tools from statistical time series analysis (Vishwakarma et al. 2021). The prime motivator for using statistical time series analysis tools in anomaly detection is that they give an estimate of probability for each measurement, and an anomaly is per definition an improbable single measurement or measurement sequence.

There are several ways of classifying time series models.

- Measurement models vs hidden state models: The simplest models do not distinguish between process and measurement, and thus attaches dependencies directly on to the time series measurements themselves. Hidden state models do distinguish between the two, so that measurements are independent while the process has dependencies. With only autocorrelation as the dependency structure of the hidden process, the model is a hidden Markov-chain model.
- There are single series time series models and vector versions that allows for multiple series.
- Process models can assume equal distribution for each time step, thus assuming equidistant time steps. Continuous time process models are models that assume the process to be continuous in time and can thus deal with arbitrary time steps.

5.3.1 AR, ARIMA and SARIMA

AR stands for auto-regressive models. Auto regression simply means that the state of a process (which is also the measurements) depends on some previous values. For instance, for AR1, it depends on just the previous state, while in an AR2, it depends on the two previous states. Typically, the distribution is assumed to be normal and the regression linear, so that the next state is a linear combination of the previous state plus independent normal noise. Fox (1972) use AR models directly for anomaly detection, while Vishwakarma et al. (2021) used AR models in conjunction with neural networks. Silva et al (2018) used auto-regressive models, but with non-gaussian noise terms.

ARIMA is a larger framework of models, which combines auto-regression (AR) with integration (I) and moving average (MA). Integration means that the model describes the differences between measurements, rather than the measurements themselves, which is why one need to integrate in order to get to the measurements. There can be multiple integrations, so if I=2, then the differences in the differences of the measurements are

modelled, for instance. I=0 simply means no integration. Moving average means that the noise terms are seen as a moving average rather than just a simple independent noise term. MA=2 means for instance that in addition to a new noise term, the previous 2 noise terms are also used. An estimation method for ARIMA models was first developed by Box&Jenkins (1976). Laptev et al. (2015) and Sun et al. (2021) used ARIMA models for anomaly detection in time series.

SARIMA is another expansion, where in addition to the previous time steps, the values of the previous cycles are also used. For NVE purposes, the cycle in question is the year (though days may also be of interest for air temperature for instance). A SARIMA model with AR=1, SAR=1 and SMA=1 would utilize the last measurement, the measurement last year and the noise term of last year, for instance. As hydrological and meteorological time series usually have a clear seasonal component, this expansion can be of use. Buda et al. (2018) used SARIMA models as well as sub-categories of that (AR and ARIMA).

All such models do however assume the same distributional dependency for each time step and are thus not amenable to time resolution changes. Since the measurements are not separated from the process, gaps can also be a problem. These are also single series models (though multi series expansions exist, see later subsections). For a statistician versed in time series analysis, they are relatively transparent. Due to their transparency and linearity, they may also be amenable to regional modelling.

5.3.2 VAR

VAR stands for vector auto-regressive models. These models have the same structure as simple auto-regressive models (thus not hidden state models), but with vectors representing the process and the noise terms, and matrices representing the auto-regressive terms and possibly also the covariance of the noise terms. Qui et al. (2012) used VAR in their anomaly detection method. VAR models thus allow for interactions between various processes, both in the regressive part and possibly also the noise part. Thus, one can distinguish between causal and correlative connections between the processes. There exist also vector versions of ARIMA models.

The considerations of robustness, transparency and regional modelling are the same as for SARIMA models, but multi series treatment is what VAR models are made for.

5.3.3 PAR

PAR are autoregressive models where the model parameters depend on the season (where season is a categorical variable). Thus, the mean, the auto-correlation and even the size of the noise terms are allowed to change from one season to the next. Battaglia et al. 2020 used this type of model.

The considerations of robustness, transparency, multi series treatment and regional modelling are the same as for SARIMA models.

5.3.4 Hidden Markov-chain models

Hidden Markov-chain models are models which separate between measurements and a hidden process (or set of processes) that are described by a Markov-chain. A Markov-chain is any process model where the present depends on the past only through a finite set of the nearest past values). Thus, this is not a fixed family of models, but rather a generic term for

a lot of different time series models. Li et al. (2017) used an advanced form of hidden Markov-chain model with so-called fuzzy logic elements.

5.3.5 Linear SDEs and the layeranalyzer package

Stochastic differential equations (SDEs) are continuous time process models and are thus able to deal with arbitrary time steps. While one cannot give an analytical expression for SDEs in general, this is possible for linear SDEs. The distribution will then be normal and the dependency to past states is linear. Single process models are simply the Wiener process ("random walk" in continuous time) and the Ornstein-Uhlenbeck (OU or mean-reverting process), which generalizes the AR1 model to continuous time. However, one can also solve for vector systems of linear SDEs, which massively expands the possibilities. This allows for analyzing multiple series while taking the connections between them into account. The connections can be Granger causal (in the deterministic part of the SDE) or correlative (in the stochastic part of the SDE). (Note that Qui et al. 2012 also explored Granger causality as a tool for anomaly detection, but in a VAR setting).

The framework also allows for unmeasured processes to affect the measured ones in so-called hidden layers. This expands the modelling possibilities, even for single time series analysis. This type of modelling was first examined by the *layeranalyzer* framework (Reitan et al. 2012, Reitan&Liow 2019), which now comes in the form of an R package.

The *layeranalyzer* framework it also is in the category of hidden Markov-chain models, thus separating between measurement and process. Because of this and the continuous time element, it is robust to missing data, changes in time resolution and differences in time resolution between the various time series involved. It also has a regression type of periodicity handling, though this is still in a fairly primitive state (only periodicity in the mean by way of trigonometric functions). Fig. 7 shows an experimental use that I and Asgeir Petersen-Øverleir at Statkraft performed as an experiment. We used one support series and took away some of the data both in the time series of interest and in the support series and performed process inference (predictions on the process rather than the noisy measurements). The method is able to correctly predict an increase in discharge in that part of the series where the support series has data. In the parts where data is missing from both series, the interpolation is essentially linear.

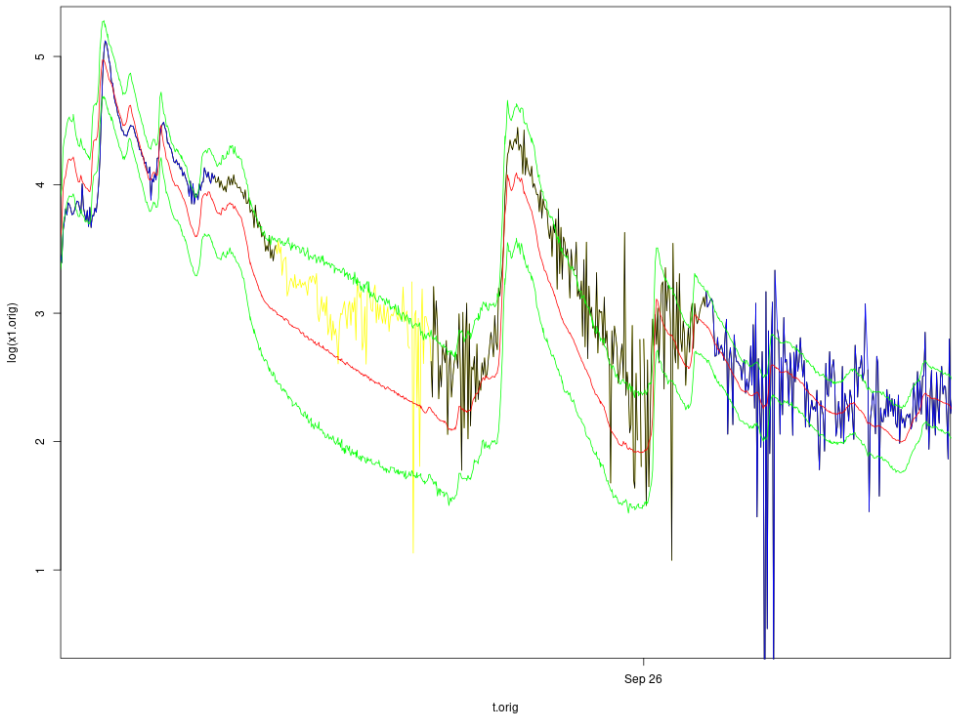


Figure 7: Interpolation with uncertainty bands using *layeranalyzer*. Another time series (not shown) were used for comparison through its connection to the time series shown. The blue line shows the actual measurements fed to the analysis, the red line shows the inferred process (the interpolation where measurements are missing), the green line shows the uncertainty, the black line is the removed data where the support series had measurements and the yellow line is the removed data where also the support series was missing. Note that the uncertainty becomes larger in the region where both time series have missing data (that is missing to the analysis).

Since the package delivers predictions with uncertainties, standardized residuals can be made, which alleviates the task of making threshold triggers for anomalies. The *layeranalyzer* package can be a bit slow during calibration, especially for large time series or large number of time series (or hidden layers). It is thus a problem that calibration and prediction is not separated at the moment. It should however be a relatively moderate task to make such a separation. The process predictions (as well as the likelihood itself) is calculated using a Kalman filter, since the hidden Markov-chain is normal and linear.

The parameters of a linear SDE may be cryptic to people not familiar with such tools, but the parameters are interpretable and due to the linearity of the model framework, it should be amenable to regional modelling if that is wanted. The main selling point, however, may be the methods robustness to time resolution changes and gaps.

5.3.6 Hydrologically motivated non-linear SDEs

SDEs can be utilized to mimic the operations of hydrological models. As a part of a statistical course at NVE, I made an example of this. A hidden humidity model (using the OU process) was thresholded to create a model for precipitation, which was then channeled into a reservoir and routed through a rating curve. OU parameters, precipitation threshold, reservoir volume and rating curve form parameter were all estimated. Since the system is non-linear, the extended Kalman filter was used for calculating likelihoods (and thus estimating the parameters) as well for doing process inference (and thus predictions). Fig. 8 shows an example of use.

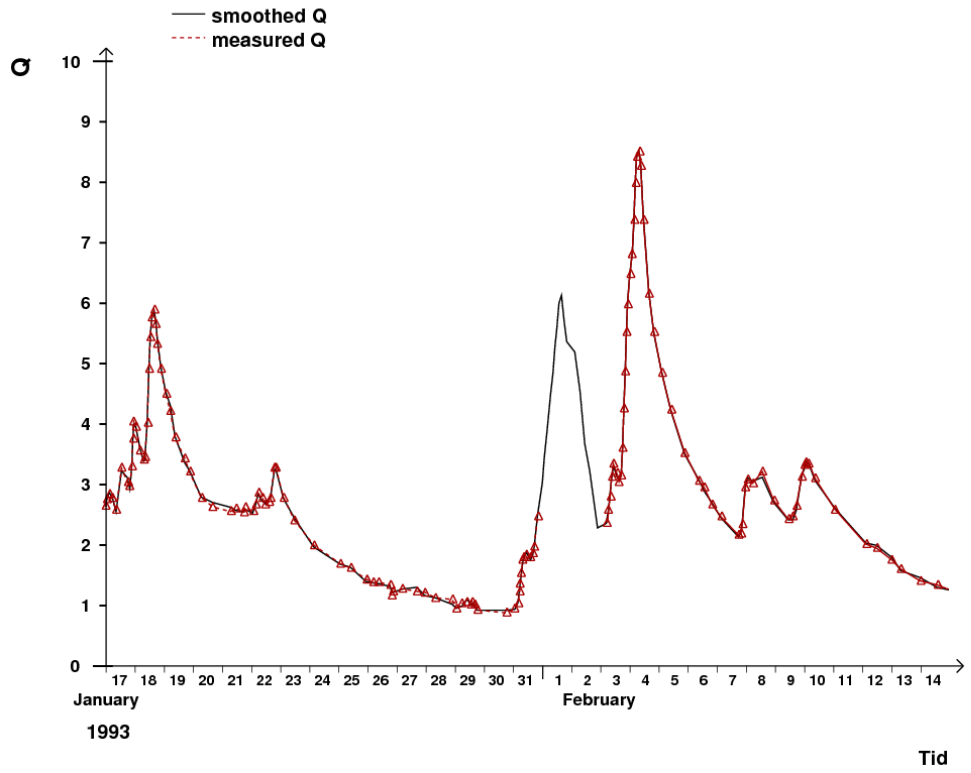


Figure 8: Process inference for a time series at Farstadelva (107.3.0.1001.1), where the topof a small flood was removed. The inferred top was very similar to the data removed. (Unfortunately not shown here.)

The model described is very simplistic in terms of hydrological modelling, but note that it is continuous in time, and thus can robustly handle time resolution and missing data issues. The framework could possibly be expanded to use precipitation series (which can be correlated to the one driving the discharge series), temperature (in order to affect the precipitation threshold as well as for ice expansions) and ground water. I do however expect that such models will be much harder to calibrate than *layeranalyzer* models.

As for robustness and multi series treatment, such a method should be as good as the *layeranalyzer* method. This method will however require much more in terms of future research.

6 A couple of initial tests on existing R packages

While the tools used for performing anomaly detection, such as time series modelling, neural networks and clustering methods, are implemented in R and Python, there are few packages that perform anomaly detection directly. I had time to look at two R packages, *tsoutliers* and *anomalize*.

6.1 The *anomalize* R package

The *anomalize* R package is a single series anomaly detection tool that decomposes time series into seasonality trend and remainder, and then checks the remainder (residuals?) for extreme values. It is an unsupervised method; thus, it will not learn how to best perform the quality control when trained on manually corrected data. The only learning is internal in the method.

I tested the method on the time series 16.122.0.1000.1 in the transition between 2017 and 2018 (an example provided by Mads-Peter Jakob Dahl at NVE), see Fig. 9. As can be seen, the *anomalize* package do detect some of the anomalies that were also detected during primary control. However, it did not mark some of the larger sequences corrected as anomalous, and marked many flood values that were deemed correct by the primary control as anomalous. The *anomalize* package thus has far too little specificity and sensitivity to be used, I think.

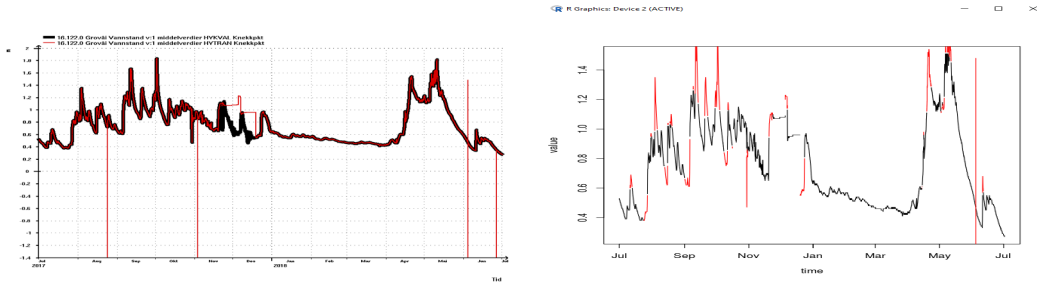


Figure 9: Figure 9a shows the difference between uncontrolled data (HYTRAN) in red, versus primary controlled data (HYKVAL) in black. Figure 9b shows how the *anomalize* package would correct the uncorrected data, with black lines deemed normal and red lines deemed as anomalies.

6.2 The *tsoutliers* R package

The *tsoutliers* R package is another unsupervised anomaly detection tool for time series. Thus, it also learns only from the uncontrolled series, not by the uncontrolled series plus the normal/anomaly label provided by previously performed quality control. The method however utilizes the SARIMA time series analysis, a more advanced statistical model than what the *anomalize* R package uses. An experiment on the same time series as for the *anomalize* package showed much of the same problems though. There are fewer false positives when it comes to anomalies, see Fig 10b, but the ones that are found are not only spikes but also floods that are probably correct. The package also fails to identify sequences where the variance is too low, such as can be seen in Fig. 10a. Due to its unsupervised nature and no hyper-parameters to adjust, neither the sensitivity nor the specificity can be improved, and are not good enough to warrant further tests, I think.

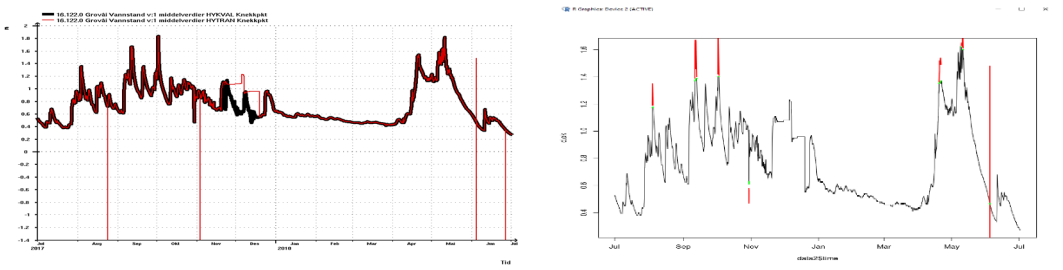


Figure 10: Figure 10a shows the difference between uncontrolled data (HYTRAN) in red, versus primary controlled data (HYKVAL) in black. Figure 10b shows how the *tsoutliers* package would correct the uncorrected data, with black lines deemed normal and red lines deemed as anomalies.

7 Conclusions

There are a great many ways of performing automatic anomaly detection on time series. The literature is certainly growing each year. The applications seem mostly to come from outside the fields of hydrology or meteorology, though. Thus, testing on hydrological time series is necessary before giving a final decision on what particular solutions to use. The testing performed so far has only been on a few methods and only on one example per method. Multiple methods and multiple examples are needed. Note also that multiple solutions may be used, since the requirements are different for different parts of the quality control pipeline.

Since NVE has a large archive of manually quality controlled time series, this suggests doing supervised rather than unsupervised learning. With some further feedback from the hydrologic engineers, the supervision can also come from new data, if the engineers are allowed to accept or reject the changes made by the automatic system. The feedback can come from the engineers in the form of accepting or rejecting the changes made by the system, and possibly by inserting their own changes (the latter requires a system for manual correction of real time data). If this can alleviate the manual primary control burden, such supervision should hopefully come as a boon rather than an extra burden.

Prediction-based anomaly detection provides methods for how to correct the values, something purely cluster-based anomaly detection does not. However, prediction-based anomaly detection can be slower than cluster-based and may not catch instances where the variation is too low (such as frozen wires would cause). It is therefore a thought that some simple cluster-based system could take care of incoming data and perhaps also the simplest corrections performed later on the real time archive. If these can catch all the instances of too low variation during the early stages of the quality control, then perhaps more sophisticated prediction-based anomaly detection can do the rest. If the simple cluster-based system could be relied on to also take care of all spikes other problems having to do with the data seen at the finest time resolution, then using time aggregation or interpolation in the later parts of the quality control pipeline to alleviate time resolution fragility could perhaps be a viable solution.

Another problem I see is that of new time series. Since machine learning is a component we want in a new quality control system, the lack of any data to learn from becomes are real problem. There can be many pragmatic solutions to this problem, such as borrowing the settings from a neighboring station or simply not performing quality control until more

data has arrived. A regional model would solve this problem more elegantly, but this is at least a moderately large research investment that comes on top of all the other research and testing needed to make a learning automatic anomaly detection system work. However, since regional models let lessons learnt in one time series be applied to others, the efficiency could be much greater than what can be achieved by local learning. Even if NVE do not start with making a regional model, the possibility of later making a regional model may perhaps be a point to consider when choosing anomaly detection methods.

One should be aware of the problems of fragility and non-transparency in a method but to what extent these problems matter, I will leave to the judgement of those reading this report.

8 References

F. Battaglia, D. Cucina (2020). Outlier identifiability in time series. The ISI's Journal of the Rapid Dissemination of Statistics Research. DOI: 10.1002/sta4.281.

F. Battaglia, D. Cucina, M. Rizzo (2020). Detection and estimation of additive outliers in seasonal time series. *Comput. Stat.* **35:** 1393–1409. DOI: 10.1007.

G.E.P. Box, G.M. Jenkins (1976). *Time series analysis: Forecasting and control*. San Francisco: Holden-Day.

T.S. Buda, B. Caglayan, H. Assem (2018). DeepAD: A Generic Framework Based on Deep Learning for Time Series Anomaly Detection. *Advances in Knowledge Discovery and Data Mining* 2018: 577-588.

V. Chandola, A. Banerjee and V. Kumar (2009). Anomaly Detection: A Survey. *ACM Computing Surveys*, **41**(3): 1–58.

W.W. Cohen (1995). Fast effective rule induction, *Proc. of the 12th Intl. Conf. on Machine Learning*: 115-123.

A.J. Fox (1972). Outliers in Time Series. *Journal of the Royal Statistical Society. Series B (Methodological)* **34**(3): 350-363.

T. Hastie, R. Tibshirani, and J. Friedman (2009), *The Elements of Statistical Learning: Prediction, Inference and Data Mining*, Second Edition, Springer Verlag.

K.J. Hsiao, K.S. Xu, J. Calder and A.O. Hero (2016). Multicriteria Similarity-Based Anomaly Detection Using Pareto Depth Analysis, *IEEE Transactions on neural networks and learning systems*, **27**(6): 1307-1321.

N. Laptev, S. Amizadeh, I. Flint (2015). Generic and Scalable Framework for Automated Time-series Anomaly Detection, Kdd '15, *Proceeding of the 21th ACM SIGKDD International Conference on Knowledge discovery and data mining*, August 2015: 1939-1947. DOI: 10.1145/2783258.2788611.

J. Li, W. Pedrycz, I. Jamal (2017). Multivariate time series anomaly detection: A framework of Hidden Markov Models. *Applied Soft Computing* **60**: 229–240.

J. Ma, S. Perkins (2003). Time-series novelty detection using one-class support vector machines, *Proceedings of the International Joint Conference on Neural Networks,* **2003**: 1741-1745. DOI: 10.1109/IJCNN.2003.1223670.

I. Melnyk, B. Matthews, H. Valizadegan, A. Banerjee, N. Oza (2016). Vector Autoregressive Model-Based Anomaly Detection in Aviation Systems. *Journal of Aerospace Information Systems*. **13**: 1-13. DOI: 10.2514/1.I010394.

D. Phung, V.S. Tseng, G.I. Webb, B. Ho, M. Ganji · Lida Rashidi (Eds.) (2018). *Advances in Knowledge Discovery and Data Mining, 22nd Pacific-Asia Conference, PAKDD* 2018, Melbourne, VIC, Australia, Part I.

H. Qiu, Y. Liu, N.A. Subrahmanya, W. Li (2012). Granger Causality for Time-Series *Anomaly Detection. IEEE 12th International Conference on Data Mining*.

A. Ray, R. Luck (1991). An introduction to sensor signal validation in redundant measurement systems, *IEEE Control Systems Magazine*, 11(2): 44-49. DOI: 10.1109/37.67675.

T. Reitan, T. Schweder, J. Henderiks (2012). Phenotypic Evolution studied by Layered Stochastic Differential Equations. *Annals of Applied Statistics* **6**(4): 1531-1551. DOI: 10.1214/12-AOAS559.

T. Reitan and L.H. Liow (2019). layeranalyzer: Inferring correlative and causal connections from time series data in R. *Methods in Ecology and Evolution*, **10**(12), 2183-2188.

D.T. Shipmon, J.M. Gurevitch, P.M. Piselli, S.T. Edwards (2017). Time Series Anomaly Detection; Detection of anomalous drops with limited features and sparse examples in noisy highly periodic data. arXiv: 1708.03665.

M.E. Silva, I. Pereira, B. McCabe (2018). Bayesian Outlier Detection in Non-Gaussian Autoregressive Time Series. *Journal of Time Series Analysis* **4**. DOI: 10.1111/jtsa.12439.

C. Sodja (2021). Detecting Anomalous Time Series by GAMLSSAkaike-Weights-Scoring, *Journal of Computational and Graphical Statistics*. DOI: 10.1080/10618600.2020.1868306.

W. Song, C. Gao, Y. Zhao, Y. A. Zhao (2020). Time Series Data Filling Method Based on LSTM—Taking the Stem Moisture as an Example. *Sensors* 2020, ***20***, 5045. DOI: 10.3390/s20185045.

Z. Sun, Q. Peng, X. Mou, Y. Wang, T. Han (2021). An artificial intelligence-based real-time monitoring framework for time series. *Journal of Intelligent & Fuzzy Systems* **40** (2021): 10401–10415. DOI: 10.3233/JIFS-200366.

I. Tinawi (2019) Machine Learning for Time Series Anomaly Detection, *Masters of Engineering in Computer Science and Engineering at the Massachusett's Institute of Technology*.

F. Vejen (ed), C. Jacobsson, U. Fredriksson, M. Moe, L. Andresen, E. Hellsten, P. Rissanen, Þ. Pálsdóttir, Þ. Arason (2002). Quality Control of Meteorological Observations, Automatic Methods Used in the Nordic Countries. *Norwegian Meteorological Institute Report* 2/2002.

G.K. Vishwakarma, P. Chinmoy, A.M. Elsawah (2021). A hybrid feedforward neural network algorithm for detecting outliers in non-stationary multivariate time series. *Expert Systems with Applications* **184**. DOI: 10.1016/j.eswa.2021.115545.

W. Wang, J. Bao, T. Li (2020). Bound smoothing based time series anomaly detection using multiple similarity measures. *Journal of Intelligent Manufacturing* **32**:1711–1727.

Y. Yu, Y. Zhu, S. Li, and D. Wan (2014). Time Series Outlier Detection Based on Sliding Window, *Mathematical Problems in Engineering*. DOI: 10.1155/2014/879736.

NVE

Norges vassdrags- og energidirektorat

MIDDE T N ATE 29
PO TBO 5091 MA OR T EN
0301 O O
TE EFON: 7 22 95 95 95

www.nve.no