AUTOMATED ANOMALY DETECTION

NIWA
Taihoro Nukurangi

Acknowledgements

This guidance was funded by MBIE's Envirolink Scheme (C01X2301) and supported by NIWA's Strategic Science Investment Fund (C01X1703—NIWA Platforms Plan). The Envirolink scheme funds research organisations (Crown Research Institutes, universities, and some not-for-profit research associations) to provide regional councils with advice and support for research on identified environmental topics and projects.

The High Frequency Water Quality Guidance Monitoring Tool is supported by:

This document was prepared by David Wood (NIWA) with contributions from the Automated Anomaly Detection Workgroup and survey participants. The workgroup and others from the water quality monitoring sector contributed to the content and design through surveys, a workshop, interviews and discussions, and contributed case study materials. They also provided feedback on the draft guidance. The input of workgroup members, survey participants and additional document reviewers is gratefully acknowledged. This chapter was reviewed by Mike Bargh and Lucy McKergow (NIWA), the workgroup and a group of external reviewers. This chapter was edited by Andrea Graves.

NIWA Report No: 2025314HN

Citation

Wood, DRR. (2025). Automated Anomaly Detection. High Frequency Water Quality Monitoring Guidance. An Envirolink Tool prepared by NIWA. NIWA Client Report 2025314HN http://www.envirolink.govt.nz/

Front cover: AAD wordcloud generated from the contents of this report.

Back cover: Through the Looking-Glass: Fouling in the housing of a dissolved oxygen sensor [Deb Hart, ORC].

CONTENTS

ABSTRACT

High-frequency water quality monitoring data generates large amounts of data, such that there are significant challenges to process and manage it. Issues such as sensor fouling, calibration drift and data transmission errors complicate the quality control (QC) process, rendering manual inspection of the data impractical. Consequently, there is an increasing need for automated tools to manage these tasks.

This chapter provides guidance on automated anomaly detection (AAD). It provides an overview of anomalies, including how they are caused and detected, and outlines the drivers and barriers to implementing AAD systems. It aims to provide a comprehensive understanding of the challenges associated with automating anomaly detection. The challenges include technical ones, plus consideration of processes and personnel. The chapter outlines a roadmap for anomaly detection, advocating for an iterative and collaborative approach to leverage and build on existing and developing expertise. It encourages the identification of quick wins, starting with one (or a small number of) water quality parameters, using simple rules-based tools that can yield substantial benefits at a low cost before progressing to more complex model-based tools.

A central finding is that expertise in Aotearoa New Zealand is widely distributed across councils, academic institutions and other organisations. By fostering collaboration and knowledge sharing, the sector can accelerate the adoption of AAD and enhance the quality and efficiency of water quality data management.

BACKGROUND

Managing high-frequency water quality (HFWQ) data is increasingly important for councils and environmental organisations. As monitoring technologies improve, the volume and rate of data collection has grown significantly. This data holds immense value, but only if it is accurate, reliable and timely. Ensuring data is of high quality is therefore critical. That means data errors or anomalies must be detected and managed. This process takes up significant time and resources, particularly if done manually, so there's a desire to develop automated processes.

This chapter is part of an Envirolink Tool project, *High Frequency Water Quality Monitoring Guidance.* It sits alongside guidance chapters on HFWQ Use Cases, Resourcing, and Sensor Selection.

PURPOSE AND SCOPE

This chapter provides a broad resource that documents the key drivers, barriers and practical considerations involved in developing, using and maintaining AAD systems. It is not possible to provide step-by-step instructions for using existing AAD tools as no single tool can meet all user needs in a plug-and-play fashion. The chapter will help regional council staff to rapidly learn about the challenges of current AAD tools and plan for successful development and implementation of an AAD system.

RELATED RESOURCES

Useful background reading that expands on the details in this chapter:

- An overview of HFWQ which places anomaly detection in the context of data collection (Rozemeijer et al. 2025).
- An overview of HFWQ data processing which links quality assurance and quality control to automated anomaly detection (Campbell et al. 2013).
- Background document for QARTOD automated data quality control (Bushnell 2022).
- Guidance manual for rules-based anomaly detection for dissolved nutrients in marine environments (IOOS 2018).
- A framework for automated anomaly detection in HFWQ data (Leigh et al. 2019).
- An automated AAD workflow (Jones et al. 2022).
- NEMS for data processing (NEMS 2023).

INTRODUCTION

The management of HFWQ data presents several challenges due to the nature of the data collection systems. Common issues include sensor fouling, calibration drift, and data transmission or storage errors (Jones et al. 2022). These errors can render the dataset unfit for its intended purpose.

Ideally, such issues are addressed at the source through robust quality assurance (QA) procedures. However, in many cases, errors persist and need to be identified during quality control (QC) after data collection. Manually detecting these errors is very time consuming and labour intensive. It can take an individual many days to manually review one year of HFWQ data. Few councils have sufficient resources to review data adequately.

Many councils and organisations are turning to automated anomaly detection (AAD) tools to reduce the time and effort required to clean up data, improve consistency, and make the QA/QC process more reproducible. These tools, often built using open-source technologies like Python and R, identify erroneous data by distinguishing it from valid data using rules-based logic and machine-learning algorithms.

However, implementing AAD is also time-consuming and challenging. Existing tools are not plug-and-play solutions. Instead, they are modular components that need to be integrated, customised and maintained. And it's important to recognise that AAD augments human judgment rather than replacing it, freeing up time for higher-value tasks.

Developing a functional AAD system involves:

- Selecting the right tools.
- Redesigning processes so that existing workflows can incorporate automated detection.
- Building or acquiring technical expertise to configure, use and maintain the system.
- Ensuring the right people are involved and that the value of AAD is understood across the organisation.

Many councils face constraints in terms of time, resources and lack of in-house expertise. Yet the processes required to support AAD can be complex and resource intensive. A roadmap (Figure 1) outlines a possible approach to developing an AAD system. The process has three components – tools, people and processes, and four stages – review, vision, plan (plus implement) and scale up.

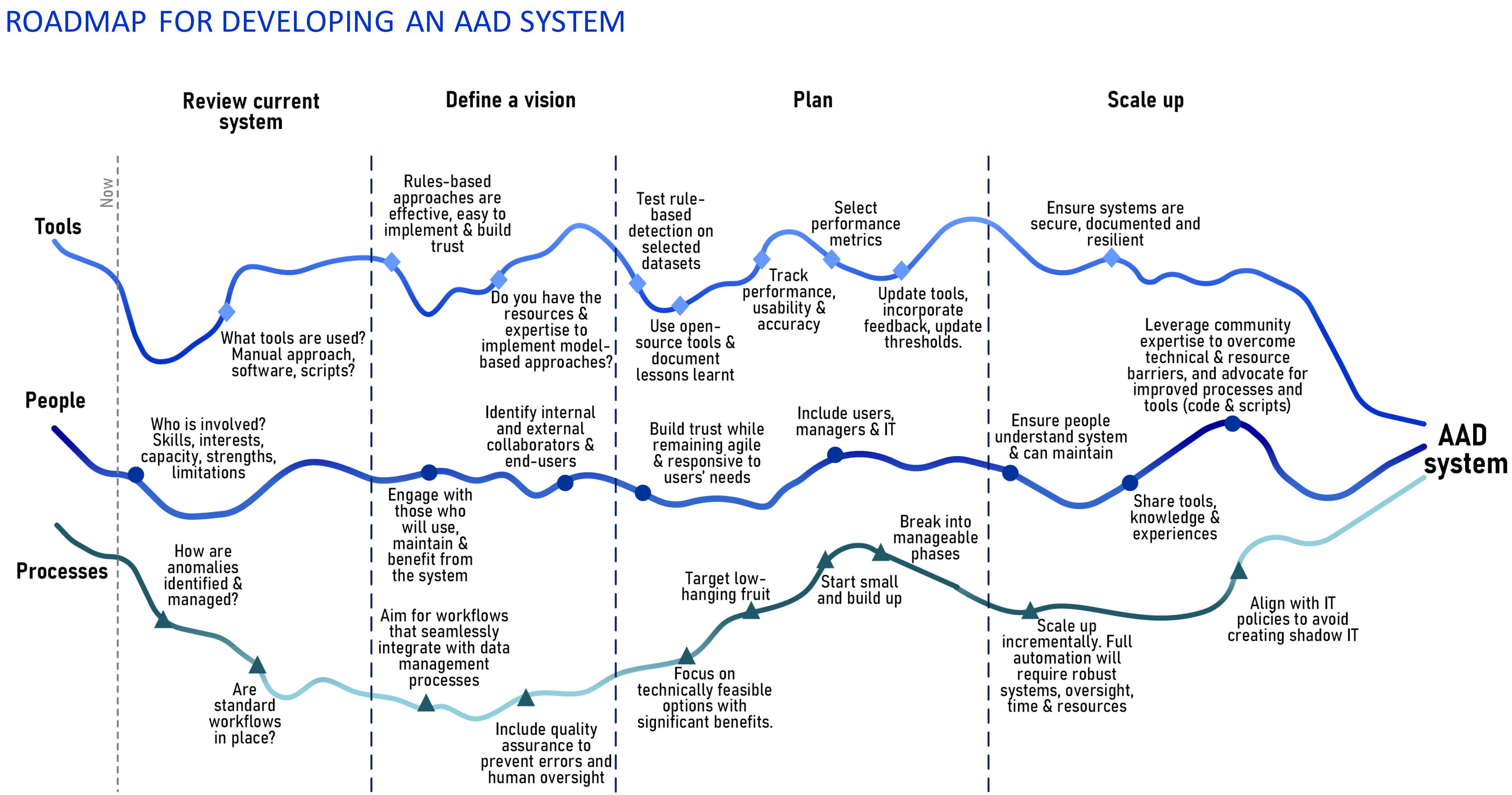

Figure 1. Roadmap for developing an AAD system from initial review to implementation. There are four main stages and three main components (Tools, People and Processes).

ANOMALY AND AUTOMATED ANOMALY DETECTION OVERVIEW

Unusual values may arise from measurement values that do not accurately reflect the parameter of interest or that originate from rare (and potentially interesting) events in the environment. Anomaly detection focuses primarily on identifying the former so that they can be investigated and managed. In practice, however, it may be difficult to distinguish between the two.

What causes data errors?

Data errors are often due to technical aberration or to local issues that make the resulting data unrepresentative of the environment being monitored. For example, when algae grow on a dissolved oxygen sensor or a turbidity sensor is buried by sediment the sensors measure accurately but do not represent the surrounding water conditions. Other common causes of technical aberrations include sensor noise or drift, power failures or spikes, and data recording or transmission errors. These errors should be identified, investigated, and sometimes corrected as part of the data post-processing and quality control procedures.

How do we identify data errors, and why are they difficult to detect?

One of the most robust approaches to identifying data anomalies is to have multiple redundant systems comprising independent sensors, recording and transmission equipment. By comparing the outputs of each system, erroneous results can be easily identified. This is sometimes done for mission-critical systems, such as flood monitoring, or to ensure that potentially hazardous waste streams do not enter the environment. But in most situations, the cost of running multiple systems in parallel outweighs the benefits.

In practice, the principal approach to identifying errors involves manual or automated inspection of data and associated metadata for anomalous values. These errors may appear as steps, spikes, gaps, poor resolution, out-of-range values, noisy periods, or unusually flat or straight segments (Figure 2).

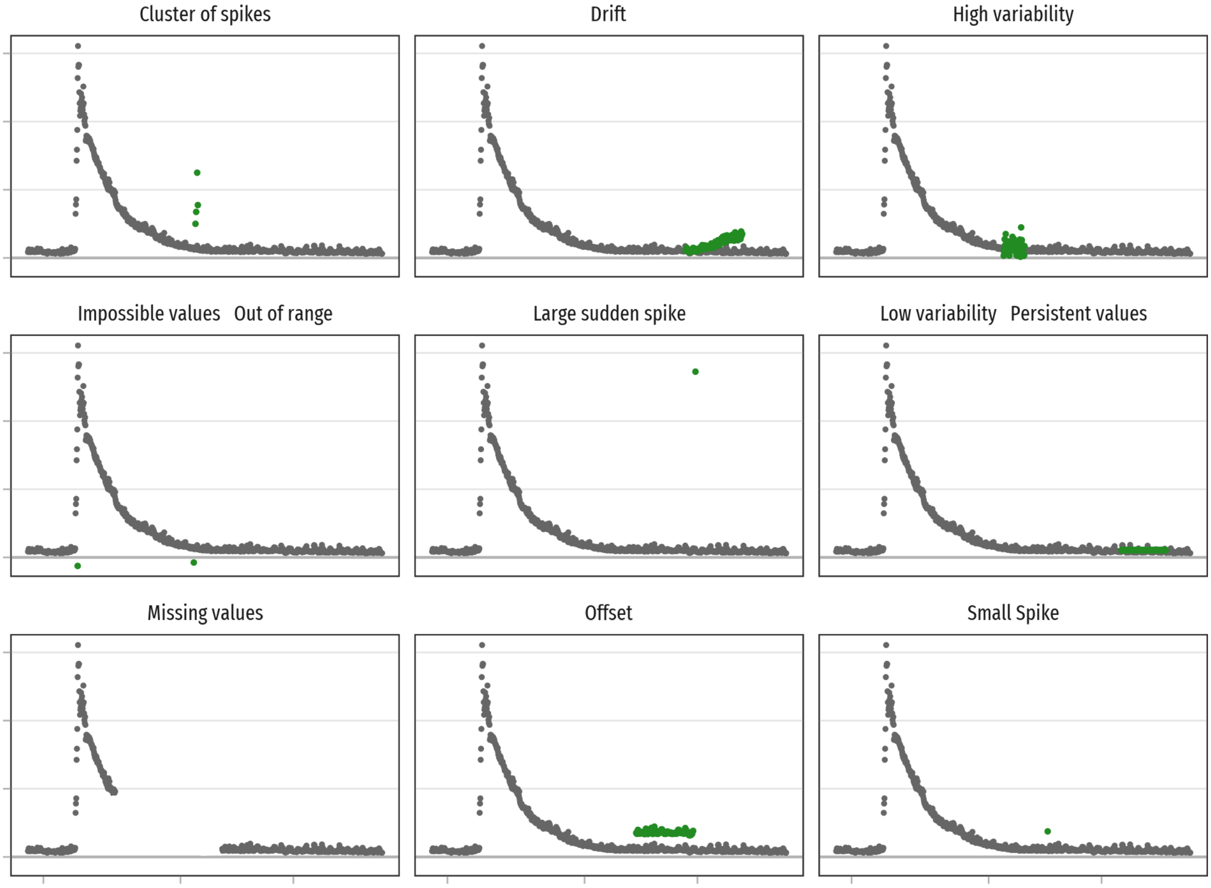

Figure 2: Example of a water quality time series featuring good quality values (grey) and anomalies highlighted in green (except missing values).

Anomalies have been described as global and local, as described by Goldstein and Uchida (2016), although other terms are also used. Values that differ significantly from the rest of the data are considered global, whereas local anomalies differ from data in their immediate vicinity but not from the overall dataset.

Even if we know what erroneous data may look like, there are several challenges in identifying it in practice, including:

- The actual/true value of the water quality indicator is unknown.
- All measurements have a degree of acceptable error associated with them.
- Unusual values may correctly represent variation in the environment rather than being erroneous.
- The causes of erroneous data are often unknown, at least until they are investigated.

It's therefore impossible to be certain whether a measured value is correct or erroneous, so a judgment must be made. Sometimes there is strong evidence that a value is incorrect, but at other times the situation can be ambiguous. This uncertainty means erroneous values will be missed (False Negatives – FN), or values may be misclassified as erroneous when they are not (False Positives – FP). No anomaly detection method can completely and accurately distinguish anomalous from non-anomalous data.

Approaches to anomaly detection in practice – manual vs automated

Detecting anomalies can be done manually or be automated, or these approaches can be combined.

The manual approach generally involves a person inspecting the data, often with visual aids to present the data graphically. It relies on people's abilities to detect patterns. They are sometimes aided by a rubric or guidelines to help them determine which values are anomalies. How this is done often depends on experience.

People can usually rapidly detect an individual anomaly, but the volume of data makes it time-consuming. The thinking is that "If a typical person can do a mental task with less than one second of thought, we can probably automate it using AI either now or in the near future," (Ng 2016). Automation relies on computer algorithms that are rules-based or model-based. The data is processed through the algorithm, which assesses whether the value is anomalous.

Both approaches have advantages and disadvantages (Table 1). In practice, when automation is used, it is commonly employed in conjunction with manual processes, and it is strongly encouraged that people are kept within the QC loop (Bushnell 2022).

Table 1. Comparison of advantages and disadvantages of manual and automated anomaly detection processes.

	Advantages	Disadvantages
Automated Anomaly Detection	Accelerates data processing. Reduces overall costs. Enhances consistency and reliability of anomaly detection. Easy to scale up – able to handle increasing amount of work. Can enable real-time data checks. Frees up resource to improve data quality.	Initial setup costs. Lack of available tools. Needs to be configured for each water quality indicator. Potentially requires new skillsets to develop, implement and use.
Manual Anomaly Detection	Low setup costs – makes use of pre-existing data processing tools. Makes use of expert knowledge. Approach can be easily transferred from one water quality parameter to another.	Highly subjective (Jones et al. 2018, Strohmenger et al. 2023). Difficult to do consistently. Time consuming. Does not scale up well.

General approaches to AAD – rules and models

Automated approaches are based on either rules or models. Rules-based approaches use predefined rules or threshold values to identify observations that diverge from the expected values. They rely on expert knowledge and are often specific to indicators, deployments and sites.

Model-based approaches are data-driven. They are also specific to indicators, sensor deployments and site (changing sensors may require the models to be retrained). Individual anomaly detection models learn from data by taking one of three approaches (Goldstein and Uchida 2016). These approaches are:

- Supervised – These methods learn from a HFWQ training dataset that has all observations classified as either anomalous or not. The resulting model can be applied to a new dataset and classifies each point as anomalous or not (Figure 3 A).
- Semi-supervised – These models use a HFWQ training dataset that's free of anomalies to learn what a normal dataset looks like (Figure 3 B). The resulting model can then identify anomalous points in a new dataset.
- Unsupervised – Unsupervised algorithms learn not from a training dataset but from the same data they classify as anomalous, identifying points that deviate from the normal data (Figure 3 C).

All three approaches use expert knowledge in different ways and degrees:

- In rules-based approaches, expert knowledge is hard-coded into an algorithm. They are ideal for detecting anomalies such as spikes and are easy for users to understand.
- Supervised models create classification rules by learning from expertly labelled data.
- Semi-supervised models learn what high-frequency data looks like in the absence of errors, based on what experts regard as normal.
- Most implementations of supervised, semi-supervised and unsupervised models have meta-parameters which, based on expert judgment, can be adjusted to improve error detection.

In an ideal world, models trained on data from one site would be transferable to another. However, transferability is usually restricted to the same site but on different collection occasions. To achieve optimal results, the training set must be representative, capturing the expected variation throughout the duration of the HFWQ data collection. This ensures that the models can accurately detect anomalies across different datasets, enhancing their reliability and effectiveness in various scenarios.

Examples of rules-based AAD

There are several examples of rules-based approaches applied to anomaly detection in high-frequency water quality. One of the most well-developed approaches is QARTOD (Quality Assurance/Quality Control for Real-Time Oceanographic Data) (IOOS 2018, Bushnell et al. 2020), which uses standardised tests, including:

- Gap tests – checks timing and ensures data points are collected when expected.
- Syntax test – ensure data is in the correct format.
- Location test – checks sample taken from the correct location.
- Gross range test – makes sure values are within the expected range.
- Climatology test – makes sure values are within seasonal range.
- Spike tests – ensures value does not exceed threshold relative to preceding observations.
- Rate-of-change tests – ensures change does not exceed a threshold value.
- Flatline tests – checks for repeated values.
- Multivariate tests – ensures values co-vary as expected.
- Attenuated signal test – tests for inadequate variation in time series data.
- Neighbour test – compares values with nearby sensors (if present).

The QARTOD approach is well-documented and was developed collaboratively with practitioners over the last few years (Bushnell 2022). The tests have been included in a well-maintained Python package called ioos_qc (Wilcox 2022). Several organisations in New Zealand have adopted these rules in their anomaly detection systems.

Examples of model-based AAD

Model-based AAD is a vast research area applied in numerous fields, including fraud detection in finance, diagnostics in healthcare, and environmental data (e.g., hydrology, HFWQ, air quality). At the time of writing, the use of model-based algorithms in New Zealand appears to be limited.

The three general modelling approaches to anomaly detection are conceptualised in Figure 3. Each has different data demands, and multiple algorithms can be used for each modelling approach.

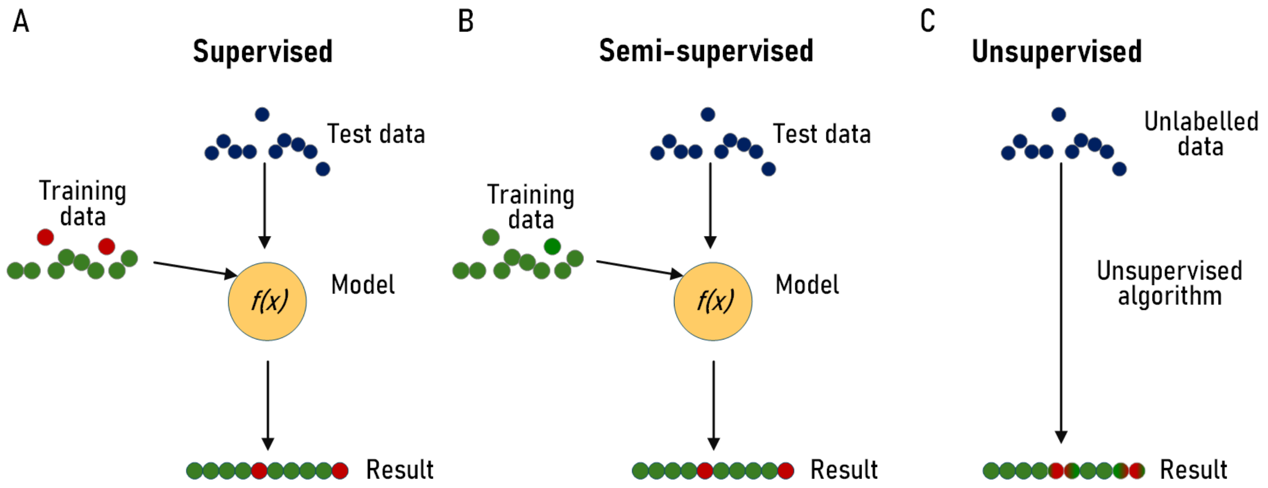

Figure 3: Conceptual description of three model-based approaches to anomaly detection (after Goldstein and Uchida 2016). (A) Supervised approach learns from a training dataset with anomalies labelled. (B) Semi-supervised learns from a training dataset without anomalies. (C) Unsupervised model does not require a training dataset to learn.

Schmidt and Kerkez (2023) showed that supervised models work well for identifying anomalies in environmental time series. From the wide variety of models that can be applied to the task, they tried Support Vector Machine, Multilayer Perceptron Neural Network, K Nearest Neighbours, Gaussian Naive Bayes, AdaBoost and Gaussian Processes. The fundamental weakness of supervised learning approaches is that they require labelled data to train the model.

Semi-supervised models include regression-based methods, with time series models such as ARIMA (Autoregressive Integrated Moving Average) being one example (Leigh et al. 2019, Jones et al. 2022). Instead of trying to obtain a non-anomalous dataset for these models to learn from, one is often created by pre-processing data using rules-based approaches to remove anomalies and replacing them with interpolated values. Unsupervised models, such as one developed by Talagala et al. (2019), require no training data and instead use statistical theory to identify unusual values.

Simple comparison of rule vs model-based approaches

A simple comparison of a rules- and model-based approach is illustrated in Figure 4. Both approaches classify the same point (red cross) as anomalous, but they do so in different ways. A straightforward rule has been applied to the data in Figure 4 A, which is that any value over 25 °C is classified as anomalous. The model in Figure 4 B was a moving average model, a semi-supervised technique useful for smoothing out random variation in time-series data. The grey band represents the prediction interval where most values are expected to lie. Any values outside this interval are flagged as anomalous.

In this instance, the rules-based approach is computationally far more straightforward than the model; however, there may be situations where the semi-supervised moving average models may identify anomalous values that the rules-based model does not.

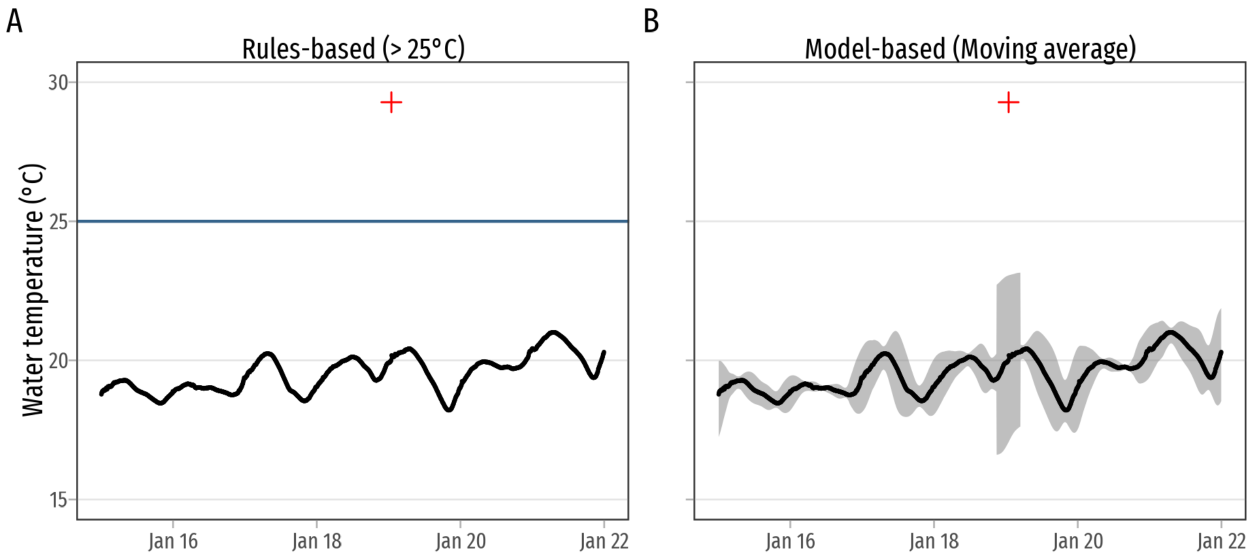

Figure 4: Side-by-side comparison of rules-based and model-based approaches to anomaly detection. The black line is the observed data, with the red cross being observed data classified as anomalous. (A) Rules-based – any value over 25 °C is anomalous. (B) Model-based – a moving average model with prediction intervals in grey. Any point outside the prediction interval is classified as anomalous.

Combining rules- and model-based approaches

The following sequence outlines a workflow for codes and scripts. The steps vary depending on whether the process is being carried out in near-real-time or there is post-processing of data. If the workflow is being applied in near real-time, then skip step 2, as data from quality checks is generally not available in real time. The recommended sequence is:

1. Do basic data checks: ensure data is in the correct format, identify and correct any timing issues (incorrect timing gap/steps, wrong time zones, etc.), and flag gaps.
2. Identify and correct for instrument drift and potential calibration errors using data available from quality checks – these types of errors are difficult to detect using rule or model-based approaches and can impact a large number of data points.
3. Use rules-based AAD – simple and understandable.
4. Use model-based AAD (optional) – more complex tools and may be able to identify nuances in data.

Limitations to Anomaly Detection

Anomaly detection algorithms have limitations. No single algorithm, or even a combination of algorithms, can be wholly relied upon to accurately identify all erroneous data within a dataset.

Some anomalies, such as sudden large spikes, are easy to identify, but others are more challenging. Drift is particularly difficult and is of concern for many HFWQ datasets. Many algorithms look for sudden changes or a lack of change, whereas drift is often a subtle bias in the data, possibly reducing sensitivity to environmental changes. Various groups have published code for drift correction (Jones et al. 2022, Schmidt et al. 2023), but it can still be difficult to identify without user intervention and verification or calibration data (IOOS 2018, Rodriguez-Perez et al. 2020).

Evaluating the performance of AAD tools

The performance of anomaly detection algorithms can be evaluated by several potential measures. The most common ones categorise the results into one of four categories:

- True Positives (TP) are points correctly classified as anomalous by the algorithm.
- True Negatives (TN) are points correctly classified as not anomalous by the algorithm.
- False Positives (FP) are points incorrectly classified as anomalous by the algorithm.
- False Negatives (FN) are points not classified as anomalous by the algorithm when they should have been.

This assumes that it is known whether each point is anomalous or not. When comparing the performance of tools on the same datasets, simple counts of the total are often used, though there are various summary statistics (Table 2). Accuracy is generally not recommended as a standalone metric in anomaly detection tasks as anomalies often constitute a small fraction of the dataset. High accuracy can be misleading if the algorithm classifies most points as normal.

Table 2. Summary statistics commonly used in AAD.

Statistic name	Calculation method
Positive predictive value	TP / (TP + FP)
Negative predictive value (or specificity)	TN / (TN + FN)
Recall (or sensitivity)	TP / (TP + FN)
Accuracy	TP + TN / (TP + TN + FP + FN)

Ideally, an anomaly detection algorithm would have positive predictive value, negative predictive value and recall of 100%. However, achieving this against all three measures is not possible. There are trade-offs between true positives, true negatives, false positives and false negatives. The precautionary approach favours false positives, where values are incorrectly flagged as anomalous, over false negatives, which implies anomalous values are missed.

The above metrics focus on how effective tools are in detecting anomalies. It is also possible to create metrics in terms of how much processing time is saved.

ATTRIBUTES OF AN IDEAL AAD FOR HFWQ

We asked workshop experts to identify an ideal AAD system and what guidance elements it should have.

The experts identified key features that an ideal AAD System would have:

- The system should be open source, allowing users to modify and extend its features as needed.
- A user-friendly interface should have good documentation, tutorials and easy-to-follow guides.
- Seamless integration with existing databases and data standards, such as NEMS, is crucial.
- The system should be able to handle large datasets efficiently and accurately detect anomalies without false positives.
- It should be flexible and extensible, allowing users to add new features and adapt the system to different applications.
- The system should provide clear visualisation of data and anomalies, and have detailed reporting capabilities.
- Ensuring security and proper version control is essential for maintaining the system's integrity.
- It's valuable to have a community support system, such as a GitHub repository, where users can share feedback and collaborate on improvements.

Key guidance elements to support AAD tool users included:

- A quick-start guide to help users get started with the system quickly.
- Documentation that comprehensively describes how to install, configure and use the system.
- Detailed tutorials and worked examples for different skill levels, from beginners to advanced users.
- Case studies that demonstrate the system's capabilities and how it can be applied to real-world scenarios.
- A frequently asked questions (FAQ) section and troubleshooting guide for common issues.
- A feedback feature so users can provide feedback and share their experiences to help improve the system.
- Online resources such as YouTube videos, wikis and forums, for additional support and learning.
- Keeping users informed about regular updates and new features added to the system.

OPEN-SOURCE CODES AND SCRIPTS FOR AAD

Numerous open-source tools and codes implement rules-based and model-based anomaly detection algorithms, and links to several examples are shown in Table 3. These algorithms automate specific anomaly detection tasks rather than the entire quality control process, which may include anomaly detection. Several tools have proposed workflows, with examples provided by Jones (2021), IOOS (2022) and Schmidt et al. (2023).

Many of these codes are available on individual or organisational Git repositories, while others can be found on repositories such as PyPI (Python Package Index) – a repository for Python software – or CRAN (Comprehensive R Archive Network). Additionally, code can be published in various formats associated with the supplementary materials of academic papers.

With Git repositories you can access the latest code, and contribute and collaborate directly with developers. Other repositories, such as PyPI and CRAN, simplify code installation by managing dependencies (where one piece of code relies on another to function) and potentially offering better versioning (tracking all changes), thus ensuring code stability. CRAN requires testing and review before publication, whereas PyPI does not. While code may function at the time of publication, it can fail over time, often due to dependency issues when related packages are updated.

Not all code is the same – some is production code while others are research code (see Case Study 2 – Assessing Open-Source Tools). Production code is intended for reuse, whereas research code often serves as a proof of concept to test an idea. These distinctions are important, as it can take considerable effort to modify and update code, particularly research code, to ensure it fits into a robust workflow.

Table 3. Some useful Git repositories.

Code name	Repository	Notes
ioos_qc	https://pypi.org/project/ioos-qc/	Python-based collection of utilities, scripts and tests to assist in automated quality assurance and quality control for oceanographic datasets and observing systems.
pyhydroqc	https://pypi.org/project/pyhydroqc/	Python package to identify and correct anomalous values in time series data collected by in situ aquatic sensors.
SaQC	https://git.ufz.de/rdm-software/SaQC	A tool/framework/application to quality control time series data. It provides a growing collection of algorithms and methods to analyse, annotate and process time series data.
hydrobot	https://github.com/HorizonsRC/hydrobot	Python package providing a suite of processing tools and utilities for Hilltop hydrological data.
HBRCWQCoder	https://github.com/hawkes-bay-rc/HBRCWQCoder.git	R-based discrete water quality coding tool using a mixture of rules-based and model-based approaches for coding data.
QartodNems	https://github.com/hawkes-bay-rc/NEMS_Autocoder_Continuous	Automated NEMS Coding of Continuous Data. A Jupyter Notebook and associated python functions aiming to combine the ioos_qc approach with NEMS.
NZ-Environmental-Informatics-Resources	https://github.com/NZ-Regional-Sector/NZ-Environmental-Informatics-Resources.git	A collection of links to courses, websites, repositories and code relevant to New Zealand Environmental data access, analysis and visualisation.

Code is not typically labelled as research or production code. Consequently, users must assess the code themselves. Indicators of production code include comprehensive documentation, testing (including unit tests), regular updates and an active community of developers.

Before using any code developed by others, it is essential to ensure that a licence is present and that you adhere to its terms. If in doubt, consult a legal expert.

AUTOMATION MATURITY – AUTOMATION OF ANOMALY DETECTION AND AUTOMATION OF THE WORKFLOW

Automated anomaly detection, particularly academic papers on the subject, has traditionally focused on automating the detection process and the efficacy of correctly classifying observations as anomalous or not. However, that is just one aspect of the entire workflow, which includes multiple related processes such as data extraction from a storage system, anomaly detection, decision-making (e.g., assigning NEMS codes), reinjecting the data into the storage system, and data publication.

While enhancing specific components like anomaly detection can lead to efficiency gains, these benefits may not be fully realised until multiple aspects of the workflow are optimised. The concept of a maturity model is a framework that helps organisations assess and improve capabilities over time (see Hammer 2007). It illustrates that organisations tend to move through stages of increasing maturity, for example, moving from one stage to another:

- Manual processing – manual analysis and inspection of data, often with the use of visual aids.
- Subtask automation – for example, automating the identification of anomalies.
- Workflow automation – integrating subtasks into an automated workflow, usually with human oversight.
- End-to-end automation – with human oversight.
- Autonomous data publishing – complete automation of processing from data collection QC to publishing, without the requirement for human oversight.

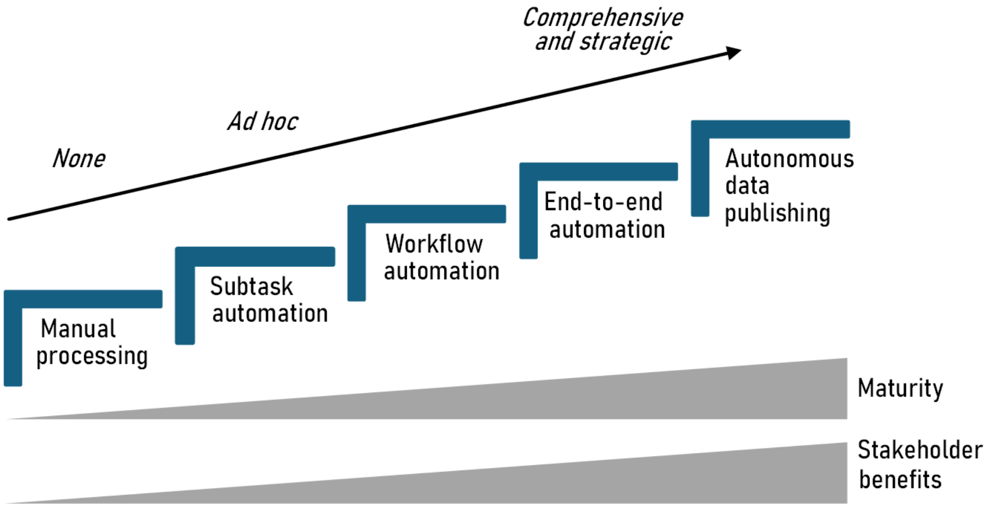

Figure 5. A simple automation maturity model. Stakeholder benefits grow as automation matures from no automation to comprehensive automation (after Hammer 2007).

This model illustrates the differences in the level and scale of automation, the skills required to make it happen, and the resources required. At the initial stages, automation is on a small scale, often *ad hoc*, driven by "evangelists", and used only for post-processing of data, but as it grows, more resources are required, and the benefits to organisations increase.

There are two common modes of anomaly detection: near real-time and post-processing of data. In the early stages of automation, post-processing seems to be the dominant use case. While these two modes may use similar anomaly detection approaches, they are distinct in several ways:

- Objectives – Near real-time detection focuses on early warnings of changes in water quality and preventing errors, ensuring quality assurance (QA). In contrast, post-processing is primarily concerned with quality control (QC) after the fact.
- Frequency – Near real-time detection is typically run more frequently, possibly daily, whereas post-processing may be conducted less frequently and on demand.
- Data availability – Information from quality checks (calibration, validation, verification), which is crucial for drift detection, is often lacking for near real-time detection, making it difficult to identify drift. Many model-based anomaly detection algorithms need data from both before and after an event. The limited post-event observations and lack of drift detection information make near real-time anomaly detection challenging, which explains the current focus on post-processing.
- Automation requirements – near real-time operation demands a high level of automation, with the entire workflow fully automated, ideally including scheduling and self-checking mechanisms to ensure the code

works as intended. Post-processing can be performed in a more ad hoc manner with subtask automation, and some tasks can be carried out manually or semi-automatically. As a result, fully automated systems for near real-time operation require a higher level of maturity to implement successfully than post-processing workflows.

There are also several drivers and barriers to automation, which are discussed further below. There are also questions about whether human oversight is required; currently, it is seen as essential.

CASE STUDY 1 – DRIVERS AND BARRIERS

Aside from the technical aspects of anomaly detection (such as algorithms, code and scripts), anyone looking to implement AAD must consider a broader range of factors. These factors include those that drive the uptake of AAD and those that act as barriers to its adoption.

Drivers can be the individuals advocating for AAD due to the potential benefits it offers. Conversely, barriers can be the obstacles and individuals or groups hindering its implementation. Barriers are the factors that resist change. The perception of these drivers and barriers can vary depending on one's perspective.

A survey on AAD drivers and barriers was circulated by email. The survey aimed to describe the current situation and provide an indication of the scale of the issues related to the use of AAD in HFWQ data. The survey may be biased towards individuals with a particular interest in AAD, as it utilised a convenience sampling method.

The survey attracted respondents who were primarily engaged in storing and managing data, as well as processing, checking, cleaning and collecting HFWQ data. Eighty per cent (24 out of 30) of respondents were involved in data management and storage, along with data processing. Sixty-three per cent (19 out of 30) participated in analysing cleaned data, while 57% (17 out of 30) focused on data collection. Many respondents performed multiple tasks directly related to the data.

Drivers

Anomaly detection in HFWQ data is primarily driven by individuals directly handling data – those involved in its processing, collection and utilisation. Middle management staff overseeing these teams also play a key role in advocating for AAD. In contrast, senior management, internal IT departments and external collaborators have limited influence, suggesting that current implementations are still in an ad hoc, immature stage. There is potential for broader development, particularly by engaging IT and leadership to unlock additional resources and capabilities.

The most valued benefits of AAD include improved data quality, accuracy and consistency in anomaly detection and editing, especially in real-time, which manual methods cannot achieve. While cost and time savings were noted, they were secondary to data integrity. Respondents also highlighted enhanced decision-making, regulatory compliance, and support for community-led water quality projects as important drivers. Free-text responses further emphasised the value of consistent data processing approaches, optimal resource use and potential improvements in job satisfaction.

Barriers

Barriers are the factors that resist change. While increasing drivers is important, removing barriers is equally crucial, though some barriers may be outside the control of any single person or council.

In order of importance, the most significant barriers were identified as:

- technical challenges
- lack of expertise
- competing priorities
- data integration issues
- lack of guidance
- high initial setup costs
- lack of awareness

Other barriers include the National Environmental Monitoring Standards (NEMS) not currently being in a format which can be readily converted into code (this transition is underway with NEMS revisions), lack of a nationally consistent approach, and potential changes to the National Policy Statement for Freshwater Management (NPS-FM) holding back councils from exploring AAD.

Technical challenges encompass a range of issues which were not explored in detail in the survey. These can include selecting appropriate algorithms, running them effectively – including debugging – and effectively integrating them within a workflow.

The development of AAD can be constrained by a lack of expertise, costs and the time/effort required to develop and use a new system.

Additionally, uncertainty in the operating environment, the lack of readily codable instructions for NEMS (Note that the QARTOD rules (Bushnell et al. 2020) have specifically been designed to be used by people creating software to automate QC tasks), and the future direction of HFWQ monitoring acts as a barrier and makes justifying investment decisions in this area more challenging. NEMS is working to remove part of this barrier by continuing to develop a data processing standard which could be automated.

Overcoming barriers

There are considerable challenges in getting some of the more sophisticated, model-based, sometimes referred to as 'black box' AAD algorithms to work (see Case Study 2 – Assessing Open-Source Tools). One way to address this is to opt for simpler, rules-based algorithms. These have the added advantage of being easy to understand, easy to turn into scripts and effective at identifying anomalies, all of which are important in building trust and demonstrating the added value of AAD.

Given that a lack of skills is an issue, there will be questions about how to make the best use of existing skills and how to access more. Some suggestions include:

- Look in-house – Many councils are looking to automate other parts of their business which are not directly related to HFWQ. It may be worthwhile reaching out to others who are working on automation projects. Even if in-house technical experts are not currently involved with automation, some will have relevant skills and experience. Collaboration makes others aware of your activities and prevents the common problem of creating an independent "shadow IT system" that is outside and possibly conflicts with the organisation's IT policies.
- Cross-organisational collaboration – The water quality sector in New Zealand has a wide range of skills and experience. Seek out those who have already solved the problem.
- Use current skills – Create plans that leverage your current skill base rather than requiring new skills. Organisations that have successfully implemented AAD are taking an agile approach to development which is iterative, incremental and works collaboratively with others who are using AAD. This is the approach described by the maturity model. Successful components are:
- Identifying low-hanging fruit, focusing on who has the need and interest in coming up with a solution.
- Focus on one (or a small number of) water quality indicators at a time; temperature and dissolved oxygen (DO) are common starting points.
- Understand who is using AAD tools and their coding capabilities. To eliminate the need for users to learn to write code, create visual front-ends. Various groups have created Jupyter notebooks, which allow code and documentation to coexist seamlessly, and taught people how to use them. In addition, listening to users' feedback and incorporating it into future iterations is a great way to improve AAD tools and ensure they are user-focused.

CASE STUDY 2 – ASSESSING OPEN-SOURCE TOOLS

During a 2023 workshop, three HFWQ AAD tools were identified for testing: *oddwater*, *pyhydroqc* and *ioos_qc*. The initial goal of this case study was to pass a series of HFWQ datasets through each tool and evaluate how well the tools identified anomalous data. The test data was sourced from literature (Leigh et al. 2019, Jones et al. 2022), NIWA and Northland Regional Council. Each test dataset was labelled (that is, the anomalies were flagged) and the performance of each tool could be compared against this reference. The anomaly rates in the labelled data varied from < 1% in one data set to 30% in another.

Code usability was a barrier. While we got the code working and passed the data through the tools, the key learnings were about open-source tool usability. Firstly, there were challenges getting the tools to run, particularly the two research tools (*oddwater* and *pyhydroqc).* Secondly, the open-source codes represent building blocks of an AAD system rather than a complete AAD system, meaning the code needs to be assembled and parameterised.

Each of the tools posed challenges:

- *oddwater* had limited functionality and insufficient documentation. The outlier threshold calculation was problematic, and the tool (including supplementary material and code) could not be run successfully. Associated code published by Leigh et al. (2019) was used instead. Although this code was executable, it was research-level code and was not structured as a package for easy reuse (it lacked modular functions, did not provide informative error messages when failures occurred, and lacked documentation).
- *pyhydroqc* faced installation issues due to code dependency problems. The code required debugging and maintenance before it could be used. This highlights the challenge of using research code, which is often not maintained and may require significant effort to turn into production-quality code.
- *ioos_qc* was the most mature tool evaluated, with extensive documentation and example scripts. It is production code, meaning it is designed to be used in the real world and is expected to be tested, bug-free and well-documented. However, its documentation (available at the time) did not account for all the latest changes to the code, potentially frustrating new users.

Overall, this case study highlights the differences between tools, as well as some of the key challenges and opportunities in using open-source code for anomaly detection.

Firstly, this case study shows that not all open-source code is the same. Some code can be considered production code; it is meant for reuse and has an active support community. In contrast, research code was created for a specific purpose; it may or may not be of high quality, and its lack of maintenance can make it hard to reuse. To use components of research code, you need to rely on your skills and resources to keep it operational.

Secondly, the open-source tools tested here are not plug-and-play; instead, they are functions, modules and scripts (building blocks) that the user must assemble into a workflow. This is both an opportunity and a challenge. The opportunity is that it allows users to design and build their own bespoke AAD systems. The challenge lies in designing and building a bespoke system. The usability of the AAD systems will depend on how the tools are designed and built.

Thirdly, though there may be challenges in using some of the open-source tools, the process of designing and building these tools has resulted in outputs that can help users build AAD systems:

- Documentation for *pyhydroqc* (Jones 2021) and *ioos_qc* (IOOS 2018) plus the paper of Leigh et al. (2019) make it clear that rules-based tools should be run before any model-based tools. Using rules-based tools (e.g., to detect calibration changes, drift correction, gaps, extreme outliers) as a first step ensures that data used for models does not include extreme outliers or artifacts.
- Some types of anomalies, in particular drift, which is a subtle anomaly, cannot easily be detected by these algorithms, and calibration or verification data is needed for them to be detected (IOOS 2018, Jones 2021).

CASE STUDY 3 – A JOURNEY IMPLEMENTING AAD

This case study reports the experiences of a regional council data manager who has been using AAD for HFWQ for several years.

The impetus to start using AAD was to move away from manual data checks and to use AAD as part of the QC process while implementing NEMS. The council is working to codify and develop scripts that align their processes with NEMS.

The initial implementation was inspired by existing tools and frameworks, particularly focusing on QARTOD, the Python-based solution *ioos_qc* (Wilcox 2022). A scientist familiar with Python helped set up the initial system, which was later refined. The refinement focused on improving the user interface and using rules; the original scientist was focusing on statistical approaches to anomaly detection.

The system primarily used a rules-based approach for anomaly detection, leveraging existing frameworks like QARTOD and NEMS (National Environmental Monitoring Standards). The system was integrated with the Hilltop Server, allowing for seamless data processing and anomaly detection. It was designed to be scalable, allowing for future enhancements and integration with other data sources and servers. Emphasis was placed on keeping a human in the loop to build trust in the system. This approach aimed to gradually increase confidence in the automated system.

There were challenges along the way, including:

- Initially finding the right thresholds for anomaly detection, which varied across different datasets and conditions.
- Ensuring data consistency and handling different types of anomalies, such as spikes and flat lines, which required continuous tweaking and refinement of the system.
- Training new staff and ensuring the system was user-friendly. The team developed Jupyter notebooks and other tools to simplify the user interface.
- Converting NEMS processes into rules that could be encoded. QARTOD provided a model for this - QARTOD rules have codable instructions (see Bushnell et al. 2020).

The system has had tangible benefits for the team, including efficiency gains and consistency. The automated system significantly reduced the time required for data processing. Tasks that previously took months can now be completed in minutes. The rules-based approach ensures consistent and repeatable results, which are crucial for maintaining data quality and reliability.

When developing the tools, the council has been collaborating with other councils and sharing ideas, even though the collaborating councils are taking different approaches.

It was noted that there are several key opportunities to improve the current system:

- There is potential for incorporating AI and machine learning techniques in the future to further enhance anomaly detection capabilities.
- This case study and Case Study 1 – Drivers and Barriers highlight the importance of collaboration and sharing of tools and knowledge across organisations to improve overall efficiency and effectiveness of AAD and data processing.

CASE STUDY 4 – IMPLEMENTING AAD INTO OPS

DevOps is the practice of integrating developing (Dev) software and deploying it into an operational environment (Ops). So, a DevOps team includes people with skills in software developer and IT operations. DevOps includes: (1) planning, building and testing software, and (2) deploying, operating and monitoring the software so feedback can be rapidly incorporated. Many organisations, like councils, have some experience in DevOps, but not necessarily within the water quality area. A NIWA expert in DevOps was interviewed and the key points are summarised.

Some key considerations for implementing an AAD system include:

- Decide whether the system will process data in batches or continuously. This choice impacts the system's design and performance.
- Code is initially written in a development environment and needs to be deployed onto a production server. This transition requires careful planning and execution.
- Ensure the system is running and flagging anomalies. Ideally, the system should have self-recovery capabilities to handle issues autonomously.
- Regularly update the code and run function tests to ensure the system continues to work as expected. This helps maintain the system's reliability and performance.
- Maintain the code for security reasons. Regular updates and patches are necessary to protect the system from vulnerabilities.
- The system should have resilience and not depend on a single person. Avoid creating shadow IT systems that operate outside the organisation's official IT infrastructure and policies.

The DevOps expert recommended some implementation steps:

- Develop the system with automation in mind to streamline processes and reduce manual intervention.
- Conduct thorough tests to ensure the system works as intended and continues to function correctly over time.
- Deploy the system to a production environment and maintain it regularly to ensure ongoing performance and security.
- Document the system comprehensively with guides that describe how to install, configure, use and troubleshoot.
- Plan for the system's lifecycle, including how to shut it down when it is no longer needed. No system lasts forever, so this planning is crucial.

SUMMARY

AAD provides clear benefits for managing HFWQ data, but developing, implementing and embedding an AAD system is a gradual process. While there are strong drivers for its adoption, significant barriers also exist.

The roadmap shown in Figure 1 offers a practical way forward. It highlights three core elements: tools, people and processes, and four development phases: review, vision, planning (and implementation), and scaling up. While tools are important, they are not enough; successful AAD depends on well-structured processes and committed people who can drive its adoption. The aim is to improve the efficiency and effectiveness of quality assurance and control, enhancing data quality within budget limits.

The first steps are to understand your current position, set a clear vision and align stakeholders. From there, create a flexible plan that focuses on early wins, tackling high-impact, achievable tasks first, before scaling up. Early development may occur in an R&D context, particularly where tools and workflows are still being tested. However, long-term success depends on integrating AAD into core IT systems to manage risks such as knowledge loss if key staff leave.

Focusing on the details of what an AAD workflow should look like, the recommended sequence for AAD checks is:

- Do basic data checks: ensure data is in the correct format, identify and correct any timing issues (incorrect timing gap/steps, wrong time zones, etc.) and flag gaps.
- Identify and correct for instrument drift and potential calibration errors using data available from quality checks – these types of errors are difficult to detect using AAD approaches and can impact a large number of data points.
- Use rules-based AAD – simple and understandable.
- Use model-based AAD – (optional) more complex tools and may be able to identify nuances in data.

This workflow is designed for post-processing. In real-time applications, quality-check data is typically unavailable, so step 2 may be omitted. Importantly, AAD does not replace the need for sensor calibration, verification or validation – it relies on and complements these processes by improving the efficiency and consistency of data review.

Collaboration can accelerate progress, reduce costs, and avoid fragmentation of environmental information systems. Shared expertise in HFWQ monitoring and data processing is a valuable resource. To harness it, the sector is encouraged to:

- Share code, documentation, user interfaces, unit tests, etc.
- Raise awareness and build support for AAD.
- Exchange knowledge, experience, and best practices.
- Grow capability and capacity across organisations.
- Form working groups, virtual teams, and communities of practice to jointly develop AAD systems, processes and tools.
- Develop generic solutions that can be adapted locally.
- Establish governance and oversight to ensure sustained progress.

By adopting an iterative, collaborative approach, the sector can transition from ad hoc solutions to robust, scalable systems, providing faster, more reliable data to support better environmental decisions.

REFERENCES

Bushnell, M. (2022) QARTOD Project Plan Update 2022–2026. https://doi.org/10.25923/1tcg-ht56

Bushnell, M., Bailey, K., Bosch, J., Burger, E., Dorton, J., Easley, R., Heitsenrether, B., King, J., Grissom, K., Tamburri, M. (2020) QARTOD-Prospects for Real-Time Quality Control Manuals, How to Create Them, and a Vision for Advanced Implementation. http://doi.org/10.25923/ysj8-5n28

Campbell, J.L., Rustad, L.E., Porter, J.H., Taylor, J.R., Dereszynski, E.W., Shanley, J.B., Gries, C., Henshaw, D.L., Martin, M.E., Sheldon, W.M., Boose, E.R. (2013) Quantity is nothing without quality: Automated QA/QC for streaming environmental sensor data. *BioScience*, 63(7): 574–585. http://doi.org/10.1525/bio.2013.63.7.10

Goldstein, M., Uchida, S. (2016) A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data. *PLOS ONE*, 11(4): e0152173. https://doi.org/10.1371/journal.pone.0152173

Hammer, M. (2007) The process audit. *Harvard business review*, 85(4). https://hbr.org/2007/04/the-process-audit

IOOS (2018) Manual for Real-Time Quality Control of Dissolved Nutrients Observations. https://cdn.ioos.noaa.gov/media/2018/03/Dissolved-Nutrients-Manual-Update_v1.1.pdf

Jones, A.S. (2021) Anomaly Detection and Correction for Aquatic Sensor Data. https://github.com/AmberSJones/pyhydroqc

Jones, A.S., Horsburgh, J.S., Eiriksson, D.P. (2018) Assessing subjectivity in environmental sensor data post processing via a controlled experiment. *Ecological Informatics*, 46: 86–96. https://doi.org/10.1016/j.ecoinf.2018.05.001

Jones, A.S., Jones, T.L., Horsburgh, J.S. (2022) Toward automating post processing of aquatic sensor data. *Environmental Modelling & Software*, 151: 105364. https://doi.org/10.1016/j.envsoft.2022.105364

Leigh, C., Alsibai, O., Hyndman, R.J., Kandanaarachchi, S., King, O.C., McGree, J.M., Neelamraju, C., Strauss, J., Talagala, P.D., Turner, R.D.R., Mengersen, K., Peterson, E.E. (2019) A framework for automated anomaly detection in high frequency water-quality data from in situ sensors. *Science of the Total Environment*, 664: 885–898. https://doi.org/10.1016/j.scitotenv.2019.02.085

NEMS (2023) Processing of Environmental Time-series Data, Version 1.1.0. https://www.nems.org.nz/documents

Ng, A. (2016) What Artificial Intelligence Can and Can't Do Right Now. *Harvard business review*, 94(11). https://hbr.org/2016/11/what-artificial-intelligence-can-and-cant-do-right-now

Rodriguez-Perez, J., Leigh, C., Liquet, B., Kermorvant, C., Peterson, E., Sous, D., Mengersen, K. (2020) Detecting Technical Anomalies in High-Frequency Water-Quality Data Using Artificial Neural Networks. *Environmental Science & Technology*, 54(21): 13719–13730. https://doi.org/10.1021/acs.est.0c04069

Rozemeijer, J., Jordan, P., Hooijboer, A., Kronvang, B., Glendell, M., Hensley, R., Rinke, K., Stutter, M., Bieroza, M., Turner, R., Mellander, P.E., Thorburn, P., Cassidy, R., Appels, J., Ouwerkerk, K., Rode, M. (2025) Best practice in high-frequency water quality monitoring for improved management and assessment; a novel decision workflow. *Environmental Monitoring and Assessment*, 197(4): 353. https://doi.org/10.1007/s10661-025-13795-z

Schmidt, J.Q., Kerkez, B. (2023) Machine Learning-Assisted, Process-Based Quality Control for Detecting Compromised Environmental Sensors. *Environmental Science & Technology*, 57(46): 18058–18066. https://doi.org/10.1021/acs.est.3c00360

Schmidt, L., Schäfer, D., Geller, J., Lünenschloss, P., Palm, B., Rinke, K., Rebmann, C., Rode, M., Bumberger, J. (2023) System for automated Quality Control (SaQC) to enable traceable and reproducible data streams in environmental science. *Environmental Modelling & Software*, 169: 105809. https://doi.org/10.1016/j.envsoft.2023.105809

Strohmenger, L., Eric, S., Bernard, C., Bonneau, J., Branger, F., Bresson, A., Brigode, P., Buzier, R., Delaigue, O., Devers, A., Evin, G., Fournier, M., Hsu, S.-C., Lanini, S., Lavenne, A., Lemaitre-Basset, T., Magand, C., Guimarães, G., Mentha, M., Thirel, G. (2023) On the visual detection of non-natural records in streamflow time series: challenges and impacts. *Hydrology and Earth System Sciences*, 27: 3375–3391. https://doi.org/10.5194/hess-27-3375-2023

Talagala, P.D., Hyndman, R.J., Leigh, C., Mengersen, K., Smith-Miles, K. (2019) A Feature-Based Procedure for Detecting Technical Outliers in Water-Quality Data From In Situ Sensors. *Water Resources Research*, 55(11): 8547–8568. https://doi.org/10.1029/2019WR024906

Wilcox, K. (2022) ioos_qc 2.1.0. https://pypi.org/project/ioos-qc/


NIWA
Taihoro Nukurangi