5-2024

Water Data Science: Data Driven Techniques, Training, and Tools for Improved Management of High Frequency Water Resources Data

Amber Spackman Jones
Utah State University, amber.jones@usu.edu

WATER DATA SCIENCE: DATA DRIVEN TECHNIQUES, TRAINING, AND TOOLS FOR IMPROVED MANAGEMENT OF HIGH FREQUENCY WATER RESOURCES DATA

by

Amber Spackman Jones

A dissertation submitted in partial fulfillment
of the requirements for the degree

of

DOCTOR OF PHILOSOPHY

in

Civil and Environmental Engineering

Approved:

_____ Jeffery S. Horsburgh, Ph.D. Major Professor	_____ Bethany T. Neilson, Ph.D. Committee Member
_____ Michelle A. Baker, Ph.D. Committee Member	_____ Belize A. Lane, Ph.D. Committee Member
_____ Tianfang Xu, Ph.D. Committee Member	_____ D. Richard Cutler, Ph.D. Vice Provost for Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2024

ABSTRACT

Water Data Science: Data Driven Techniques, Training, and Tools for Improved Management of High Frequency Water Resources Data

by

Amber Spackman Jones, Doctor of Philosophy

Utah State University, 2024

Major Professor: Dr. Jeffery S. Horsburgh
Department: Civil and Environmental Engineering

Sensors deployed to aquatic and terrestrial environments measure environmental phenomena at high frequency. As sensors have become more affordable, the body of sensor-based data has grown, offering detailed information to better understand and predict natural processes, detect environmental changes, assess impacts, track natural disasters, determine compliance with standards, and reduce health risks. However, growing volumes of environmental sensor data come with data management and analysis challenges that require scientists to consider additional tools and skillsets that fall within the realm of data science. Incorporating data science methodologies into traditional hydrologic science can improve manipulation, visualization, and analysis workflows for datasets of increased size and complexity. Major needs include ensuring data quality and incorporation of large datasets into scientific investigations and hydrologic data workflows. All of these can be difficult for domain scientists who lack data science expertise. This work addresses challenges associated with the increased volume and complexity of high frequency water data by: 1) advancing techniques for automating data

review, 2) assessing gaps and presenting solutions for water data science instruction aimed at building the technical expertise of water data managers, and 3) reviewing, prototyping, and recommending options for sensor data management.

High-quality data and robust derived products are necessary for using high frequency datasets in advanced analyses. Quality control post processing for sensor data is typically performed manually, which is a tedious and time consuming, but necessary, step for researchers and practitioners operating sensor networks. A Python code package, pyhydroqc, was developed to help automate this process. pyhydroqc uses data science approaches to automatically detect and correct anomalies in aquatic sensor time series, providing tools for reviewing large volumes of environmental sensor data prior to use and advancing workflows that require little manual human intervention. The pyhydroqc package is available for implementation and reuse by scientists and practitioners engaged in aquatic monitoring.

A major reason that domain scientists lack relevant data science skillsets is that traditional water resources engineering and science courses generally do not address how to collect, manage, and use increasingly large and complex datasets with modern technology. Newer courses in hydroinformatics or water data science have emerged to teach tools and approaches for working with large data. To assess and address challenges in effectively teaching data science techniques and tools in the context of the hydrology and water resources engineering domain, instructors of related courses were surveyed and interviewed. Based on survey and interview responses, a set of online educational modules were developed and made available as a shared educational resource for students and teachers to demonstrate how many of the articulated challenges in providing this type

of instruction can be addressed. These modules focus on accessing, managing, and analyzing high frequency sensor data and could be incorporated into existing courses or could be the foundation of a dedicated course aimed at advancing the data science skillsets of students pursuing degrees in hydrology or water resources engineering.

In addition to the necessary skills for working with high resolution sensor data, scientists and engineers need robust cyberinfrastructure for storing, managing, and sharing hydrologic observations data. Hydrologic information systems (HIS) meet this need and enable the automatic flow of high frequency observational data from observing infrastructure deployed in the hydrologic environment to centralized data storage infrastructure and then to data consumers via multiple access and dissemination tools. As a means of connecting domain scientists and engineers with data science skillsets and related cyberinfrastructure for data management, all of which are required for effective collection and use of high resolution sensor data, this dissertation provides a review of HIS technologies and systems, details the progression of software applications and standards that comprise HIS, and extracts and presents the essential architectural components of HIS. Functional details of these components are presented along with persistent challenges that are being addressed by modern HIS.

The results of this dissertation include improved practices, software and cyberinfrastructure tools for using high frequency sensor data, and educational resources aimed at elevating students' data science skillset. Goals include ensuring that data are high quality, improving instruction for prospective data collectors and users, and effective data management to better enable understanding of hydrologic processes.

(237 pages)

PUBLIC ABSTRACT

Water Data Science: Data Driven Techniques, Training, and Tools for Improved Management of High Frequency Water Resources Data

Amber Spackman Jones

Electronic sensors can measure water and climate conditions at high frequency and generate large quantities of observed data. This work addresses data management challenges associated with the volume and complexity of high frequency water data. We developed techniques for automatically reviewing data, created materials for training water data managers, and explored existing and emerging technologies for sensor data management.

Data collected by sensors often include errors due to sensor failure or environmental conditions that need to be removed, labeled, or corrected before the data can be used for analysis. Manual review and correction of these data can be tedious and time consuming. To help automate these tasks, we developed a computer program that automatically checks the data for mistakes and attempts to fix them. This tool has the potential to save time and effort and is available to scientists and practitioners who use sensors to monitor water.

Scientists may lack skillsets for working with sensor data because traditional engineering or science courses do not address how work with complex data with modern technology. We surveyed and interviewed instructors who teach courses related to "hydroinformatics" or "water data science" to understand challenges in incorporating

data science techniques and tools into water resources teaching. Based on their feedback, we created educational materials that demonstrate how the articulated challenges can be effectively addressed to provide high-quality instruction. These materials are available online for students and teachers.

In addition to skills for working with sensor data, scientists and engineers need tools for storing, managing, and sharing these data. Hydrologic information systems (HIS) help manage the data collected using sensors. HIS make sure that data can be effectively used by providing the computer infrastructure to get data from sensors in the field to secure data storage and then into the hands of scientists and others who use them. This work describes the evolution of software and standards that comprise HIS. We present the main components of HIS, describe currently available systems and gaps in technology or functionality, and then discuss opportunities for improved infrastructure that would make sensor data easier to collect, manage, and use.

In short, we are trying to make sure that sensor data are good and useful; we're helping instructors teach prospective data collectors and users about water and data; and we are making sure that the systems that enable collection, storage, management, and use of the data work smoothly.

ACKNOWLEDGMENTS

This work is dedicated to my children – Pearl, Ashton, and Griffin. You are my greatest joy and inspiration. I have been working on this degree for a large chunk of your lives – it's a big thing for me, but a small thing for the world. My hope is that my modest contribution can inspire you to dare greatly and to work to make the world a better place. I am grateful to my entire family for the encouragement, support, and patience. To my parents for instilling me with high expectations and encouraging me to pursue my interests. To my mother-in-law for the thoughtful and caring help. To my husband, Tanner Jones – thank you for collaborating, commiserating, and celebrating with me. Your support and love amplify any success of my own.

This work would not have been possible without the long-standing support and creative vision of my advisor Dr. Jeff Horsburgh. Thank you for the generously sharing experience and expertise, asking hard questions, persistently encouraging, and offering constructive feedback. I appreciate my committee members, Dr. Michelle Baker, Dr. Belize Lane, Dr. Bethany Neilson, and Dr. Tianfang Xu for thoughtful suggestions that helped shape this work and for support as my focus shifted. I appreciate those who were coauthors on chapters in this dissertation. I am grateful to the instructors that participated in interviews and surveys, and to the technicians and students who have worked to maintain the Logan River Observatory and to review and maintain the data. Thank you to Andrea Carroll and the staff at the UWRL for helping work through logistics.

I acknowledge my fellow graduate students Camilo Bastidas-Pacheco, Madison Haacke, Nour Atallah, Betsy Morgan, and Joseph Brewer – thank you for being great officemates, for coursework support, and for commiserating on setbacks and successes.

To my colleagues at the USGS – thank you for your interest and encouragement in my work and for offering resources and insight. I also want to thank my dear friends Emma and Laura – you have been there from the start and kept me sane through thoughtfulness, real talk, and healthy doses of adventure.

This research was primarily funded by the United States National Science Foundation (NSF) under grant number 1931297, Advancing Data Science and Analytics for Water (DSAW). Additional support was provided by the FAIR Cyber Training Fellowship program at Purdue University corresponding to NSF grant number 1829764. Additional funding was provided by the National Oceanic & Atmospheric Administration (NOAA), awarded to the Cooperative Institute for Research to Operations in Hydrology (CIROH) through the NOAA Cooperative Agreement with The University of Alabama (NA22NWS4320003). Additional support and funding were provided by the Utah Water Research Laboratory at Utah State University. Any opinions, findings, and conclusions or recommendations expressed are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Amber Spackman Jones

CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER 1

INTRODUCTION

The availability of hydrologic and water-related data has rapidly grown as new datasets have been generated and existing datasets have been shared and published. As sensors and related technology have decreased in price, the number of high temporal frequency water-related observations has increased (Pellerin et al., 2016; Rode et al., 2016). The resolution of data creates opportunities to increase understanding of hydrologic processes, but the associated volume presents challenges for managing the data and extracting useful information (Campbell et al., 2013; Gries et al., 2014; Jones et al., 2017; Gibert et al., 2018). Water scientists and engineers often need to perform data manipulation, visualization, and analysis tasks that may be difficult to apply to larger and more complex datasets. Traditional engineering or science curricula may not have prepared them with the data science tools they need to tackle these types of data workflows (Merwade and Ruddell, 2012; Burian et al., 2013; Gibert et al., 2018; Habib et al., 2019). Thus, two major needs are evident: 1) improved tools and techniques to ensure the quality of high frequency data, establish methods for creating derived data products, and make those data available for further analyses; and 2) data intensive scientific methods and expertise around their use to enable incorporation and analysis of large datasets into scientific and management investigations to gain better understanding of hydrologic processes.

This dissertation provides enhanced tools in the form of reproducible algorithms, code packages and notebooks, and online resources that address the needs articulated above and that will be of use to researchers, educators, practitioners, and water managers

who monitor water systems with *in situ* sensors, manage the data, perform subsequent analyses, and who instruct on these topics. To guide the research, the following objectives were identified. Each of the objectives is addressed within one of the chapters of this dissertation.

Objective 1: Advance tools for automatically detecting and correcting anomalies in environmental sensor data using data science and machine learning approaches.

Although increases in data size and availability invite advanced analyses, to effectively gain insights into hydrologic systems and water resources, data must be robust with regard to quality (Campbell et al., 2013; Gibert et al., 2018). As datasets grow in size and complexity and as the number of data collection sites expands, data management overhead also increases. Time series of observations from environmental sensors generally need to be examined for validity by technicians or scientists who are familiar with the sensors, the monitoring sites, and the phenomena of interest (Campbell et al., 2013; Horsburgh et al., 2015; Jones et al., 2018). Performing review and corrections to account for errors and ensure high data quality is a significant resource cost for obtaining high temporal resolution data (Jones et al., 2017). Rules and algorithms exist for identifying some types of anomalous values in sensor data streams (Dereszynski and Dietterich, 2007; Hill et al., 2009; Taylor and Loescher, 2013); however, detecting subtle anomalies, classifying anomalies, and applying corrections typically require technician expertise (Fiebrich et al., 2010; White et al., 2010). Data science and machine learning approaches may be effective tools for streamlining this process and improving data robustness for subsequent analyses.

With the increase in data availability, computational resources, and programming software with associated code libraries and tools, data science opportunities have expanded in most domains. Data science applies analytical methods and computational power with subject understanding to transform data to decisional knowledge (Gibert et al., 2018) and, according to a widely accepted formulation, requires a combination of coding, math and statistics, and scientific expertise (Conway, 2013). Machine learning is often equated with data science; indeed, it comprises the overlap between math and statistics and computing (Conway, 2013). Machine learning algorithms are frameworks that use computational methods to "learn" a model based on datasets without using a complex set of rules or predetermined equations as might be done with traditional statistical or domain focused model development (Géron, 2017). Compared to more conventional techniques, machine learning uses patterns in actual data to build models rather than fitting data to a defined model. One strength of machine learning is its ability to handle interactions and nonlinearity in relationships between inputs and targeted outputs without the constraints of human preconceptions (Shen, 2018a).

Machine learning approaches show promise toward advancing automation of quality control of sensor data streams (Leigh et al., 2018; Talagala et al., 2019), and approaches for sensor time series anomaly detection and correction from other fields may have relevance to environmental sensor observations. In an effort toward streamlining the process for quality control post processing of aquatic sensor data, we explored several data science and machine learning approaches with an aim for detecting and correcting anomalies in aquatic sensor data. This work combines concepts of rules-based anomaly detection (Sheldon et al., 2008; Horsburgh et al., 2015) with machine learning anomaly

detection (e.g., Leigh et al., 2018) and expands on previous efforts by testing algorithms on long periods of actual high frequency observations that were reviewed, labeled, and corrected by technicians and by releasing a software package with functions that could be applied to other datasets.

Objective 2: Advance materials and instructional approaches for teaching hydroinformatics and water data science.

A major reason that domain scientists and engineers struggle to effectively use high resolution data from environmental sensors is that they lack the data science skillsets needed for collection, management, quality control, and analysis of the large volume of data sensors produce. Traditional water resources engineering and science curricula predate both the growth in water data made possible by the proliferation of *in situ* sensors as well as the increased prevalence of data science approaches. However, students need preparation and training in techniques and tools to effectively work with complex datasets and in the appropriate application of data science methods to real-world, open-ended problems (Merwade and Ruddell, 2012; Burian et al., 2013; Gibert et al., 2018; Maggioni et al., 2020; Ngambeki et al., 2012). Instructors are challenged to incorporate technical tools in their courses and to find water-related datasets and examples for applying data science techniques that are accessible for students (Habib et al., 2019; Lane et al., 2021). Courses in hydroinformatics, or the application of technical tools to water related data (Burian et al., 2013; Chen and Han, 2016; Vojinovic and Abbott, 2017; Makropoulos, 2019), and in water data science, which packages data science topics with water related applications (Gibert et al., 2018; McGovern and Allen, 2021), can better equip students for solving emerging data-intensive problems. Educational content shared

in online community platforms can support instructors and students with lessons and examples to enable data-driven learning (Habib et al., 2019; Maggioni et al., 2020; Lane et al., 2021). Furthermore, the shift to online formats in the wake of the COVID-19 pandemic increased interest in alternative approaches to teaching and learning technical content (Beason-Abmayr et al., 2021; Rapanta et al., 2021).

To address challenges in providing effective instruction in hydroinformatics and water data science areas, we surveyed instructors of existing courses, documented challenges and successes, and developed and shared online educational modules that demonstrate how many outstanding challenges can be addressed. While information on implementing statistical and data science techniques in common coding environments is generally available, examples of application to specific problems in the water resources domain are specialized and emergent (Shen, 2018a), and providing accessible data science case studies is of high value to instructors and students.

Objective 3: Synthesize the current state of practice and existing standards and approaches for sharing, delivering, and integrating hydrologic time series observations.

The large quantities of data generated by environmental sensors necessitate cyberinfrastructure to manage data collection, storage, and publication (Horsburgh et al., 2008; Benson et al., 2010; Dow et al., 2015; Horsburgh et al., 2019). Hydrologic Information Systems (HIS) are comprehensive hardware and software for managing data derived from sensors (Mason et al., 2014; Jones et al., 2015). HIS are essential for effectively managing data, reducing the time between data collection and analysis, and linking data collection to scientific research and water resource management (Muste et

al., 2015; McGuire et al., 2016; Samourkasidis et al., 2019). HIS are crucial for disseminating water data to diverse user groups including government agencies, research organizations, and citizen scientists. Community-developed tools and standards have made HIS more accessible to scientists and practitioners that are monitoring with sensors (Horsburgh et al., 2011; Ames et al., 2012; Jones et al., 2015). However, even though HIS have evolved for better data interoperability (Goodall et al., 2008), challenges remain related to standardized data models for encoding the data, vocabularies for describing the data, and exchange protocols for transmitting data; and new challenges have emerged as technology has advanced and as the needs and priorities of monitoring networks have changed.

In an effort to advance HIS, we reviewed existing systems and technologies and extracted key components that we presented as the foundation for a generalized HIS architecture. We then identified challenges and opportunities for improved functionality and described how existing or emerging HIS are addressing these challenges and opportunities to offer guidance for researchers and practitioners collecting and using high frequency sensor data. Understanding established principles of HIS is essential to building a next generation HIS with modern technology. This is a key objective of HydroServer, a current effort at Utah State University (USU) to support a national network of cooperative monitoring sites that complement national agency monitoring efforts and contribute to a national water model. The review, generalized architecture, and identification of challenges and advancement opportunities are directly informing development of HydroServer as a next generation HIS.

This dissertation leverages data science methods with high temporal frequency

environmental observations to develop and implement tools and methods for ensuring high quality data for scientific analysis and education. The specific research questions addressed focus on automatically post processing hydrologic sensor data, effective approaches for instruction on water data science, and options for systems for managing time series of sensor observations. The outline of this dissertation is as follows:

Chapter 2 addresses *Objective 1: Advance tools for automatically detecting and correcting anomalies in environmental sensor data using data science and machine learning approaches* by developing and presenting a set of tools and techniques for automating post processing of aquatic sensor data. The algorithms are encapsulated as functions within a Python package that is open source and available for use by scientists and practitioners who manage data observed with environmental sensors. The algorithms were developed and tested on high frequency water quality data collected in the Logan River Observatory in northern Utah, USA.

Chapter 3 addresses *Objective 2: Advance materials and instructional approaches for teaching hydroinformatics and water data science* through a presentation of educator perspectives on the subject, options and requirements for online platforms for sharing educational materials, and examples of shared educational modules that address many of the challenges identified through our surveys with educators. More specifically, the chapter reports the results of a survey of instructors who teach hydroinformatics, water data science, or related courses. It articulates the challenges and successes experienced by instructors in providing effective instruction for building students' technical skillsets and readiness to work in data-intensive engineering and science environments. This chapter also describes the development and sharing of online instruction modules that

demonstrate how many of the challenges expressed by instructors can be overcome. These outcomes demonstrate a path forward for bringing more rigorous and comprehensive hydroinformatics and water data science instruction to domain scientists and engineers through educational materials shared online.

Chapter 4 addresses *Objective 3: Synthesize the current state of practice and existing standards and approaches for sharing, delivering, and integrating hydrologic time series observations* by reviewing the functionality of existing commercial, open-source, and government HIS to illustrate common patterns and techniques, which are represented by a generalized, high-level HIS architecture. It also identifies and describes challenges that have persisted through HIS development and operation along with new challenges that have emerged as technology has advanced. This chapter describes how new development efforts, including the HydroServer software platform at USU, are addressing these challenges and as a specific outcome serves as a foundation and blueprint from which modernized HIS can be advanced.

REFERENCES

Ames, D.P., Horsburgh, J.S., Cao, Y., Kadlec, J., Whiteaker, T., Valentine, D., 2012. HydroDesktop: Web services-based software for hydrologic data discovery, download, visualization, and analysis. Environ. Model. Softw. 37, 146–156. https://doi.org/10.1016/j.envsoft.2012.03.013

Beason-Abmayr, B., Caprette, D.R., Gopalan, C., 2021. Flipped teaching eased the transition from face-to-face teaching to online instruction during the COVID-19 pandemic. Adv. Physiol. Educ. 45, 384–389. https://doi.org/10.1152/advan.00248.2020

Benson, B., Bond, B., Hamilton, M., Monson, R., Han, R., 2010. Perspectives on next-generation technology for environmental sensor networks. Front. Ecol. Environ. 8, 193–200. https://doi.org/10.1890/080130

Burian, S.J., Horsburgh, J.S., Rosenberg, D.E., Ames, D.P., Hunter, L.G., Strong, C., 2013. Using interactive video conferencing for multi-institution, team-teaching. ASEE Annu. Conf. Expo. Conf. Proc. https://doi.org/10.18260/1-2--22706

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, Wade, M., Boose, E.R., 2013. Quantity is Nothing without Quality. Bioscience 63, 574–585. https://doi.org/10.1525/bio.2013.63.7.10

Chen, Y., Han, D., 2016. Big data and hydroinformatics. J. Hydroinformatics 18, 599–614. https://doi.org/10.2166/hydro.2016.180

Conway, D., 2013. The DataScience Venn Diagram [WWW Document].

Dereszynski, E.W., Dietterich, T.G., 2007. Probabilistic Models for Anomaly Detection in Remote Sensor Data Streams, in: Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence (UAI2007). pp. 75–82.

Dow, A.K., Dow, E.M., Fitzsimmons, T.D., Materise, M.M., 2015. Harnessing the environmental data flood: A comparative analysis of hydrologic, oceanographic, and meteorological informatics platforms. Bull. Am. Meteorol. Soc. 96, 725–736. https://doi.org/10.1175/BAMS-D-13-00178.1

Fiebrich, C.A., Morgan, C.R., McCombs, A.G., Hall, P.K., McPherson, R.A., 2010. Quality assurance procedures for mesoscale meteorological data. J. Atmos. Ocean. Technol. 27, 1565–1582. https://doi.org/10.1175/2010JTECHA1433.1

Géron, A., 2017. Hands-On Machine Learning with Scikit-Learn & TensorFlow. O'Reilly.

Gibert, K., Horsburgh, J.S., Athanasiadis, I.N., Holmes, G., 2018. Environmental Data Science. Environ. Model. Softw. 106, 4–12. https://doi.org/10.1016/j.envsoft.2018.04.005

Goodall, J.L., Horsburgh, J.S., Whiteaker, T.L., Maidment, D.R., Zaslavsky, I., 2008. A

first approach to web services for the National Water Information System. Environ. Model. Softw. 23, 404–411. https://doi.org/10.1016/j.envsoft.2007.01.005

Habib, E., Deshotel, M., Guolin, L.A.I., Miller, R., 2019. Student perceptions of an active learning module to enhance data and modeling skills in undergraduate water resources engineering education. Int. J. Eng. Educ. 35, 1353–1365.

Hill, D.J., Minsker, B.S., 2010. Anomaly detection in streaming environmental sensor data: A data-driven modeling approach. Environ. Model. Softw. 25, 1014–1022. https://doi.org/10.1016/j.envsoft.2009.08.010

Horsburgh, J.S., Caraballo, J., Ramírez, M., Aufdenkampe, A.K., Arscott, D.B., Damiano, S.G., 2019. Low-cost, open-source, and low-power: But what to do with the data? Front. Earth Sci. 7, 1–14. https://doi.org/10.3389/feart.2019.00067

Horsburgh, J.S., Tarboton, D.G., Hooper, R.P., Zaslavsky, I., 2014. Managing a community shared vocabulary for hydrologic observations. Environ. Model. Softw. 52, 62–73. https://doi.org/10.1016/j.envsoft.2013.10.012

Horsburgh, J.S., Tarboton, D.G., Maidment, D.R., Zaslavsky, I., 2011. Components of an environmental observatory information system. Comput. Geosci. 37, 207–218. https://doi.org/10.1016/j.cageo.2010.07.003

Horsburgh, J.S., Tarboton, D.G., Maidment, D.R., Zaslavsky, I., 2008. A relational model for environmental and water resources data. Water Resour. Res. 44. https://doi.org/10.1029/2007wr006392

Horsburgh, J.S., Tarboton, D.G., Piasecki, M., Maidment, D.R., Zaslavsky, I., Valentine, D., Whitenack, T., 2009. An integrated system for publishing environmental observations data. Environ. Model. Softw. 24, 879–888. https://doi.org/10.1016/j.envsoft.2009.01.002

Horsburgh, J.S., Reeder, S.L., Jones, A.S., Meline, J., 2015. Open source software for visualization and quality control of continuous hydrologic and water quality sensor data. Environ. Model. Softw. 70, 32–44. https://doi.org/10.1016/j.envsoft.2015.04.002

Jones, A.S., Aanderud, Z.T., Horsburgh, J.S., Eiriksson, D.P., Dastrup, D., Cox, C., Jones, S.B., Bowling, D.R., Carlisle, J., Carling, G.T., Baker, M.A., 2017. Designing and Implementing a Network for Sensing Water Quality and Hydrology across Mountain to Urban Transitions. J. Am. Water Resour. Assoc. https://doi.org/10.1111/1752-1688.12557

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

Jones, A.S., Horsburgh, J.S., Reeder, S.L., Ramírez, M., Caraballo, J., 2015. A data management and publication workflow for a large-scale, heterogeneous sensor network. Environ. Monit. Assess. 187, 348. https://doi.org/10.1007/s10661-015-

4594-3

Lane, B., Garousi-Nejad, I., Gallagher, M.A., Tarboton, D.G., Habib, E., 2021. An open web-based module developed to advance data-driven hydrologic process learning. Hydrol. Process. 1–15. https://doi.org/10.1002/hyp.14273

Leigh, C., Alsibai, O., Hyndman, R.J., Kandanaarachchi, S., King, O.C., McGree, J.M., Neelamraju, C., Strauss, J., Talagala, P.D., Turner, R.S., Mengersen, K., Peterson, E.E., 2018. A framework for automated anomaly detection in high frequency wate r-quality data from in situ sensors. Sci. Total Environ. 664, 885–898. https://doi.org/10.1016/j.scitotenv.2019.02.085

Maggioni, V., Girotto, M., Habib, E., Gallagher, M.A., 2020. Building an online learning module for satellite remote sensing applications in hydrologic science. Remote Sens. 12, 1–16. https://doi.org/10.3390/RS12183009

Makropoulos, C., 2019. Urban Hydroinformatics: Past, Present and Future. Water 11. https://doi.org/https://doi.org/10.3390/w11101959

Mason, S.J.K., Cleveland, S.B., Llovet, P., Izurieta, C., Poole, G.C., 2014. A centralized tool for managing, archiving, and serving point-in-time data in ecological research laboratories. Environ. Model. Softw. 51, 59–69. https://doi.org/10.1016/j.envsoft.2013.09.008

McGovern, A., Allen, J., 2021. Training the Next Generation of Physical Data Scientists. Eos (Washington. DC). 102, 1–9. https://doi.org/10.1029/2021EO210536

McGuire, M.P., Roberge, M.C., Lian, J., 2016. Channeling the water data deluge: a system for flexible integration and analysis of hydrologic data. Int. J. Digit. Earth 9, 272–299. https://doi.org/10.1080/17538947.2015.1031715

Merwade, V., Ruddell, B.L., 2012. Moving university hydrology education forward with community-based geoinformatics, data and modeling resources. Hydrol. Earth Syst. Sci. 16, 2393–2404. https://doi.org/10.5194/hess-16-2393-2012

Muste, M. V, Asce, M., Bennett, D.A., Secchi, S., Schnoor, J.L., Kusiak, A., Arnold, N.J., Mishra, S.K., Asce, S.M., Ding, D., Rapolu, U., 2013. End-to-End Cyberinfrastructure for Decision-Making Support in Watershed Management 565–573. https://doi.org/10.1061/(ASCE)WR.1943-5452

Ngambeki, I., Thompson, S.E., Troch, P.A., Sivapalan, M., Evangelou, D., 2012. Engaging the students of today and preparing the catchment hydrologists of tomorrow: student-centered approaches in hydrology education. Hydrol. Earth Syst. Sci. Discuss. 9, 707–740. https://doi.org/10.5194/hessd-9-707-2012

Pellerin, B.A., Stauffer, B.A., Young, D.A., Sullivan, D.J., Bricker, S.B., Walbridge, M.R., Clyde, G.A., Shaw, D.M., 2016. Emerging Tools for Continuous Nutrient Monitoring Networks: Sensors Advancing Science and Water Resources Protection. JAWRA J. Am. Water Resour. Assoc. 20460, 1–16. https://doi.org/10.1111/1752-1688.12386

Rapanta, C., Botturi, L., Goodyear, P., Guàrdia, L., Koole, M., 2021. Balancing Technology, Pedagogy and the New Normal: Post-pandemic Challenges for Higher Education. Postdigital Sci. Educ. 3, 715–742. https://doi.org/10.1007/s42438-021-00249-1

Rode, M., Wade, A.J., Cohen, M.J., Hensley, R.T., Bowes, M.J., Kirchner, J.W., Arhonditsis, G.B., Jordan, P., Kronvang, B., Halliday, S.J., Skeffington, R.A., Rozemeijer, J.C., Aubert, A.H., Rinke, K., Jomaa, S., 2016. Sensors in the Stream : The High-Frequency Wave of the Present. Environ. Sci. Technol. https://doi.org/10.1021/acs.est.6b02155

Samourkasidis, A., Papoutsoglou, E., Athanasiadis, I.N., 2019. A template framework for environmental timeseries data acquisition. Environ. Model. Softw. 117, 237–249. https://doi.org/10.1016/j.envsoft.2018.10.009

Shen, C., 2018. Deep Learning : A Next-Generation Big-Data Approach for Hydrology. Eos (Washington. DC). 1–4. https://doi.org/https://doi.org/10.1029/2018EO095649

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 Resour. Res. 55, 8547–8568. https://doi.org/10.1029/2019WR024906

Taylor, J.R., Loescher, H.L., 2013. Automated quality control methods for sensor data: a novel observatory approach. Biogeosciences 10, 4957–4971. https://doi.org/10.5194/bg-10-4957-2013

Vojinovic, Z., Abbott, M.B., 2017. Twenty-five years of hydroinformatics. Water (Switzerland) 9, 1–11. https://doi.org/10.3390/w9010059

White, D.L., Sharp, J.L., Eidson, G., Parab, S., Ali, F., Esswein, S., 2010. Real-Time Quality Control (QC) Processing, Notification, and Visualization Services, Supoorting Data Management of the Intelligent River, in: Proceedings of the 2010 South Carolina Water Resources Conference. p. 4.

CHAPTER 2

TOWARD AUTOMATING POST PROCESSING OF AQUATIC SENSOR DATA[1]

Abstract

Sensors measuring environmental phenomena at high frequency commonly report anomalies related to fouling, sensor drift and calibration, and datalogging and transmission issues. Suitability of data for analyses and decision making often depends on manual review and adjustment of data. Machine learning techniques have potential to automate identification and correction of anomalies, streamlining the quality control process. We explored approaches for automating anomaly detection and correction of aquatic sensor data for implementation in a Python package (pyhydroqc). We applied both classical and deep learning time series regression models that estimate values, identify anomalies based on dynamic thresholds, and offer correction estimates. Techniques were developed and performance assessed using data reviewed, corrected, and labeled by technicians in an aquatic monitoring use case. Auto-Regressive Integrated Moving Average (ARIMA) consistently performed best, and aggregating results from multiple models improved detection. pyhydroqc includes custom functions and a workflow for anomaly detection and correction.

2.1 Introduction

Observation of environmental phenomena using in situ sensors is increasingly common as sensors and related peripherals become more affordable and as cyberinfrastructure and expertise to support their operation have grown (Hart and

[1] Jones, A.S., Jones, T.L., Horsburgh, J.S., 2022. Toward Automating Post Processing of Aquatic Sensor Data. Environmental Modelling and Software 151, 105364. https://doi.org/10.1016/j.envsoft.2022.105364

Martinez, 2006; Pellerin et al., 2016; Rode et al., 2016). Sensors are subject to environmental factors that affect measurements and their suitability for subsequent analyses. Data from environmental sensors include anomalous points and biases that are artifacts of instrument noise or drift, power failures, transmission errors, or unusual ambient conditions (Horsburgh et al., 2015; Wagner et al., 2006). Protocols for ensuring quality of environmental sensor data (quality assurance) and mechanisms for performing data post processing (quality control) are challenges and key components of sensor network cyberinfrastructure (Campbell et al., 2013; Gries et al., 2014; Jones et al., 2015). As the quantity of sensor data increases, there is a commensurate need for practices that ensure resultant data are of high quality for subsequent analyses and exploration (Campbell et al., 2013; Gibert et al., 2016).

In current practice, quality control post processing of sensor data is expensive and tedious. Tools exist to assist practitioners and technicians in reviewing data and performing corrections (Gries et al., 2014; Horsburgh et al., 2015; Sheldon, 2008); however, quality control remains a time consuming and manual process consisting of an interactive sequence of steps. Performing corrections generally requires expert knowledge about the sensor and the phenomena being observed as well as conditions at the monitoring location (Fiebrich et al., 2010; White et al., 2010). Furthermore, the quality control process involves subjectivity as individual technicians may make different correction decisions (Jones et al., 2018). As a result, it is difficult to transfer the institutional knowledge required to post-process data, and even for trained and experienced technicians, quality control remains a daunting task as datasets grow in size and complexity for environmental observatories with ongoing data collection. For one

network, a substantial delay of approximately six months between data collection and availability of reviewed and processed datasets allowed for thorough review and correction (Jones et al., 2017). For cases where observations are used for real time decisions related to public health and water treatment, the impacts of anomalous data are costly.

As sensor datasets continue to grow, it is not tenable for scientists and technicians to manually perform quality control tasks (Gibert et al., 2018), neither is it advisable to use or publish data without performing corrections to mitigate for errors. As a result, there is a recognized need for automating and improving quality control post processing for high frequency in situ sensor data. In this vein, automated, data driven techniques to detect anomalies in streaming sensor data are documented in the realm of research (Hill and Minsker, 2010; Leigh et al., 2018; Russo et al., 2020; Talagala et al., 2019); however, they are unfamiliar to practitioners, generally lack robust and accessible software implementations, and are not typically reproducible. Furthermore, while basic checks and more complex algorithms may identify and flag potentially erroneous values (e.g., Dereszynski and Dietterich, 2007; Hill et al., 2009; Taylor and Loescher, 2013), these procedures are generally not capable of applying corrective actions. Thus, the specific questions we pursued with this research are: 1) how can data-driven methods be applied to automatically detect and correct anomalies in aquatic sensor data, and 2) how can these methods be packaged into an overall workflow and reusable software for general application?

Regression models are one class of data-driven techniques that can be used as anomaly detectors for time series data by making a prediction based on previous data

(either univariate or multivariate) and comparing the residual of the modeled and observed values to a threshold. Because regression models produce an estimate, they are well-suited for detection and correction of anomalous data. Although it is a substantial step in quality control post-processing, automated anomaly correction has not been widely examined. A handful of studies replaced raw data with modeled forecasts to exclude anomalies from model input but did not generate a corrected version of the dataset (Hill and Minsker, 2010; Leigh et al., 2018). In this work, we implemented and compared several regression models for anomaly detection and explored new approaches for anomaly correction.

Although effectively implemented for specific case studies, none of the techniques described in the cited studies have been packaged as accessible software for broad application and dissemination. Without reusable code, the specifics of the algorithms as implemented with environmental data cannot be examined, further tested, or applied to other datasets. Rather than a model calibrated to a specific variable/site combination, practitioners need tools that can be applied to a broad suite of variables and/or monitoring locations documented in a reusable and reproducible way. Thus, we sought to package the tools we developed as open-source software that could easily be deployed in a commonly available analytical environment.

In this paper, we present a Python package (pyhydroqc) that implements a set of methods for data- driven anomaly detection and correction for high temporal frequency aquatic sensor data. Our approach includes machine learning algorithms for detection, labeling, and correction of anomalous points. Multiple years of aquatic monitoring data from the Logan River Observatory (LRO) that have been reviewed and corrected by

trained technicians were used as a case study for developing and testing automated detection and correction methods. The algorithms are encapsulated in a Python package that is publicly available and open-source (see Software and Data Availability section). Example scripts are also shared as Jupyter Notebooks that can be run with case study data to demonstrate the functionality and performance of the tools we developed. As there are many potential approaches to anomaly detection, additional techniques can be incorporated by adding new functions to the package that can be incorporated to the workflow. Thus, the specific contributions of this work include: 1) advancing the algorithms and methods for automated quality control of aquatic sensor data, and 2) developing and demonstrating software tools that can make the process more approachable for data technicians and scientists.

Section 2.2 outlines the methods we implemented for detecting anomalies and performing corrections in the context of the structure and design of the pyhydroqc Python package, including a description of the case study that drove the implementation. In Section 2.3, we report the performance of the techniques on case study data and offer recommendations for next steps, followed by conclusions in Section 2.4. Appendix A contains related background including an overview of relevant literature and additional motivation for the work reported.

2.2 Methods

2.2.1 pyhydroqc Software Design and Implementation

This work implements methods for anomaly detection and correction for environmental time series data within a Python-based software package. A subset of data-driven regression models are situated within an overall workflow that includes practical

steps to facilitate anomaly detection and correction. The following sections describe the approaches for anomaly detection and correction, including details of how the software supports the workflow.

While many classes of algorithms could be used for detecting anomalies in aquatic sensor data, we selected time series regression models that were relatively straightforward to implement and that we anticipate will meet the needs and considerations of many applications. Specifically, we investigated auto-regressive integrated moving average (ARIMA), several types of long short-term memory (LSTM), and Facebook Prophet. ARIMA has been successfully implemented to detect anomalies in environmental data (Hill and Minsker, 2010; Leigh et al., 2018; Papacharalampous et al., 2019). LSTM is a class of Artificial Neural Networks (ANNs), and though applications to environmental data anomalies are limited, studies from other fields have detected anomalies with LSTM models (Hundman et al., 2018; Lindemann et al., 2019; Malhotra et al., 2016; Yin et al., 2020). Prophet was investigated but not included in the Python package. Because Prophet is geared toward social media and business applications (Taylor and Letham, 2018), we found that its applicability to environmental data is limited. It failed to capture seasonal shifts in the timing of daily cycles, and model features did not represent environmental phenomena. This paper focuses on a subset of models, but the modular design of the Python package allows for the implementation of additional techniques.

The software design and development were driven by the following steps as a workflow for anomaly detection and correction (Figure 2.1), and each is described in more detail in the sections that follow.

1. Import raw sensor data into a memory-resident data structure.
2. Perform rules-based anomaly detection and correction as a first pass at quality control, including addressing sensor calibration.
3. Build one or more models for predicting observed values:
    a. Determine model hyperparameters.
    b. Transform and scale data if necessary.
    c. Build and fit models.
    d. Execute the model to determine model predictions and residuals.
4. Post-process model results:
    a. Determine dynamic thresholds based on model residuals and user-defined parameters.
    b. Detect anomalies where the absolute value of the model residual exceeds the defined threshold.
    c. Widen and index anomalous events.
5. Compare technician labeled and detected anomalous events (rules-based and model- based detections, inclusive) to assign confusion matrix categories and report metrics. (This step is only applicable if labeled data are available.)
6. Combine detections identified by multiple models for an aggregate anomaly detection (if rules-based detection has been performed, those detections are included).
7. Perform model-based correction for points identified as anomalous.

In addition to performing the workflow steps, requirements that drove our design included: 1) open- source software development to facilitate deployment and use by

others; 2) cross-platform compatibility for use on Windows, MacOS, and Linux platforms; 3) modular and extensible architecture that enables each workflow step to be executed independently along with integration of new/additional functionality; and 4) simple deployment. A Python package was selected as the platform for software implementation. The Python language meets the open-source and cross- platform requirements, and existing tools and libraries in Python support steps in the workflow, including loading and manipulating large datasets and developing data-driven models. In a Python package, functions that comprise each step in the workflow can be called by scripts in a modular manner. Each of the steps can be performed independently, facilitating flexibility in use. A Python package also supports extensibility as new functions can be added without impacting existing functionality. Finally, Python packages can be published to the Python Package Index (PyPI, https://pypi.org/) making deployment straightforward and ensuring that algorithms can be applied in any Python coding environment.

The anomaly detection and correction workflow steps are encapsulated by functions in the pyhydroqc Python package described in the following sections. High level workflow wrapper functions ('ARIMA_detect', 'LSTM_univar_detect', and 'LSTM_multivar_detect') call more granular functions specific to each data and model type to perform steps 2-7 (Figure 2.1) and generate objects of the 'ModelWorkflow' class. For clarity, each function is named and described in this paper; however, most users will use the overarching workflow function calls. Example Python scripts and Jupyter Notebooks (see Software Availability section) illustrate how the workflow functions are implemented for the data use case described in this paper. A full list of

functions with inputs and outputs is found in Appendix B and with the package documentation.

2.2.1.1 Data Format and Import

pyhydroqc operates on pandas data frames, which are high performance, two-dimensional, tabular data structures for representing data in memory (pandas Development Team, 2008). Data frames can be created and saved or output as comma separated values (CSV) files. For pyhydroqc to perform anomaly detection and correction, input data need to be formatted as a data frame for each variable of interest indexed by date/time with a column of raw data. If technician labels or corrections are available, they are included as additional columns in the data frame. Technician labels are only needed for determining anomaly detection metrics.

It is common to report environmental sensor data as one table or file with a single date/time column and multiple columns of measurements – one for each sensor output. For flat files with this structure, the pyhydroqc 'get_data' function wraps the 'read_csv' function from the pandas library to import data into Python and parse into separate pandas data frames for each variable as required by the anomaly detection and correction functions.

2.2.1.2 Rules Based Detection and Correction

Rules-based detection is an important precursor to detection using models (Leigh et al., 2018; Taylor and Loescher, 2013), and the results of this step contribute to the overall set of detected anomalies. Whether a result of sensor failure or another cause, some anomalies are "low hanging fruit" that can be detected by rules-based preprocessing that performs a first pass of the data. Preprocessing the data is motivated, in part, by the

need to train models on a dataset absent of extreme outliers or artifacts that models cannot capture. By first applying rules-based anomaly detection and correction, a first degree of correction is made for subsequent input into data driven models. We created Python functions with basic rules to detect and correct out of range and persistent data. Furthermore, some aquatic sensors commonly exhibit drift, which requires sensor calibration and subsequent data correction. Because calibration shift and the preceding drift are subtle and difficult for any type of model to detect, we developed a rules-based routine that attempts to identify and these events. Basic correction methods for these anomaly types were also implemented as Python functions.

2.2.1.2.1 Range and Persistence Checks

The function ‘range_check’ adds a column to the data frame and populates it with an anomalous label if the observation is outside of user defined thresholds or a valid label if it is within the thresholds. Ranges should be determined specific to each sensor based on physics and the environment in which the sensor is deployed and can be refined based on site specific patterns. Data persistence refers to instances where the same value is repeated by a sensor, which is unlikely in natural systems, although sensors may report repeated values due to limitations in resolution. For the ‘persistence’ function, the user defines a minimum duration of repeated values for data to be considered anomalous. If repeated values exceed that duration, the points are classified as anomalous by populating the column from the ‘range_check’ function. Beyond these basic checks, additional rules of increasing complexity could be added to the pyhydroqc package and the anomaly detection workflow. Examples include ranges that vary seasonally, rate of change checks, and differencing checks.

Once anomalous points are identified by the Python functions that implement these rules, labels are carried through to the model-based detection steps. Labeled points are omitted from model training, either by logical exclusion, or, for models requiring an unbroken time series for training, by interpolating between valid points. Linear interpolation is performed (using the 'interpolate' function) over the entire time series as a preliminary correction step so that model input is more valid. If the complete workflow is followed, values initially corrected using linear interpolation are replaced by the model-based correction described in Section 2.2.1.9.

2.2.1.2.2 Calibration and Drift Correction

Environmental sensors commonly drift, and many aquatic sensors (specific conductance, pH, dissolved oxygen) require regular calibration to known standards to minimize drift. Drift causes a gradual increasing or decreasing trend separate from daily and seasonal patterns, and a calibration event manifests as a localized shift that corrects subsequent data up or down. These trends and shifts can be subtle and difficult to identify without a detailed record of calibration dates. In preliminary work, the model-based detectors described in subsequent sections were unable to consistently identify these data patterns. Detected shifts due to calibration events were undiscernible from other localized anomalies. Thus, it is important to address calibration events early in the quality control process because it is preferable that model-based detectors be trained on data that are free from drift.

For calibration and drift correction, we implemented functions to mimic a typical manual workflow. Performing post-processing correction for drift and calibration involves review of data, comparison of field records to data shifts to identify points

corresponding to calibrations, and application of a drift correction that uses start and end points and the gap of the calibration shift to retroactively correct data between two calibrations. In our experience, calibration events are typically reviewed and corrected one at a time.

While recognizing the difficulty of definitively identifying calibration events in an automated way, we designed functions for detection (functions ‘calib_edge_detect’, ‘calib_detect’, ‘calib_overlap’) and correction (functions ‘find_gap’, ‘lin_drift_cor’) of data affected by drift and calibration. The algorithms take advantage of characteristics of calibration events, specifically that events only occur during certain hours of the day, they may involve a shift in observed data, and that when returned to the water, sensors may report the same values for several time steps until the sensor stabilizes. Two separate approaches identify calibration events: 1) where there is a discernable shift in the data, or 2) persistence occurs over a limited window of points. Both are restricted to hours and days when technicians would be in the field.

Given dates of calibration, a gap value needs to be specified for correcting past data. A function ‘find_gap’ identifies the greatest shift for a given window of time to determine a gap value and the precise point that should be shifted. The function accounts for outlier spikes that are commonly associated with calibrations. A function for linear drift correction, ‘lin_drift_cor’, corrects for drift and calibration events given start and end dates for the period to be corrected and a gap value of the calibration shift. A list of calibration start and end times and gap values can be input to the linear drift correction function to correct multiple instances of drift and calibration. While the calibration event detectors may not adequately identify events, requiring technician review or input, this

process is a step toward automation as it evaluates gap values according to a set of rules rather than arbitrary determination by technicians (as illustrated in Jones et al. (2018)) and allows for bulk correction of calibration events.

2.2.1.3 Model-Based Detection Using ARIMA

ARIMA is a time series forecasting model where inputs correspond to past time steps of the variable of interest, and the output is a predicted value for that variable at the next time step. ARIMA uses three parameters to define a linear model (Equation 1):

$$y_t = \sum_{i=1}^{p} \varphi_i y_{t-\mathrm{i}} + \varepsilon_t - \sum_{i=1}^{q} \theta_i \varepsilon_{t-\mathrm{i}} \quad (1)$$

where y_t is the model output or the prediction for time step t, p is the number of previous points in the series to be used in the model, q is the number of moving average terms to include, φ_i are the fitted coefficients for auto-regression, θ_i are fitted model coefficients for the moving average, and ε_t is the moving average error term. Not shown in the equation is the term d, which is the order of differencing applied to the data y before this equation is evaluated. The parameters (p, d, q) can be determined manually or automatically. Manual parameter determination involves time series decomposition and the review of auto-correlation plots, which is tedious for numerous data series. Automatic determination of the parameters is effective but can be computationally demanding. pyhydroqc includes a function 'pdq' for automated determination using the pmdarima package (Smith, 2017). Given (p, d, q), model training involves determining the values of the coefficients for the terms in the linear equation (φ_i and θ_i) based on actual data.

In pyhydroqc, the function 'build_arima_model' constructs and trains an ARIMA model given input time series data and input parameters (p, d, q). It relies on the sarimax function from the statsmodel package (Seabold and Perktold, 2010) to fit an ARIMA

model (based on Equation 1), make model predictions for each time step, and compare predictions to observations. Input data should be free from gaps, so the anomaly detection workflow uses output of the rules-based detection with linear interpolation of any identified anomalies as input for ARIMA modeling. Scaling and transforming data are not necessary, so data are kept in the original units.

2.2.1.4 Model-Based Detection Using LSTM

LSTM is a type of neural network model architecture specifically designed for time-dependent and sequenced data. LSTM models consist of recurrent "cells" or units, each corresponding to one time step. A cell uses "gates" to control the flow of information in and out of the cell and how much of the past data that the cell "remembers" for computing output. To train an LSTM model, the weights of the connections within and between the gates are iteratively refined based on training data.

There are many variations of LSTM architecture (Greff et al., 2017), and layers of LSTM can be stacked. For our implementation, we compared several LSTM model types that are appropriate to time series data modeling for anomaly detection: vanilla and bidirectional, univariate and multivariate. In contrast with other neural network architectures, for which many layers are advised for fitting data, more shallow LSTM have been used because of the internal complexity of LSTM cells (Géron, 2017; Greff et al., 2017; Hundman et al., 2018). Other model types could be constructed, model layers and complexity could be added, and the input parameters could be tuned to each time series. Parameters can be defined by users and can be adjusted to investigate sensitivity, and we describe our approach for parameter selection in Section 2.3.1.4. The objective of this work was not to achieve the best time series model, but rather to detect anomalies, so

fine-tuning models was not required or pursued. Instead, comparisons were made between a few basic LSTM variations with the same parameter settings.

As mentioned, pyhydroqc workflow functions call multiple lower-level functions. For LSTM models, each type is implemented within the workflow function by an associated model wrapper function ('LSTM_univar', 'LSTM_multivar', 'LSTM_univar_bidir', 'LSTM_multivar_bidir'), which calls functions specific to that model type for preprocessing, model building, model training, and model evaluation (shown in Figure 2.1 and described in the Jupyter Notebook example script). The model wrappers return objects of the class 'LSTMModelContainer,' containing model predictions and residuals for each time step, similar to the output of 'build_arima_model.'

2.2.1.4.1 Vanilla and Bidirectional LSTM

pyhydroqc implements the "vanilla" type of LSTM model (Greff et al., 2017), which consists of a single layer LSTM in a sequence-to-one manner, i.e. the model returns a single output based on a sequence of inputs. Given a user-specified number of past time steps, the model output is a single value for the next point in time. "Bidirectional" LSTM models use observations both before and after the point of interest to provide information for model prediction, which is appropriate if immediate, real-time anomaly detection is not a requirement. By encoding a vanilla LSTM model with a bidirectional wrapper, input data are traversed both forward and backward in sequence, and model output is the value to have occurred in the middle of the sequence. In pyhydroqc, parallel functions structure input data to contain a user specified number of time steps prior to the point of interest for vanilla LSTM and prior to and following the point of interest for bidirectional LSTM (functions further described in Section 2.2.1.5.3).

2.2.1.4.2 Univariate and Multivariate LSTM

Either univariate or multivariate input data may be used for vanilla and bidirectional LSTM through the LSTM workflow functions and model wrapper functions. The workflow functions ('LSTM_detect_univar' and 'LSTM_detect_multivar') prepare data and report results for univariate or multivariate data and call the associated model wrapper functions ('LSTM_univar' and 'LSTM_univar_bidir' for univariate, 'LSTM_multivar_bidir' and LSTM_multivar' for multivariate). For multivariate data, the models use data for all observed variables as input and output estimates of the same variables for the point of interest. Model errors are examined for each variable, and independent thresholds are set for anomaly detection.

2.2.1.4.3 LSTM Preprocessing, Model Building, and Training

The functions for preprocessing, model building, and model training are compiled as sequenced steps in the LSTM model wrapper functions (Figure 2.1). Preprocessing for LSTM models involves scaling, reshaping, and ensuring that training data are valid, which is facilitated by using the output of the rules based detection. Data must be scaled so that extreme values do not have an outsized impact on the model, and pyhydroqc includes a function for scaling ('create_scaler') based on the standardscaler function from the scikitlearn package, which subtracts the mean and divides by the standard deviation to scale the data (Pedregosa et al., 2011). Reshaping data creates a sequence of immediately previous points (i.e., model input) for each data value (i.e., model output). pyhydroqc functions ('create_sequenced_dataset' and 'create_bidir_sequenced_dataset') reshape data based on a user defined number of past time steps.

To build a model structure, the pyhydroqc functions 'create_vanilla_model' and

‘create_bidir_model’ use the Sequential model from the Keras package (Keras Development Team, n.d.) with model layers (LSTM, Dense, and Bidirectional) and the suite of user-specified hyperparameters accepted by the Sequential model. To train the model, the functions ‘create_training_dataset’ and ‘create_bidir_training_dataset’ select a subset of data based on a user defined number of random points, ensuring that none were identified as anomalous by the rules-based detection. These points are reshaped and used for training the LSTM model. The function ‘train_model’ uses the Keras early stopping feature so that model training ceases when the error of the test and validation sets (randomly selected by the algorithm) are approximately equal.

2.2.1.5 Post Processing: Dynamic Threshold Determination and Anomaly Detection

A key component of model-based anomaly detection using regression approaches is determination of the threshold that regulates whether a point is marked as anomalous or valid. Aquatic data vary seasonally, daily, and with environmental events, changes that may not be adequately captured by a model. A dynamic threshold has the potential to improve detection accuracy by applying a narrower range (i.e., higher sensitivity) when the model predictions are more precise and a wider range when model predictions are more variable. In particular, by using a dynamic threshold, we hoped to identify localized outliers that are within the absolute expected range of values but are relatively distinct for a narrower time window and which were undetectable with a constant threshold.

pyhydroqc implements a dynamic threshold following the format of confidence intervals and prediction intervals used in other studies (Hundman et al., 2018; Leigh et al., 2018). For each data point, a threshold is determined based on a moving window of points (Equation 2):

$$T = \begin{cases} \mu \pm z_{\alpha/2}\sigma, & if\ z_{\alpha/2}\sigma < min \\ \mu \pm min, & otherwise \end{cases} \quad (2)$$

where *T* is the threshold, μ is the mean of the user defined moving window model residuals, σ is the standard deviation of the moving window model residuals, α is a user defined value to adjust the width of the threshold, $z_{\alpha/2}$ is the *α/2* quantile of a normal distribution, and *min* is a user defined parameter for the minimum threshold value. Note that *min* may be set to zero (having no effect) or to a non-zero value to prevent too many false positives - i.e., detections that are not anomalies. This can occur when model residuals are low over an extended period and the dynamic threshold is smaller than the resolution or uncertainty inherent in the sensor.

Given a time series of model residuals, the 'set_dynamic_threshold' function in pyhydroqc determines upper and lower thresholds for each point in a series using Equation 2 with a user defined moving window – the number of points used to calculate μ and σ. The 'detect_anomalies' function then compares the dynamic threshold values to the residuals for each time step to determine whether a point is anomalous. Assuming rules based detection was performed, the anomalies detected in that step are propagated through the workflow and are included in the detections output by this step.

Because anomalies are sparse relative to the total number of data points, the datasets are considered imbalanced (Chandola et al., 2009). Counts of true negatives are overwhelming, resulting in high accuracy, which may make it difficult to compare between models (Tan et al., 2019). As a result, anomaly detection focuses on true positives, false positives, and false negatives. Anomaly detection requires a balance between increasing true positives while reducing both false negatives and false positives, objectives that may be mutually exclusive and depend on model sensitivity. Our preferred

approach is to err on the side of sensitivity in the detector to minimize false negatives (along with maximizing true positives) even at the expense of increased false positives. Automating post processing reduces the overall number of data points to be reviewed so that, even with some false positives, review of detected anomalies by a technician will still be faster than a manual review of the entire dataset. The F2 score supports this aim by more heavily weighting false negatives while the F1 score equally weights true negatives and false negatives (Cook et al., 2020).

2.2.1.6 Post Processing: Anomaly Events and Widening

In comparing anomalies identified by the model-based detectors to anomalies labeled by technicians, we observed mismatches related to resolution and lags in model approximations related to model smoothing. When an anomaly is identified, either the technician or the algorithm must determine how many points to label. To address this in a systematic way, pyhydroqc generalizes anomalies into numbered "events" consisting of groups of anomalous points. By widening the detection window to include points before and after anomalies detected by the algorithm as well as points labeled by the technician, overlap between the two is more likely. In pyhydroqc, the 'anomaly_events' function groups contiguous anomalous points as events by adding a column to the data frame with incrementing numbers as an index for each anomalous event. To perform widening for each anomalous event, the function assigns the event's index to points before and after the event (the number of points is user defined), effectively adding those points to the event.

2.2.1.7 Performance Metrics

For data with technician labels, the function 'compare_events' determines valid

and invalid detections by comparing events detected by the algorithm to those labeled by the technician. Each point is classified as true positive, true negative, false positive, or false negative. When there is any overlap between detected events and labeled events (i.e., any portion of a labeled event is detected), all points are classed as true positives to indicate that the labeled event was detected. For accuracy, the points assigned as anomalous on the edges of events by widening are removed from the event as part of this step.

A confusion matrix compares model classifications to actual data to evaluate overall performance by reporting total true positives, true negatives, false positives, and false negatives (Leigh et al., 2018; Tan et al., 2019). Additional metrics that are commonly reported include positive predictive value (precision), negative predictive value, accuracy, recall, and F scores (Li et al., 2017). In pyhydroqc, the function 'metrics' determines the performance metric outputs in Table 2.1. As aggregates of precision and recall, F scores combine true positives, false positives, and false negatives into a single assessment score to assess models (Cook et al., 2020). The F1 score gives equal weight to false positives and false negatives while the F2 score gives greater weight to false negatives. F scores range from 0 to 1, with 1 being the upper bound.

2.2.1.8 Aggregate Detections

In this paper, we tested and compared the performance of ARIMA and several LSTM models for anomaly detection. In applying multiple models, rather than select the single best performing model, a robust approach is to aggregate results so that a point identified by any of the models as anomalous is considered a detection. To address this, pyhydroqc includes a function 'aggregate_results' for combining anomalies detected by

the different model types into a single column of detected anomalies. Because rules-based detections are propagated through the workflow and are present in the detections associated with each model, the aggregation automatically includes the rules-based detections.

2.2.1.9 Model-Based Correction

A primary goal of this work was to suggest corrections for anomalous points, which is enabled by using time series regression methods for anomaly detection. While the model predictions used to determine anomalies could be simply substituted as corrections, the prevalence of consecutive anomalous points means that anomalous points would be used to determine corrections. To prevent this, correction models were implemented at a more granular scale. A function ‘generate_corrections’ was developed that implements piecewise ARIMA models using the following steps:

1. Given a data frame of observations with anomalies detected, assign consecutive points with either valid or anomalous labels to alternating groups. The function ‘group_bools’ adds a column populated with 0 for valid points and assigns each anomalous event a unique integer.
2. Ensure that sets of valid data points are large enough to generate forecast predictions. Where valid data points are in between anomalous points and the duration is too small to use as model input, the function ‘ARIMA_group’ merges them with previous and subsequent anomalous points into one anomalous group by resetting the group’s incrementing index.
3. For each anomalous group, beginning with the group of shortest duration and progressing in order of increasing duration, develop 2 ARIMA models: one based

on the preceding valid points and one based on subsequent valid points (using a specified maximum number of points for model development). Use the piecewise models to make forecasts and backcasts and blend them using the function 'xfade' to get a single correction estimate for each point in the anomalous group.

4. In the data frame, populate a new column with the correction estimates for points in anomalous groups and with the observations for the points in valid groups.

To blend the forecast and backcast, the values are weighted according to the proximity to each end point of the anomalous event, as shown in Equation 3, which is encoded in the function 'xfade':

$$y_k = A_k \frac{N-k}{N+1} + B_k \frac{k+1}{N+1} \quad (3)$$

where y_k is the correction estimate for each time step k in the anomalous group, N is the total number of data points in the anomalous group to be corrected ($k = 0 \ldots N\text{-}1$), and A_k and B_k are the ARIMA forecasted and backcasted values, respectively. Examples in Section 2.3.4 illustrate this concept. Because the ARIMA correction is based on points immediately proximate, instead of using the hyperparameters and model generated for the dataset as a whole, each forecast and backcast is an individual ARIMA model with hyperparameters and model fit based on the window of valid data. Using more granular models allows models to be tuned to that local time window and helps prevent errors that might arise from not having enough valid data points to estimate a point (e.g., if $p = 9$ for the time series as a whole, at least 9 valid data points are required). To avoid overfitting and to conserve computational resources, the 'generate_corrections' function includes a user defined limit on the duration of data used to develop and train piecewise models to generate the forecasts and backcasts.

Instead of applying corrections sequentially, the correction function first corrects the events of shortest length and then corrects events of increasing duration. In this manner, corrected estimates are available as model inputs when needed for correcting longer events. This helps ensure that the period of valid data before or after an anomalous event is sufficient to capture patterns.

2.2.2 Experimental Use Case: Logan River Observatory Data

The primary objective of this work was to advance automation of quality control post processing specifically for environmental sensor data. As an extensive test case, we used data collected within the LRO where high frequency monitoring is conducted at several climate and aquatic sites within the Logan River watershed, located in northern Utah, USA (http://lro.usu.edu, Neilson et al., 2021). Monitoring sites were established and infrastructure was originally deployed using protocols described by Jones et al. (2017). The LRO is similar to many research sites throughout the world where *in situ* monitoring of aquatic, climatic, and terrestrial variables is performed in support of research activities. Utah State University manages the monitoring network including site maintenance and data dissemination (available at http://lrodata.usu.edu/).

The upper Logan River watershed consists of mountainous forest and rangeland with limited development while the lower watershed is agricultural and urban with multiple agricultural diversions. Hydrology is generally driven by snowmelt, and the upper watershed is characterized by karst topography. Aquatic monitoring sites are located in both the upper mountain/canyon and lower urban/agricultural sections and include sensors for water level, water temperature, pH, dissolved oxygen, specific conductance, and turbidity (Figure 2.2).

2.3 Results and Discussion

2.3.1 Preprocessing and Settings

The following subsections present the parameters, configuration, and settings used by each anomaly detection and correction procedure. Anomalies detected by the combination of rules (range and persistence) and models with thresholds (ARIMA and LSTM) are reported together in Section 2.3.3.

2.3.1.1 Rules Based Detection and Correction: Range and Persistence Checks

For the LRO data, range thresholds were determined specific to each sensor based on manufacturer reported ranges and were further refined according to past observations at each site (Table 2.2). The maximum allowable persistence durations were also based on review of raw observations and varied with sensor. Initially, persistence durations were set lower (~5-10 time steps); however, those durations resulted in many false positives as sensors regularly reported repeated values for more than 10 time steps. We observed that repeated values are often caused by limitations in sensor resolution, so persistence durations were increased (Table 2.2). Anomalies detected by these functions retained labels through subsequent steps, so the metrics resulting from rules-based detection are reported with the overall anomaly detection results in Section 2.3.3.

Anomalies detected by the range and persistence checks were initially corrected by linear interpolation, which is identical to the LRO protocol used by technicians to manually correct over short periods. However, in the pyhydroqc anomaly detection and correction workflow, the linear interpolation correction is an intermediate step to facilitate more accurate model development. These points retain an anomalous label through subsequent steps of the workflow and are eventually corrected using the model

correction algorithm. Consequently, the final correction is performed by the model overwriting the interpolated points in the final, corrected dataset.

2.3.1.2 Rules-Based Detection and Correction: Calibration and Drift Correction

Results from the calibration detection algorithms were compared to calibration events identified and corrected by technicians for all sensors at one site (Main Street). The persistence functions (calib_detect and calib_overlap) identified about 25% of the calibration events with a high false positive rate (5X). The persistence we observed following a calibration may be specific to the sensors used in the LRO (YSI multiparameter sondes) and not broadly applicable. The edge detection function (calib_edge_detect) identified about 40% of calibrations for pH but was less successful (<10%) for specific conductance and dissolved oxygen. Additional effort could be applied to improve calibration event detection and to refine the parameters of the edge detection function (threshold and width). In theory, the model algorithms should identify these local shifts as anomalies; however, although the observed values may deviate from the modeled, the residuals were often within the dynamic thresholds (as defined in Table 2.2) and so were not detected as anomalies. Adjusting threshold settings may identify more calibration events but cause oversensitivity. Furthermore, the corrective action required for calibration events is different from that of other anomaly types, so the detection step should be separate.

Although calibration events were not automatically detected with high accuracy, the function for finding gap values was effective at determining valid gap values and end times for calibration shifts. In a review of the results of the ‘find_gap’ function, out of 100 distinct calibrations (the total for all variables at Main Street), revision was made for

only 6 instances. With calibration dates and gap values as inputs, the function for linear drift correction was executed for all calibrated sensors (specific conductance, pH, dissolved oxygen) for the Main Street site. Many of the automatically determined gap values approximated the values used by the technician for correction, in which case the linear drift correction was comparable to the technician correction. Some automatically determined values were judged as preferable to the technician selected gap value (e.g., Figure 2.3).

In our experience, selecting a viable gap value and performing drift correction can be the most time consuming aspect of manual quality control. So, although the algorithms we designed were not successful in identifying a majority of calibration events, technicians typically record the dates of calibration, and automatically determining the gap value and performing drift correction in batch is a significant improvement. Furthermore, using an algorithm for this step increases consistency – the range of gap values selected by multiple technicians was the primary source of quality control subjectivity identified by Jones et al. (2018).

Based on our testing using the LRO data, our recommended workflow for addressing drift and calibration events is to: 1) identify a list of calibration dates (generally from field notes, although the pyhydroqc functions may be useful); 2) determine gap values and associated times using the 'find_gap' function; 3) review those shifts and make any adjustments; and 4) use the dates and gap values as inputs to the linear drift correction function. Code for performing these steps including generating plots of gap values for review are demonstrated in example notebooks.

2.3.1.3 Model Based Detection and Correction: Threshold Determination

The dynamic threshold used to evaluate differences between simulated and observed values directly impacts which observations are detected as anomalous or valid. For the LRO data, we used trial and error to settle on window sizes, alpha values, and minimum range values for determining thresholds (Table 2.2). The same threshold settings were used for all model types. We found that moving windows longer than a single day resulted in too much smoothing to the threshold and introduced artifacts due to daily patterns in model residuals. In general, window sizes of 5-10 hours (corresponding to 20-40 time steps) were selected to balance between over-smoothing of longer windows and highly dynamic thresholds of shorter windows. An added benefit of smaller window sizes is that fewer computational resources are required to determine thresholds. Relatively small alpha values were selected (0.001-0.00001) to create a sufficiently high threshold range. With larger alpha values, the narrow threshold range was overly sensitive, resulting in too many false positives. Minimum values were similar for all sensors across sites, with a few exceptions. As illustrated in Figure 2.4, the pattern of spread in thresholds tracks with the variability in model residuals, and residuals that exceed the threshold are detected anomalies.

2.3.1.4 Model-Based Detection and Correction: Model Parameters and Settings

To create ARIMA models, (p, d, q) were determined for each LRO data series over the full duration of data (Table 2.2). To build, compile, and train LSTM models, consistent parameters and settings were used for all of the LRO data series and the several varieties of LSTM models (Table 2.3). Default settings and commonly used parameters (Géron, 2017; Keras Development Team, n.d.) were selected with minimal

tuning to achieve the goal of satisfactory rather than perfect models. Models were trained with 20,000 randomly selected data points from each data series, corresponding to approximately 10% of the points within each data series. Anomalous events in both technician-labeled data and model-detected data were widened by a single point (widening factor = 1). This setting was used for all data series and all model types.

2.3.2 Anomaly Detection Example

Examples help demonstrate the performance of the workflow for both successful and unsuccessful anomaly detection (Figure 2.5; additional examples in Appendix C). On 2018-11-11, the ARIMA model detected an event that was not labeled by the technician (false positive). Although this is a false positive, the model with a dynamic threshold behaved as designed in detecting a localized outlier. The events on 2018-11-12 and 2018-11-13 consist of points both detected by the algorithm and labeled by the technician (true positive). Not all points labeled by the technician were detected as anomalies by the model; however, performing widening and considering the overlapping sets of points as anomalous events resulted in true positives for all of these points. The event on 2018-11-14 was not detected by the algorithm but was labeled by the technician (false negative). There is nothing in the original data to indicate that something was amiss, so it is unclear why the points were labeled as anomalous by the technician. The technician has expert knowledge or is following protocol that the algorithm is unable to discern. In assessing algorithm performance, we defer to technician labels as a benchmark. However, the quality control process is subjective (Jones et al., 2018) and data are not perfectly labeled, making reliance on technician labels as a gold standard problematic (Russo et al., 2020). In the LRO data, we identified numerous cases where it was unclear why some data

points were labeled and others were not (see Appendix C), which may be due to multiple technicians and evolving protocols, among other reasons.

2.3.3 Combined Anomaly Detection Results

The F2 scores for all time series (Table 2.4) combine true positives, false positives, and false negatives to indicate overall performance for each model type, rules-based detection, and an aggregate of all models. Higher scores indicate better model performance (F2 = 1 would be a perfect score). Figure 2.6 is a visual illustration of the confusion matrix where each panel corresponds to a time series and each bar to a model type. The bottom portion of each bar (light blue) represents true positives, the middle portion (orange) represents false negatives, and the sum of those is equivalent to all technician labeled points. The top portion of each bar (purple) represents false positives. The dashed lines distinguish the proportion of anomalies identified by rules based detection. True positives below the lower dashed line (black) were detected by rules while those above it were only detected by models. Likewise, false positives below the upper dashed line (gray) were detected by rules, and false positives above it were detected by only models. Anomalies detected by rules (those below each line) may have also been detected by models, so there may be overlap. The results illustrate some general trends regarding the performance of both rules based and model based detection.

2.3.3.1 Detections Due to Rules and Threshold Settings

For several time series, the rules-based algorithm accounts for the majority of anomaly (true positive) detections (e.g., temperature at several sites, dissolved oxygen at Franklin Basin). In these cases, the model detection did not provide many additional detections. In other cases (e.g., temperature at Tony Grove, all pH time series, most

specific conductance and dissolved oxygen time series), the true positives are split between rules-based and model-based, indicating that the models capture anomalous events that the rules-based detection misses. This demonstrates the value of using both approaches in tandem.

In some cases, the success of the model(s) in detecting anomalies (true positives) is offset by a large number of false positives. Particularly high counts of false positives indicate oversensitivity, due to either persistence durations that are too short or to thresholds that are too tight, both of which may result in too many detections. In particular, dissolved oxygen at Franklin Basin and Mendon and specific conductance at Blacksmith Fork exhibit high rates of false positives. Given that most are under the rules-based line, the false positives are attributable to oversensitivity in rules (range check or persistence duration) rather than inadequate threshold settings. The similar rates of false positives between models for many time series indicates that using the same threshold settings for all model types is acceptable.

Cases with a large portion of false negatives (undetected anomalies) across models indicate that the models were not sensitive enough (e.g., temperature at Main Street and Blacksmith Fork). Better detection might occur with tighter thresholds or adjusted rules-based settings. Practitioners need to consider the tradeoffs with model sensitivity in determining threshold settings. Under the assumption that anomalies identified by the algorithm would be further reviewed by a technician, the thresholds can be set to capture more potential anomalies, erring on the side of false positives. However, sensitivity must be balanced to avoid excessive false positives from narrow thresholds.

2.3.3.2 Model Comparison

The detections between all models were generally comparable (e.g., temperature at most sites, pH at most sites, dissolved oxygen at several sites), although, for a few time series, there were distinct variations in results between models (e.g., specific conductance at Franklin Basin and Tony Grove, dissolved oxygen at Tony Grove). ARIMA models gave the best average F2 score (Table 2.4) – they generally outperformed LSTM models for the cases with differences in model performance and were often slightly better than the LSTM models for the time series with comparable results. ARIMA was generally more sensitive – detecting more true positives than the LSTM models at the expense of detecting more false positives. Results from the LSTM models varied without a discernable pattern. In one case, the univariate bidirectional model excelled (temperature at Main Street), while in other cases the multivariate vanilla was preferred (specific conductance at Tony Grove, dissolved oxygen at the Water Lab).

Differences in anomaly detection between the model types could be due to several factors. ARIMA and LSTM models have inherently different structures with distinct processes for hyperparameter tuning and model training. ARIMA models use a limited number of hyperparameters (three), which were tuned by automated optimization, while LSTM models include several hyperparameters for which minimal tuning was performed. It is possible that LSTM models could be improved with additional tuning; however, the process may not be worth the effort given that the objective of modeling was to detect anomalies rather than generate a perfect model. As one example, we observed LSTM models consistently biased toward the overall time series mean, which was reduced when developed with input sequences containing fewer previous data points (5 versus 10).

Another possible explanation for the poorer performance of LSTM models is a result of the training process. LSTM models were trained on a randomized subset of available data. Due to the stochastic nature of training data selection and initialization of weights, a new model is developed each time the algorithm is run (although pyhydroqc can save models for future use). If a distinct set of training data was used or learning converges to a local minimum, it may cause the seemingly arbitrary failure of some LSTM models on certain time series. To test this, LSTM models were regenerated. The resulting metrics were similar to those reported in Table 2.4. This indicates that the size of the training sets is sufficient so that the strength of the model does not depend on the specific, randomized subset of data used for training. Independently developing and training multiple models on the same time series is a straightforward check for training data robustness.

Although we tested across a range of sites that span elevation, land use, and hydrologic regime within the LRO, these locations do not represent the full spectrum of sites across the world. Investigating the suitability of the algorithm to additional physical settings is an important next step. More directly examining the performance of each model type related to physical characteristics of locations may help inform transferability of the techniques.

2.3.3.3 Model Aggregation

The comparability of most of the results suggests that using any one of the models may be acceptable; however, rather than select a single model, aggregating detections by the multiple models may improve results. F2 scores of aggregated anomaly detection (Table 2.4) indicate overall good performance for most time series (F2>0.8), also

illustrated by confusion matrix plots (Figure 2.7). For some time series, the aggregation does not add high value, presumably because the same points were detected by multiple models. However, for a few time series in particular, aggregating detections of multiple models had a synergistic effect such that the aggregate F2 score is higher than that of any single model (e.g., temperature at Main Street, dissolved oxygen at Tony Grove). Lower F2 scores (<0.8) that persist after aggregating model detections are a result of either high rates of false positives (dissolved oxygen at Franklin Basin) or false negatives (temperature at Blacksmith Fork), both of which could be addressed by tuning rules and threshold settings as described rather than perfecting models.

The results affirm that time series regression methods with dynamic thresholds and widening are an effective tool for automating anomaly detection and correction, and implementing these techniques can streamline the quality control process. Without the models, a technician would need to review 200,000+ data points for each of the time series used in this case study. By using the pyhydroqc anomaly detection workflow, the number of data points for review (referring to combined rules and model detections) is reduced by at least an order of magnitude (e.g., ~20,000 for pH at Franklin Basin), even for cases with high rates of false positives (e.g., ~4,000 for dissolved oxygen at Franklin Basin).

2.3.4 Model-Based Correction Examples

The model-based anomaly correction implemented in pyhydroqc generally resulted in smooth data profiles without outstanding nonlinearities (Figure 2.8). The method offers a viable path for correcting many anomalous events, although results varied depending on the duration, the variable, the season, and the reliability of anomaly

detection. For shorter durations (e.g., approximately 2 hours, Figure 2.8a), the model corrected data are similar to the technician correction (i.e., linear interpolation). For longer periods, the blended forecasts and backcasts can estimate patterns (diurnal cycles, Figure 2.8b and 2.8c) that would not be practical for a technician to approximate. In these cases, technicians did not attempt corrections but set data to a no data value (-9999). In other cases, the model did not capture data patterns, particularly for extended periods (see Appendix C for examples). Some models overgeneralized and missed patterns while others focused on a single dominant feature. Overall, the correction algorithm better captured diurnal patterns in temperature and pH data while regular patterns in specific conductance and dissolved oxygen were less consistently approximated.

2.3.5 Combined Correction Results

Quantifying the overall performance of the correction algorithm for each time series is impractical because no gold standard exists for comparison. Algorithm-corrected data cannot be quantitatively compared to technician corrected data because the technician corrected data are subjective, contain correction and labeling errors, and include many periods where the values were set to a designated “no data value” (e.g., -9999 for the LRO). For correcting LRO data, technicians followed one of the following paths: 1) linear interpolation for periods less than 4 hours, or 2) setting values to -9999 for longer periods where interpolation was deemed unreasonable. Technicians also performed linear drift correction between identified calibration events. The model-based correction algorithm is not designed to correct for drift, which was performed as part of the rules-based steps (Section 2.3.1.2).

Without a benchmark, correction algorithm performance cannot be definitively

measured for each time series, leaving evaluation to be done qualitatively on a case-by-case basis (Section 2.3.4 and Appendix C). We considered simulating artificially introduced anomalies, which are then corrected and compared to valid raw data; however, it is unclear what frequency and duration of artificial anomalies would be appropriate and how to propagate artificial anomalies through multiple concurrently measured variables (i.e., in the case of multivariate models). We determined that analysis to be outside the scope of this work. In an attempt to assess the value of the correction algorithm in terms of relative accuracy, we considered the total number of points in each series that were altered from the raw data by the technician or the algorithm and that were set to values outside of a valid range (Table 2.5). Ranges specific to each time series were adopted from the range checks in rules-based preprocessing (Table 2.2) to determine whether altered points were valid. Technician corrections resulting in invalid values generally correspond to data changed to the no data value of -9999. Causes of invalid values produced by the correction algorithm may include periods where anomaly detection was not adequately inclusive, so the points corrected by the algorithm were overly influenced by anomalous points that were not labeled as such (Figure C5). In another scenario, anomalous data may be close to the range limits resulting in forecasts, backcasts, and corrections outside of the valid range (e.g., the estimations of peaks in Figure 2.8b exceed the upper limit for that time series).

For most cases, the algorithm correction resulted in significantly fewer invalid values than the technician correction. For 16 out of 24 time series (most of the temperature, specific conductance, and pH series), the number of invalid points produced by the algorithm correction was less than 100 (out of 200,000+ total points) while the

number of invalid points produced by the technician was significantly higher (ranging from 22 to 8541). For five of the time series (primarily dissolved oxygen), the algorithm correction resulted in a higher number of invalid values. For some of these series, the anomaly detection was also less performant (e.g., dissolved oxygen at Franklin Basin, Tony Grove, and Mendon – Figure 2.7). These results highlight the need to review anomaly detections and refine settings to improve anomaly detection. Although the corrections classed as valid were within an acceptable range for that time series, the correction may not have approximated observed data patterns, so review of proposed algorithm corrections is necessary.

The overarching benefit of the correction in pyhydroqc is that the algorithm may capture diurnal patterns to suggest values that a technician could not estimate. However, anomalous events need review prior to correction, as do correction suggestions. Adjacent data may be inadequate to generate correction estimates for the full duration of an anomalous event. A more complete workflow could offer correction options for each anomalous event for review and selection by a technician.

2.4 Conclusions

We developed a new Python package, pyhydroqc, that enables application of rules based and time series regression techniques coupled with dynamic thresholds as part of a workflow to detect and correct anomalies in aquatic sensor data. Functions to implement the models and supporting steps in the workflow are contained in the Python package and documented within the GitHub repository. Available functions include rules-based anomaly detection, calibration detection and drift correction, model development and estimation, threshold determination, anomaly detection and widening, performance

metrics reporting, and model-based correction. Although this workflow advances the automation of sensor data post processing, a Python package and scripts may not be intuitive tools for some technicians. A graphical user interface offering more interactive review could be built on top of the underlying functionality contained in pyhydroqc.

Based on our case study of 24 time series from the LRO, the anomaly detection workflow enabled by pyhydroqc was successful with high detection rates. ARIMA models were more performant, likely due to differences in model structure and development. Rather than using constant thresholds, dynamic thresholds allowed for responsiveness to data variability. A correction algorithm used blended forecasts and backcasts of local models to make correction estimates that follow data patterns for events of up to several days for some observed variables. These approximations surpass a technician's ability to correct anomalous data, but each corrected event needs review. A rules-based approach was successful in determining calibration gap values and performing linear drift correction with calibration dates as input. Though not completely automated, this work helps to streamline the process of quality control related to sensor drift and calibration.

Manual detection and correction performed by technicians is an extended process that overlaps with other tasks. To perform quality control for 3-6 month durations of a single time series takes multiple days of dedicated effort. In comparison, implementing the complete pyhydroqc workflow for anomaly detection and correction for all variables at a single site for a single year of data takes a few hours to run in the background on a personal computer. A technician will still need to review results; however, we submit that the package and workflow offer significant resource savings.

Throughout this process, the technician was treated as an ‘oracle’ with technician labels dictating algorithm performance. The subjectivity inherent in manual quality control and uneven application of labels by technicians highlight the need for improving consistency in quality control, which is an important driver of automating post processing given that computers are not subjective in their decisions.

As the volume of environmental sensor data continues to increase, so does the need for performing post processing quality control. This work contributes tools and approaches that can be used to streamline and automate the quality control process to reduce the costs of manual quality control; facilitate a post processing workflow that is reproducible, defensible, and consistent; and provide reliable data for analysis and decision making.

2.5 Acknowledgments

This research was primarily funded by the United States National Science Foundation under grant number 1931297. Any opinions, findings, and conclusions or recommendations expressed are those of the authors and do not necessarily reflect the views of the National Science Foundation. Additional funding support was provided by the Utah Water Research Laboratory at Utah State University. Ongoing data collection and management in the LRO is supported by funding from the Utah State Legislature administered by the Utah Division of Water Resources. Additional funding for the LRO has been provided by Utah State University, Logan City, and the Cache Water District. We gratefully acknowledge the work of the many technicians and students who have participated in maintaining and operating the LRO instrumentation along with producing the quality-controlled sensor data that were used in testing the software we developed.

REFERENCES

Ahmad, S., Lavin, A., Purdy, S., Agha, Z., 2017. Unsupervised real-time anomaly detection for streaming data. Neurocomputing 262, 134–147. https://doi.org/10.1016/j.neucom.2017.04.070

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, Wade, M., Boose, E.R., 2013. Quantity is Nothing without Quality. Bioscience 63, 574–585. https://doi.org/10.1525/bio.2013.63.7.10

Chandola, V., Banerjee, A., Kumar, V., 2009. Survey of Anomaly Detection. ACM Comput. Surv. 41, 1–72. https://doi.org/10.1145/1541880.1541882

Christ, M., Braun, N., Neuffer, J., Kempa-Liehr, A.W., 2018. Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package). Neurocomputing 307, 72–77. https://doi.org/10.1016/j.neucom.2018.03.067

Conde, E.F., 2011. Environmental Sensor Anomaly Detection Using Learning Machines. Learning. Utah State University.

Cook, A., Misirli, G., Fan, Z., 2020. Anomaly Detection for IoT Time-Series Data: A Survey. IEEE Internet Things J. 1–1. https://doi.org/10.1109/jiot.2019.2958185

Dereszynski, E.W., Dietterich, T.G., 2007. Probabilistic Models for Anomaly Detection in Remote Sensor Data Streams, in: Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence (UAI2007). pp. 75–82.

Fiebrich, C.A., Morgan, C.R., McCombs, A.G., Hall, P.K., Mcpherson, R. a., Morgan, Y.R., McCombs, A.G., Hall, P.K., Mcpherson, R. a., 2010. Quality assurance procedures for mesoscale meteorological data. J. Atmos. Ocean. Technol. 27, 1565–1582. https://doi.org/10.1175/2010JTECHA1433.1

Galarus, D., Angryk, R., Sheppard, J., 2012. Automated weather sensor quality control. Proc. 25th Int. Florida Artif. Intell. Res. Soc. Conf. FLAIRS-25 388–393.

Géron, A., 2017. No hand-On Machine Learning with Scikit-Learn & TensorFlow. O'Reilly.

Gibert, K., Horsburgh, J.S., Athanasiadis, I.N., Holmes, G., 2018. Environmental Data Science. Environ. Model. Softw. 106, 4–12. https://doi.org/10.1016/j.envsoft.2018.04.005

Gibert, K., Sànchez-Marrè, M., Izquierdo, J., 2016. A survey on pre-processing techniques: Relevant issues in the context of environmental data mining. AI Commun. 29, 627–663. https://doi.org/10.3233/AIC-160710

Giustarini, L., Parisot, O., Ghoniem, M., Hostache, R., Trebs, I., Otjacques, B., 2016. A user-driven case-based reasoning tool for infilling missing values in daily mean river flow records. Environ. Model. Softw. 82, 308–320. https://doi.org/10.1016/j.envsoft.2016.04.013

Greff, K., Srivastava, R.K., Koutnik, J., Steunebrink, B.R., Schmidhuber, J., 2017. LSTM: A Search Space Odyssey. IEEE Trans. Neural Networks Learn. Syst. 28, 2222–2232. https://doi.org/10.1109/TNNLS.2016.2582924

Gries, C., Henshaw, D., Brown, R.F., Cary, R., Downing, J., Jones, C., Kennedy, A., Laney, C.M., Martin, M., Morse, J., Porter, J., Read, J.S., Rettig, A., Sheldon, W., Strachan, S., Zdravkovic, B., 2014. Sensor and sensor data management best practices released. LTER Databits Spring 201.

Hart, J.K., Martinez, K., 2006. Environmental Sensor Networks: A revolution in the earth system science? Earth-Science Rev. 78, 177–191. https://doi.org/10.1016/j.earscirev.2006.05.001

Hill, D.J., Minsker, B.S., 2010. Anomaly detection in streaming environmental sensor data: A data-driven modeling approach. Environ. Model. Softw. 25, 1014–1022. https://doi.org/10.1016/j.envsoft.2009.08.010

Hill, D.J., Minsker, B.S., 2006. Automated fault detection for in-situ environmental sensors, in: 7th International Conference on Hydroinformatics.

Hill, D.J., Minsker, B.S., Amir, E., 2009. Real-time Bayesian anomaly detection in streaming environmental data. Water Resour. Res. 45, 1–16. https://doi.org/10.1029/2008WR006956

Horsburgh, J.S., Tarboton, D.G., Maidment, D.R., Zaslavsky, I., 2008. A relational model for environmental and water resources data. Water Resour. Res. 44. https://doi.org/10.1029/2007wr006392

Horsburgh, J.S., Reeder, S.L., Jones, A.S., Meline, J., 2015. Open source software for visualization and quality control of continuous hydrologic and water quality sensor data. Environ. Model. Softw. 70, 32–44. https://doi.org/10.1016/j.envsoft.2015.04.002

Hundman, K., Constantinou, V., Laporte, C., Colwell, I., Soderstrom, T., 2018. Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding. Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. 387–395. https://doi.org/10.1145/3219819.3219845

Jones, A.S., Aanderud, Z.T., Horsburgh, J.S., Eiriksson, D.P., Dastrup, D., Cox, C., Jones, S.B., Bowling, D.R., Carlisle, J., Carling, G.T., Baker, M.A., 2017. Designing and Implementing a Network for Sensing Water Quality and Hydrology across Mountain to Urban Transitions. J. Am. Water Resour. Assoc. https://doi.org/10.1111/1752-1688.12557

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

Jones, A.S., Horsburgh, J.S., Reeder, S.L., Ramírez, M., Caraballo, J., 2015. A data management and publication workflow for a large-scale, heterogeneous sensor

network. Environ. Monit. Assess. 187, 348. https://doi.org/10.1007/s10661-015-4594-3

Keras Development Team, n.d. Keras [WWW Document]. URL https://keras.io/about/ (accessed 2.5.21).

Leigh, C., Alsibai, O., Hyndman, R.J., Kandanaarachchi, S., King, O.C., McGree, J.M., Neelamraju, C., Strauss, J., Talagala, P.D., Turner, R.S., Mengersen, K., Peterson, E.E., 2018. A framework for automated anomaly detection in high frequency wate r-quality data from in situ sensors. Sci. Total Environ. 664, 885–898. https://doi.org/10.1016/j.scitotenv.2019.02.085

Li, J., Pedrycz, W., Jamal, I., 2017. Multivariate time series anomaly detection: A framework of Hidden Markov Models. Appl. Soft Comput. J. 60, 229–240. https://doi.org/10.1016/j.asoc.2017.06.035

Lindemann, B., Fesenmayr, F., Jazdi, N., Weyrich, M., 2019. Anomaly detection in discrete manufacturing using self-learning approaches. Procedia CIRP 79, 313–318. https://doi.org/10.1016/j.procir.2019.02.073

Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., Shroff, G., 2016. LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection.

Moatar, F., Miquel, J., Poirel, A., 2001. A quality-control method for physical and chemical monitoring data. Application to dissolved oxygen levels in the river Loire (France). J. Hydrol. 252, 25–36. https://doi.org/10.1016/S0022-1694(01)00439-5

Mourad, M., Bertrand-Krajewski, J.-L., 2002. A method for automatic validation of long time series of data in urban hydrology. Water Sci. Technol. 45, 263 LP – 270.

Pandas Development Team, 2008. Pandas Documentation [WWW Document]. URL https://pandas.pydata.org/docs/ (accessed 2.5.21).

Papacharalampous, G., Tyralis, H., Koutsoyiannis, D., 2019. Comparison of stochastic and machine learning methods for multi-step ahead forecasting of hydrological processes, Stochastic Environmental Research and Risk Assessment. Springer Berlin Heidelberg. https://doi.org/10.1007/s00477-018-1638-6

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine Learning in Python. J. Mach. Learn. Res. 12, 2825–2830.

Pellerin, B.A., Stauffer, B.A., Young, D.A., Sullivan, D.J., Bricker, S.B., Walbridge, M.R., Clyde, G.A., Shaw, D.M., 2016. Emerging Tools for Continuous Nutrient Monitoring Networks: Sensors Advancing Science and Water Resources Protection. JAWRA J. Am. Water Resour. Assoc. 20460, 1–16. https://doi.org/10.1111/1752-1688.12386

Rode, M., Wade, A.J., Cohen, M.J., Hensley, R.T., Bowes, M.J., Kirchner, J.W., Arhonditsis, G.B., Jordan, P., Kronvang, B., Halliday, S.J., Skeffington, R.A.,

Rozemeijer, J.C., Aubert, A.H., Rinke, K., Jomaa, S., 2016. Sensors in the Stream : The High-Frequency Wave of the Present. Environ. Sci. Technol. https://doi.org/10.1021/acs.est.6b02155

Russo, S., Lürig, M., Hao, W., Matthews, B., Villez, K., 2020. Active Learning for Anomaly Detection in Environmental Data. Environ. Model. Softw. https://doi.org/10.1016/j.envsoft.2020.104869

Seabold, S., Perktold, J., 2010. statsmodels: Econometric and statistical modeling with python., in: Proceedings of the 9th Python in Science Conference.

Sheldon, W.M., 2008. Dynamic, Rule-based Quality Control Framework for Real-time Sensor Data, in: Gries, C., Jones, M.B. (Eds.), Proceedings of the Environmental Information Management Conference. Albuquerque, NM, pp. 145–150.

Smith, T.G., 2017. pmdarima: ARIMA estimators for Python.

Smolyakov, D., Sviridenko, N., Ishimtsev, V., Burikov, E., Burnaev, E., 2019. Learning Ensembles of Anomaly Detectors on Synthetic Data. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) 11555 LNCS, 292–306. https://doi.org/10.1007/978-3-030-22808-8_30

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 Resour. Res. 55, 8547–8568. https://doi.org/10.1029/2019WR024906

Tan, P.-N., Steinback, M., Karpatne, A., Kumar, V., 2019. Introduction to Data Mining, second. ed. Pearson, New York.

Taylor, J.R., Loescher, H.L., 2013. Automated quality control methods for sensor data: a novel observatory approach. Biogeosciences 9, 18175–18210. https://doi.org/10.5194/bg-10-4957-2013

Taylor, S.J., Letham, B., 2018. Forecasting at Scale. Am. Stat. 72, 37–45. https://doi.org/10.1080/00031305.2017.1380080

Tran, L., Fan, L., Shahabi, C., 2019. Outlier Detection in Non-stationary Data Streams 25–36. https://doi.org/10.1145/3335783.3335788

Wagner, R.J., Boulger, R.W., Oblinger, C.J., Smith, B.A., 2006. Guidelines and Standard Procedures for Continuous Water-Quality Monitors: Station Operation, Record Computation, and Data Reporting, U.S. Geological Survey Techniques and Methods 1-D3.

White, D.L., Sharp, J.L., Eidson, G., Parab, S., Ali, F., Esswein, S., 2010. Real-Time Quality Control (QC) Processing, Notification, and Visualization Services, Supoorting Data Management of the Intelligent River, in: Proceedings of the 2010 South Carolina Water Resources Conference. p. 4.

World Meteorological Organization, 2008. Guide to meteorological instruments and

methods of observation. WMO-No 8. Geneva.

Yin, C., Zhang, S., Wang, J., Xiong, N.N., 2020. Anomaly Detection Based on Convolutional Recurrent Autoencoder for IoT Time Series. IEEE Trans. Syst. Man, Cybern. Syst. 1–11. https://doi.org/10.1109/tsmc.2020.2968516

TABLES

Table 2.1 Performance metrics calculated in pyhydroqc and associated equations.

Metric	Definition	Equation
True Positives (TP)	Count of data points from valid detection events where model detection events overlap with labeled anomalous events.	
False Positives (FP)	Count of data points from invalid detections where model detection events did not overlap with labeled anomalous events.	
True Negatives (TN)	Count of data points which did not belong to either labeled events or model detection events.	
False Negatives (FN)	Count of data points from labeled events which were not detected by model(s).	
Positive Predictive Value (PPV)	Ratio of true positives to total positives.	$PPV = \frac{TP}{TP + FP}$
Negative Predictive Value (NPV) (or Specificity)	Ratio of true negatives to total negatives.	$NPV = \frac{TN}{TN + FN}$
Accuracy	Ratio of correctly identified points to all data points.	$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$
Recall (or Sensitivity)	Ratio of True Positives to the total number of labeled anomalies.	$Recall = \frac{TP}{TP + FN}$
F1	Assessment score that combines true positives, false positives, and false negatives. Perfect score = 1.	$F1 = \frac{2 * PPV * Recall}{PPV + Recall}$
F2	Assessment score that combines true positives, false positives, and false negatives. Gives greater weight to false negatives than does F1. Perfect score = 1.	$F2 = \frac{5 * TP}{5 * TP + 4 * FN + FP}$

Table 2.2 Input parameters for each time series. Persistence duration and window size refer to the number of time steps: 20 = 5 hours, 30 = 7.5 hours, 40 = 10 hours, 45 = 11.25 hours.

Observed Variable	Parameter	Franklin Basin	Tony Grove	Water Lab	Main Street	Mendon	Black smith Fork
Temperature (degrees C)	Maximum range	13	20	18	20	28	28
	Minimum range	-2	-2	-2	-2	-2	-2
	Persistence duration	30	30	30	30	30	30
	Window size	30	30	30	30	30	30
	alpha	1E-04	1E-05	1E-04	1E-05	1E-04	1E-04
	Threshold minimum	0.25	0.4	0.4	0.4	0.4	0.4
	(p, d, q)	(1, 1, 3)	(10, 1, 0)	(0, 1, 5)	(0, 0, 0)	(3, 1, 1)	(1, 1, 0)
Specific Conductance (µS/cm)	Maximum range	380	500	450	2700	800	900
	Minimum range	120	175	200	150	200	200
	Persistence duration	30	30	30	30	30	30
	Window size	30	40	40	40	40	20
	alpha	1E-04	1E-05	1E-04	1E-06	1E-05	1E-02
	Threshold minimum	4	5	5	5	5	4
	(p, d, q)	(10, 1, 3)	(6, 1, 2)	(7, 1, 0)	(1, 1, 5)	(9, 1, 4)	(0, 0, 5)
pH	Maximum range	9.2	9	9.2	9.5	9	9.2
	Minimum range	7.5	8	8	7.5	7.4	7.2
	Persistence duration	45	45	45	45	45	45
	Window size	30	40	40	20	20	30
	alpha	1E-05	1E-05	1E-05	1E-04	1E-04	1E-05
	Threshold minimum	0.02	0.02	0.02	0.03	0.03	0.03
	(p, d, q)	(10, 1, 1)	(8, 1, 4)	(10, 1, 0)	(3, 1, 1)	(0, 1, 2)	(0, 1, 4)
Dissolved Oxygen (mg/L)	Maximum range	13	14	14	15	15	14
	Minimum range	8	7	7	5	3	2
	Persistence duration	45	45	45	45	45	45
	Window size	30	30	30	30	30	30
	alpha	1E-04	1E-04	1E-05	1E-05	1E-03	1E-04
	Threshold minimum	0.15	0.15	0.15	0.25	0.15	0.15
	(p, d, q)	(0, 1, 5)	(10, 1, 0)	(1, 1, 1)	(1, 1, 1)	(10, 1, 3)	(0, 0, 5)

Table 2.3 LSTM model parameters and settings selected for the LRO case study. Defaults were used for all other settings and parameters not listed here. See Géron (2017) and Keras Development Team (n.d.) for additional details.

Parameter	Function	Setting	Details
Time steps	model.add	5	The number of past data considered as input for prediction. For the LRO data, more time steps (10, 15, 20) biased results toward the mean. Reduced time steps (5) gave greater accuracy and improved computational time.
Units/cells	model.add	128	Number of cells or nodes in the model architecture. There is no rule for finding the perfect number of cells. We chose a high number and used early stopping and dropout to prevent overfitting. For processing purposes, it is generally preferred to have network dimensions in multiples of 32.
Dropout	model.add	0.2	A fraction of cells that are randomly ignored during training. Using dropout improves the model by reducing overfitting, but the number usually matters little. 20% is often used to balance accuracy and overfitting.
Optimizer	model.compile	adam	Algorithm for training. Adam (adaptive movement estimation) is commonly selected for training LSTM models for being computationally efficient, requiring little memory, and handling large amounts of data.
Loss	model.compile	Mean absolute error	The quantity to be minimized during training. Mean absolute error computes the mean of the difference between observations and predictions.
Epochs	model.fit	100	The number of rounds to train the model. We opted for a high number that is truncated by early stopping that ends training when the model is sufficiently fit.
Validation split	model.fit	0.1	Fraction of training data to be used as validation data on which the loss is evaluated at the end of each epoch.
Callbacks	model.fit	Early stopping	Interrupts training when performance on the validation set drops.
Patience	model.fit	6	Number of epochs with no improvement after which training will be stopped.
Shuffle	model.fit	False	Whether to shuffle training data before each epoch. Set to false because the order of training data matters for these data.

Table 2.4 F2 score comparisons. Scores are reported for ARIMA and LSTM models for each time series as well as rules based detection and the aggregate of all of the models. F2 = 1 would be a perfect score.

Monitoring Site	ARIMA	LSTM univar	LSTM univar bidir	LSTM multi	LSTM multi bidir	Rules Based	Aggregate
			Temperature				
Franklin Basin	0.926	0.840	0.842	0.840	0.841	0.764	0.920
Tony Grove	0.966	0.966	0.966	0.966	0.966	0.066	0.966
Water Lab	0.970	0.909	0.922	0.895	0.923	0.888	0.975
Main Street	0.546	0.571	0.650	0.569	0.625	0.548	0.709
Mendon	0.992	0.992	0.992	0.991	0.992	0.867	0.992
Blacksmith Fork	0.615	0.605	0.605	0.607	0.607	0.448	0.616
			Specific Conductance				
Franklin Basin	0.985	0.403	0.410	0.977	0.723	0.176	0.986
Tony Grove	0.978	0.383	0.264	0.884	0.501	0.127	0.978
Water Lab	0.952	0.809	0.810	0.822	0.919	0.370	0.957
Main Street	0.935	0.876	0.884	0.872	0.904	0.155	0.928
Mendon	0.945	0.836	0.836	0.943	0.856	0.424	0.966
Blacksmith Fork	0.845	0.736	0.776	0.839	0.807	0.134	0.806
			pH				
Franklin Basin	0.967	0.852	0.849	0.945	0.839	0.317	0.968
Tony Grove	0.946	0.654	0.638	0.658	0.632	0.064	0.945
Water Lab	0.966	0.954	0.932	0.934	0.929	0.175	0.969
Main Street	0.983	0.982	0.982	0.983	0.980	0.186	0.984
Mendon	0.995	0.983	0.848	0.849	0.847	0.396	0.995
Blacksmith Fork	0.989	0.983	0.982	0.958	0.955	0.125	0.990
			Dissolved Oxygen				
Franklin Basin	0.496	0.467	0.457	0.470	0.459	0.429	0.497
Tony Grove	0.705	0.404	0.256	0.263	0.256	0.140	0.827
Water Lab	0.892	0.879	0.880	0.967	0.881	0.064	0.980
Main Street	0.967	0.943	0.942	0.946	0.944	0.194	0.968
Mendon	0.873	0.736	0.823	0.750	0.735	0.107	0.879
Blacksmith Fork	0.912	0.964	0.918	0.919	0.963	0.204	0.965
Average	0.889	0.780	0.769	0.827	0.795		

Table 2.5 Technician and algorithm invalid changed data points. Counts represent the number of points where raw data were corrected to values outside of the valid range for that time series. The total number of data points for each series is ~200,000.

Monitoring	**Temperature**		**Specific Conductance**		**pH**		**Dissolved Oxygen**	
Site	Technician	Algorithm	Technician	Algorithm	Technician	Algorithm	Technician	Algorithm
Franklin Basin	584	8	3123	92	11259	837	568	1656
Tony Grove	44	8	1517	13	482	0	692	1185
Water Lab	22	0	7527	59	4169	35	906	0
Main Street	168	0	632	0	6454	121	1171	271
Mendon	1459	2339	8541	0	8187	0	1678	3149
Blacksmith Fork	502	0	1202	0	1208	0	385	507

FIGURES

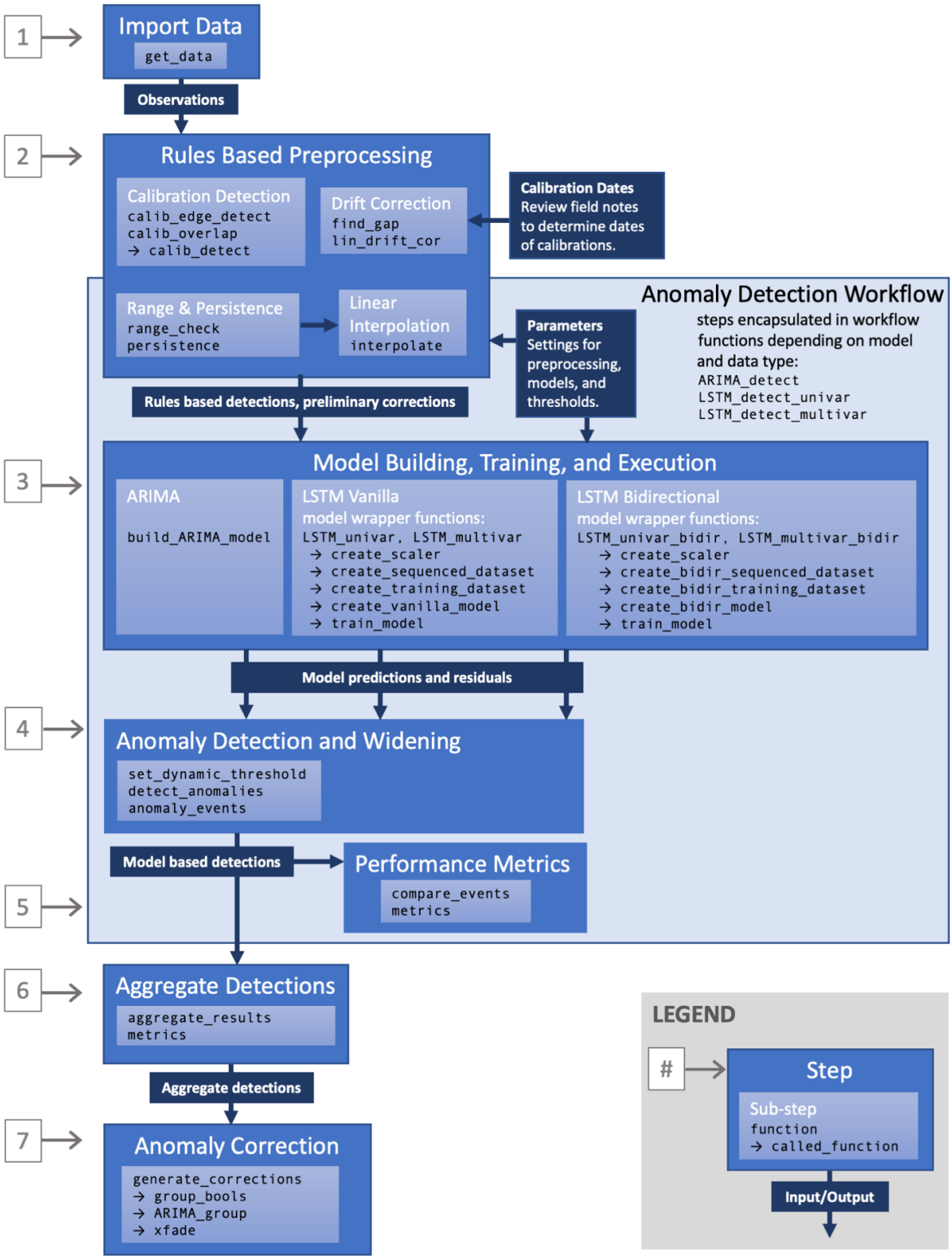

Figure 2.1 Workflow for steps and functions in pyhydroqc. Numbers on the left correspond to steps in the process listed in Section 2.2.1.

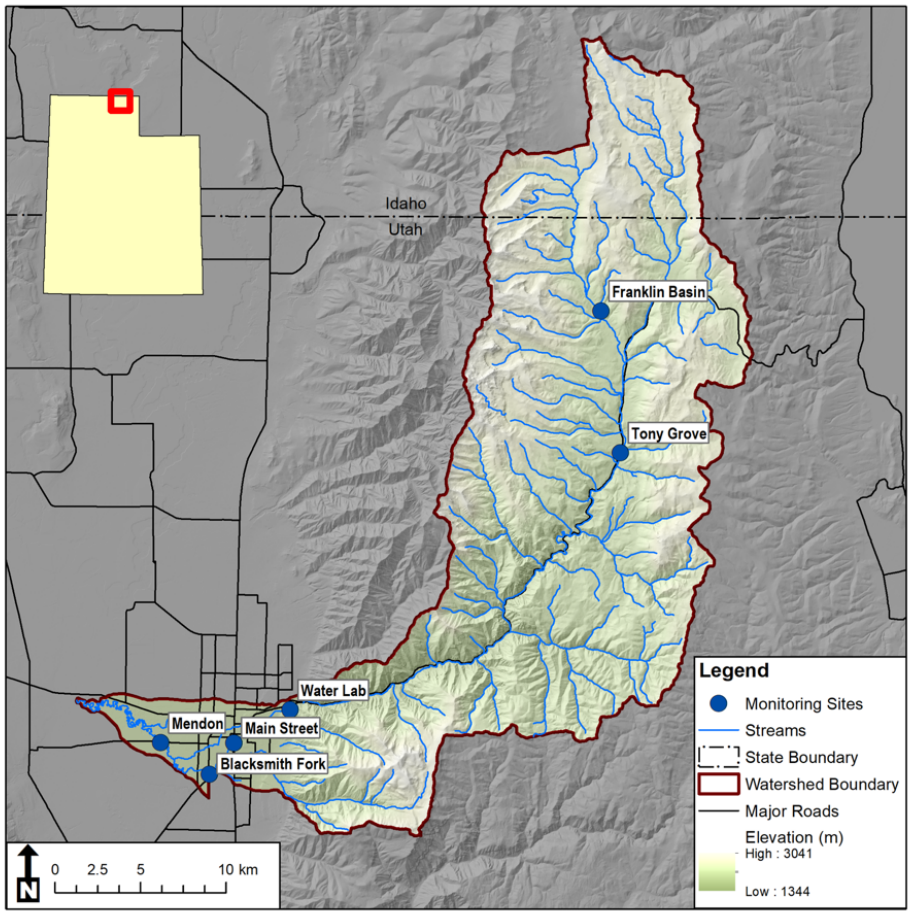

Figure 2.2 Logan River Observatory showing locations of aquatic monitoring sites.

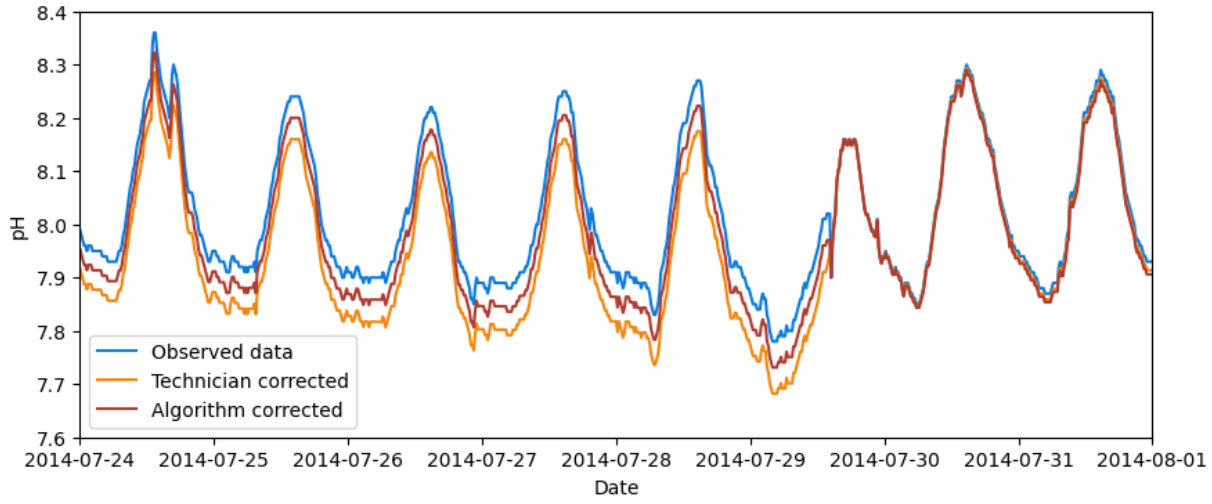

Figure 2.3 Example of gap values and linear drift correction for pH at Main Street. A calibration shift occurred 2014-07-29. The data at the calibration were shifted by a gap value – determined either by the algorithm or by the technician, and data before the calibration were adjusted proportionately.

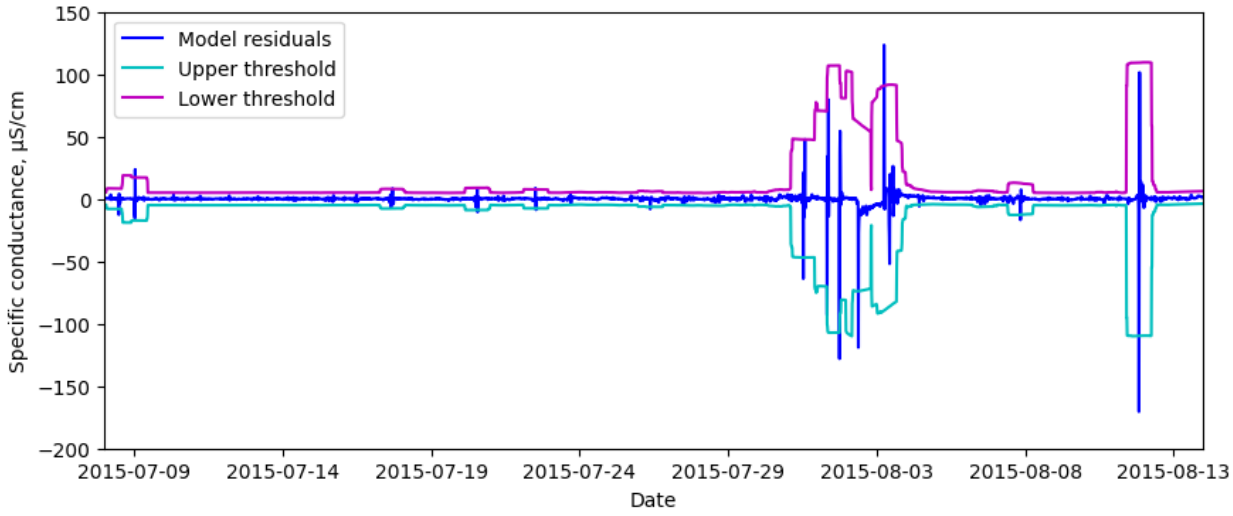

Figure 2.4 Example of model residuals and dynamic thresholds for specific conductance at Main Street.

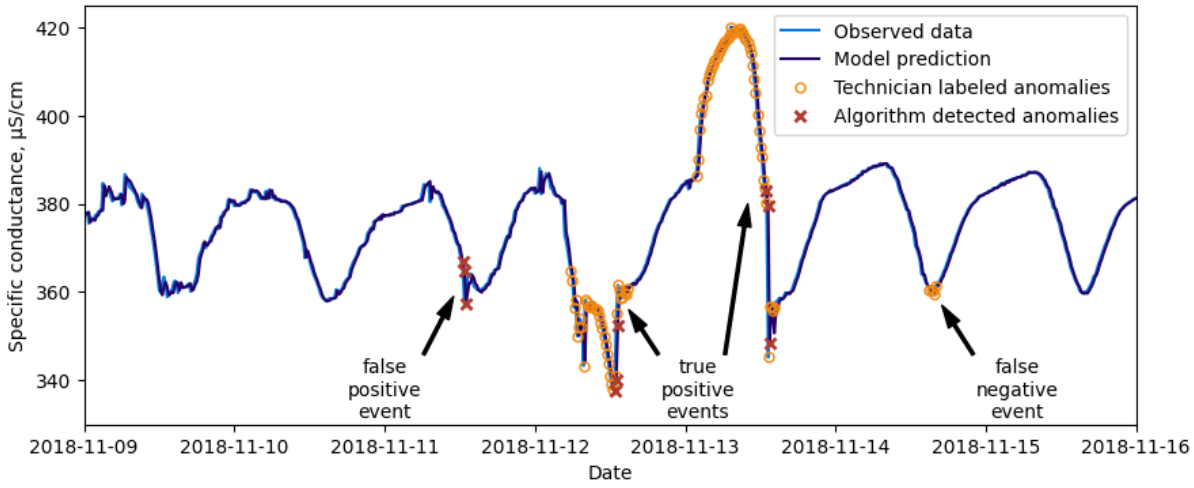

Figure 2.5 Examples of anomalies detected using an ARIMA model for specific conductance at Tony Grove.

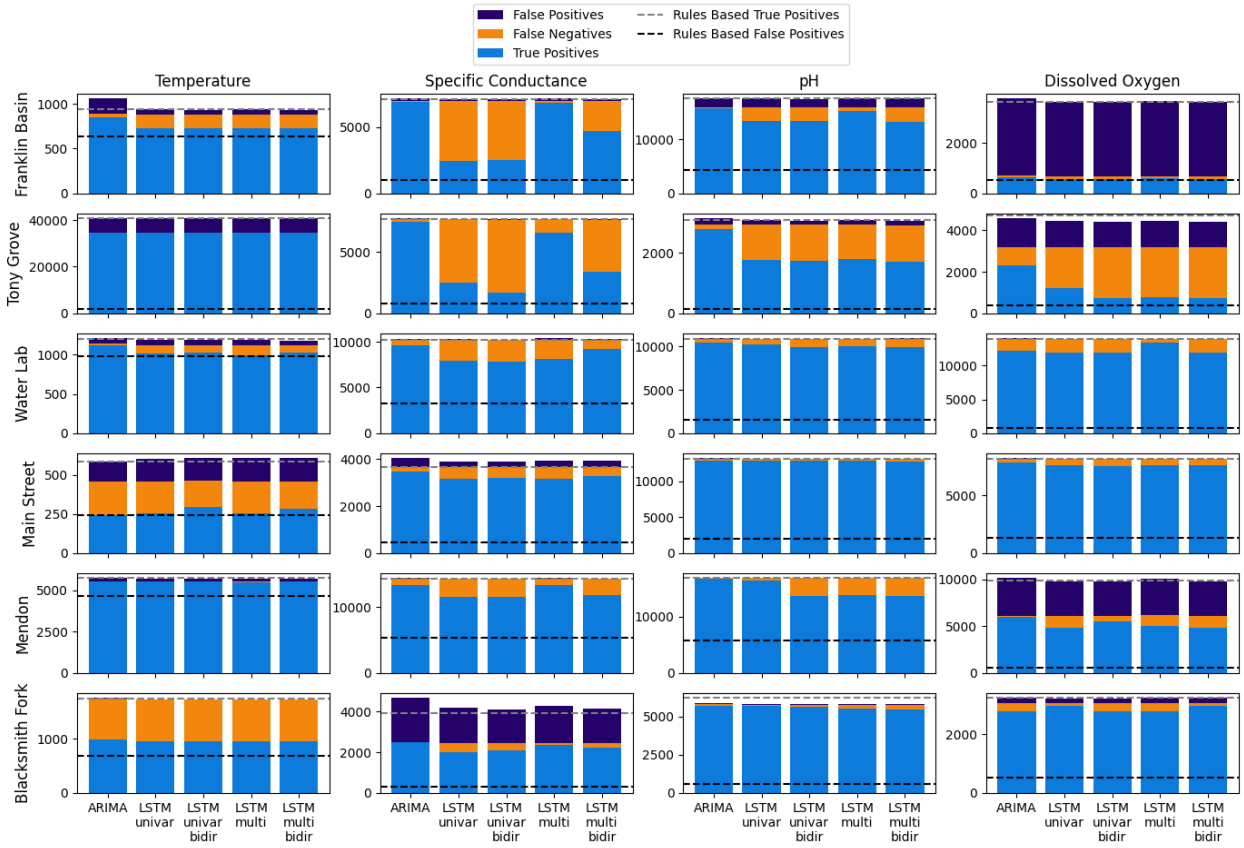

Figure 2.6 Detection confusion matrix values for all time series (panels) and models (bars). y-axis values represent the count of observations that fall within each category shown in the legend. Dashed lines differentiate the proportions of detections from the rules based detection and the model based detection.

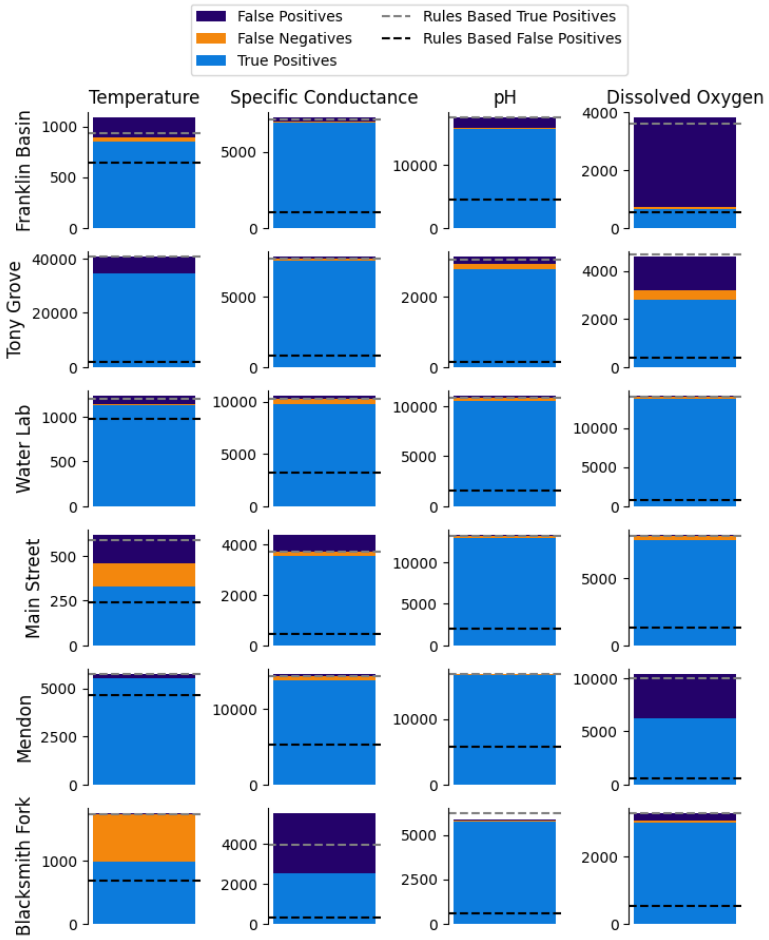

Figure 2.7 Detection confusion matrix values for aggregate results for all time series. Symbology is as described for Figure 6.

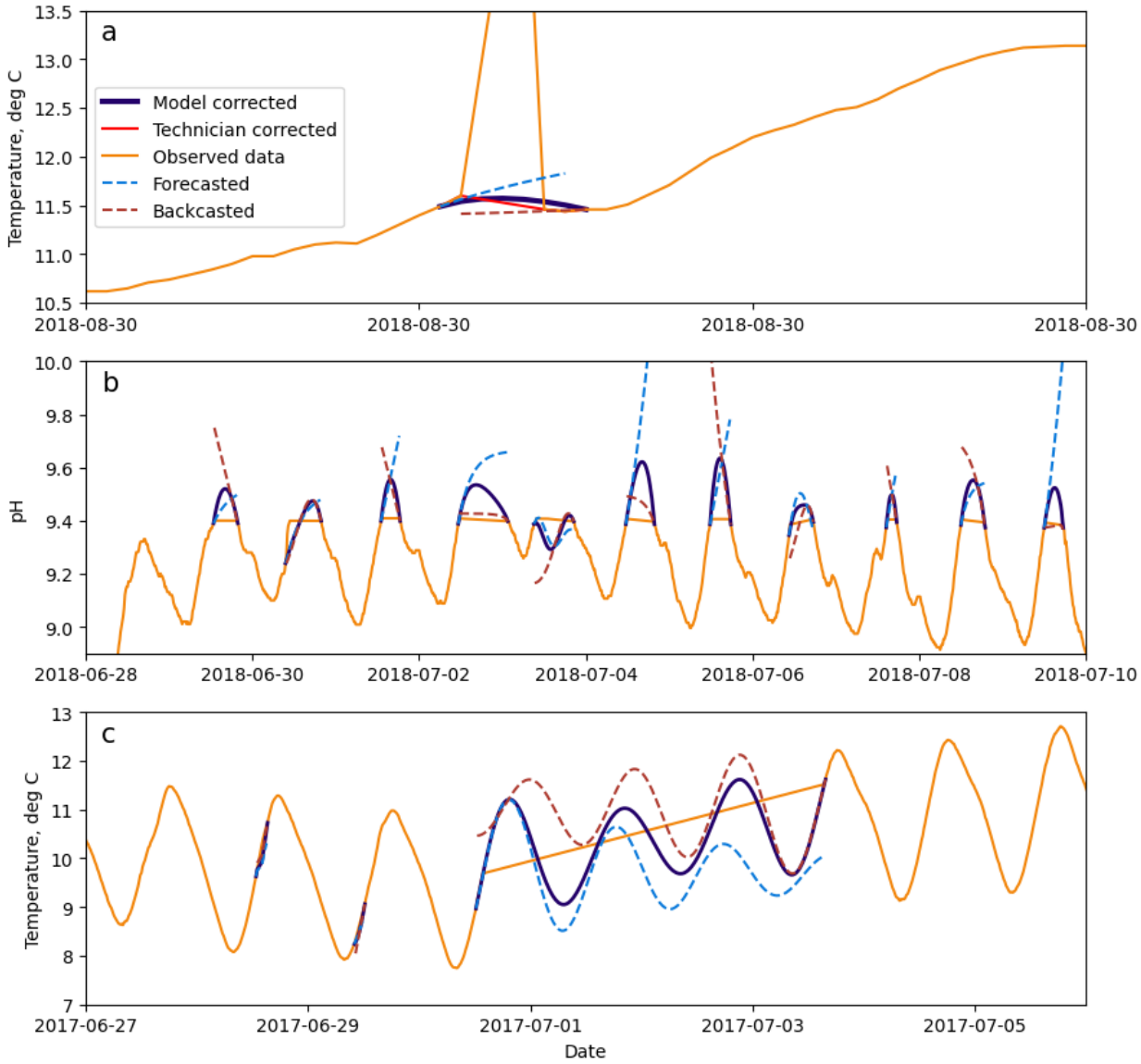

Figure 2.8 Examples of successful correction using piecewise ARIMA models and the cross-fade technique. 8a: temperature at Water Lab, 8b: pH at Main Street, 8c: temperature at Water Lab.

CHAPTER 3

ADVANCING HYDROINFORMATICS AND WATER DATA SCIENCE INSTRUCTION: COMMUNITY PERSPECTIVES AND ONLINE LEARNING RESOURCES[2]

Abstract

Hydroinformatics and water data science topics are increasingly common in university graduate settings through dedicated courses and programs as well as incorporation into traditional water science courses. The technical tools and techniques emphasized by hydroinformatics and water data science involve distinctive instructional styles, which may be facilitated by online formats and materials. In the broader hydrologic sciences, there has been a simultaneous push for instructors to develop, share, and reuse content and instructional modules, particularly as the COVID-19 pandemic necessitated a wide scale pivot to online instruction. The experiences of hydroinformatics and water data science instructors in the effectiveness of content formats, instructional tools and techniques, and key topics can inform educational practice not only for those subjects, but for water science generally. This paper reports the results of surveys and interviews with hydroinformatics and water data science instructors. We address the effectiveness of instructional tools, impacts of the pandemic on education, important hydroinformatics topics, and challenges and gaps in hydroinformatics education. Guided by lessons learned from the surveys and interviews and a review of existing online

[2] Jones, A.S., Horsburgh, J.S., Bastidas-Pacheco, C.J., Flint, C. G., Lane, B.A. Advancing Hydroinformatics and Water Data Science Instruction: Community Perspectives and Online Learning Resources. Frontiers in Water. 4. https://doi.org/10.3389/frwa.2022.901393

learning platforms, we developed four educational modules designed to address shared topics of interest and to demonstrate the effectiveness of available tools to help overcome identified challenges. The modules are community resources that can be incorporated into courses and modified to address specific class and institutional needs or different geographic locations. Our experience with module implementation can inform development of online educational resources, which will advance and enhance instruction for hydroinformatics and broader hydrologic sciences for which students increasingly need informatics experience and technical skills.

3.1 Introduction

In an increasingly data intensive world, researchers and practitioners in water sciences need to apply data-driven analyses to address emerging problems, to explore theories and models, and to leverage growing datasets and computational resources. Within hydrology and related fields in environmental and geosciences, observational data are increasing in scope, frequency, and duration, and computational technologies are essential to solving complex problems (Chen and Han, 2016). Without training, students are unprepared to work or conduct research centered around large and complex data, questions, and tools (Merwade and Ruddell, 2012). To meet this need, hydroinformatics and water data science have been growing as specific topics of instruction, both in university programs and in community education settings (e.g., Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI) Virtual University and University of Washington WaterHackWeek) (Burian et al., 2013; Popescu et al., 2012; Wagener et al., 2021). In parallel, incorporation of technical tools in traditional water science courses is growing, though uptake has been uneven and lags

behind what many see as needed (Habib et al., 2019; Lane et al., 2021). Hydroinformatics and water data science both combine computational tools and water-related data to achieve actionable knowledge. Although the fields are overlapping, there are subtle differences, and both terms are used throughout this paper.

Within the geosciences, there is increased focus on reusability and reproducibility of research data, code, and results, as well as educational materials (Ceola et al., 2015). Several online spaces have emerged as hubs for storing and sharing lectures, code, examples, and scripts developed by instructors in hydrology, water resources, and other geosciences (Habib et al., 2019, 2012; Lane et al., 2021). The widespread shift to online education resulting from the COVID-19 pandemic illustrated the value of online instructional materials and rapidly accelerated development and transition to online formats (Beason-Abmayr et al., 2021; Rapanta et al., 2021). Community educational resources, online platforms, and increased accessibility of digital tools offer an opportunity to more fully incorporate informatics tools and techniques for data-driven hydrologic applications into water science education.

This paper reports on the current state of hydroinformatics and water data science education in the United States based on available literature and qualitative interviews and surveys with instructors of relevant courses. Another objective of this work was development of online educational modules and evaluation of the implementation platform to share insights with other instructors. Study participants offered information about key topics and technologies, formats and methods of delivery, challenges and gaps, and impacts of COVID-19 on instruction. In addition to the results of the survey, we performed a functional review of online educational platforms based on participants'

criteria. Their perspectives and our evaluation were used to inform the development of online learning modules that address some of the identified challenges and gaps while demonstrating existing tools. The modules are community resources that can be incorporated into any related course, workshop, or educational program. They are a step toward sharing educational resources for reuse not only by instructors that specialize in hydroinformatics, but to incorporate informatics skills and topics more broadly in water science courses. The lessons learned from platform feature evaluation and module implementation are valuable for instructors sharing content and for further platform development.

In Section 3.2, we present a literature review of hydroinformatics and water data science education, including best practices for sharing educational content and outstanding gaps. Section 3.3 outlines the procedures and literature-informed questions of the surveys/interviews and the methodology for development of educational modules. In Section 3.4, we present survey results and the key points that drove the design and implementation of learning modules. Section 3.4 also covers a review of existing online platforms and module implementation successes and challenges. Section 3.5 offers conclusions and an outlook for the future of hydroinformatics and water data science instruction.

3.2 Background

3.2.1 Hydroinformatics and Water Data Science

In an early conceptualization, hydroinformatics was described as encompassing computational tools to transform water related data and information into useful and actionable knowledge (VanZuylen et al., 1994). Although hydroinformatics may be

technical in nature, water issues are inherently social, and consideration of human factors for the presentation and dissemination of results and information is a key component (Celicourt et al., 2021; Makropoulos, 2019; Vojinovic and Abbott, 2017). More recently, the definition of hydroinformatics is broadening to encapsulate water science, data science, and computer science (Burian et al., 2013; Chen and Han, 2016; Makropoulos, 2019; Vojinovic and Abbott, 2017). The objective of data science is application of analytical methods and computational power with domain understanding to transform data to decisional knowledge (Gibert et al., 2018; McGovern and Allen, 2021). When applied to the water domain, this definition is very close to that of hydroinformatics, and for most practical purposes, it is difficult to draw boundaries between hydroinformatics and water data science.

Based on the increasing volume, variety, and availability of data sources and the advancement of software and hardware tools, there is opportunity and need for the application of data science to water, environmental, and geoscience domains (Burian et al., 2013; Gibert et al., 2018). Hydrologic science is shifting from collecting data to support existing conceptual models toward analyses based on models derived from observational data (Chen and Han, 2016). In this paper, we report on how current instructors of hydroinformatics and water data science define their fields and the topics and technologies that are growing in importance in these fields.

3.2.2 Hydroinformatics and Water Data Science Education

Without training in data intensive approaches with modern technological tools, students will be unprepared to solve emerging water problems (Lane et al., 2021; Merwade and Ruddell, 2012). Technology integration and data and model-driven

curriculum are key components for advancing hydrology education (Ruddell and Wagener 2015). Many have recommended educational pedagogies for hydrology that are "student-centered" or "problem-based", which describe applications that deepen learning by connecting to real-world contexts (Habib et al., 2019; Maggioni et al., 2020; Ruddell and Wagener, 2015; Wagener and McIntyre, 2007). Students need to learn using real-world datasets, actual tools, and open-ended problems, also referred to as "ill-defined", "authentic", or "experiential" (Burian et al., 2013; Lane et al., 2021; Maggioni et al., 2020; Ngambeki et al., 2012).

Hydroinformatics was initially taught in the mid-1990s to enable engineers to apply information technology to complex water problems (Abbott et al., 1994). Specific programs have since developed including courses for professionals (Popescu et al., 2012) and graduate students (Burian et al., 2013) and complete doctoral programs (Wagener et al., 2021). However, hydroinformatics courses remain limited, and to gain informatics skills, students often rely on technology incorporated into traditional hydrology courses, pursue self-learning (e.g., online courses, tutorials, etc.), or enroll in computer centric courses that do not address the focused set of topics with domain-specific applications covered by hydroinformatics.

Training in data science is typically separate from domain sciences; however, data science curricula cannot adequately address domain knowledge, so students are expected to rely on their own "substantive expertise" (Grus, 2015). Voices in industry and academia are calling for well-rounded and technology-literate water scientists (Chen and Han, 2016; McGovern and Allen, 2021), which may be achieved by packaging informatics and/or data science topics with real-world water science applications (Gibert

et al., 2018; Wagener et al., 2021). In this paper, we use information gathered from instructors to understand how courses are being taught, what techniques are successful, and what would be useful going forward.

3.2.3 Sharing Educational Content

As technology and applications advance, books and even online content may become outdated quickly, and hydroinformatics and water data science instructors are challenged to keep up (Maggioni et al., 2020; Makropoulos, 2019; Wagener et al., 2007). Given shifts toward big data, open data sources, reproducible research, and data-driven analysis, many have called for advancement in content for teaching water science and methods for delivery of that content (Habib et al., 2019; Seibert et al., 2013). The COVID-19 pandemic caused many courses to be moved to virtual platforms, prompting evaluations of instructional formats and a call for additional online educational material (Maggioni et al., 2020).

Community platforms and resources can advance water science instruction by facilitating data-driven learning and offering common principles and approaches for teaching (Makropoulos, 2019; Merwade and Ruddell, 2012; Popescu et al., 2012; Wagener et al., 2012). Although water science modules have been shared and published online (e.g., Gannon and McGuire, 2022; Habib et al., 2012; Merck et al., 2021; Wagener et al., 2012), without integration within a common platform, modules are difficult to identify, access, and implement. In 2012, Merwade and Ruddell noted that an appropriate system was not yet in place, and there remains no single clearinghouse of educational resources in the field. More recently, Lane et al. (2021) and Maggioni et al. (2020) developed and published course content via HydroLearn (https://www.hydrolearn.org/).

Lane et al. (2021) made the case that online educational materials should be supported by active learning, basic templates, adaptation, multiple content types, and pedagogical tools, which are emphasized in the HydroLearn platform. To these functional capabilities, we add that systems need to offer persistence as we were unable to access many of the online resources that were reported in the literature. They were either missing completely, lacking crucial metadata, or using outdated software or systems.

Our review of the literature identified key components, guidelines, and best practices for sharing educational content along with gaps and opportunities to improve. In this paper, we also consider key components to successful online modules as identified by hydroinformatics and water data science instructors, which we used as criteria to select an online educational platform. Based on these findings, we describe the development and implementation in an online system for four modules focused on hydroinformatics and water data science, which are available for instructors to adapt into courses and may serve as examples to the community.

3.3 Methods

3.3.1 Survey and Interview Methodology

We developed survey and interview questions that focused on the instructors' courses and their perspectives on the future of the field (Table 3.1). Participant responses were analyzed to identify common themes surrounding key research questions: 1) What is the current state of instruction in hydroinformatics and water data science, including the effectiveness of tools being used for in-person and online instruction?; 2) How has the COVID-19 global pandemic affected instruction?; 3) Which topics comprise hydroinformatics education and what topics are growing in importance?; 4) What are the

major challenges in hydroinformatics instruction?; and 5) How can shared instructional resources be beneficial for instructors and students? Although this analysis was primarily qualitative, where commonalities emerged, we were able to tally responses and present quantitative results.

Potential participants were initially identified via investigator connections, review of relevant literature, and information on institutional and personal websites discovered by Internet searches. Target participants were selected based on their experience teaching hydroinformatics, water data science, or related subject matter at an institution of higher education. We used email to invite contacts to participate, and participants elected to respond to questions either via online survey or recorded interview. During each interview or survey, participants were asked to identify any additional instructors who might be a good fit for the project.

While the questions for surveys and interviews were the same, both approaches were used so that participants could choose their preferred mechanism to respond. We acknowledge that the different modes for data collection may have influenced the length or character of the responses, but we made this decision to maximize the potential for participation. We observed that content specificity did not differ greatly between surveys and interviews. The survey was composed using Qualtrics software and administered with links personalized for each participant. Interviews were conducted over Zoom, recorded, and subsequently transcribed. Each interview lasted approximately 45-60 minutes. Notes were taken during all interviews in case of issues with audio. A total of 18 instructors participated in interviews (n=7) or responded via survey (n=11). Herein, we refer to interview and survey participants as “participants” and do not differentiate

between the mode in which they participated. Procedures were approved by the Utah State University Institutional Review Board for Human Subjects Research with participation limited to instructors within the United States.

3.3.2 Review of Educational Platforms and Modules

From participants and our own review, we identified several existing online platforms for sharing educational content. Using the survey and interview responses, we extracted characteristics that participants considered important in an online platform for depositing materials and used these to assess available options. We identified specific instances of educational materials from the hydroinformatics community that are available online for each of the considered platforms.

3.3.3 Module Development

We evaluated educational platforms based on the criteria identified in interview and survey results to determine the repository and format to use for depositing the educational modules developed as part of this work. At a minimum, we required that modules be implemented in an open access format. Our selection of a particular platform does not signify that it should be preferred for all instructors, courses, or learning situations, and we anticipate that instructors will adapt content to their preferred interface.

We used the suggestions from participants to inform the topics for the educational modules developed as part of this work. Given the breadth of suggested topics, our team could not develop modules to comprehensively cover all areas. This points to the need for community resources to take advantage of the varied teaching and research expertise of instructors. Rather than serve as a complete and unified set of educational content, the modules we developed act as a demonstration and a launching point for sharing content.

Our conceptual model of a learning module independent of any specific technological implementation consists of the following elements: 1) learning objectives, 2) narrative, 3) example code, and 4) technical assignment. The learning objectives guide the content that is presented through the other elements and may be contained separate from or as part of the narrative. The narrative covers the core of the concepts and topics and is communicated through various formats – e.g., slides, documents, and/or video. Example code may take the form of scripts, formatted markdown or text, or an interactive code notebook. Technical assignments consist of authentic, open-ended tasks based on real-world data that require students to implement code and write a descriptive summary. Authentic tasks are high cognitive-demand activities designed to reflect how knowledge is used in real life and to simulate the type of problems that a professional might tackle. Authentic tasks have no single answer and thus avoid concerns with publicly available solutions and achieve higher level learning objectives. Each assignment includes a grading rubric to ensure that expectations and evaluation criteria are clearly defined and activities are aligned with learning objectives, outcomes and assessment, referred to as constructive alignment (Kandlbinder, 2014).

3.4 Results and Discussion

3.4.1 Survey and Interview Results

Each instructor's definition of the terms "hydroinformatics" or "water data science" was unique, but all centered on common themes of using computers and informatics tools to solve water problems, including data collection, storage, sharing, interpretation, analysis, synthesis, and modeling. One participant simply defined hydroinformatics as "*data and water*". The following quote summarizes the motivation

for teaching these subjects:

> *"We have…talented, quantitatively savvy people…engineers and geologists and hydrologists and scientists that live and breathe data analysis and are limited by the tools they use. And we also have increasing data volume and aging infrastructure, emerging pollutants, drought, climate change. There [are] so many challenges our field faces. So, the goal is to give people modern tools to deal with modern water data challenges."*

The interviews and surveys generated a rich body of results, which we distilled in view of our core research questions. The current state of instruction in hydroinformatics and water data science is addressed in Section 3.4.1.1, including platforms, modes of delivery, and impacts related to the COVID-19 pandemic. As the pandemic prompted shifts to online platforms, Section 3.4.1.2 focuses on the effectiveness of tools for online instruction. Section 3.4.1.3 reports on the topics and technologies that comprise hydroinformatics education. Challenges and future directions of hydroinformatics instruction are covered in Section 3.4.1.4. Section 3.4.1.5 addresses interest, considerations, and potential benefits of shared instructional resources. In the following results, the number of participants (out of 18 total) that correspond to each response is reported parenthetically.

3.4.1.1 Courses, Platforms, and Modes of Delivery

The courses taught by participants include hydroinformatics and related courses with emphases on data science, research computing, and data and analysis tools (see Table 3.2). Most of the courses taught by participants are directed to university graduate students (14), though a few are undergraduate Introduction to Data Science classes (2), several courses are a mix of undergraduate and graduate students (4), and a few are designed for professionals (2). Most of the graduate classes permit some undergraduate

enrollment, and several instructors noted that students at their institutions are exposed to some hydroinformatics topics in lower-level hydrology or geographic information system (GIS) classes.

Most of the courses are conducted in-person, although some had an online component even prior to COVID-19. In total, 12 out of 18 participants teach courses in person. Of these, most moved to an online format because of the COVID-19 pandemic. A few instructors (4) did not teach during this period due to buyout, sabbatical, or changing institutions. Multiple instructors (3) developed courses during the pandemic that would normally be held in-person. Of the courses offered fully online (6), one is a course for professionals, one was offered through an online community college, one was designed for a virtual university, and the remaining 3 are taught through universities.

Of those participants who moved from in-person to online because of COVID-19, most did not significantly change course structure but continued to use a format consisting of lectures with slides and coding demonstrations. Some instructors held synchronous classes over Zoom while others recorded lectures for asynchronous viewing. Generally maintaining course content with some changes to modalities was a commonly reported adaptation to the global pandemic (Beason-Abmayr et al., 2021; Smith and Praphamontripong, 2021). Additional modifications to address challenges of online learning are described in Section 3.4.1.2. Although hydrology and hydroinformatics have been identified as well-suited for online instruction (Merwade and Ruddell, 2012; Popescu et al., 2012; Wagener et al., 2012), even technologically savvy instructors with informatics-focused curriculum were generally returning to in-person formats even before the COVID-19 pandemic was over. The return to in-person instruction may be

related to institutional expectations and instructors' preferences rather than ineffectiveness of tools and technologies (Rapanta et al., 2021). However, several instructors perceived benefits to online aspects and reported adjusting their teaching formats accordingly. A handful plan to shift modalities to alternate in-person and online classes or to a flipped format where lectures are recorded and viewed asynchronously while in-person class periods are work sessions. One participant was pleased with outcomes from online instruction and planned to continue with a purely online format. This is consistent with literature from other fields reporting that a flipped teaching format eased the transition between in-person and online education (Beason-Abmayr et al., 2021). Furthermore, the forced transition to online instruction can facilitate a deliberate integration of online and in-person instruction that is beneficial to active learning (Rapanta et al., 2021).

Instructors reported implementing a wide range and multiple layers of educational platforms to support instruction and handle course materials. Out of 18 participants, most (16) used a learning management system (e.g., Canvas, Blackboard, Brightspace, Sakai) for grading and assignment submission. For messaging with students, some used Canvas (or similar), though several instructors reported success in transitioning all course communication to Slack (2). For some, the learning management system was used to share files, while others stored and shared code and datasets with repositories in GitHub (6) and HydroShare (4), and a few reported using email or Google Drive. All these platforms were generally reported to be effective for both in person and online instruction, and several instructors planned to continue using Slack when returning to in-person instruction.

Most of the participants reported conducting live coding during lectures, whether synchronous or asynchronous, online or in-person. Some instructors switch between traditional teaching material (e.g., slides, videos) and live coding while others exclusively use coding interfaces for instruction. Many instructors (6) reported teaching with code notebooks (e.g., Jupyter) that can be launched from a web browser and include text and images as scaffolding to explain and support the code. Some instructors reported advantages to using GitHub and Jupyter notebooks:

> *"Jupyter notebooks enable us and our students to have a conversation with a problem and link to resources, like audio, video, images, visualizations and implement water resources projects step by step."*
>
> *"Jupyter notebooks work great for teaching either online or in person… They are especially nice for students working through in-class exercises. We…share screens while the instructor or students work through problems."*
>
> *"…copying [the assignment] to my private [GitHub repository] for grading and…deleting …the code that the students need to fill out but leaving the results…then committing those to the public repo [is]…a great tool…because [they] know what the answer should look like. … there's…self-training and…self-evaluation…by…working on their code until they get it to look like what it should."*

3.4.1.2 Challenges and Benefits of Online Delivery

The most reported challenges for online delivery were interpersonal and not unique to hydroinformatics or water data science. Instructors were concerned about meaningful engagement with students, lack of feedback and participation during lectures, and students struggling without the camaraderie and accountability of an in-person instructor and classmates. The paucity of in-person interaction and decreased student engagement have been reported as common concerns with the abrupt shift to online learning (Daniels et al., 2021; Godber and Atkins, 2021).

> *"…a lot of tactile things…are lost in a virtual format, and that can be very frustrating for students and instructors and really slow the course down."*

> *"You ask a question, and there's no feedback. You don't see anybody's faces. You don't hear any response. …you have to force those interactions and knowledge checks through some other mechanism."*

Instructors also reported difficulties with determining the best formats and technologies for rapidly pivoting to online instruction and the time-consuming nature of creating high quality online content. Reduced interaction and the time required for instructors to develop content are established drawbacks to online learning (Habib et al., 2019; Wagener et al., 2021), especially with the rapid shift that occurred in 2020 (Godber and Atkins, 2021; Rapanta et al., 2021).

A concern expressed by multiple instructors (6) specific to computer-based classes was the difficulty of troubleshooting and reviewing code and errors without being able to crowd around the screen, consistent with challenges reported by Gannon and McGuire (2022). Another issue for several instructors was getting hardware and sensors into the hands of students.

> *"…during the hands-on lab, I stop by each student and see if they're following and if they can finish that specific section of the code. …But in Zoom, it's relatively harder to see all the screens and then go back to each one…a classroom environment is often very engaging and more hands on for students. They can easily talk to the person next to them and get some help."*

> *"Live coding is challenging because students don't often have multiple screens, so typing code while watching the lecture requires some careful window manipulation."*

To address these challenges, instructors adjusted to hold more office hours and help sessions and increase communication opportunities, which was also important for Smith and Praphamontripong (2021) in transitioning a coding class online.

> *"I polled students [to ask] what's going on? What are the pain points? …they really enjoyed being able to watch stuff on their own time. So instead of doing a live lecture, I ended up doing recordings and then during the lecture times I [held] office hours. In fact, I started doing…office hours at…9pm, 10pm. It was crazy how busy they were."*

> *"We do a lot of office hours due to COVID so that we can connect, look at their screen…What's the problem with their code? I increased [office hours], but also, I schedule meetings with students if they have a [specific] problem…it's not really that engaging as in person, but still, we try to support the missing pieces…through some online meetings."*

Participants reported that communicating expectations for online classes and deliberately facilitating interaction helped ensure student engagement.

> *"We make it a point to tell students that being in an online class is no different than being face-to-face in terms of being engaged or not. …This helps the students get to know each other and learn how to navigate online meetings, which is a great professional skill to develop. We are also more intentional in encouraging community in the online class; I have an "ice breaker" question related to data science each day, and many students submit their answers in the chat window."*

Despite the challenges of online delivery, instructors deemed several aspects of online instruction as beneficial. Zoom was an effective technology for interactive remote instruction, and several participants preferred live coding via Zoom rather than in the classroom because students could more easily follow along and screenshare their own work. For some participants, Zoom breakout rooms facilitated group work. Others reported benefits of live coding with screen sharing as well as online breakout rooms (Beason-Abmayr et al., 2021; Smith and Praphamontripong, 2021).

> *"If anything, the class may have gone more smoothly this way because everyone was sitting at a computer all the time so we could more easily screen share and debug and demonstrate across the instructor and student machines."*

> *"There are some elements of being online that work really well for this class. …The course is …flipped, so each professor prepares…videos for*

> *the students to watch in advance, and they also prepare a set of in-class exercises. During class, we split the students into breakout groups of 4-5 students each, and they work on the exercises. The professors and TA circulate through the rooms answering questions. At the end of the class period, we reconvene to discuss interesting problems or issues that arose while the students worked."*

Even with a return to in-person instruction, some are retaining approaches that were successful during the online period. These adjustments include non-traditional modalities for synchronous/asynchronous lecture and work sessions and increasing the use of tools and platforms such as Zoom, Slack, and Jupyter notebooks. This reflects the recommendations made by Rapanta et al. (2021) to retain effective aspects of online learning when blending with in-person modalities so that digital technologies support rather than hinder active learning.

3.4.1.3 Content, Technology, and Topics

All participants reported creating custom materials for their course and/or adapting content from other sources. A majority (13) created most of the instructional materials for their course. Only a handful (4) used any textbook: one hydroinformatics text, one modeling text, one statistics text, and one converted an existing coding book to water resources examples. A reported challenge is the rapidly evolving nature of the field in which the technology and applications change faster than published textbooks can account for. Several instructors (4) borrowed, exchanged, or modified material from each other.

> *"I have created all of my own course materials. I do not use a text. Most materials were drawn directly from my own research and project experience or that of my close colleagues."*

> *"We have built up the course material from scratch…we were not aware of a…textbook that would teach the students at the level that we wanted*

> *and with the types of R programming that we wanted while illustrating with the water-related data that we wanted."*

Regarding technologies emphasized, almost all instructors teach coding in Python (10) or R (6). In addition, instructors cover structured query language (SQL) (4), ArcGIS (3), Arduino (3), and web technologies (i.e., PHP, JavaScript, HTML, CSS) (3). For several cases, the course evolved from using Matlab to R to Python so that students have experience in a non-proprietary coding language that they can use in subsequent settings regardless of affiliation.

> *"I had a student who was just an outstanding computationalist. ...got a great job…came back and she said…I really loved your class and I wish I still had…the ability to do those kinds of analyses, but our company won't pay for the MATLAB license…it was just heartbreaking because…think about what your company is missing out on by you not being able to do that…I [determined I] really…need to move this to Python or something that they're going to continue to have access to, regardless of where they work in the future."*

Although hydroinformatics is centered on tools, rather than emphasizing specific technologies, participants emphasized teaching students how to learn new informatics tools, a finding that echoes the emphasis of Burian et al. (2013). Several instructors noted that hydroinformatics technologies continue to advance, which makes it hard to settle on a set of tools to use in teaching a course and highlights the need to teach students how to recognize which tools to use in different scenarios.

> *"Students might never use those specific tools again, but have skills to learn new tools."*

> *"I do not expect that students leaving my class will be experts in any of these skills. However, they should have explored each of them and developed a level of proficiency that they know which of them will be the most useful in their research and future careers and which may be the most important for them to invest further time and effort into becoming more proficient."*

> *"I think we have reached a point where there are relatively good cyberinfrastructure components out there in the hydroinformatics domain and now one of the bigger problems is composability - e.g., how can students and researchers learn all of the available tools and then decide which tools to put together in composing a research, data analysis, data science, modeling, etc. workflow."*

Other instructors emphasize data and project management skills, which are agnostic to specific technologies or tools.

> *"My expectations for the informatics skills…are…more about…habits of mind and computational practices around…reproducibility and…sustainable code…making sure that their code is under version control, making sure that they're using things like Jupyter notebooks to provide…traceable and reproducible demonstrations of their workflows, more so than any kind of specific technique that they're using."*

An important skill repeated by participants was appropriate troubleshooting, including understanding documentation and finding help through forums and other resources.

> *"We…encourage students to use the internet to help them work through problems and troubleshoot coding errors (e.g., Google, StackOverflow)."*

Each instructor and each course have specific emphases. While there is variety in what is taught, the overlap of common subjects illustrates key topics and themes that currently comprise hydroinformatics instruction (Figure 3.1). Most instructors (13) focus on scripting and coding basics (in Python, R, or Matlab) with emphases on data formatting, manipulation, and wrangling (12) and data visualization and plotting (11). Data science (10), basic statistics (7), and machine learning topics (7) were commonly mentioned. About half of participants covered geospatial topics such as mapping (7) and spatial analysis (10), which some instructors view as essential while others exclude these topics as they are covered by other courses. Several participants (6) include instruction on

workflows, reproducibility, and best practices for coding. Other topics mentioned by multiple instructors included databases, data models, and SQL; dataloggers and sensors; modeling; the data life cycle and metadata; Git; and web services and web mapping tools.

Because of the open-ended nature of the questions, these numbers should be interpreted generally – e.g., more instructors may include content on metadata but did not explicitly mention it. Similarly, "modeling" is a broad term with various meanings and implementations. Despite these limitations, we can identify a few important takeaways. First, hydroinformatics is broadening its focus from modeling with custom tools and graphical user interfaces (GUIs) (as described in many of the papers we reviewed) to more strongly emphasize data management, visualization, and analysis using open-source scripting tools. These capabilities provide a broader path for addressing water-related challenges and questions.

> *"[The] basics of how to organize, use, and process data has not changed, but the technology to do that keeps changing. For example, we no longer use interface or GUI… The term workflow was not used earlier but is now used frequently. There is more use of internet-based tools and publicly available/open-source tools."*

> *"Things are becoming more standard; the tools keep getting better. We are now able to use mostly open-source mainstream languages and tools for our specialized environmental informatics work; 20 years ago we needed to build and use clunky, custom-purpose tools. This is much better now. It also means, however, that there is less need for 'hydroinformatics' specific tools and methods."*

Second, a primary objective for many of the instructors was to ensure that students are comfortable working in one scripting language and understanding the basic concepts of functions, conditional statements, iteration, logical operation, data management, querying, and visualization. Any modeling being taught is within the context of open-source scripting environments. We observed that data science, statistics,

and machine learning topics are generally being taught in the water data science courses while databases, sensors, and spatial analyses are being taught in strictly hydroinformatics classes. However, the crossover between these topics is growing, and the boundaries between hydroinformatics and water data science are fuzzy.

Third, several instructors emphasize communicating scientific data and results, and others focus on enabling students to translate the skills gained in the course to resume entries or digital code portfolio.

> *"I'm big on science communication…that was the first time that they had ever really had someone be pedantic enough to talk about presentation of data, quality of graphs, quality of the writing."*
>
> *"I try to work with them to put it on their resume in a way they can explain it. …they're getting some really cool jobs…they wouldn't have gotten, as a result…So it basically opens up career trajectories that are not just typical civil and environmental consulting."*
>
> *"At the end of the class I'm hoping that they have…a GitHub repository that has…Jupyter notebooks that are their problem sets that they feel comfortable sharing on their LinkedIn profile or their CV that [is] a small e-portfolio of a demonstration of things [they] can do computationally."*

3.4.1.4 Challenges And Future Directions

There was little consensus in identified challenges and future directions (Figure 3.2), which reflects our finding that instructors are developing their own content based on their own definition of the field, drawing from their own research and experience. Many participants identified machine learning, deep learning, and/or artificial intelligence as increasingly relevant, reflecting the growing use of these techniques in water science (McGovern and Allen, 2021; Nearing et al., 2020; Shen, 2018). Beyond covering those topics broadly, some instructors offered specific ideas, including better understanding why some techniques do or do not work for some datasets, addressing correlation in data,

and using data-driven modeling with physics-informed machine learning. Sensors and hardware-related subjects were identified as important by many participants, including managing high frequency data, low power and ubiquitous sensing, and smart sensors with controls and feedback for real-time decision making. Participants also mentioned electronics, drones, and satellite data. Data management aspects included data quality, reproducible analyses, big data, database schemas and SQL, and collaborative version control (e.g., GitHub).

> *"So there's always going to be an importance in a baseline proficiency in working with tabular and spatial data within water resources data science. …as data volumes increase, then you need…database skills, so creating schemas, interacting with databases, whether that's Postgres on a cloud or [SQLite] on your local computer. …something [that will] hold really big volumes of data, and then interact with it in a structured query language."*

One participant noted that web applications are overtaking desktop applications, further evidenced by several participants identifying cloud computing and technologies as an area of growing importance. For geospatial topics, emerging applications include open technology and platforms (e.g., Google Earth Engine) and open remote sensing products. Although visualization is covered in most of the courses, several participants noted that creative, interactive visualization tools and dashboards are increasingly important.

The range of responses regarding topics of growing importance demonstrate that these subjects are broad and varied, and that the tools, technologies, and topics continue to evolve, compelling instructors and courses to be agile. The challenge of defining and teaching a moving target was reiterated by several participants. Despite the long list of possible topics to cover in a course, one participant suggested that simplifying to cover fewer tools and models is preferable. Given the inflexibility of most engineering and

science degree curricula and class structures, it is unlikely, outside of specifically focused degree programs, that additional hydroinformatics and water data science classes will proliferate in most university settings. However, it is feasible, and arguably preferable, that hydroinformatics and data science topics be better incorporated into other existing courses.

> *"Students have told me previous versions of this course was foundational for their PhD/MS and that it was 'the most useful course I have ever taken'. They appreciated…the hidden curriculum (stats/R/programming) was brought to the forefront in my classes."*

> *"Students get very little, if any, exposure to hydroinformatics with their undergraduate degrees. I am in a Civil and Environmental Engineering department, and our undergraduate curriculum is so tight that students have very few options for tailoring their undergraduate degrees. Thus, many…show up in graduate school lacking the preparation for making advances in hydroinformatics."*

A major gap reported by participants is students' lack of baseline programming experience. Most of the courses expect some level of domain knowledge but do not require programming skill. However, getting students up to speed consumes precious time, and instructors would prefer programming/scripting at earlier levels (i.e., undergraduate). Participants reported difficulty in approaching advanced topics when students are learning to program for the first time, similar to Lane et al. (2021). Although computational skills are critical to water science and hydrology fields (Merwade and Ruddell, 2012), students are often expected to figure them out without explicit instruction (i.e., the "hidden curriculum").

> *"Mainly I think hydroinformatics concepts could be introduced earlier or at all in undergraduate education. These things are so critical to the field that I think a solely analog hydrology course is a disservice to students."*

> *"If students don't come prepared with coding competency and conceptual*

fluency in computer science, they struggle to learn the applications to environmental fields."

3.4.1.5 Shared Resources

Participants unanimously indicated moderate to high interest in sharing and exchanging teaching materials, and several reported already depositing educational content online. However, the materials are spread out in various formats over multiple platforms, and we were unable to locate some of the resources reported to be available. There is no single centralized platform, and implementations range from files uploaded to a personal website to a fully interactive online course. Reported interest and rate of uptake is uneven. One participant prepared and posted course content in a public repository with no knowledge of reuse while another shared content in an interactive website and received feedback from multiple external users. Even so, the level of reuse is modest relative to what some participants consider necessary for high impact.

"You have to make it easy and provide a venue where a significant number of students or other faculty will pick up on content."

Despite universal interest in sharing materials, some participants expressed hesitancy to rely on others' content, to personalize and adapt it to fit their class, and to invest the time to gain the expertise to present others' materials.

"I don't know that…I would have grabbed someone else's material and…taught…a course. There's a lot of value I found as an instructor in having to prepare all the material from scratch myself as a way of making sure I actually know what I'm talking about. …it is very nice to have other resources [as a] stencil of what a class might look like, and what good topics would be…I would probably still have to spend the time to develop…a copy of that myself so that I actually knew what I was doing."

A barrier to exchanging materials is the difficulty of knowing what modules or case studies exist, so an ideal system would facilitate discovery. Other desirable qualities

of a platform, as identified by participants, include complete descriptions/metadata, a navigable interface, straightforward functionality for adding content, and separate teacher/student access.

> *"Some website where it is easy to search and find modules. It should be easy to navigate and easy to add new contributions. It would be cool if you could see how other faculty members have put together modules to create their own course."*

For shared resources, instructors are interested in portable programming examples, particularly: 1) Jupyter notebooks consisting of code and supporting theory and instructions in markdown, and 2) GitHub repositories that can be cloned and adapted. Other suggestions included slide decks, videos, handouts, example assignments, HydroShare resources, and ArcGIS online content. Participants wanted modular, self-contained exercises that can be modified and swapped into classes.

> *"Self-contained coding exercises that maybe on the first iteration can address a single problem, but then the instructor themselves can develop the sequence of problems that are the deeper dives after that. Something that can be easily plug and played into an existing curriculum or into an existing lecture, and then…would encourage ownership of the content."*

Similar to topics of increasing importance, topics of interest for shared resources varied (e.g., databases, interactive visualization, data-driven hydrologic models, cloud computing, etc.). Regardless of topic, domain specific datasets were consistently mentioned as a key need for shared resources.

> *"The biggest [need] is domain specific data that works for the kind of examples that we need to show…datasets that are large, complex, have hidden components in them that we're going to find, can be used to make a case for or against something…that can serve as good examples. And it's a slippery slope because either the dataset is too simple and it's silly. It's like 10 data points and we're drawing a line through it. Or it's…somebody's PhD dissertation and good luck getting that like into*

some sort of format where an undergrad can actually use it in the class."

"Datasets that are ready to be used for illustration in class. These must have associated metadata that describes why the data was collected, what the researchers hoped to achieve with it, what each of the variables is, the sampling frequency, and what the data can be used to illustrate (i.e., clustering, visualization, regression, etc.)."

Several participants recognized that licenses with clear conditions for reuse and citation would help instructors understand limitations and expectations for repurposing content.

"…one of the best ways to learn is to look through other people's well-documented code, so open-sourcing the code and data used for scientific research, and using FAIR data standards to improve documentation and usability, is very important."

"I think a GitHub with data with notebooks…that has a clear Creative Commons license for both the data and the notebook. And so I know I can use it, change it without getting a nasty gram…from someone's legal department seven years later."

Regarding barriers for exchanging resources, the most common response was that credit could motivate instructors to publish instructional material. This may take the form of counting toward tenure and promotion decisions, citations to document the contribution, or monetary payment – e.g., a grant related to platform or repository development.

"Support from universities for "teaching" efforts beyond the…classroom, and consideration of these efforts and outcomes (e.g., pageviews/downloads) for hiring & tenure decisions."

"Money - there's a lot I think we'd all do for a small amount of money. If you pay professors for their time, they will engage."

Normalizing sharing teaching materials and developing a community around the exchange was another commonly repeated suggestion. Reciprocity was mentioned as crucial so that the exchange is mutually beneficial rather than a one-way offering.

> *"…if there are ways to, outside of the traditional incentive structure of writing research papers, to incentivize…technologically savvy researchers, postdocs, faculty to contribute lessons like this, then you'll see more participation… it has to be made important and valued by…the community somewhere."*

> *"[I would] go through the trouble of sharing…my resources if I knew that others were sharing theirs and that there could be an exchange from which I could benefit. All of my course materials have been online and openly available for a long time. Others have asked if they could use them, and I have always said yes. I've never had anyone offer to let me use modules they have developed, so the 'exchange' part of this would be important for me."*

Collaboration via feedback and edits on shared content was suggested, and multiple participants mentioned that workshops would be helpful to exchange ideas and build rapport.

> *"This course material is available to only 25 students per year. And seeing that it is used by many more…by different instructors and different institutes would be a nice…outcome of all these efforts. We really put a lot of effort for these materials to be created and used and refined throughout the years. ...potentially giving feedback to these material and…seeing some updated versions of it by other instructors...a community level refinement of the course materials, and creating new versions and better, maybe more up to date versions of these slides will be…useful."*

> *"It would…motivate me if I knew that my contribution would be widely viewed and/or utilized. A workshop that drew educators/contributors together to share could be a helpful place to start."*

3.4.2 Building Educational Modules for the Future

Using information gathered on online educational platforms and examples of hydroinformatics educational content from study participants and our own search, we reviewed existing online platforms considering participant-identified attributes and selected HydroLearn for module implementation, covered in Section 3.4.2.1. Section 3.4.2.2 describes the modules developed by this work and how they address identified gaps. Module implementation is related in Section 3.4.2.3, including the mapping of

module components to HydroLearn concepts and the benefits and challenges of implementing modules in online platforms such as HydroLearn.

3.4.2.1 Online Educational Platforms and Materials

There was no consensus among instructors on the preferred approach for sharing hydroinformatics educational material (Table 3.3). Some of these platforms are growing in popularity in the hydrologic science community but have not gained traction with the hydroinformatics instructors that we surveyed. The options include systems specifically designed for sharing and publishing educational content (HydroLearn, MyGeoHub, eddie, ECSTATIC), more generic repositories for data or code (HydroShare, GitHub), and customizable interfaces (personal websites, Canvas, or online courses). We reviewed these options with respect to characteristics extracted from the literature and our survey results (Table 3.4). Desirable characteristics include flexibility for hosting various types of materials, compatibility with open data practices, formal pedagogical structure, structured metadata, review and curation of content, and separate faculty and student access (Lane et al., 2021; Makropoulos, 2019; Merwade and Ruddell, 2012; Popescu et al., 2012; Wagener et al., 2012).

The major tradeoffs between the identified platforms are the level of control for creators versus structure to support education-specific content. Whereas personal websites and custom online courses allow for a great deal of specialization, regular updating, and customizable interfaces, they do not include the searchability, structured metadata, curation, and educational support offered by several of the education focused platforms. A particularly attractive feature for hydroinformatics and water data science instruction is the ability to launch and run code notebooks. Two of the platforms that we

examined have Jupyter servers and can launch notebooks: MyGeoHub and HydroShare. Potential challenges with these platforms include scalability for use with classes of students, inclusion of data files that accompany code, and installing desired software packages. Although existing systems currently do not support all desired functionality, we anticipate those limitations will be overcome with future development.

In deciding which platform to use for the educational modules of this work, we considered the factors in Table 3.4 with a focus on reuse and collaboration. We deposited materials in HydroLearn as it facilitates export and adaptation of courses and includes metadata, citation, curation, and pedagogical structure. HydroLearn is a repository for instructional material related to hydrology and water resources. Developed on the edX learning management system, HydroLearn is designed to support collaboration around instructional content, reuse and adaptation of materials, and flexibility for implementation in organized courses or by self-paced learners. Although it is relatively new, several cases observed enhanced learning of concepts and technical skills by students using HydroLearn and its precursors (Habib et al., 2019; Lane et al., 2021; Merck et al., 2021). Although it does not natively support launching and running notebooks, Lane et al. (2021) demonstrated linking notebooks via HydroShare.

3.4.2.2 Online Module Development

Based on the survey results, online educational materials are being used and modules have potential to address challenges in hydroinformatics and water data science education. However, there is substantial variety in topics and methods of instruction. While a unified curriculum and approach to the subject matter may be appealing, it does not match the reality of a rapidly changing field with dynamic courses and instructors.

Instead, we sought to develop and publish example educational modules that focus on addressing gaps identified by participants and to illustrate an approach for additional online content creation and sharing.

The online modules were designed to address key challenges/gaps in hydroinformatics and water data science education reported by instructors. These gaps relate to: 1) content, 2) platform, and 3) organization. Regarding content, there is a lack of data-driven and problem-based learning that uses datasets from the water domain. Instructors requested notebooks for online coding examples, and there is a need for baseline levels of instruction in coding and scripting. To address the content gap, online educational content should include interactive code with water-related data and problems. Currently, instructors use various platforms for hosting educational content, and participants repeated the need for a system to facilitate upload, discovery, and community involvement. The platform gap may be addressed by publishing and publicizing resources in a system that meets many of the criteria in Table 3.4. We add that active and ongoing support are essential to ensure that the resources are not siloed or lost. Finally, the organization gap can be addressed by ensuring that the content is designed and structured to be modular and adaptable to different instructors, courses, and modes of delivery.

For our online modules, we worked to follow these recommendations to address the needs of hydroinformatics and water data science education. The modules address four topics: (1) Programmatically accessing water data via web services, (2) The sensor data life cycle and sensor data quality control, (3) Relational databases and SQL querying, and (4) Machine learning for classification (Table 3.5). These topics were selected based on survey and interview results indicating the need for reproducible code

and the growing importance of high frequency sensor data, data quality control, databases, big data, web technologies, and machine learning. In conceptualizing these modules, we drew from our own expertise and datasets generated or used as part of our research efforts. The datasets are available for reuse, or instructors could apply the examples to data from other locations.

3.4.2.3 Online Module Implementation

HydroLearn facilitates a "Backward Design" approach wherein desired outcomes are first defined, then authentic tasks are crafted to meet outcomes, then instructional content is designed to present necessary information (Maggioni et al., 2020). Although in our case, development did not proceed in this order, the essential elements in our module design methodology correspond to backward design concepts and specific HydroLearn components: 1) learning objectives map to desired outcomes, 2) narrative maps to instructional content, 3) example code maps to both instructional content and authentic tasks (i.e., learning activities in HydroLearn), and 4) technical assignment maps to authentic tasks (learning activities). Implementation of each of the components in HydroLearn is reported in the following subsections.

3.4.2.3.1 Structure and Organization

Each HydroLearn course contains "modules" or "sections", which is the level to which we matched our modules. Although our modules stand alone, we included them under a single course umbrella (Hydroinformatics – USU 6110) to fit the HydroLearn schema. Modules consist of "subsections" comprised of "units". The subsections are only titles, whereas content is contained as components (e.g., text, discussions, problems, HTML code, videos) within units. In HydroLearn, users have control over using either

many components within fewer units, which makes interaction with content more vertical (i.e., scrolling on a single page), or using many units, which makes interaction with content more horizontal (i.e., navigating from unit to unit). While this provides flexibility in presenting content, we found that navigation between subsections and the different levels of each module was not always clear.

Figure 3.3 illustrates the organization of a module implemented in HydroLearn. While this is an intuitive structure, it imposes hierarchical levels that may be overly strict for some users. For example, we found "subsection" to be an unnecessary level for some modules and would have preferred to directly use "units" under the module level – or to have had control over the hierarchical levels. Granularity and organization are persistent questions for many repositories, regardless of content type (Horsburgh et al., 2016), and developers of many data repositories determined to leave organization and structure up to the user (e.g., FigShare, HydroShare, Zenodo). Although there are benefits to imposed structure, there is no single prescriptive pattern, and users may prefer different organizational levels. We identified degree of control as the main distinction between platforms, and giving users more control over organization and structure may improve the appeal and uptake of HydroLearn (and similar platforms). Despite these limitations, we were able to fit our module content to the HydroLearn structure.

3.4.2.3.2 Learning Objectives

Learning objectives are the desired outcomes of instruction and are ideally action-oriented, specific, and measurable. As a major part of its pedagogical emphasis (Lane et al., 2021), HydroLearn facilitates the creation of learning objectives, which can be entered manually or developed using a wizard according to an established structure

(Maggioni et al., 2020). Although our learning objectives were defined prior to using HydroLearn, the wizard helped improve their specificity and robustness. HydroLearn functionality can directly connect module learning objectives to other module components (e.g., rubrics).

3.4.2.3.3 Narrative

For each module, the narrative was created in slides with text and images, then content was transferred to HydroLearn. Because study participants reported commonly using slides for lectures, the modules include linked slide deck files. Overall, we were successful in translating our content to HydroLearn components. Despite it being somewhat tedious to adapt text to HTML and to import and export images from slides to HydroLearn, we found it straightforward to edit content, to duplicate and modify components, to reorder units, and to publish changes. Building the course from the foundation of a HydroLearn template offered helpful organization and instructions.

3.4.2.3.4 Example Code

Each module contains 3-6 example scripts, each of which illustrates a task or piece of functionality (Table 3.5). There may be redundancy as examples build on each other, and instructors may choose to use fewer examples than provided. Code examples are shared in Jupyter notebooks as part of HydroShare resources that can be opened and run via the CUAHSI JupyterHub Server. We opted to use the CUAHSI JupyterHub because: 1) common Python packages are pre-installed, and additional packages can be installed by request, both of which are dependencies in our examples, and 2) data files can be called by code, which is essential for our modules. If data files are necessary to examples, they accompany the code notebooks in the HydroShare resources.

HydroShare resources containing notebooks and data can be linked and opened in a separate browser window or embedded as iFrames in HydroLearn units (Lane et al., 2021). We used links that directly launch the CUAHSI JupyterHub (Figure 3.3). From the link in HydroLearn, a user is prompted to sign into HydroShare and choose a coding environment and then is taken to their server directory where the notebooks are ready to be launched. This simplifies deployment of example code as learners do not have to install software or match a particular coding environment to view, execute, or manipulate code.

3.4.2.3.5 Technical Assignment

The technical assignments were conceptualized to meet recommendations in educational literature for open-ended, ill-defined, problem-based learning. For each assignment, students are expected to synthesize the narrative and code examples and apply the data and analysis tools to real-world applications. Each assignment requires coding and a written summary report to communicate and defend the results and conclusions. Within each module in HydroLearn, the assignment is a unit with components that specify the assigned tasks and expected deliverable. Assignments are accompanied by a customized rubric that sets expectations for students and facilitates objective grading for instructors. We adapted rubrics developed by a team of hydroinformatics instructors to each assignment (Burian et al., 2013). In another approach to assessment, HydroLearn offers rubric templates that connect the degree of student performance related to each learning objective (Lane et al., 2021).

3.4.2.3.6 Platform Challenges and Opportunities

Our experience with HydroLearn shows that it contains functionality that

addresses each of the needs for online sharing and content organization that we identified in surveys and interviews with study participants. We also experienced challenges that present opportunities for continued advancement of educational platforms. We acknowledge that others who use HydroLearn may have varied experiences, and while it is beyond the scope of this effort, there is opportunity to gain further insight by soliciting feedback from users of HydroLearn and/or other platforms. In this section, we describe our experience using HydroLearn with respect to identified criteria, and each of the following paragraphs corresponds to a category in Table 3.4. While these outcomes may be specific to HydroLearn, we anticipate that other platforms face similar challenges and may require further development to support online educational resources.

Discoverability refers to locating content using keyword searches from Internet browsers and search functionality within a platform. After creating a course on HydroLearn, it appeared in the results of basic Internet searches. Within HydroLearn, we were able to search for the course and within the course. The platform could enhance discoverability by including keywords as part of the metadata for each course or module and filtering courses on keywords.

Metadata are displayed on the course landing page. The course template suggests metadata elements, which we used (e.g., target audience, tools needed, suggested citation), but elements are optional. HydroLearn could better standardize metadata by requiring certain elements and by automatically generating elements where possible. Creating metadata requires editing HTML code, and HydroLearn could improve usability through webforms or markdown.

Navigability of HydroLearn courses is dictated by the hierarchical structure

described in Section 3.4.2.3.1. Even with a logical organization for content, moving between sections and knowing how to proceed through the module sequentially can be challenging for beginners. This may be improved by adding text to the icons in the navigation bar and by displaying a course outline and navigation in a persistent sidebar.

In Table 3.4, content refers to the types of files that are supported by the platform. We were able to use HydroLearn to share text, images, interactive websites, and to link files for download. Videos, equations, code snippets, and other HTML components are also supported. Supporting either a JupyterHub for launching notebooks or more directly integrating with the CUAHSI JupyterHub would strengthen the platform's ability to support code files.

Separate access for students and instructors is supported by HydroLearn. Course creators can elect to restrict access of certain content to course staff. Other instructors can access restricted content by exporting the course or by contacting course creators, though that may be unreliable. Although we used open-ended assignments, some require specific coding tasks. In these cases, we created scripts or notebooks as a solution key to the assignment, and we were able to use this functionality to restrict access without separating the solution from course materials.

Licenses can be specified by creators at the course level. HydroLearn supports Creative Commons licenses (e.g., Attribution, Noncommercial, No Derivatives, Share Alike), and related icons and messaging are displayed on course subsection pages. Licensing could be made clearer if displayed prominently on the course landing page.

Scalability refers to the ability for multiple users (e.g., classes of students) to use the materials or program. We have not yet tested HydroLearn in the context of multiple

simultaneous users, but we are not aware of any limitations. It is built on an established online learning platform (edX), which offers robustness. There may be scaling issues with many users running notebooks on the CUAHSI JupyterHub, for which Lane et al. (2021) observed student frustration related to losing server connection and authentication.

Reusability of educational materials is an intent of HydroLearn, and modules are expected to be designed with consideration for uptake by other instructors. While the modules described here have not yet been reused, we found it straightforward to export and customize a HydroLearn course, and Lane et al. (2021) report that adaptation of a HydroLearn course by instructors at other institutions was straightforward. Reusability is facilitated by licenses and citations, and the course metadata template includes "Adapted From" to acknowledge source material. HydroLearn courses have been used for both online and in-person instruction and can be designed to be student-paced or with an imposed schedule making them compatible to the mix of modalities reported by study participants.

Citations are a recommended (but optional) metadata element for HydroLearn courses. Creators can structure the citation as desired, and it is displayed on the course landing page. There is opportunity for the platform to standardize by automatically generating a citation for each course or module, as is done for data and code resources in HydroShare (Horsburgh et al., 2016).

Curation of courses is not required in HydroLearn, and instructors may deposit and share content without review. However, most of the modules currently available on HydroLearn were developed through intensive summer hackathons including substantive instruction on pedagogical best practices and feedback from the HydroLearn team

(Maggioni et al., 2020; Gallagher et al *in prep*). As a result, much of the educational content shared on HydroLearn meets their criteria for high quality modules. However, there is no long-term system in place for module review and curation by the project team. As our modules were developed outside of the formal hackathons, we requested the feedback of a HydroLearn team member who was able to review and offer helpful suggestions. The approach of offering but not requiring curation balances increased overhead with fostering high quality content. Also, compensating fellows increases their motivation to deposit high quality material, as noted by study participants.

Educational support refers to assistance with teaching pedagogy and tasks, and is provided by HydroLearn through multiple features. HydroLearn emphasizes learning objectives throughout course development and includes functionality for various problem types to assess student learning (e.g., multiple choice questions, open responses, advanced mathematical expressions). Following templates and recommendations, capitalizing on features, and taking advantage of review by HydroLearn staff offers an approach that will result in a robust pedagogy. Although we did not tap into all these capabilities in developing modules, this is major benefit of HydroLearn.

Collaboration is facilitated in HydroLearn through the inclusion of multiple instructors who share editing abilities and co-authorship on a course. HydroLearn also has the ability give feedback through comments. It was uncomplicated to add instructors to our course and for all authors to edit materials; however, we did not experiment with feedback.

3.4.3 Outlook for the Future of Hydroinformatics and Water Data Science Instruction

In light of the transition to online courses precipitated by the COVID-19 pandemic as well as the growing prevalence of material online, instructors may need to consider how to best bring value to their course offerings. As expressed by one interview participant:

> *"…the incentive, the value proposition of the classroom is fundamentally altered after COVID. …No matter how good somebody is at explaining something, there's always somebody better on the internet. …what really is the role of the instructor…and modern classroom? … Obviously in person, it's made easier by the fact that [students are] there. But then the question is, is it you or is it the fact that they can be around each other? …online [content] is growing and dismissing it [is naïve]."*

Several participants indicated that the merit of an organized course for students is interaction with an instructor curating content and facilitating learning. Despite the possibility of learning from purely online materials, a knowledgeable and engaged instructor still has much to offer. This echoes Rapanta et al. (2021) in identifying a teacher's role to organize and curate the learning process and recommending that instructors increase technology expertise to adapt to changing educational environments.

> *"…engagement, pre and post class discussions, office hours, a tailored curriculum to the class. …my class changes every semester based on…what I'm perceiving in lecture and what I'm hearing in office hours."*

> *"We're in an era where it's not necessarily the content that's most valuable to the students, it's me facilitating their use of the content. And so, I think that the content should be shared as broadly as possible."*

Access to educational material that is current, flexible, and reusable can help instructors adapt to the rapidly evolving field. The modules presented in this work are a first step and an invitation to the community to continue development and sharing of

content online. In this way, instructors can address the gaps we identified related to content, platform, and organization of community materials. As instructors consult the list of topics of growing importance in the field and consider which of their materials and datasets may be most useful as community resources, we envision that they will deposit modules that include relevant water-related datasets and accessible code examples with ideas for problem-based learning.

This work illustrated that materials deposited in HydroLearn are modular and adaptable, and as HydroLearn advances and usage increases, it may address the platform gap related to limited community and siloed resources. This vision depends not only on sharing content, but also on uptake by other instructors implementing, reviewing, and engaging with shared material. As articulated by study participants, reciprocity, credit, and feedback will all motivate sharing and reuse of content, which will help advance instruction in hydroinformatics and water data science. Further implementation of online educational modules may help corroborate our experience in meeting identified criteria and may point to additional challenges or gaps.

3.5 Conclusion

We interviewed and surveyed instructors that teach hydroinformatics and water data science at collegiate and professional levels to assess the current state of practice regarding topics, teaching tools, shifts to online instruction related to COVID-19, and the potential for shared online resources. Results indicated a mix of online and in-person modalities. Although nearly all courses moved online because of COVID-19, there was a strong preference for in-person learning, and most were returning to in-person teaching. However, instructors are retaining some virtual aspects that facilitated instruction,

particularly related to live coding. Student feedback and interaction were lacking in purely online modalities, leading to the conclusion that even successful online resources and tools require deliberate interpersonal components.

Instructors generally customized teaching materials to meet the demands of a rapidly developing field. Results show variety in topics currently taught and topics of growing importance, with consensus around emphasizing reproducible code development in open-source languages and competence regarding learning and selecting informatics tools. Live coding for online and in-person settings was facilitated by the growing use of online code notebooks. A key finding was a common need for technical skill development earlier in students' college experience.

We found high interest in shared online educational content, although a lack of recognition, reciprocity, community, and credit were deterrents to sharing. Although participants currently use multiple layers of miscellaneous educational platforms, there was an expressed need for common community resources. Participants reported gaps and challenges to hydroinformatics instruction related to content (water-related datasets, online notebooks, and data-driven problems), platform (community-based, facilitates discovery), and organization (modular, adaptable).

The educational modules we developed attempt to address these challenges, center around subjects of growing importance in the field, and were developed and deposited in HydroLearn, a platform for water-related educational modules. We found that HydroLearn was successful in meeting participants' criteria for a community content platform. HydroLearn has robust functionality for educational tools and pedagogy, and its scaffolding supports content sharing (i.e., metadata, citation, discoverability,

collaboration, reusability). The major drawbacks were related to an imposed hierarchical structure, and improvements could be made regarding minimum metadata requirements. These modules are a step toward developing a rich set of online resources and an active community of instructors to meet the advancements in hydroinformatics and water data science.

In conclusion, shared online resources hold promise for overcoming challenges in hydroinformatics and water data science education. As instructors are already accustomed to tailoring content for their courses, adapting online modules with a water emphasis is accessible. Current and flexible resources would help instructors keep pace with the rapid development of technology and topics in the field and maintain the value of their course and teaching for students.

3.6 Author Contributions

ASJ, JSH, and BAL conceptualized the presentation of survey and interview results with associated educational modules. ASJ formulated the survey and interview design with support from JSH and CGF. ASJ facilitated all surveys and interviews and analyzed the responses. ASJ, JSH, and CJBP created the educational modules and published them with support from BAL. ASJ wrote the manuscript with consultation and contributions from JSH, CGF, BAL, and CJBP.

3.7 Funding

This research was primarily funded by the United States National Science Foundation under grant number 1931297. Additional support for the educational/training modules was provided by the FAIR Cyber Training Fellowship program at Purdue University corresponding to National Science Foundation grant number 1829764. We

gratefully acknowledge the support we received from the HydroLearn team in setting up and sharing our educational modules. HydroLearn is supported by National Science Foundation grant number 1726965. Any opinions, findings, and conclusions or recommendations expressed are those of the authors and do not necessarily reflect the views of the National Science Foundation. Additional funding support was provided by the Utah Water Research Laboratory at Utah State University.

3.8 Acknowledgments

We gratefully acknowledge the input and expertise of the instructors who were participants in the surveys and interviews reported in this paper.

3.9 Data Availability Statement

The materials generated by and reported by this work are publicly available. The survey responses and interview transcripts are available via HydroShare (Jones et al., 2021). The educational modules are published via HydroLearn (Jones, A.S. et al., 2022) along with code and associated datasets in HydroShare (Jones et al., 2022).

REFERENCES

Abbott, M.B., Minns, A.W., Van Nievelt, W., 1994. Education and training in hydroinformatics. J. Hydraul. Res. 32, 203–214. doi:10.2166/hydro.1999.0002

Bader, N.E., Meixner, T., Gibson, C.., O'Reilly, C., Castendyk, D.N., 2015. Stream Discharge Module. https://doi.org/10.25334/V96B-NM56

Bandaragoda, C., Wen, T., 2020. Data Science in Earth and Environmental Sciences. HydroLearn. https://edx.hydrolearn.org/courses/course-v1:SyracuseUniversity+EAR601+2020_Fall/about

Beason-Abmayr, B., Caprette, D.R., Gopalan, C., 2021. Flipped teaching eased the transition from face-to-face teaching to online instruction during the COVID-19 pandemic. Adv. Physiol. Educ. 45, 384–389. https://doi.org/10.1152/advan.00248.2020

Burian, S.J., Horsburgh, J.S., Rosenberg, D.E., Ames, D.P., Hunter, L.G., Strong, C., 2013. Using interactive video conferencing for multi-institution, team-teaching. ASEE Annu. Conf. Expo. Conf. Proc. https://doi.org/10.18260/1-2--22706

Celicourt, P., Rousseau, A.N., Gumiere, S.J., Camporese, M., 2021. Editorial: Hydro-Informatics for Sustainable Water Management in Agrosystems. Front. Water 3, 1–3. https://doi.org/10.3389/frwa.2021.758634

Ceola, S., Arheimer, B., Baratti, E., Blöschl, G., Capell, R., Castellarin, A., Freer, J., Han, D., Hrachowitz, M., Hundecha, Y., Hutton, C., Lindström, G., Montanari, A., Nijzink, R., Parajka, J., Toth, E., Viglione, A., Wagener, T., 2015. Virtual laboratories: New opportunities for collaborative water science. Hydrol. Earth Syst. Sci. 19, 2101–2117. https://doi.org/10.5194/hess-19-2101-2015

Chen, Y., Han, D., 2016. Big data and hydroinformatics. J. Hydroinformatics 18, 599–614. https://doi.org/10.2166/hydro.2016.180

Daniels, L.M., Goegan, L.D., Parker, P.C., 2021. The impact of COVID-19 triggered changes to instruction and assessment on university students' self-reported motivation, engagement and perceptions. Soc. Psychol. Educ. 24, 299–318. https://doi.org/10.1007/s11218-021-09612-3

Flores, A., 2021. Open and Reproducible Computing. GitHub. https://github.com/LejoFlores/Open-And-Reproducible-Research-Computing

Gannon, J., 2021. Hydroinformatics at VT. GitHub. https://vt-hydroinformatics.github.io/

Gannon, J.P., McGuire, K.J., 2022. An Interactive Web Application Helps Students Explore Water Balance Concepts. Front. Educ. 7, 1–7. https://doi.org/10.3389/feduc.2022.873196

Garousi-Nejad, I., Lane, B.A., 2021. Hydrologic Statistics and Data Analysis (M1). HydroShare. https://www.hydroshare.org/resource/bd0b38fc5d1e4d5c895dc484ceeb2c2a/

Gibert, K., Horsburgh, J.S., Athanasiadis, I.N., Holmes, G., 2018. Environmental Data Science. Environ. Model. Softw. 106, 4–12. https://doi.org/10.1016/j.envsoft.2018.04.005

Godber, K.A., Atkins, D.R., 2021. COVID-19 Impacts on Teaching and Learning: A Collaborative Autoethnography by Two Higher Education Lecturers. Front. Educ. 6, 1–14. https://doi.org/10.3389/feduc.2021.647524

Gorelick, D., Characklis, G., 2019. Introductory R for Water Resources - Fall 2019 - Univeristy of North Carolina at Chapel Hill. ECSTATIC. https://digitalcommons.usu.edu/ecstatic_all/86/

Grus, J., 2015. Data Science from Scratch. O'Reilly.

Habib, E., Deshotel, M., Guolin, L.A.I., Miller, R., 2019. Student perceptions of an active learning module to enhance data and modeling skills in undergraduate water resources engineering education. Int. J. Eng. Educ. 35, 1353–1365. https://www.researchgate.net/publication/336812803_Student_Perceptions_of_an_Active_Learning_Module_to_Enhance_Data_and_Modeling_Skills_in_Undergraduate_Water_Resources_Engineering_Education

Habib, E., Ma, Y., Williams, D., Sharif, H.O., Hossain, F., 2012. HydroViz: Design and evaluation of a Web-based tool for improving hydrology education. Hydrol. Earth Syst. Sci. 16, 3767–3781. https://doi.org/10.5194/hess-16-3767-2012

Hamilton, A., 2021. Python for Environmental Research. MyGeoHub. https://mygeohub.org/courses/Environmentalresearch/overview

Horsburgh, J.S., 2019. Fa19 CEE 6110-001. Utah State Univ. Canvas. https://usu.instructure.com/courses/545625/pages/hydroinformatics

Horsburgh, J.S., Morsy, M.M., Castronova, A.M., Goodall, J.L., Gan, T., Yi, H., Stealey, M.J., Tarboton, D.G., 2016. HydroShare: Sharing Diverse Environmental Data Types and Models as Social Objects with Application to the Hydrology Domain. J. Am. Water Resour. Assoc. 52, 873–889. https://doi.org/10.1111/1752-1688.12363

Horsburgh, J.S., Tarboton, D.G., Maidment, D.R., Zaslavsky, I., 2008. A relational model for environmental and water resources data. Water Resour. Res. 44. https://doi.org/10.1029/2007wr006392

Jones, A.S., Horsburgh, J.S., Bastidas Pacheco, C.J., 2022. Hydroinformatics and Water Data Science. HydroLearn. https://edx.hydrolearn.org/courses/course-v1:USU+CEE6110+2022/about

Jones, A.S., Horsburgh, J.S., Bastidas Pacheco, C.J., 2022. Hydroinformatics Instruction Modules Example Code. HydroShare. http://www.hydroshare.org/resource/761d75df3eee4037b4ff656a02256d67

Jones, A.S., Horsburgh, J.S., Flint, C.G., 2021. Hydroinformatics and Water Data Science Instructor Interviews and Surveys. HydroShare. http://www.hydroshare.org/resource/15b1a61f47724a6e8deb100789353df2

Kerkez, B., 2019. CEE575 Sensors, Data, and Intelligent Systems. Univ. Michigan. http://www-personal.umich.edu/~bkerkez/courses/cee575/

Lane, B., Garousi-Nejad, I., Gallagher, M.A., Tarboton, D.G., Habib, E., 2021. An open web-based module developed to advance data-driven hydrologic process learning. Hydrol. Process. 1–15. https://doi.org/10.1002/hyp.14273

Maggioni, V., Girotto, M., Habib, E., Gallagher, M.A., 2020. Building an online learning module for satellite remote sensing applications in hydrologic science. Remote Sens. 12, 1–16. https://doi.org/10.3390/RS12183009

Makropoulos, C., 2019. Urban Hydroinformatics: Past, Present and Future. Water 11. https://doi.org/https://doi.org/10.3390/w11101959

McGovern, A., Allen, J., 2021. Training the Next Generation of Physical Data Scientists. Eos (Washington. DC). 102, 1–9. https://doi.org/10.1029/2021EO210536

Merck, M.F., Gallagher, M.A., Habib, E., Tarboton, D., 2021. Engineering Students' Perceptions of Mathematical Modeling in a Learning Module Centered on a Hydrologic Design Case Study. Int. J. Res. Undergrad. Math. Educ. https://doi.org/10.1007/s40753-020-00131-8

Merwade, V., Ruddell, B.L., 2012. Moving university hydrology education forward with community-based geoinformatics, data and modeling resources. Hydrol. Earth Syst. Sci. 16, 2393–2404. https://doi.org/10.5194/hess-16-2393-2012

Nearing, G.S., Kratzert, F., Sampson, A.K., Craig, S., Frame, J.M., Klotz, D., Gupta, H. V, 2020. What Role Does Hydrological Science Play in the Age of Machine Learning ? Water Resour. Res. 1–17. https://doi.org/10.31223/osf.io/3sx6g

Ngambeki, I., Thompson, S.E., Troch, P.A., Sivapalan, M., Evangelou, D., 2012. Engaging the students of today and preparing the catchment hydrologists of tomorrow: student-centered approaches in hydrology education. Hydrol. Earth Syst. Sci. Discuss. 9, 707–740. https://doi.org/10.5194/hessd-9-707-2012

Peek, R., Pauloo, R., 2021. R for Water Resources Data Science. https://www.r4wrds.com/

Popescu, I., Jonoski, A., Bhattacharya, B., 2012. Experiences from online and classroom education in hydroinformatics. Hydrol. Earth Syst. Sci. 16, 3935–3944. https://doi.org/10.5194/hess-16-3935-2012

Rapanta, C., Botturi, L., Goodyear, P., Guàrdia, L., Koole, M., 2021. Balancing Technology, Pedagogy and the New Normal: Post-pandemic Challenges for Higher Education. Postdigital Sci. Educ. 3, 715–742. https://doi.org/10.1007/s42438-021-00249-1

Ruddell, B.L., Wagener, T., 2015. Grand Challenges for Hydrology Education in the 21st Century. J. Hydrol. Eng. 20, 1–8. https://doi.org/10.1061/(asce)he.1943-5584.0000956

Seibert, J., Uhlenbrook, S., Wagener, T., 2013. Preface Hydrology education in a changing world. Hydrol. Earth Syst. Sci. 17, 1393–1399. https://doi.org/10.5194/hess-17-1393-2013

Shen, C., 2018. Deep Learning : A Next-Generation Big-Data Approach for Hydrology. Eos (Washington. DC). 1–4. https://doi.org/https://doi.org/10.1029/2018EO095649

Smith, C., Praphamontripong, U., 2021. Analysis of the transition to a virtual learning semester in a college software testing course, EASEAI 2021 - Proceedings of the 3rd International Workshop on Education through Advanced Software Engineering and Artificial Intelligence, co-located with ESEC/FSE 2021. Association for Computing Machinery. https://doi.org/10.1145/3472673.3473967

VanZuylen, H.J., Dee, D.P., Mynett, A.E., Rodenhuis, G.S., Moll, J.., Ogink, H.J.M., Most, H.V.D., Gerritsen, H., Verboom, G.K., 1994. Hydroinformatics at Delft Hydraulics. J. Hydraul. Res. 32, 83–136. https://doi.org/https://doi.org/10.1080/00221689409498806

Vojinovic, Z., Abbott, M.B., 2017. Twenty-five years of hydroinformatics. Water (Switzerland) 9, 1–11. https://doi.org/10.3390/w9010059

Wagener, T., Kelleher, C., Weiler, M., McGlynn, B., Gooseff, M., Marshall, L., Meixner, T., McGuire, K., Gregg, S., Sharma, P., Zappe, S., 2012. It takes a community to raise a hydrologist: The Modular Curriculum for Hydrologic Advancement (MOCHA). Hydrol. Earth Syst. Sci. 16, 3405–3418. https://doi.org/10.5194/hess-16-3405-2012

Wagener, T., McIntyre, N., 2007. Tools for teaching hydrological and environmental modeling. Comput. Educ. J. 17, 16–26. https://coed.asee.org/wp-content/uploads/2020/08/3-Tools-for-Teaching-Hydrological-and-Environmental-Modeling.pdf

Wagener, T., Savic, D., Butler, D., Ahmadian, R., Arnot, T., Dawes, J., Djordjevic, S., Falconer, R., Farmani, R., Ford, D., Hofman, J., Kapelan, Z., Pan, S., Woods, R., 2021. Hydroinformatics education-the Water Informatics in Science and Engineering (WISE) Centre for Doctoral Training. Hydrol. Earth Syst. Sci. 25, 2721–2738. https://doi.org/10.5194/hess-25-2721-2021

Wagener, T., Weiler, M., McGlynn, B.L., Gooseff, M., Meixner, T., Marshall, L., McGuire, K., McHale, M., 2007. Taking the pulse of hydrology education. Hydrol. Process. 21, 1789–1792. https://doi.org/10.1002/hyp

Ward, A.S., Herzog, S., Bales, J., Barnes, R., Ross, M., Jefferson, A., Basu, N., Yoder, L., Covino, T., Habib, E., Maertens, J., 2021. Educational Resources for Hydrology & Water Resources. HydroShare. http://www.hydroshare.org/resource/148b1ce4e308427ebf58379d48a17b91

TABLES

Table 3.1 Survey/interview questions

Survey/Interview Questions
The term "hydroinformatics" is used throughout. If your course or program uses a different title or term (e.g., "water data science"), consider that term instead.
Course Details
What is the name of the hydroinformatics-related course/program at your institution?
Is this course/program taught at a graduate level?
Are any hydroinformatics topics taught at an undergraduate level?
How is "hydroinformatics" defined in the context of the course/program offered at your institution?
What are the objectives for the hydroinformatics related course/courses/or programs offered at your institution?
Course Expectations
What prerequisite informatics skills are expected of students?
Do most students exhibit the prerequisite informatics skills at the start of the course?
What informatics skills (and level of skill) are students expected to attain in this course?
What benefits have students derived from taking the course? This could be quantitative or anecdotal.
Formats
What are the sources of the teaching materials used for the course/program?
What is the course/program format? (e.g., in-person, online, etc.) Please clarify if this changed due to COVID.
What platforms or instructional tools are being used in course delivery? (e.g., Canvas, HydroLearn, MyGeoHub, HydroShare, etc.) Please clarify if this changed due to COVID.
Did the COVID pandemic impact instruction related to hydroinformatics courses at your institution? If so, how?
What platforms or instructional tools have proven effective for in person versus online instruction (if your course has been offered online)?
If your courses have been offered online (due to covid or other reasons), what were the biggest challenges in delivering online instruction?
Topics and Technologies
What topics are emphasized in the hydroinformatics courses at your institution? (e.g., machine learning, databases and data models, numerical modeling)
What informatics technologies are emphasized? (e.g., Python, R, MySQL, ArcGIS)
What (if any) geospatial data and techniques are covered in the hydroinformatics course(s) at your institution?
How have the topics and technologies changed over the time that the course(s) have been taught?
What topics and technologies are growing in importance in hydroinformatics?
What are the gaps in existing hydroinformatics instruction/education?
Shared Resources
What **types** of shared community resources for instruction would be useful? (e.g., online modules that could be incorporated into courses)
In developing shared resources, what **topics** would be helpful in addressing gaps and challenges?
What **formats** would be conducive to shared resources?
What informatics **technologies** would be useful for shared resources?
What is your level of interest in sharing and exchanging teaching resources and materials with the community? (Very Interested, Interested, Moderately Interested, Slightly Interested, Not Interested)
What would motivate hydroinformatics instructors to participate in sharing/exchanging teaching resources?
In your view, what resources would a useful shared educational module consist of?
Wrap Up
Do you know of any other instructors who would be a good fit for this survey/interview? Please provide a name, institution, and email address (if known).

Table 3.2 Courses taught by study participants.

Course Titles	Count	Audience
Hydroinformatics	5	Graduate (4), Undergraduate and Graduate (1)
Informatics for Sustainable Systems	1	Graduate
Physical Hydrology (with a Hydroinformatics Unit)	1	Undergraduate and Graduate
Intro to Environmental Data Science	1	Graduate
Water Resource Data Science Applications	1	Graduate
Earth Data Science	1	Graduate
Ecological and Environmental Data and Tools	1	Graduate
Introduction to Data Science	2	Undergraduate and Professional
R for Water Resources Data Science	1	Professional
R for Water Resources Research	1	Undergraduate and Graduate
Python for Environmental Research	1	Graduate
Research Computing in Earth and Environmental Sciences	1	Graduate
Modeling Earth and Environmental Systems	1	Graduate
Computational Watershed Hydrology	1	Undergraduate and Graduate
Data Analysis for Water Quality Management	1	Graduate
Sensing and Data	1	Graduate

Table 3.3 Educational platforms and instances of hydroinformatics or related implementations.

Platform	Description	Examples
HydroLearn https://www.hydrolearn.org/	Specifically designed for instructors to post and share educational modules for hydrology and water resources	(Bandaragoda and Wen, 2020)
MyGeoHub https://mygeohub.org/courses	Hosts groups, datasets, tools, and educational content for geoscience research and education	(Hamilton, 2021)
environmental data-driven inquiry and exploration (eddie) https://serc.carleton.edu/eddie/index.html	Repository for classroom modules and datasets for environmental subjects	No hydroinformatics or water data science modules. Stream Discharge Module: (Bader et al., 2015)
Excellence in Systems Analysis Teaching and Innovative Communication (ECSTATIC) https://digitalcommons.usu.edu/ecstatic/	Repository for water resources systems analysis teaching and communication materials	(Gorelick and Characklis, 2019)
HydroShare https://www.hydroshare.org/	Repository for sharing water related data, models, and code. HydroShare is generally focused on data and code, but several instructors have also used it for educational materials.	(Garousi-Nejad and Lane, 2021; Ward et al., 2021)
GitHub https://github.com/	Repository for software and code with version control	(Flores, 2021)
Personal or institutional website	Users determine structure	(Kerkez, 2019)
Canvas (or similar)	Institutional learning management system	(Horsburgh, 2019)
Customized books/websites	Users determine structure. Some programming languages have packages to convert code to an online book or website.	(Gannon, 2021; Peek and Pauloo, 2021)

Table 3.4 Characteristics of educational platforms related to instructor-defined criteria.

Platform	Discoverability	Metadata	Navigability	Content	Student/Instructor Access	Licenses	Scalability	Reusability	Citation	Curation	Education Support	Collaboration
HydroLearn	Searchable, indexed for Internet search	User-defined metadata	Hierarchical structure. Expandable navigation menu.	Text, videos, links to files and webpages	Supports separate access	Creative commons licenses	Not expected to be an issue	Expected	User-defined	Available but optional	Learning objectives, discussions, many problem types	Comment and creati derivative supported
MyGeoHub	Searchable, keywords, indexed for Internet search	Basic description	Courses with modules containing files	Any file type. Natively run Jupyter notebooks	Not explicit support, but could be achieved with groups	Creative commons licenses	Some issues reported for multiple users running notebooks	Unclear	Citation generated but not obvious on landing page	Approval required for uploading files	Quizzes, exams, homework, discussions	Participan may comment
eddie	Searchable, filterable, indexed for Internet search	Detailed outline	Outline with links to files	Any file type	Supports separate access	Unclear	Unclear	Expected	Unclear	Multistep review process	Structured around teaching objective	Unclear
ECSTATIC	Searchable, filterable by type	Abstract and keywords	All content in zip file	Any file type	No	Present on landing page	No issues	Expected	Included	Very light review	None	None
HydroShare	Searchable, filterable, indexed for Internet search	Abstract and keywords		Any file type. Natively run Jupyter notebooks with data files.	Could be achieved using different privacy levels	Present on landing page	Could occur if there are many users on the Jupyter Hub server	Expected	Included	None	None	Comment and group
GitHub	Searchable, but difficult	Minimal metadata required	Creators can structure files as desired	Any file type. Code and markdown rendered.	Could be achieved using different privacy levels	Available but not required	No issues	Expected	Can be generated	None	None	Facilitate by forkin another repository
Canvas (or similar)	Only if user knows what to look for	Creators can include as much as desired	Predetermined structure with some customization	Any file type	Separate access for creator but not for resuse	Possibly	No issues	Unclear	Possibly	None	Quizzes, exams, homework, discussions	Potential collaborat
Customized books or websites	Only if user knows what to look for	Creators can include as much as desired	Creators can structure files as desired	Any file type	Separate access for creator but not for reuse	Possibly	No issues	Unclear	Possibly	None	None	None

Table 3.5 Educational modules developed and deployed as part of this work with descriptions of essential components and datasets. Modules are accessed at Jones, A.S. et al., (2022).

Module	Programmatic data access	Sensor data quality control	Databases and SQL	Machine learning classification
Topics	• Open web technology • High frequency data • Visualization • Big data	• High frequency data • Data quality • Big data • Machine learning	• Databases and SQL • High frequency data • Big data	• Machine learning • Smart sensors • High frequency data
Narrative	• The United States Geological Survey (USGS) National Water Information System (NWIS) • Web services for accessing data	• Data life cycle for *in situ* aquatic sensor data • Sensors, hardware, and infrastructure • Sensor data quality assurance and quality control	• Data models and database implementation • SQL queries (e.g., selecting, joining, and aggregating data) • Observations Data Model (ODM, Horsburgh et al., 2008)	• Common machine learning approaches, concepts, and algorithms • Python package scikit-learn Problem of labeling residential water end use event data
Code Examples	• Use the Python dataretrieval package • Import and plot data via USGS NWIS web service endpoints • Examine local hydrology using flow statistics	• Import and plot a time series • Use the Python pyhydroqc package • Perform rules-based and model-based anomaly detection	• Use SQL to select data, sort results, perform joins between tables, aggregate and group data	• Explore data features • Apply basic machine learning model • Compare multiple algorithms • Hyperparameter tuning and optimization
Assignment	Retrieve data, calculate statistics, and generate plots to explain the impact and severity of drought conditions	Apply package algorithms and determine performance metrics to consider using the software in an observatory quality control workflow	Construct SQL queries to compare data to state water quality criteria and identify potential water temperature impairment	Apply machine learning models to develop guidance for using smart meters to collect residential water use data
Dataset	Water data collected by national agency available via web. Similar data/methods may be available for data from other agencies.	Flat files in containing high frequency Logan River aquatic data with raw data and technician labels. Posted on HydroShare.	SQLite ODM database with high frequency water temperature data for several sites in the Logan River. Posted on HydroShare.	Flat file of labeled residential water use event data. Posted on HydroShare.

FIGURES

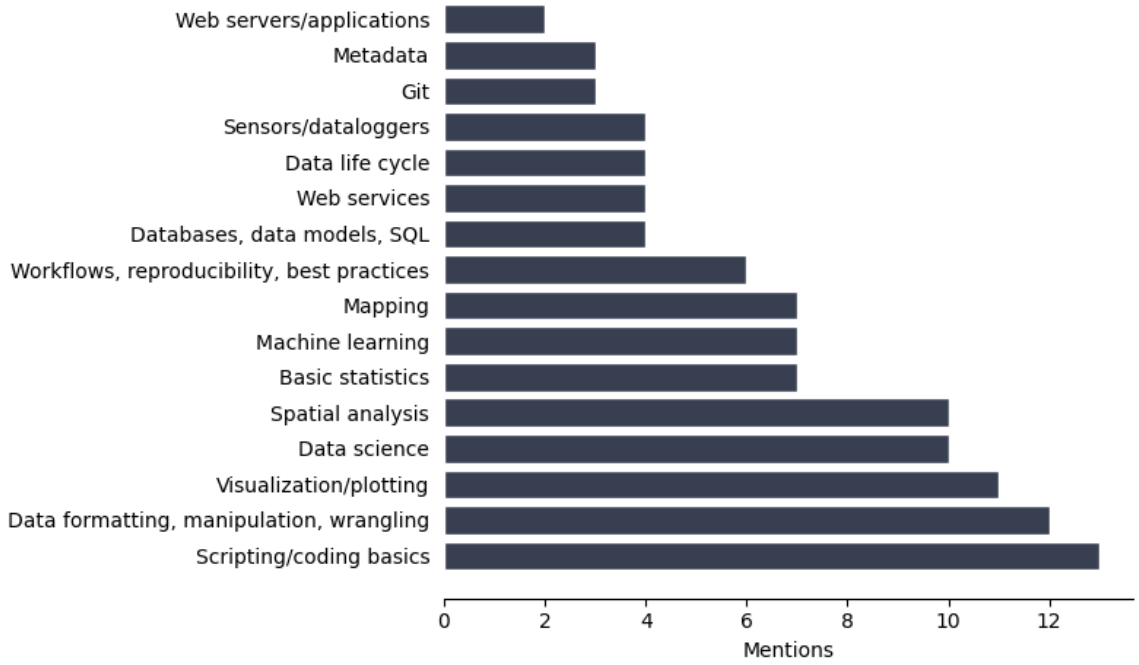

Figure 3.1 Count of mentions related to subjects taught by participants.

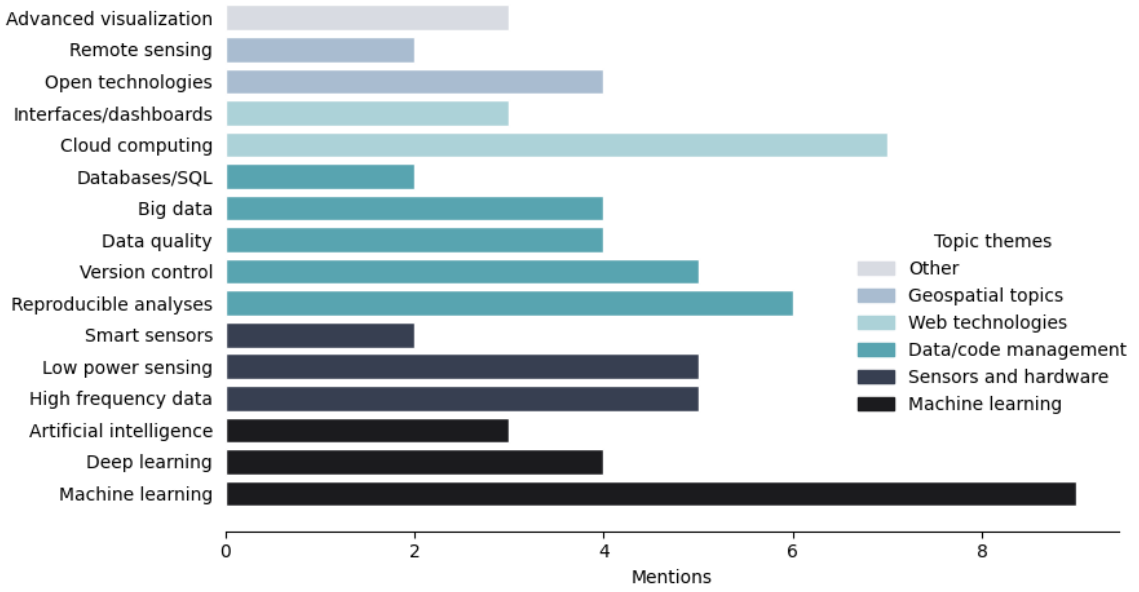

Figure 3.2 Count of mentions related to subjects of growing importance sorted by thematic topics.

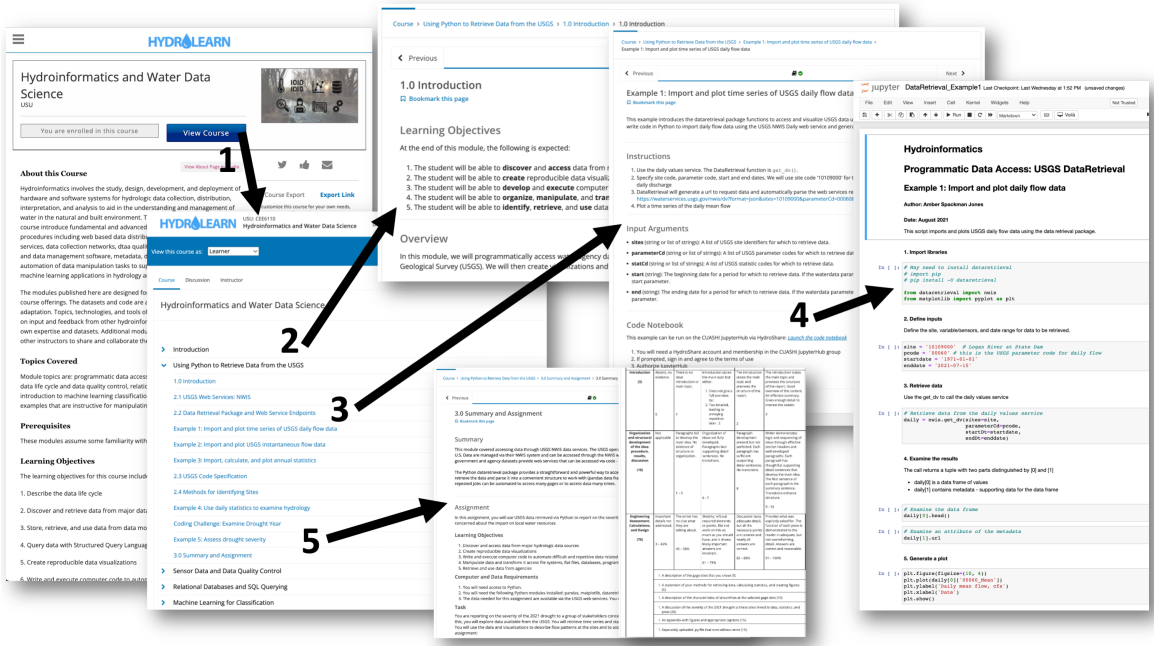

Figure 3.3 Module implementation in HydroLearn. The numbered steps indicate the order of workflow and the location of essential module elements: 1) the course landing page contains metadata and links to a course outline, 2) learning objectives in the module introduction, 3) the narrative consists of text, links, images, tables, and code snippets, 4) code examples are interactive notebooks in the CUAHSI JupyterHub linked from HydroLearn, and 5) the technical assignment and associated rubric are a separate module component.

CHAPTER 4

HYDROLOGIC INFORMATION SYSTEMS: AN INTRODUCTORY OVERVIEW[3]

Abstract

Hydrologic Information Systems (HIS) integrate hardware and software to support collection, management, and sharing of hydrologic observations data. Successful HIS facilitate hydrologic monitoring, scientific investigation, watershed management, and communication of hydrologic conditions. Furthermore, HIS support the day-to-day data operations that are essential to organizations that monitor hydrologic systems. As an introductory overview of HIS, this paper reviews the history of HIS development and identifies and describes key components. Based on past HIS literature, patterns emerged for universal and generic HIS functionality and componentry. The main data pools are collection/acquisition, operational storage, and sharing/publication/dissemination with data flux occurring between pools. Persistent and contemporary challenges for HIS are identified, and examples of current and emerging HIS are described in the context of how they are addressing these challenges. Opportunities remain for coordinated community efforts to address outstanding barriers, advance HIS, and further enable hydrologic science.

4.1 Introduction

Hydrologic Information Systems (HIS) consist of the integrated hardware and software systems used to collect, store, manage, and disseminate water observations data and metadata to support monitoring, modeling, and management of water resources

[3] Co-authored by Amber Spackman Jones, Jeffery S. Horsburgh

(Bandaragoda et al., 2006; Hooper et al., 2004; Maidment, 2005; Soh et al., 2006). As barriers to environmental monitoring have decreased (e.g., cost, communications, power, expertise), water observations data are being generated at greater volumes, at finer spatial and temporal resolutions, and over longer durations and greater spatial extents (Benson et al., 2010; Bieroza et al., 2023; Blaen et al., 2016; Dow et al., 2015; Horsburgh et al., 2019; Laney et al., 2015; Pellerin et al., 2016; Porter et al., 2012; Turner et al., 2020). Supporting the full life cycle of these data requires cyberinfrastructure applications and tools deployed as hardware and software (Mason et al., 2014). Successful HIS enable scientific investigations and analysis as well as water resource management by providing reliable and accessible water data and information (Dow et al., 2015; McGuire et al., 2016; Muste et al., 2013) and reducing the time and effort between data collection and analysis (Samourkasidis et al., 2019). HIS support data dissemination to end users, usually with visualization capabilities that aid in data exploration, interpretation, and communication. As water data are being collected by diverse groups with distinct needs – e.g., national, state, and local agencies; research groups; nonprofit organizations; and citizen groups – scientists and technicians engaged in water data collection need HIS to support their needs for collecting, managing, and sharing hydrologic data. HIS also serve data consumers who rely on monitoring data for assessment and operations.

In this paper, we present a synthesis and overview of HIS specifically focused on cyberinfrastructure to support time series of observational data at fixed monitoring stations. We acknowledge that some cyberinfrastructure systems integrate heterogeneous data types including ecological data (Rüegg et al., 2014; Rundel et al., 2009), spatial data (Henzen et al., 2016; Ruddell et al., 2014; Yang et al., 2010), discrete measurements (Hsu

et al., 2017), mobile platforms (Coopersmith et al., 2007; Viqueira et al., 2020), and modeling processes and results (Hill et al., 2011; McGuire et al., 2016; Muste et al., 2013). While systems supporting multiple types of earth-based observations may provide useful flexibility (Horsburgh et al., 2016), they often do so at the cost of significant complexity. In this overview, we chose to emphasize systems focused specifically on observational time series as an important data type with key functionality for many water monitoring systems (Soh et al., 2006). The objectives of this paper are to: 1) establish an introductory overview of HIS, defining universal, generic HIS functionality and componentry based on a review of existing literature, 2) describe implementations of HIS in current and emerging systems, and 3) identify gaps and deficiencies as opportunities for improving operational HIS. Section 4.2 describes the methods we used in tracing the evolution of HIS, and Section 4.3 presents the resulting history and review of HIS. Informed by this review, Section 4.4, identifies and explains the essential components and functions of HIS. Section 4.5 presents persistent challenges for HIS along with approaches that some current and developing HIS are using the address them. Finally, in Section 4.6, we offer an outlook for HIS going forward.

4.2 Methods

To identify important components and functionality of HIS, we performed a review of relevant literature. Initially, a few articles on HIS that were known to the authors were considered, and additional relevant articles were identified by reviewing reference lists and by tracking the citations of those key articles. Papers were also selected via Google Scholar and Scopus using relevant keywords: “hydrologic information systems”, “water data management”, and “hydroinformatics”. Papers were

considered germane and included in the review if observational time series were supported as a key data type. The literature search included conference proceedings but did not include presentation abstracts due to the lack of detailed information. However, we acknowledge that, given the operational nature of HIS, presentations and talks at conferences are emblematic of much HIS-related development work that may not have been published more formally because it was not viewed as publishable research. In general, when research reported in the scientific literature involves hydrologic data, the methods associated with collection, management, quality control, processing, and dissemination are often presented only cursorily (Jones et al., 2017; Lundquist et al., 2015) because the focus is usually on new hydrologic or process understanding rather than systems used to collect and manage the data. These factors limit the available literature to describe operational HIS in detail.

We also sought to gather information from currently operational implementations of HIS not formally described in the research literature. To identify these systems, we contacted representatives of water monitoring networks and included information on systems that the authors have worked on (e.g., the United States Geological Survey (USGS) National Water Information System (NWIS)). In all cases, we limited the search to systems that support fixed location time series, although the reported HIS may support multiple data types. HIS have been implemented internationally; however, we found more literature describing systems within the United States, and our experience is in the U.S., thus the discussion in this paper is mostly U.S.-focused. We acknowledge that the coverage of HIS by this paper is not comprehensive; however, the systems described and reported here cover major known networks and bracket existing functionality that we

believe to be sufficiently representative.

4.3 Hydrologic Information Systems: History and Review

Most published literature related to HIS either describes specific components of HIS or implementations of HIS for a specific workflow, use case, or data type. As a result, it is impractical to directly compare systems described by each paper. Instead, this section provides important background and history of HIS, and the following section presents key functional components of HIS based on a review and synthesis of the literature. Scopus identified 65 documents (articles, conference abstracts, etc.) with the keyword "hydrologic information systems" (Figure 4.1) dating back to 2004, which marked a major push to apply advances in information technology to the hydrology domain (Hooper et al., 2004).

Prior to 2004, references to "hydrologic information systems" (n=10) were closely tied to spatial data management. HIS was a natural derivative of the established "geographic information systems" (GIS) (Lee et al., 2004), and the ArcHydro data model and related tools were developed to support hydrologic time series and analysis within GIS (Maidment, 2002). Limited early references to HIS for time series data include a relational database that separated time series data from site metadata and supported graphing and statistical tools (Gandolfi and Wethner, 1987) and a description of satellite data transmission for the USGS (Shope Jr., 1987). Initially released in the mid-1990s, USGS NWIS was an early HIS that integrated data from USGS stream gages, wells, and water quality sites and made it uniformly accessible to the public over the Internet (Blodgett et al., 2016a). While not well-represented in the literature because it was built as an operational system rather than a research effort, NWIS was foundational to HIS

development. Making so much data easily available to the public was a major milestone in water data management, and NWIS remains the world's largest enterprise HIS (Blodgett et al., 2016a). NWIS illustrated that making data accessible facilitates advancement of hydrologic research and understanding, which has been exemplary to the development of HIS for other agencies and monitoring efforts around the world (Shukla et al., 2019). Based on our review, we conclude that the definition of "hydrologic information systems" as stated in the first paragraph of this paper has not changed significantly over time, although technological advancements have influenced the mechanisms and applications of HIS.

The majority of the publications from 2004-2009 are related to development of an HIS by the Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI), an effort funded by the U.S. National Science Foundation (Hooper et al., 2004). CUAHSI set out to design and develop the first comprehensive HIS, which they defined as a database coupled with tools for data acquisition and tools for data analysis, visualization, and modeling (Maidment, 2008; Hooper et al., 2004). Along with university partners, CUAHSI designed and implemented a national scale HIS that initially deployed HIS tools for a set of "test bed" water monitoring networks (Coopersmith et al., 2007; Horsburgh et al., 2011). Following that initial deployment, numerous practitioners and new users were consulted to achieve broader buy-in from the hydrologic science community (Bandaragoda et al., 2006). Identified challenges included mediating across disparate data formats and organizations, creating consistent metadata, supporting data from various geographic areas, and documentation of data quality (Hooper et al., 2004; Bandaragoda et al., 2006). To address these challenges, CUAHSI

development focused on a community observations data model (ODM) (Horsburgh et al., 2008, 2005), web services for data transfer (Goodall et al., 2008; Maidment et al., 2006), standardized terminology (Horsburgh et al., 2014; Piasecki and Beran, 2009), and tools for data access and visualization (Ames et al., 2012; Beran and Piasecki, 2009; Maidment, 2008).

As the CUAHSI HIS matured, hydrologic monitoring networks implemented or adapted CUAHSI-developed components along with custom software applications to support data workflows, described by several publications from 2010-2015 (e.g., Conner et al., 2013; Horsburgh et al., 2011; Jones et al., 2015; Mason et al., 2014; Muste et al., 2013). Distributed data sources registered their data with CUAHSI's HIS Central, which was a metadata catalog and source for data discovery and access for the federated system (Ames et al., 2012). During this period, commercial products for hydrologic data management were developed concurrently with the academic and community approaches, including offerings from Aquatic Informatics and Kisters. Some commercial products offer a comprehensive HIS, while others focus on specific components or functions in an HIS (e.g., software from a sensor or datalogger manufacturer to support acquisition of data from their commercial devices) (ESIP EnviroSensing Cluster, 2014). In parallel to efforts in the hydrology community, collaborative and/or large-scale monitoring networks in ecology (e.g., the Long Term Ecological Research (LTER) Network and National Ecological Observing Network (NEON)) built operational systems that incorporated monitoring of water with sensors and similarly grappled with data management. Because sensor data are part of larger ecological datasets and data collection efforts that contain diverse classes of data, the metadata and data management requirements resulted in

approaches less specifically focused than other HIS development. However, similar to other operational cyberinfrastructure development and data management work, there are few detailed publications available to describe these activities.

In more recent years (2016-2023), several HIS described in the literature were designed to support increasingly popular low cost sensing platforms using off-the-shelf microcontrollers as alternatives to traditional sensing and data logging technology (Celicourt et al., 2023; Horsburgh et al., 2019; Mao et al., 2019; Sadler et al., 2016). During this period, monitoring networks began using emerging protocols and standards for data transfer (Ventura et al., 2019), including technologies associated with the Internet of Things (IoT), which refers to devices or sensors connected to the cloud and to each other and supporting communication technology (Wong and Kerkez, 2016). Thus, advances in available technology have influenced the mechanisms used for sensing, logging, communicating, and managing the data after collection, including the development of specific standards for low-power, Internet connected sensing devices (e.g., the Open Geospatial Consortium's SensorThings API specification (Open Geopatial Consoritum, 2021)).

We postulate that the rise in published papers during the 2004-2014 period and the more recent decline indicates that many research challenges were addressed, that software and system developments moved from research to operations, and that there was a subsequent decline in HIS-related research literature. Another possible confounding factor is that with newer technologies and approaches (e.g., IoT), the terminology used to describe systems may have changed sufficiently that newer literature may not be recognized as describing "HIS". Indeed, while the need for operational HIS has not

changed, the technology landscape has changed. Many of the systems described in this review were developed using technologies and/or standards that are becoming outdated, and none of them has emerged as a de facto standard that is widely used and maintained by broad communities of scientists collecting and managing time series of hydrologic observations. These drivers warrant a new push for research and development focused on modernizing HIS. Along with synthesizing key components of HIS (Section 4.4), this paper describes how some monitoring systems are currently implementing HIS and identifies opportunities for the next steps for HIS development and modernization (Section 4.5).

4.4 Generalized HIS

Based on reviewed studies of past HIS efforts, commonalities emerged in the functional components of HIS (Braud et al., 2022; Jones et al., 2015; Samourkasidis et al., 2019; Slawecki et al., 2017; Varadharajan et al., 2019; Wong and Kerkez, 2016), which we synthesize here as a generalized HIS and data workflow (Figure 4.2). We consider components to be either "data pools" in which data reside or "data fluxes" in which data are transferred. We identify the following as major data pools in HIS, and data flux between pools is mediated by data transformation and transmission steps:

Collection and Acquisition: data are observed, stored on sensors and/or dataloggers, and input to operational storage systems via sensor-to-storage transfer.

Operational Storage: in this component, which is typically centralized, data are stored along with metadata and accessed for operations such as management, curation, and processing - steps that may involve data transfer.

Publication, Sharing, and Exchange: data are accessed by and output to end

users who create and store a copy for their use. Storage-to-sharing transfer is enabled by web user interfaces or web services for automated retrieval and may be facilitated by data registries and catalogs.

These data pools are parallel to steps in the generic data life cycle (Rüegg et al., 2014; Ventura et al., 2019) as well as the components of environmental data infrastructures identified in a review by Viqueira et al. (2020), which they described as data acquisition and integration, data and metadata storage, and data searching and browsing. Braud et al. (2022) also describe data transfer from sources to storage and then to users as information flux. Other time series HIS emphasize "shepherding" data from generation to publication (Mason et al., 2014), and different practitioners that we have spoken with refer to supporting hydrologic time series data from "gage to page" or from "stream to screen". Adjacent and parallel to these data pools are steps for data management, curation, and processing, important HIS functions (ESIP EnviroSensing Cluster, 2014; Horsburgh et al., 2011; Jones et al., 2015; Shukla et al., 2019), which, along with the major pools, are described in a subsection below.

For both data storage (pools) and data transmission (fluxes), software applications are generally based on a common information model that describes the data in the domain - in this case, time series of hydrologic observations at fixed monitoring locations. The information model can be physically implemented in various data models or data encodings depending on the functional requirements - i.e., to support storage, exchange, or cataloging. The following subsections describe each generalized HIS component including how past studies have applied the components, considerations for implementing an information model for fixed point time series observations, and ongoing

challenges and developments.

4.4.1 Collection and Acquisition

Hydrologic time series are observed at fixed locations using a variety of sensing technologies. Sensor observations are typically recorded and stored on dataloggers either onboard the sensor or on independent platforms. Datalogger platforms range from proprietary hardware systems (Horsburgh et al., 2011) to low cost, custom microcontrollers (Mao et al., 2019). Data are stored on dataloggers at spatially distributed sites and are transmitted to centralized systems at user-defined time intervals using automated telemetry (e.g., satellite, spread spectrum radios, cellular modems, etc.) or manually (Jones et al., 2017). Once data are transferred to a server, they are loaded into operational data stores automatically (Gries et al., 2016; Jones et al., 2015; Wong and Kerkez, 2016) or manually (Conner et al., 2013). Other HIS acquire time series data from external sources (e.g., public agencies) (McGuire et al., 2016) using web-scraping mechanisms such as source-specific templates or configurations that are processed to ingest the source data with associated metadata (Samourkasidis et al., 2019).

In an HIS described by Horsburgh et al. (2011), sensors, dataloggers, and radios were used to collect, record, and transmit data from field sites to central servers where they were initially stored in comma-separated values (CSV) text files. Data were then automatically loaded from those files into relational databases using a Streaming Data Loader (SDL) software application. Developed for automating the loading of data into a relational database (Observations Data Model: ODM, Section 4.4.2), the SDL was automatically executed as a scheduled job on the server after a data manager mapped metadata for each data stream (i.e., data column in a CSV datalogger file). As freely

available, open source software, the SDL was implemented as the sensor-to-server component for multiple environmental monitoring networks (Jones et al., 2017, 2015; Muste et al., 2013). Similarly, Gries et al. (2016) used custom R scripts for insertion of data from the proprietary LoggerNet software into central data stores. In another approach, Ventura et al. (2019) triggered loading of new data into operational data stores by adding new data files to a strict file structure with metadata for observed properties at each site stored in associated XML configuration files.

For the USGS NWIS, observations are recorded on various commercial dataloggers and transmitted via several different telemetry types. The majority of sites are equipped with satellite telemetry systems - either Geostationary Operational Environmental Satellite (GOES) government-operated weather satellites that transmit to local readout ground stations, or Iridium commercially operated satellites. A small percentage (~10 percent) of USGS' approximately 12,000 telemetered sites use cellular or radio telemetry. The USGS NWIS uses a custom decoding application (known within USGS as DECAP) that integrates observations from all telemetry sources, applies mappings of metadata, and parses the data into the central data store.

Advancements in off-the-shelf microcontrollers and low cost sensors prompted alternative methods for automating data acquisition. One approach used PHP scripts to retrieve observations from platforms (Arduino microcontrollers connected to cell phone modems) and parse data into a structured query language (SQL) query for insertion to a relational database (Sadler et al., 2016). In another implementation, monitoring data from remote sites were pushed to a relational database by using web service HTTP POST requests through a representational state transfer (REST) web service application

programming interface (API) (Horsburgh et al., 2019). Similarly Ventura et al. (2019) used a REST API to insert sensor observations from remote sites into a centralized relational database. To avoid mapping data columns after data collection, as is done in many of these examples, Celicourt et al. (2023) prototyped a system with metadata specified prior to sensor deployment via Python modules installed on low cost microcontrollers, one of which may act as a field base station to handle data operations including aggregating data from other remote sites rather than having every site transmit to a central server.

For the various approaches to acquiring data in distributed systems, an important distinction is between pull- and push-based architectures. In pull-based architectures, a central system initiates a request to the remote logger or device, and data are returned only when requested. Pull-based architectures are predictable, usually use some sort of static addressing system for connected monitoring sites, may use proprietary communication protocols (e.g., Campbell Scientific's PakBus networking), may require data caching at the measurement site if data are to be recorded between pulls, and work well for situations with intermittent or periodic access to telemetry (Gries et al., 2016). Many commercially available dataloggers are capable of pull-based architectures, and web-scraping approaches also use pull-based architectures. Conversely, in push-based architectures, remote data collection nodes initiate data transfer to a centralized data store. With many push-based approaches, dataloggers or sensors use standardized and open communication protocols like HTTP to push data, but they may also use more specific protocols like message queuing telemetry transport (MQTT, OASIS, 2019). Push-based systems can have lower power requirements because they do not need to

listen for requests from a central system. Many IoT and related applications use push-based architectures, and they are especially important for systems requiring event-based or real time data (e.g., reporting an event when it happens rather than based on a predefined data download interval).

4.4.2 Operational Storage

An operational data store involves a location, structure, and technology for storing and describing data. In many reported cases, a centralized server is the physical location of storage, and the structure is an implementation of an information model. An early information model for hydrologic time series, ArcHydro Time Series, related time series data to geographic features in GIS based on the observed variable and datetime of observation (Maidment, 2005, 2002). This combination of time-site-variable to support each observation was used as the basis for multiple specific implementations for data storage and data encoding but with additional metadata because the attributes in ArcHydro were insufficient to describe data for unambiguous interpretation. Similarly, the Open Geospatial Consortium later released Observations and Measurements (O&M) as a standard to support consistent description and encoding of observed data (Cox, 2013). The O&M information model has served as the basis for several environmental data management cyberinfrastructures with more specific profiles that adapt O&M's generic information model to a more specific domain and use cases (e.g., OGC's Sensor Web Enablement suite (Botts et al., 2013) and WaterML (OGC, 2014)).

A major objective of data storage and data encoding is addressing syntactic (i.e., data and file structure) and semantic (i.e., terminology and vocabularies) heterogeneity in hydrologic observations data. The Observations Data Model (ODM) was designed to

provide a consistent format for point observation data storage and was implemented in relational database management systems (RDBMS) with defined entities, attributes, and relationships to represent observations (Horsburgh et al., 2008). In ODM, the concepts of sites (locations at which observations are made), variables (phenomena that are observed), methods (how the observations are made), sources (who is responsible for creating the observations), and quality control levels (the degree of quality control, processing, or aggregation observations have been subject to) are the key metadata characteristics that define each observation, along with the associated date and time. ODM's concepts parallel those of OGC's O&M (now referred to as Observations, Measurements, and Samples), including feature of interest, observed property, observing procedure, and observer. Using a consistent data model between sources and projects streamlines data organization, enhances data interoperability, and facilitates development of software applications for loading, managing, processing, and sharing data. ODM was adopted as a de facto community standard data model across several different initiatives. CUAHSI implemented ODM as the data storage format in their federated community HIS (Ames et al., 2012), and many HIS used ODM or a modified version of the ODM data model in RDBMS with additional tables or fields for network or project-specific needs (Mason et al., 2014; Varadharajan et al., 2019; Winslow et al., 2008).

Although ODM was well-suited to the structure of time series data, some implementers found that the performance for large and frequently updating time series was deficient for some use cases related to both the structure of the data model and its physical implementation within RDBMS. These limitations, along with a need to support more diverse and complex data types from other geoscience domains, motivated the

development of ODM2, an information and data model for spatially discrete earth observations (Horsburgh et al., 2016a). While ODM included concepts similar to those in the OGC's O&M standard (Cox, 2013), ODM2 explicitly profiled O&M for spatially discrete earth observations. In ODM2, time series are represented as a specific result type. While there was some variability, our review showed that most HIS implementations have used OGC's O&M, ODM, or a derivative of these two information models to implement the data model for their operational data store.

Physical implementations of operational data stores varied across HIS we reviewed and included RDBMS Microsoft SQLServer, PostgreSQL, SQLite, and MySQL. To support smaller scale efforts or to facilitate open-source, low cost sensing platforms, Conner et al. (2013) and Sadler et al. (2016) deployed "HydroServer Lite", which implements ODM in MySQL, a freely available RDBMS. As an operational data store for time series data contributed from low cost, off-the-shelf modular sensing equipment, Horsburgh et al. (2019) implemented an ODM2 database in PostgreSQL. In efforts to improve performance for data retrieval and presentation via web applications, the dedicated time series database, InfluxDB, was added to cache time series data with a subset of metadata. InfluxDB offers high performance for indexing and retrieving time-based data, which is an increasingly common use case for HIS. To insert metadata, this system employed a web user interface for citizen scientists to register and manage data collection sites and create metadata for associated sensors and observed variables.

Additional approaches to storing data include custom data models (Samourkasidis et al., 2019), proprietary systems (Ventura et al., 2019), and document models (Braud et al., 2022; McGuire et al., 2016). Well known proprietary HIS (e.g., 52 North, Aquatic

Informatics Aquarius, or Kisters Wiski (ESIP EnviroSensing Cluster, 2014)) implement data models in a RDBMS (e.g., PostgreSQL) and may take advantage of open, community-developed standards. Other systems have used document models (Braud et al., 2022), which reference objects as key-value pairs most often in JSON documents stored within a database system like MongoDB that supports document data models (McGuire et al., 2016). This approach may provide the flexibility needed for integration of multiple data types (i.e., time series, geospatial raster and vector) at multiple spatial and temporal scales; however, the specificity and granularity particularly suited to time series data may be sacrificed.

For operational storage and metadata management, the USGS NWIS uses a combination of in-house designed software applications and a commercial vendor data management system. The USGS NWIS data are stored and managed within the Aquarius water data management software suite, which includes an underlying PostgreSQL relational database that stores a cache of information about monitoring locations, parameters (i.e., observed properties or variables), and methods. Custom software applications built by USGS are used as authoritative sources for information on monitoring sites from which Aquarius extracts the information it requires. Similarly, reference lists managed centrally by USGS are used to define parameters and methods. USGS' implementation illustrates that rather than a single operational database, multiple databases or layers may be required to meet the needs of operational storage - especially when legacy software systems are combined with software from commercial vendors. A single central data store or a single proprietary software system may not meet all needs, and for NWIS, when the proprietary system was adopted, connecting to existing systems

was necessary.

4.4.3 Publication, Sharing, and Exchange

Users of HIS include both data publishers and data consumers, and the following subsections describe tools for each role. Consumers of hydrologic time series access data using software applications that interface with underlying storage systems to enable data discovery, exchange, visualization, and analysis (Goodall et al., 2008). Internal users (i.e., those within the organization collecting and managing the data) may directly access data in the operational store (e.g., SQL queries to a RDBMS) (Shukla et al., 2019). However, this method entails security risks when sharing data with many users or with users external to the organization collecting the data. Thus, these use cases usually require provision of external-facing software applications to meet user needs while minimizing security concerns. HIS may deploy a number of different tools for providing access to hydrologic data, including web services that enable programmatic access to and encode data for interoperability and transfer, web applications that provide a graphical user interface in a web browser, or client software applications that communicate with web services and are paired with analysis environments (e.g., a Python client library paired with a Python development environment) to support data visualization and analysis.

4.4.3.1 Data Publisher Tools

Multiple standards-based web service interfaces and standardized data transfer encodings have been developed that may be used for encoding and transferring hydrologic time series data over the Internet (i.e., from data publishers to data consumers). As part of the CUAHSI HIS effort, WaterOneFlow (WOF) web services were developed as a standardized web service interface for data querying and retrieval,

and Water Markup Language (WaterML) was developed as a machine-readable eXtensible Markup Language (XML) schema for encoding water observation data for communication over the Internet (Zaslavsky et al., 2007). A subsequent version of WaterML, WaterML 2.0, was developed and adopted as a standard by the OGC as an implementation of the O&M information model (OGC, 2014). The WaterML 2.0 standard is multi-part and includes a specification for encoding time series, a second part for ratings, gagings, and sections, a third part for surface hydrology features, and a fourth part for encoding groundwater data. Additionally, there is a profile of WaterML 2.0 for encoding water quality data. The OGC also developed Sensor Web Enablement (SWE), a suite of standards for exchange of data collected by environmental sensor networks (Botts et al., 2013). SWE includes multiple standards related to sensor data management, and components relevant to HIS include Sensor Observation Service (SOS) and SensorML. SOS is a web service interface that allows querying observations that are then encoded for response using WaterML 2.0 or O&M. Associated sensor metadata and representation of observed features are encoded using SensorML, another OGC standard.

For the CUAHSI HIS, WOF web services were deployed with a direct mapping to ODM databases to extract data and metadata and encode them for transmission using WaterML (Maidment, 2008). WOF web services were also deployed for data from multiple U.S. government agencies by using the same web service interface and data encoding but with proxy web services that first retrieved data from agency websites and then reformatted the data using WaterML (e.g., streamflow data from the USGS National Water Information System) (Goodall et al., 2008). The common information model concepts from ArcHydro and ODM (site, variable, date/value) were used as the basis for

defining an observation using WOF/WaterML, and the primary web service methods were GetSites and GetVariables as data discovery functions and GetValues, as a data access function (Goodall et al., 2008). Although the SOS standard was not strictly mapped to a specific information model, opting to use OGC's O&M as default but allowing other data formats, it has been profiled for use with hydrologic data (Andres et al., 2014) through encoding using WaterML 2.0 and uses primary web service methods similar to those of WOF: GetFeatureofInterest, DescribeSensor, and GetObservation.

Deploying the same web service interface on top of multiple individual databases allows for consistent data access across distributed sources (Blodgett et al., 2016b; Slawecki et al., 2017). Through the adoption of a common web service interface for data from multiple sources (WOF), the CUAHSI HIS demonstrated how data can be retrieved from any provider and how software applications could be applied that were agnostic of the data store (Ames et al., 2012). For example, CUAHSI developed a metadata catalog called "HIS Central" that was designed to regularly fetch and centrally store metadata about available time series data from all registered WOF services so that data discovery services could be provided. Multiple client applications, including a desktop client called HydroDesktop (Ames et al., 2012) and a web browser client called HydroClient (https://data.cuahsi.org), were developed to search the HIS Central catalog and enable download of discovered time series data. Search was facilitated by consistent metadata provided by each registered WOF service, and download was facilitated by each WOF service having a consistent interface and data encoding – regardless of where each service was hosted. For multiple data providers, this ensures data are delivered in a consistent format and avoids tedious data formatting and transformations for data

consumers.

As part of a federally sponsored Open Water Data Initiative (OWDI) and related interoperability experiments, additional examples, opportunities, and best practices for aggregating and synthesizing data from multiple sources were identified (Blodgett et al., 2016a, 2016b; Slawecki et al., 2017). The USGS NWIS implemented web service protocols in a RESTful API ("Water Services"), which returns results in WaterML 1.1 or 2.0 as well as tab-delimited or JSON formats. Although NWIS does not use WOF, this illustrates that existing systems may map to and deliver data using standardized encodings even if the systems are not easily compatible with exchange protocols. Another USGS example is the National Groundwater Monitoring Network (https://cida.usgs.gov/ngwmn/), which is a catalog that uses web services to communicate with and integrate data from several sources (Blodgett et al., 2016a). Although it is focused on discrete data, the Water Quality Portal (https://www.waterqualitydata.us/) is an important system to mention as it integrates data from the Environmental Protection Agency, the USGS, states, tribes, and other agencies through an exchange protocol known as Water Quality Exchange (WQX) (Read et al., 2017).

Client usage of web services is independent of programming language and computing platform, and the WOF web service interface was implemented using several software development environments. The CUAHSI HIS project developed WOF using the Microsoft .Net development environment (Zaslavsky et al., 2007). Realizing the need for an implementation using freely available software, Conner et al. (2013) created an implementation using PHP. Another group developed a version based on Python (https://github.com/ODM2/WOFpy), which has been implemented by several HIS

(Celicourt et al., 2023; Horsburgh et al., 2019), and at least one commercial HIS implemented WOF (Newswire, 2011). Similarly, there are multiple implementations of the SOS web service interface (e.g., 52 North, Kisters). The CUAHSI HIS was initially designed as a federated system with each data providing organization hosting their own WOF web service instance. However, many organizations that participated initially and others who wanted to use the CUAHSI HIS to share data did not have the resources or personnel to run, manage, or sustain a publishing service on their local computer infrastructure. Thus, CUAHSI provided additional capability for data publishing organizations to upload data to a cloud instance of an ODM database with a connected WOF service hosted by CUAHSI and registered with CUAHSI's HIS Central metadata catalog.

Monitoring networks of varying scales have incorporated web services for data publication and sharing following similar approaches (Conner et al., 2013; Horsburgh et al., 2011, 2009; Jones et al., 2017, 2015; Muste et al., 2013; Sadler et al., 2016). In each case, WOF web services were deployed on operational data stores and registered with the CUAHSI HIS Central metadata catalog making data accessible via CUAHSI's search, discovery, and visualization tools. Data were encoded for delivery in WaterML 1.1 format, and publishers deployed public facing custom websites that interpreted data encoded using WaterML (e.g., Horsburgh et al., 2016). HIS that either modified the ODM data model to add metadata elements or deployed custom operational data stores for additional data types also used WOF and WaterML to make compatible data available through the CUAHSI HIS (e.g., Mason et al., 2014). In a similar approach, a web application called the ODM2 Data Sharing Portal (Horsburgh et al., 2019) was created

for enabling upload of streaming sensor data from low-cost monitoring stations deployed by citizen scientists. The ODM2 Data Sharing Portal enables creation of site, observed variable, and other metadata; streaming of data from field deployed stations into an operational data store via a web service API, and visualization and access to contributed data. The ODM2 Data Sharing Portal also included an instance of the WOF web services for programmatic data access.

For HIS that used OGC SWE standards, web services were implemented that returned data using the SOS web service interface and encoded data for transfer using either the XML encoding of the O&M standard (Samourkasidis et al., 2019; Samourkasidis and Athanasiadis, 2017; Ventura et al., 2019); or WaterML 2.0 (Slawecki et al., 2017). The underlying operational data stores for these services were either mapped to SOS and the respective data encoding using a data model developed by 52 North or using the Kisters software and its underlying data model. For a catalog application that accessed data sources implementing SOS (analogous to CUAHSI's HIS Central), Slawecki et al. (2017) developed a custom metadata mapping in PostgreSQL with accompanying REST API services. Other data publishers developed APIs specific to their databases that were accessed by custom web portals or interfaces for dissemination (Samourkasidis et al., 2019; Varadharajan et al., 2019).

For their sensor network, Wong and Kerkez (2016) implemented IoT technologies consisting of relatively low cost systems that support REST APIs for web services and data transfer (Hart and Martinez, 2015). Because IoT technologies are used across many application domains (e.g., environmental sensing, smart homes and buildings, supply chain management, etc.), the data communication protocols are generic, and similar to

SOS, a mapping must be made between metadata required to describe hydrologic time series and generic metadata concepts used by IoT information models. However, in some cases IoT information models may not be complete enough to enable a mapping that adequately describes hydrologic data.

4.4.3.2 Data Consumer Tools

Data access functionality for consumers typically involves some combination of software tools that enable data download, programmatic access, and data visualization for exploration, interpretation, and communication. Many of the reviewed HIS deployed project-based web applications/portals that included, at minimum, a map for displaying the locations of monitoring sites and a time series plot viewer for visualizing time series of observed variables at selected locations (Demir and Krajewski, 2013; Jones et al., 2015; Mason et al., 2014; Muste et al., 2013; Slawecki et al., 2017). For example, Water Data for the Nation is the umbrella effort for the USGS's water data consumption tools including the National Water Dashboard (https://dashboard.waterdata.usgs.gov/), which displays a national map with symbology representing current conditions for each site for streamflow, groundwater, water quality, and meteorological data. A page for each monitoring location displays adjustable time series plots of monitored parameters along with options for data download.

As hydrologic time series are observed at specific locations, mapping is essential for spatial reference and context (Braud et al., 2022; Horsburgh et al., 2016b; Soh et al., 2006), and national scale data products need spatial relationships to enable interpretation (Sullivan et al., 2018). Basic time series plots are fundamental for data exploration and preliminary analyses necessary for determining quality, characteristics, and appropriate

uses for the data (Muste et al., 2013). More complex and specialized plotting types may be implemented, such as box and whisker, histograms (Horsburgh et al., 2016), flood frequency and duration curves (Xu et al., 2022), multidimensional plots (Mason et al., 2014), and visualizations that integrate other data types (Demir and Krajewski, 2013). Visualization types are driven by user needs, and website/portal developers may need to consider the metadata dimensions by which users will want to query or filter results (Horsburgh et al., 2016b; Slawecki et al., 2017; Soh et al., 2006).

As describe above, the HydroDesktop and HydroClient software programs were developed to facilitate data discovery, download, and visualization for data sources that published data with WOF web services and registered with CUAHSI's HIS Central metadata catalog. In both HydroClient and HydroDesktop, data discovery was accomplished through map-based and keyword-based searches. When a user searched for data, the software applications accessed the HIS Central metadata catalog facilitated by WOF web services. Discovered time series that met the user's search criteria could then be downloaded via their WOF web service. Time series were written either to CSV files for download (HydroClient) or a SQLite database on the user's local computer (HydroDesktop), and both software applications offered visualization with built in plotting tools. Similarly, in a pilot catalog of water quality time series data, Slawecki et al. (2017) ingested metadata from sources implementing SOS to populate a metadata catalog, which was used for search and discovery within a web interface along with download of CSV data files.

To support automated/programmatic data access and querying by data consumers, packages in common programming languages were developed that used the available

web services to fetch data and return them in performant data formats (e.g., R or Python pandas data frames). Examples include HydroR, an R package for accessing data from CUAHSI HIS Central (Horsburgh and Reeder, 2014), WaterML R, an R package for querying and retrieving data from WOF (Kadlec et al., 2015), and data retrieval packages in R, Python, and Julia for accessing USGS NWIS data (Hodson et al., 2023). The ULMO package for Python (https://github.com/ulmo-dev/ulmo/) accesses data from multiple hydrologic and environmental data sources (e.g., USGS NWIS, the National Climate Data Center, CUAHSI WOF, several state agencies). Ventura et al. (2019) developed an R package with functions that directly queried their HIS database with a Shiny App to perform functions in a web user interface. In an operational context, it is important to have tools like these that facilitate programmatic data access/retrieval to make software application development easier and to enable the types of data retrieval for which a web user interface is superfluous (e.g., supplying realtime observational data to a forecast model).

Another use case for data sharing and publication is long term archival to support access to historic data records (ESIP EnviroSensing Cluster, 2014). Web portals and operational data stores may be deprecated after a project ends or technology ages (Ruddell et al., 2014), so data access may need to be preserved by submitting data to an archival repository. In addition to using web services and a project website to share data, Jones et al. (2017) published hydrologic time series as CSV files to resources in the HydroShare repository using scripts that automatically exported CSV files from the operational data store. HydroShare is a community repository for heterogeneous data types that can act as an archival HIS (Horsburgh et al., 2016b), a distinction from the

operational HIS described in this paper. Using a similar approach, Varadharajan et al. (2019) archived hydrologic time series by publishing snapshots with persistent identifiers in the ESS-DIVE repository, the Department of Energy's approved repository. Other long term repositories may be used that are either data type agnostic (e.g., FigShare) or are more specific (e.g., to a region, a project, or a data type).

4.4.4 Management, Processing, and Curation

In parallel to data moving between the collection, storage, and publication pools, HIS commonly include operations to manage, process, and curate data. Operations may include sensor/equipment management and tracking, monitoring and notification of data issues, data review and quality control, and generation of derived products (e.g., discharge from stage, aggregated data products or statistics). These operational steps are often performed on data that have been loaded to the operational data store but may be executed on the data collection platform (datalogger) or after extracting data from the operational data store using publicly accessible web services or API (Gries et al., 2016). Some of these functions only read data from the storage pools while others necessarily modify data or write new or derived data back to the operational data store. Execution of these steps may be automated as part of the data management workflow for an HIS or may be performed offline and asynchronously, possibly even manually. While most data managers have similar management, processing, and curation needs, in the literature and systems we reviewed we noted significant heterogeneity in both workflows to meet these needs and in how they are executed, often requiring data managers to string together multiple software tools to accomplish their workflows. While some agencies and monitoring networks may have documented guidelines and practices (e.g., ESIP

EnviroSensing Cluster, 2014; Jones et al., 2017; Wagner et al., 2006), operations that fall in this category are seldom described with much detail in available literature, and no widely used standards exist.

Technicians that operate sensor networks typically track equipment characteristics, deployments, and maintenance. Some groups may track and document this in an ad hoc way while others use more formal approaches. In either case, capturing the linkages between a sensor (e.g., its specific manufacturer, model, serial number, etc.), its maintenance (e.g., deployment, cleaning, calibration), and resulting observations can inform data quality (ESIP EnviroSensing Cluster, 2014; Jones et al., 2017), especially when sensors malfunction or when cleaning or calibration of sensors introduce data artifacts that must later be corrected. Several HIS extended the ODM data model to include records of field activities and sensor characteristics (Jones et al., 2015; Mason et al., 2014) to facilitate accurate tracking of equipment deployment and maintenance activities. The USGS does not currently have an integrated application or guidelines for documenting and tracking sensor assignments, and the approach depends on local office practices and the equipment type.

For many operational uses of real time sensor data, data quality is a concern due to high volumes of sensor data and the challenges of ambient monitoring (ESIP EnviroSensing Cluster, 2014). In some cases, automated or manual tools to support quality assurance and quality control (QAQC) are implemented in the HIS data flow (Campbell et al., 2013). Automated data quality monitoring systems and algorithms can enable responsive diagnostics and repair when sensors or peripheral equipment fail (Benson et al., 2010; Wong and Kerkez, 2016). To monitor the latest data values for

potential concerns, scripts and algorithms may be implemented on any of the data stores that alert technicians (Jones et al., 2015; Shukla et al., 2019), automatically flag data (ESIP EnviroSensing Cluster, 2014), or trigger corrective action (Varadharajan et al., 2019). For example, the USGS operates custom-built software applications that monitor data streams, apply basic algorithms to check data status, and send alerts to field staff. Furthermore, technicians use agency-specific tools and the public websites to visually review data on a near-daily basis.

Even with manual and/or automated data monitoring, time series data from in situ sensors often include anomalies or drift related to ambient conditions (e.g., fouling) or sensor drift that need to be corrected. Custom desktop applications were developed for performing quality control post processing for data in flat files or ODM databases (Horsburgh et al., 2015; Sheldon, 2008). Several studies developed R or Python scripts to support data QAQC processing that retrieve data from the operational data store via the database API and perform automated and semi-automated data checking and correction (Varadharajan et al., 2019; Ventura et al., 2019). For the USGS NWIS data, data management, review, and processing are conducted within the Aquarius data management software, a web browser-based interface that connects to underlying databases. Aquarius' visualization and editing tools are used by technicians to remove anomalies, set qualifiers, apply corrections, and make estimates. Aquarius is also used in the process for officially approving data.

4.5 Challenges and Opportunities

After reviewing the literature and considering operational HIS systems, we identified many challenges that, if solved, could advance HIS capabilities. We narrowed

our focus and selected several challenges for which we believe solutions would have the highest impact on utility and functionality for HIS. Some of these challenges have persisted from earlier work on HIS while others are new and emerging. We focused on challenges that are not specific to deployment of monitoring equipment or data collection sites because, while these are important considerations that affect the ease with which data can be collected (e.g., power, availability of communications, ease of installation, etc.), products to address them exist and may be implemented depending on resources and priorities. Instead, we selected challenges that are mainly downstream of monitoring site and equipment considerations in the data workflow. In the sub-sections below, we articulate a challenge and illustrate how it is being addressed by one or more existing systems. These challenges are not fully solved by any one system, but this section highlights current work being done by various organizations and opportunities for advancement. As many of these HIS are new and may not have been significantly documented in existing literature, we rely on our own experience and on discussions with the people and teams developing and managing these systems.

Challenge 1: Non-interoperability of sensing and datalogging systems. A challenge with data collection and acquisition in HIS includes non-interoperability of sensing, datalogging, and data communication systems that are often proprietary. While several standard communication protocols exist for dataloggers to read data from sensors (e.g., RS232, Modbus, SDI-12, I2C, etc.), making it possible to integrate most sensors with dataloggers from a variety of manufacturers that support these standards, the same is not true for communication between dataloggers and downstream data management systems. Sensor and datalogger manufacturers may offer access to data communication

and sharing software, but methods for doing so have not been standardized to date. Many datalogger manufacturers use proprietary software systems for these purposes and are not incentivized to make a shift to open, standards-based software as they may potentially lose a competitive business advantage. As a result, retrieving data from a field-deployed datalogger still often requires proprietary software or custom middleware.

As an example of advances in integration between open and proprietary software, the Dendra system (https://github.com/DendraScience) uses a microservices architecture to develop targeted functionality for individual vendors. Dendra has developed code for integrating with Campbell Scientific's proprietary LoggerNet software to load data into Dendra's operational data stores. LoggerNet communicates with and retrieves data from remote monitoring sites using Campbell Scientific's proprietary communication and networking technologies. Dendra then connects to LoggerNet's Logger Data Monitoring Protocol (LDMP) server to retrieve data records for a datalogger over a TCP socket. While this is an example of successful interoperability between two systems, a generalizable solution has not emerged to enable interoperability with software provided by multiple different vendors. Additionally, it can be difficult to make these connections between systems as firewalls may block TCP connections.

Another way that monitoring systems are addressing this challenge is by using microcontrollers as dataloggers, which allow for flexibility with communication protocols. These dataloggers will generally operate push-based systems, requiring an Internet connection (e.g., via a cellular data). This is the approach used in the Monitor My Watershed HIS wherein hundreds of Arduino-based microcontrollers are deployed by citizen scientists and the data are acquired via HTTP POST requests initiated by the

microcontrollers (Horsburgh et al., 2019). Some commercial dataloggers (e.g., Campbell Scientific) are also capable of connecting to and pushing data to any system with an API via HTTP POST requests.

The team developing the HydroServer HIS (https://github.com/hydroserver2) is focusing on APIs through which data can be pushed using common protocols. Internet-connected dataloggers, regardless of manufacturer, can push observations directly into an instance of HydroServer through the OGC SensorThings API. While it is increasingly common for environmental sensing sites to be Internet-connected, there are still many situations where Internet connectivity is impossible. For those sites, HydroServer relies on whatever proprietary software is communicating with the datalogger to deliver observations to a base station in a CSV file, which can then be loaded automatically through the SensorThings API by the HydroServer Streaming Data Loader software. The HydroServer team is developing a Python/Django implementation of the SensorThings API, and others who have embraced a similar approach are using the FROST Server implementation of SensorThings API (https://github.com/FraunhoferIOSB/FROST-Server).

Challenge 2: Heterogeneity in data models, metadata, and vocabularies. Although some standardized data models have been widely adopted for operational storage, a rigid data model does not meet all needs or use cases, and some systems prefer custom data representations. While most time series HIS use site and observed property to describe observations, there is inconsistency in the implementation of additional metadata attributes. Furthermore, even when data collection organizations use the same metadata elements, if they do not use standardized vocabularies for the values of those

attributes, semantic heterogeneity persists. Earlier efforts at using community moderated controlled vocabularies (e.g., ODM and ODM2) have shown a path forward that can minimize semantic heterogeneity; however, implementation and adoption of controlled vocabularies has been inconsistent.

Several recent efforts that integrate time series from several data sources into a single HIS have tackled this issue by mapping metadata terms and vocabularies from separate sources to authoritative lists or standards. Slawecki et al. (2017) mapped data providers' parameter names to EPA's Substance Registry Service (SRS), an inventory of chemicals, organisms, and other substances of interest. Given that this was a pilot project, the scalability of mapping observed properties with the EPA SRS is untested. In a national-scale catalog of data sources in France (Braud et al., 2022), a custom hierarchical vocabulary system was developed based on several standards, linked to a number of external thesauri, and implemented so that data sources either adopt the project vocabulary or map their terms to the project vocabulary. A similar approach was used by the CUAHSI HIS Central metadata catalog to better facilitate data discovery services for data sources that did not adopt the ODM controlled vocabularies. For a data integration effort to be successful, a common vocabulary or technologies that enable mediation across vocabularies must be used, but most are too complex for typical domain scientists and practitioners to use and may require modification for usability (Varadharajan et al., 2019). A system is needed that can support searches with approximate matching and that is straightforward to implement and reference.

Both the HydroServer and Dendra system described above have incorporated controlled vocabularies from the ODM information model (ODM CVs for Dendra, and

updated ODM2 CVs for HydroServer, http://vocabulary.odm2.org). These curated vocabularies are useful aids in describing hydrologic observations, but they do not satisfy the needs of custom data models (e.g., those with important metadata for which no ODM CV exists). Additionally, in efforts to prioritize flexibility and ease of use, both of those systems offer the terms from the controlled vocabularies as recommendations but do not rigidly enforce their use.

Even within a single organization, heterogeneity in how observations are described can be an issue. USGS has historically used a system of "parameter codes" that mixes the name of the observed property with methodological information and, in some cases, measurement units. This leads to multiple parameter codes associated with the same observed property, but differing by method or units (e.g., code 00910 – "Calcium, water, unfiltered milligrams per liter as calcium carbonate" versus code 00915 – "Calcium, water filtered, milligrams per liter"). As a result, there is a gap in relating NWIS-specific parameter codes to community supported vocabularies and ontologies (Blodgett et al., 2016a). USGS is currently undergoing modifications to the system for identifying observed properties to eliminate ambiguity, reduce duplication, and better align with the Water Quality Portal and WQX. Because multiple groups are already submitting data via WQX, there is some momentum behind the system. When released, the revised NWIS parameter lists could serve as a guide for vocabulary systems, although additional functionality (e.g., machine readable formats, search capabilities, ability to add terms) might need to be implemented for integration into dispersed HIS.

Challenge 3: Modernizing existing systems. Advances in the availability of technology (e.g., cloud systems, IoT, etc.) has increased options for infrastructure,

software development, deployment and hosting, data storage mechanisms, and data transfer protocols and services. HIS that were implemented in the early efforts by CUAHSI and others are now operating using dated technology - e.g., using web services based on simple object access protocol (SOAP) and XML data encodings rather than more modern representational state transfer (REST) and JavaScript object notation (JSON) data encodings. However, while newer standards have emerged (e.g., OGC's SensorThings), no new standard has been fully embraced by the environmental sensing community, and community recommendations and best practices vary. Furthermore, technical debt associated with legacy systems is a challenge for many HIS implementations. Some HIS may be badly in need of upgrades to one piece of their data workflow but integration with older products or software, which are dependent on outdated or deprecated technologies, infrastructure, or programming languages can present significant challenges - especially when HIS implementers are users and not software developers themselves.

OGC's SensorThings API, associated information model, and commercial cloud technologies hold promise here. The HydroServer development team has mapped the ODM/ODM2 time series information model to the SensorThings information model and has developed an implementation of an operational data store that can be deployed in a straightforward way using commercial cloud technologies. This provides a path forward for migrating existing, ODM-based HIS to a modernized web service interface and data encoding via SensorThings with deployment in the commercial cloud (e.g., Amazon Web Services). Discussions with Aquatic Informatics indicate that Aquarius software systems will move to commercial cloud deployments for new customers, with relatively few still

running Aquarius software on-premise.

To aid data providers and HIS implementers, CUAHSI has historically focused on data publication through web services, the HIS Central metadata catalog, and select cloud hosting options. Presently, CUAHSI is actively revamping its publication and cataloging services and expanding its capabilities to provide operational support. As part of this modernization effort, CUAHSI is working to integrate the Dendra system as an operational component with its existing suite of offerings for data providers. Additionally, in a collaborative effort, CUAHSI and Dendra are planning to employ SensorThings as an API and as an encoding for data publication.

In an enterprise context, the USGS is not immune to the need for modernization. The USGS is currently undergoing a major effort to modernize the software applications that comprise NWIS to address technical debt on out-of-date software and infrastructure and to improve the data workflow. This modernization effort aims to create software applications that communicate with each other via APIs and that use modern technologies and programming languages (e.g., deploying applications on the cloud, RESTful APIs that return JSON). For example, USGS has been prototyping using SensorThings' RESTful API for data ingest and egress.

Challenge 4: Multiple different web service interfaces and exchange standards. In a survey of water data systems, about half used web services to transfer data in machine friendly formats (Dow et al., 2015). As shown in this paper, multiple web service interfaces (SOS, WOF, SensorThings) and encodings for data exchange (WaterML 1.1., WaterML 2.0, O&M XML, SensorML, SensorThings) have been developed and approved as standards, yet there is no consistent guidance on applicability

or which standard should be used for which situation and by whom (Dow et al., 2015). In one example of addressing this challenge, the catalog of French monitoring networks (Braud et al., 2022) identified multiple metadata standards necessary to cover desired functionality and created a project-specific data model that combined entities from several standards and maps between them. The result is, in effect, yet another standard. In contrast, some data producers may not have the expertise to implement standards-based publication. Because options for open source and commercial implementations are limited, data producers may buy into commercial systems or develop one-off systems rather than deal with the complexity of adopting standards.

SensorThings was designed to support lightweight and flexible IoT web services and exchange protocols for sensor data and information. SensorThings includes a RESTful API definition and an information model that defines key components of observing systems: Things (i.e., monitoring stations/platforms in the context of hydrologic time series), Observed Properties (what is observed), Sensors (the physical instruments and/or procedures used to create observations), Observations, Datastreams (a time series of Observations for an Observed Property at a Thing), and Locations (the physical location of a Thing). SensorThings is a more modern web service standard that uses REST bindings and JSON encodings rather than SOAP and XML, with REST and JSON being highly preferred by programmers. Some groups are transitioning from SOS to SensorThings to modernize and simplify data access (e.g., Kotsev et al., 2018), but operational use of SensorThings is still growing and there is still no definitive guidance on the best web service interface and data encoding standards for future use.

For HIS that do implement standards, there are multiple software implementations

of some standards with inconsistent levels of development and support. These limitations can result in adoption of software for which an organization lacks specific development or hosting expertise. As a specific example, there are currently several open-source and commercial implementations of the OGC SensorThings API (e.g., FROST, SensorUp, 52 North, HydroServer). Written in Java, the FROST server implementation is likely the most complete. HydroServer's implementation is less mature but is written in Python, a more commonly used language. An organization with Python web application expertise is forced to choose between the less complete HydroServer implementation for which they may be able to fix bugs or add features or the FROST implementation that is more complete but for which they have no specific development expertise.

Challenge 5: Mapping data to exchange standards. Newer exchange standards like OGC's SensorThings are increasingly flexible with constructs that are more loosely defined and that can be associated with various data models (e.g., ODM/ODM2's time series information model can be mapped to the SensorThings data model, which was derived from the O&M information model). However, because existing data stores and data exchange functions and encodings have often been developed independently, mapping of data and metadata to the exchange standard is necessary. This mapping process can be challenging, and may, in some cases be lossy (i.e., metadata from an existing data store may not be fully captured in the exchange encoding) as some exchange standards may not be expressive enough to unambiguously describe hydrologic time series.

The HydroServer development group found this to be the case with the SensorThings information model. The original HydroServer software stack upon which

the CUAHS HIS was based used the ODM/ODM2 time series information model for expressing metadata about hydrologic time series data. WOF web services and WaterML were designed for the ODM time series information model, so there was effectively a 1:1 correspondence between the storage implementation of the information model in a RDBMS and the exchange implementation of the information model (WaterML). In contrast, the SensorThings information model is not hydrology specific and relies on a relatively small number of entities with a few attributes – some of which are extendable. This extensibility enabled the addition of important ODM/ODM2 time series metadata elements to the SensorThings information model as extended attributes. This included adding new entities for concepts like units of measure, processing levels, and people/organizations because the SensorThings data model lacks these concepts entirely. Each "Thing" (e.g., a monitoring site) has a "name" and "description" and a "properties" element to which attributes like latitude and longitude might be mapped. While this worked reasonably well for the ODM/ODM2 time series metadata, there may be other custom HIS data models for which the mapping is not as straightforward.

It may also be important to consider a distinction between the metadata needed to enable discovery of time series data versus the metadata needed to describe the dataset for unambiguous interpretation and use. Our preliminary analysis of the SensorThings metadata encoding suggests that it can serve both purposes, and different systems may be made interoperable by using the SensorThings API and metadata encodings. However, in other projects, there is growing momentum around using the Schema.org vocabulary for dataset discovery metadata to help make datasets more universally discoverable on the Internet using tools like Google's search engine. Use of Schema.org would require yet

another metadata translation.

Challenge 6: Automating data processing and review. Due to the large volumes of data being collected, processes in HIS should be as automated as possible (Muste et al., 2013; Ruddell et al., 2014), which has generally been achieved for major data fluxes. However, data processing and review can be subjective and often requires local knowledge (Jones et al., 2018), which impedes full implementation of automated QAQC measures on streaming data and often prevents consistency in application. This is a major limitation for HIS systems focused on producing timely and accurate data – e.g., HIS that supply data to real time modeling or forecasting systems.

Automating data processing continues to be an active area of research, which presents an opportunity for potential incorporation into HIS. Various rules-based, regression-based, and feature-based approaches have been tested for anomaly detection in hydrologic time series, with mixed results (Leigh et al., 2018; Santos-Fernandez et al., 2023; Schmidt and Kerkez, 2023; Schmidt et al., 2023; Talagala et al., 2019). While these algorithms and approaches show promise, there is a gap in moving from research to operations and scaling to larger monitoring networks. Success is dependent on characteristics of the site, observed property, sensor technology, anomaly cause or type, and identified anomalies may still require technician review. Furthermore, testing and training algorithms also present a challenge as sufficiently and consistently labeled "gold standard" datasets may not exist. Some have used data simulations for this purpose (Santos-Fernandez et al., 2023). Recent developments include packages (e.g., Python or R) that implement both heuristic rules based algorithms relying on user input for initial set up (e.g., thresholds) along with data driven algorithms based on past data to determine

whether data are valid or anomalous (Jones et al., 2022; Schmidt et al., 2023). Functions from these packages could be incorporated into an HIS data workflow to automatically identify anomalies, but evaluation by a technician would likely still be necessary in many cases.

There is opportunity to streamline data processing. Local knowledge can be encoded as rules with increasing complexity that are then automated to identify anomalies. More complex algorithms that are trained on past data can identify potential anomalies, and a choice must be made either by a technician or by an algorithm to determine whether data are anomalous. While it is unclear whether algorithms could be sufficiently trained to consistently make choices on par with trained technicians, if the anomaly detection workflow is streamlined, then review by technicians may become straightforward. We recommend that HIS components be developed to support flexible implementation of various algorithms under the premise that there is no "one size fits all" solution. Other potential improvements include recommending algorithms, rules, parameters, or settings based on data characteristics; development of software applications with a visual graphical user interface for technicians with less programming experience; standardized approaches and benchmark datasets for training algorithms and reporting performance; and ensuring that processing steps are traceable and reproducible.

Commercial HIS are also implementing approaches that automatically determine rules for data editing and corrections based on the behavior of past data. At the USGS, progress toward streamlining processing within Aquarius includes automatically applying corrections to data related to fouling or drift based on field readings, and a current effort focuses on automating estimates of discharge based on hydrologically related sites rather

than technician estimates. The USGS has also explored approaches for automating anomaly detection, but concerns remain with high levels of false positives resulting in an extra burden on technicians reviewing data.

4.6 Conclusions and Outlook

HIS are comprised of hardware and software that support the life cycle of hydrologic data from observation in the field to end user dissemination and include functionality for data storage, transfer, management, curation, processing, and publication. HIS are key to the advancement of hydrologic science – the systems by which data are stored and transferred and the ease with which management is performed can either facilitate or inhibit subsequent analyses. Innovations and improvements to the technology deployed as part of HIS have enabled advances in the understanding of hydrologic systems and will continue to do so going forward. Although the core concepts and parameters of interest in hydrologic research may not have changed, our ability to collect, support, and manage the resulting data accelerates scientific discovery.

As technology has developed, HIS have evolved to support data of increased volume and complexity, a trend we expect must continue. For fixed point time series data, the USGS NWIS set a precedent for data to be broadly available in consistent formats. Community efforts by CUAHSI and others advanced HIS with regard to standards and protocols for data structure, semantics, and data transfer. But, we are now at a critical juncture in which additional development is needed to support innovations in sensing technologies, expectations for data timeliness, and data driven analyses.

For fixed point time series data, the key components of HIS support data collection and acquisition; operational storage; publication, sharing, and exchange; and

the movement of data between these pools. In parallel, data management, processing, and curation tasks are important to most HIS. Researchers and practitioners have deployed systems that perform these tasks at a range of scales, but challenges persist, some of which are being addressed with varying degrees of success by current systems and for which larger community efforts could establish effective solutions.

Interoperability between sensing and datalogging systems and downstream HIS components could be advanced through generic communication solutions that are agnostic to the hardware manufacturer. This may be achieved by using the same encoding for both data ingest and egress (e.g., SensorThings) with a standard protocol (e.g., HTTP). Addressing heterogeneity between different HIS requires an authoritative source for information models, metadata, and vocabularies with buy in and uptake from a critical mass of practitioners. Similarly, momentum around a smaller number of accepted standards has potential to reduce duplication of resources and expertise and would ensure that modernized or newly developed software applications use interoperable interfaces and data encodings. Such an approach would enable more 'plug and play' type functionality, promoting integration and interoperability between components. Comprehensive guidance is essential to navigate the options of multiple web service and exchange standards, and community discussions can help determine whether a consensus can be reached for a singular or at least a smaller number of standards. Time and testing may be required for a smaller number of highly endorsed standards to emerge. Community-endorsed mappings are also vital in overcoming the challenge of aligning information models for hydrologic time series data with those of broader, more generic exchange standards (e.g., mapping ODM/ODM2's time series information model to that

of SensorThings. Finally, improving the automation of data processing and review can be achieved by transitioning algorithms from research to operational use with a common set of software libraries and accepted best practices for integrating these approaches into technicians' workflows.

Research groups, citizen science efforts, and small municipalities may not have the resources, expertise, or bandwidth to effectively deploy HIS and sustain them on their own over the long term. Even for larger agencies like the USGS, efforts to maintain and modernize the software applications that keep data flowing require ongoing support. Data producers and providers would benefit from access to a more complete suite of both commercial and openly available HIS software applications that use community-endorsed standards to facilitate compatibility and interoperability. Public release of agency produced HIS software applications (e.g., those used in practice by USGS) could help meet some existing needs. Implementation guidance, community moderation, and consensus is needed to establish current recommended standards and systems for vocabularies, data semantics, and exchange protocols. Leading efforts to bring together academic, agency, practitioner, and commercial parties to achieve these goals is a significant undertaking and requires funding and social capital. CUAHSI and OGC are examples of organizations with experience and success in similar endeavors from past initiatives, and community efforts in other fields (e.g., LTER and Earth Science Information Partners (ESIP)) may serve as examples. Although numerous challenges may pose difficulties across various HIS use cases, we conclude that these hurdles, while substantial, are not insurmountable. With concerted community efforts, we are poised to advance the next generation of HIS and further the progress of hydrologic science.

4.7 Acknowledgements

Funding for this project was provided by the National Oceanic & Atmospheric Administration (NOAA), awarded to the Cooperative Institute for Research to Operations in Hydrology (CIROH) through the NOAA Cooperative Agreement with The University of Alabama (NA22NWS4320003). Additional funding and support were provided by the Utah Water Research Laboratory at Utah State University.

REFERENCES

Ames, D.P., Horsburgh, J.S., Cao, Y., Kadlec, J., Whiteaker, T., Valentine, D., 2012. HydroDesktop: Web services-based software for hydrologic data discovery, download, visualization, and analysis. Environ. Model. Softw. 37, 146–156. https://doi.org/10.1016/j.envsoft.2012.03.013

Andres, V., Jirka, S., Utech, M., 2014. Open Geospatial Consortium OGC Sensor Observation Service 2 . 0 Hydrology Profile. Open Geospatial Consort. Best Pract. Pap. 1–36.

Bandaragoda, C., Tarboton, D.G., Maidment, D.R., 2006. Hydrology's Efforts Toward the Cyberfrontier. Eos (Washington. DC). 87, 2–3.

Benson, B., Bond, B., Hamilton, M., Monson, R., Han, R., 2010. Perspectives on next-generation technology for environmental sensor networks. Front. Ecol. Environ. 8, 193–200. https://doi.org/10.1890/080130

Beran, B., Piasecki, M., 2009. Engineering new paths to water data. Comput. Geosci. 35, 753–760. https://doi.org/10.1016/j.cageo.2008.02.017

Bieroza, M., Acharya, S., Benisch, J., ter Borg, R.N., Hallberg, L., Negri, C., Pruitt, A., Pucher, M., Saavedra, F., Staniszewska, K., van't Veen, S.G.M., Vincent, A., Winter, C., Basu, N.B., Jarvie, H.P., Kirchner, J.W., 2023. Advances in Catchment Science, Hydrochemistry, and Aquatic Ecology Enabled by High-Frequency Water Quality Measurements. Environ. Sci. Technol. 57, 4701–4719. https://doi.org/10.1021/acs.est.2c07798

Blaen, P.J., Khamis, K., Lloyd, C.E.M., Bradley, C., Hannah, D., Krause, S., 2016. Real-time monitoring of nutrients and dissolved organic matter in rivers: Capturing event dynamics, technological opportunities and future directions. Sci. Total Environ. https://doi.org/10.1016/j.scitotenv.2016.06.116

Blodgett, D., Lucido, J., Kreft, J., 2016a. Progress on water data integration and distribution: A summary of select US Geological Survey data systems. J. Hydroinformatics 18, 226–237. https://doi.org/10.2166/hydro.2015.067

Blodgett, D., Read, E., Lucido, J., Slawecki, T., Young, D., 2016b. An Analysis of Water Data Systems to Inform the Open Water Data Initiative. J. Am. Water Resour. Assoc. 52, 845–858. https://doi.org/10.1111/1752-1688.12417

Botts, M., Percivall, G., Reed, C., Davidson, J., 2013. OGC Sensor Web Enablement: Overview and High Level Architecture. Open Geospatial Consort. White Pap. 713–723. https://doi.org/10.1007/978-3-540-79996-2

Braud, I., Chaffard, V., Coussot, C., Galle, S., Juen, P., Alexandre, H., Baillion, P., Battais, A., Boudevillain, B., Branger, F., Brissebrat, G., Cailletaud, R., Cochonneau, G., Decoupes, R., Desconnets, J.C., Dubreuil, A., Fabre, J., Gabillard, S., Gérard, M.F., Grellet, S., Herrmann, A., Laarman, O., Lajeunesse, E., Le Hénaff, G., Lobry, O., Mauclerc, A., Paroissien, J.B., Pierret, M.C., Silvera, N., Squividant,

H., 2022. Building the information system of the French Critical Zone Observatories network: Theia/OZCAR-IS. Hydrol. Sci. J. 67, 2401–2419. https://doi.org/10.1080/02626667.2020.1764568

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, Wade, M., Boose, E.R., 2013. Quantity is Nothing without Quality. Bioscience 63, 574–585. https://doi.org/10.1525/bio.2013.63.7.10

Celicourt, P., Sam, R.D., Piasecki, M., 2023. Rapid Prototyping of An Automated Sensor-to-Server Environmental Data Acquisition. J. Environ. Informatics 41, 1–15. https://doi.org/10.3808/jei.202300483.same

Conner, L.G., Ames, D.P., Gill, R.A., 2013. HydroServer Lite as an open source solution for archiving and sharing environmental data for independent university labs. Ecol. Inform. 18, 171–177. https://doi.org/10.1016/j.ecoinf.2013.08.006

Coopersmith, E., Minsker, B., Maidment, D.R., Hodges, B., Conner, J., Ojo, T., Montagna, P., 2007. An Environmental Information System for Hypoxia in Corpus Christi Bay: A WATERS Network Testbed, in: Restoring Our Natural Habitat - Proceedings of the 2007 World Environmental and Water Resources Congress. ASCE.

Cox, S., 2013. OGC Abstract Specification - Geographic information - Observations and measurements. Open Geospatial Consort.

Demir, I., Krajewski, W.F., 2013. Towards an integrated Flood Information System: Centralized data access, analysis, and visualization. Environ. Model. Softw. 50, 77–84. https://doi.org/10.1016/j.envsoft.2013.08.009

Dow, A.K., Dow, E.M., Fitzsimmons, T.D., Materise, M.M., 2015. Harnessing the environmental data flood: A comparative analysis of hydrologic, oceanographic, and meteorological informatics platforms. Bull. Am. Meteorol. Soc. 96, 725–736. https://doi.org/10.1175/BAMS-D-13-00178.1

ESIP EnviroSensing Cluster, 2014. Community Wiki Document on Best Practices for Sensor Networks and Sensor Data Management [WWW Document]. Fed. Earth Sci. Inf. Partners. URL http://wiki.esipfed.org/index.php/EnviroSensing_Cluster (accessed 1.1.16).

Gandolfi, C., Wethner, H., 1987. A hydrologic information system for the valtellina region. Environ. Softw. 2, 89–93. https://doi.org/10.1016/0266-9838(87)90006-2

Goodall, J.L., Horsburgh, J.S., Whiteaker, T.L., Maidment, D.R., Zaslavsky, I., 2008. A first approach to web services for the National Water Information System. Environ. Model. Softw. 23, 404–411. https://doi.org/10.1016/j.envsoft.2007.01.005

Gries, C., Gahler, M.R., Hanson, P.C., Kratz, T.K., Stanley, E.H., 2016. Information management at the North Temperate Lakes Long-term Ecological Research site — Successful support of research in a large, diverse, and long running project. Ecol.

Inform. 36, 201–208. https://doi.org/10.1016/j.ecoinf.2016.08.007

Hart, J., Martinez, K., 2015. Toward an environmental Internet of Things. Earth Sp. Sci. https://doi.org/10.1002/2014EA000044

Henzen, D., Mueller, M., Jirka, S., Senner, I., Kaeseberg, T., Zhang, J., Bernard, L., Krebs, P., 2016. A scientific data management infrastructure for environmental monitoring and modelling. Environ. Model. Softw. Support. a Sustain. Futur. Proc. - 8th Int. Congr. Environ. Model. Software, iEMSs 2016 1, 218–225.

Hill, D.J., Liu, Y., Marini, L., Kooper, R., Rodriguez, A., Futrelle, J., Minsker, B.S., Myers, J., McLaren, T., 2011. A virtual sensor system for user-generated, real-time environmental data products. Environ. Model. Softw. 26, 1710–1724. https://doi.org/10.1016/j.envsoft.2011.09.001

Hodson, T.O., DeCicco, L.A., Hariharan, J.A., Stanish, L.F., Black, S., Horsburgh, J.S., 2023. Multi-language retrieval of United States hydrologic data. https://doi.org/10.22541/essoar.169264772.27243384/v1

Hooper, R.P., Maidment, D.R., Helly, J., Kumar, P., Piasecki, M., 2004. CUAHSI Hydrologic Information Systems. Bull. Am. Meteorol. Soc.

Horsburgh, J.S., Aufdenkampe, A.K.., Mayorga, E., Lehnert, K.A., Hsu, L., Song, L., Jones, A.S., Damiano, S.G., Tarboton, D.G., Valentine, D., Zaslavsky, I., Whitenack, T., 2016a. Observations Data Model 2: A community information model for spatially discrete Earth observations. Environ. Model. Softw. 79, 55–74. https://doi.org/10.1016/j.envsoft.2016.01.010

Horsburgh, J.S., Caraballo, J., Ramírez, M., Aufdenkampe, A.K., Arscott, D.B., Damiano, S.G., 2019. Low-cost, open-source, and low-power: But what to do with the data? Front. Earth Sci. 7, 1–14. https://doi.org/10.3389/feart.2019.00067

Horsburgh, J.S., Jones, A.S., Ramírez, M., Caraballo, J., 2016b. Time series analyst: Interactive online visualization of standards based environmental time series data. Environ. Model. Softw. Support. a Sustain. Futur. Proc. - 8th Int. Congr. Environ. Model. Software, iEMSs 2016 1, 162–169.

Horsburgh, J.S., Morsy, M.M., Castronova, A.M., Goodall, J.L., Gan, T., Yi, H., Stealey, M.J., Tarboton, D.G., 2016c. HydroShare: Sharing Diverse Environmental Data Types and Models as Social Objects with Application to the Hydrology Domain. J. Am. Water Resour. Assoc. 52, 873–889. https://doi.org/10.1111/1752-1688.12363

Horsburgh, J.S., Reeder, S.L., 2014. Data visualization and analysis within a Hydrologic Information System: Integrating with the R statistical computing environment. Environ. Model. Softw. 52, 51–61. https://doi.org/10.1016/j.envsoft.2013.10.016

Horsburgh, J.S., Tarboton, D.G., Hooper, R.P., Zaslavsky, I., 2014. Managing a community shared vocabulary for hydrologic observations. Environ. Model. Softw. 52, 62–73. https://doi.org/10.1016/j.envsoft.2013.10.012

Horsburgh, J.S., Tarboton, D.G., Maidment, D.R., 2005. A Community Data Model for

Hydrologic Observations, in: CUAHSI Hydrologic Information System Workshop. CUAHSI, Durham, NC, pp. 102–135.

Horsburgh, J.S., Tarboton, D.G., Maidment, D.R., Zaslavsky, I., 2011. Components of an environmental observatory information system. Comput. Geosci. 37, 207–218. https://doi.org/10.1016/j.cageo.2010.07.003

Horsburgh, J.S., Tarboton, D.G., Maidment, D.R., Zaslavsky, I., 2008. A relational model for environmental and water resources data. Water Resour. Res. 44. https://doi.org/10.1029/2007wr006392

Horsburgh, J.S., Tarboton, D.G., Piasecki, M., Maidment, D.R., Zaslavsky, I., Valentine, D., Whitenack, T., 2009. An integrated system for publishing environmental observations data. Environ. Model. Softw. 24, 879–888. https://doi.org/10.1016/j.envsoft.2009.01.002

Horsburgh, J.S., Reeder, S.L., Jones, A.S., Meline, J., 2015. Open source software for visualization and quality control of continuous hydrologic and water quality sensor data. Environ. Model. Softw. 70, 32–44. https://doi.org/10.1016/j.envsoft.2015.04.002

Hsu, L., Mayorga, E., Horsburgh, J.S., Carter, M.R., Lehnert, K.A., Brantley, S.L., 2017. Enhancing interoperability and capabilities of earth science data using the Observations Data Model 2 (ODM2). Data Sci. J. 16, 1–16. https://doi.org/10.5334/dsj-2017-004

Jones, A.S., Aanderud, Z.T., Horsburgh, J.S., Eiriksson, D.P., Dastrup, D., Cox, C., Jones, S.B., Bowling, D.R., Carlisle, J., Carling, G.T., Baker, M.A., 2017. Designing and Implementing a Network for Sensing Water Quality and Hydrology across Mountain to Urban Transitions. J. Am. Water Resour. Assoc. https://doi.org/10.1111/1752-1688.12557

Jones, A.S., Horsburgh, J.S., Eiriksson, D.P., 2018. Assessing subjectivity in environmental sensor data post processing via a controlled experiment. Ecol. Inform. 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. Environ. Model. Softw. 151, 105364. https://doi.org/10.1016/j.envsoft.2022.105364

Jones, A.S., Horsburgh, J.S., Reeder, S.L., Ramírez, M., Caraballo, J., 2015. A data management and publication workflow for a large-scale, heterogeneous sensor network. Environ. Monit. Assess. 187, 348. https://doi.org/10.1007/s10661-015-4594-3

Kadlec, J., StClair, B., Ames, D.P., Gill, R.A., 2015. WaterML R package for managing ecological experiment data on a CUAHSI HydroServer. Ecol. Inform. 28, 19–28. https://doi.org/10.1016/j.ecoinf.2015.05.002

Kotsev, A., Schleidt, K., Liang, S., van der Schaaf, H., Khalafbeigi, T., Grellet, S., Lutz,

M., Jirka, S., Beaufils, M., 2018. Extending INSPIRE to the internet of things through sensorthings API. Geosci. 8, 1–22. https://doi.org/10.3390/geosciences8060221

Laney, C.M., Pennington, D.D., Tweedie, C.E., 2015. Filling the gaps : sensor network use and data-sharing practices in ecological research. Front. Ecol. Environ. 13, 363–368. https://doi.org/10.1890/140341

Lee, K., Meng, C., Chiang, S., Chung, Y., 2004. An Inquiry System for Design Discharge in Small and Midsize Watersheds, in: Processdings of the Watershed Management Symposium. p. 243.

Leigh, C., Alsibai, O., Hyndman, R.J., Kandanaarachchi, S., King, O.C., McGree, J.M., Neelamraju, C., Strauss, J., Talagala, P.D., Turner, R.S., Mengersen, K., Peterson, E.E., 2018. A framework for automated anomaly detection in high frequency wate r-quality data from in situ sensors. Sci. Total Environ. 664, 885–898. https://doi.org/10.1016/j.scitotenv.2019.02.085

Lundquist, J.D., Wayand, N.E., Massmann, A., Clark, M.P., Lott, F., Cristea, N.C., 2015. Diagnosis of insidious data disasters. Water Resour. Res. 3815–3827. https://doi.org/10.1002/2014WR016585

Maidment, D.R., 2008. Bringing Water Data Together. J. Water Resour. Plan. Manag. 134, 95–96. https://doi.org/10.1061/(asce)0733-9496(2008)134:2(95)

Maidment, D.R., 2005. Hydrologic Information System Status Report. Report 214.

Maidment, D.R., 2002. Arc Hydro: GIS for Water Resources. ESRI Press.

Maidment, D.R., Zaslavsky, I., Horsburgh, J.S., 2006. Hydrologic Data Access Using Web Services. Southwest Hydrol. 5, 16–17.

Mao, F., Khamis, K., Krause, S., Clark, J., Hannah, D.M., 2019. Low-Cost Environmental Sensor Networks: Recent Advances and Future Directions. Front. Earth Sci. 7, 1–7. https://doi.org/10.3389/feart.2019.00221

Mason, S.J.K., Cleveland, S.B., Llovet, P., Izurieta, C., Poole, G.C., 2014. A centralized tool for managing, archiving, and serving point-in-time data in ecological research laboratories. Environ. Model. Softw. 51, 59–69. https://doi.org/10.1016/j.envsoft.2013.09.008

McGuire, M.P., Roberge, M.C., Lian, J., 2016. Channeling the water data deluge: a system for flexible integration and analysis of hydrologic data. Int. J. Digit. Earth 9, 272–299. https://doi.org/10.1080/17538947.2015.1031715

Muste, M. V, Asce, M., Bennett, D.A., Secchi, S., Schnoor, J.L., Kusiak, A., Arnold, N.J., Mishra, S.K., Asce, S.M., Ding, D., Rapolu, U., 2013. End-to-End Cyberinfrastructure for Decision-Making Support in Watershed Management 565–573. https://doi.org/10.1061/(ASCE)WR.1943-5452

Newswire, 2011. KISTERS Makes It OFFICIAL, Joins CUAHSI As Corporate Member.

Newswire.

OASIS, 2019. MQTT Version 5.0.

OGC, 2014. OGC WaterML 2.0: Part 1- Timeseries. Open Geospatial Consort. Implemenation Stand.

Open Geopatial Consoritum, 2021. SensorThings API Part 1: Sensing Version 1.1. https://doi.org/http://www.opengis.net/doc/is/sensorthings/1.1

Pellerin, B.A., Stauffer, B.A., Young, D.A., Sullivan, D.J., Bricker, S.B., Walbridge, M.R., Clyde, G.A., Shaw, D.M., 2016. Emerging Tools for Continuous Nutrient Monitoring Networks: Sensors Advancing Science and Water Resources Protection. JAWRA J. Am. Water Resour. Assoc. 20460, 1–16. https://doi.org/10.1111/1752-1688.12386

Piasecki, M., Beran, B., 2009. A semantic annotation tool for hydrologic sciences. Earth Sci. Informatics 2, 157–168. https://doi.org/10.1007/s12145-009-0031-x

Porter, J.H., Hanson, P.C., Lin, C., 2012. Staying afloat in the sensor data deluge. Trends Ecol. Evol. 27, 121–129. https://doi.org/10.1016/j.tree.2011.11.009

Read, E.K., Carr, L., De Cicco, L., Dugan, H.A., Hanson, P.C., Hart, J.A., Kreft, J., Read, J.S., Winslow, L.A., 2017. Water quality data for national-scale aquatic reserach: The Water Quality Portal. Water Resour. Res. 1735–1745. https://doi.org/10.1002/2016WR019993.Received

Ruddell, B.L., Zaslavsky, I., Valentine, D., Beran, B., Piasecki, M., Fu, Q., Kumar, P., 2014. Sustainable long term scientific data publication: Lessons learned from a prototype Observatory Information System for the Illinois River Basin. Environ. Model. Softw. 54, 73–87. https://doi.org/10.1016/j.envsoft.2013.12.015

Rüegg, J., Gries, C., Bond-Lamberty, B., Bowen, G.J., Felzer, B.S., McIntyre, N.E., Soranno, P. a, Vanderbilt, K.L., Weathers, K.C., 2014. Completing the data life cycle: using information management in macrosystems ecology research. Front. Ecol. Environ. 12, 24–30. https://doi.org/10.1890/120375

Rundel, P.W., Graham, E. a., Allen, M.F., Fisher, J.C., Harmon, T.C., 2009. Environmental sensor networks in ecological research. New Phytol. 182, 589–607. https://doi.org/10.1111/j.1469-8137.2009.02811.x

Sadler, J.M., Ames, D.P., Khattar, R., 2016. A recipe for standards-based data sharing using open source software and low-cost electronics. J. Hydroinformatics 18, 185–197. https://doi.org/10.2166/hydro.2015.092

Samourkasidis, A., Athanasiadis, I.N., 2017. A miniature data repository on a raspberry pi. Electron. 6, 1–13. https://doi.org/10.3390/electronics6010001

Samourkasidis, A., Papoutsoglou, E., Athanasiadis, I.N., 2019. A template framework for environmental timeseries data acquisition. Environ. Model. Softw. 117, 237–249. https://doi.org/10.1016/j.envsoft.2018.10.009

Santos-Fernandez, E., Ver Hoef, J.M., Peterson, E.E., Mcgree, J.M., Villa, C.A., Leigh, C., Turner, R., Roberts, C., Mengersen, K., 2023. Unsupervised anomaly detection in spatio-temporal stream network sensor data Unsupervised anomaly detection in spatio-temporal stream network sensor data. https://doi.org/10.13140/RG.2.2.33200.74241

Schmidt, J.Q., Kerkez, B., 2023. Machine Learning-Assisted, Process-Based Quality Control for Detecting Compromised Environmental Sensors. Environ. Sci. Technol. 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. Environ. Model. Softw. 169, 105809. https://doi.org/10.1016/j.envsoft.2023.105809

Sheldon, W.M., 2008. Dynamic, Rule-based Quality Control Framework for Real-time Sensor Data, in: Gries, C., Jones, M.B. (Eds.), Proceedings of the Environmental Information Management Conference. Albuquerque, NM, pp. 145–150.

Shope Jr., W.G., 1987. U. S. Geological Survey's National Real-Time Hydrologic Information System Using GOES Satellite Technology.

Shukla, A., Shukla, S., Kinsman, G.H., Crowell, M.L., 2019. Evolution of hydroinformatics at a state water management agency. Hydrol. Sci. J. 00, 1–14. https://doi.org/10.1080/02626667.2019.1661418

Slawecki, T., Young, D., Dean, B., Bergenroth, B., Sparks, K., 2017. Pilot implementation of the US EPA interoperable watershed network. Open Geospatial Data, Softw. Stand. 2, 1–11. https://doi.org/10.1186/s40965-017-0025-4

Soh, L.K., Samal, A., Zhang, J., 2006. A task-based approach to user interface design for a web-based hydrologic information systems. Trans. GIS 10, 417–449. https://doi.org/10.1111/j.1467-9671.2006.01005.x

Sullivan, D.J., Joiner, J.K., Caslow, K.A., Landers, M.N., Pellerin, B.A., Rasmussen, P.P., Sheets, R.A., 2018. U. S. Geological Survey Continuous Monitoring Workshop — Workshop Summary Report. Reston VA. https://doi.org/https://doi.org/10.3133/ofr20181059

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 Resour. Res. 55, 8547–8568. https://doi.org/10.1029/2019WR024906

Turner, B., Hill, D.J., Caton, K., 2020. Correction to: Cracking "Open" Technology in Ecohydrology C1–C1. https://doi.org/10.1007/978-3-030-26086-6_25

Varadharajan, C., Faybishenko, B., Henderson, A., Henderson, M., Hendrix, V.C., Hubbard, S.S., Kakalia, Z., Newman, A., Potter, B., Steltzer, H., Versteeg, R., Agarwal, D.A., Williams, K.H., Wilmer, C., Wu, Y., Brown, W., Burrus, M.,

Carroll, R.W.H., Christianson, D.S., Dafflon, B., Dwivedi, D., Enquist, B.J., 2019. Challenges in Building an End-to-End System for Acquisition, Management, and Integration of Diverse Data from Sensor Networks in Watersheds: Lessons from a Mountainous Community Observatory in East River, Colorado. IEEE Access 7, 182796–182813. https://doi.org/10.1109/ACCESS.2019.2957793

Ventura, B., Vianello, A., Frisinghelli, D., Rossi, M., Monsorno, R., Costa, A., 2019. A methodology for heterogeneous sensor data organization and near real-time data sharing by adopting OGC SWE standards. ISPRS Int. J. Geo-Information 8. https://doi.org/10.3390/ijgi8040167

Viqueira, J.R.R., Villarroya, S., Mera, D., Taboada, J.A., 2020. Smart environmental data infrastructures: Bridging the gap between earth sciences and citizens. Appl. Sci. 10. https://doi.org/10.3390/app10030856

Wagner, R.J., Boulger, R.W., Oblinger, C.J., Smith, B.A., 2006. Guidelines and Standard Procedures for Continuous Water-Quality Monitors: Station Operation, Record Computation, and Data Reporting, U.S. Geological Survey Techniques and Methods 1-D3.

Winslow, L., Benson, B., Chiu, K., 2008. Vega: a flexible data model for environmental time series data, in: Gries, C., Jones, M. (Eds.), Proceedings of the Environmental Information Management Conference. Albuquerque, NM.

Wong, B.P., Kerkez, B., 2016. Real-time environmental sensor data: An application to water quality using web services. Environ. Model. Softw. 84, 505–517. https://doi.org/10.1016/j.envsoft.2016.07.020

Xu, H., Berres, A., Liu, Y., Allen-Dumas, M.R., Sanyal, J., 2022. An overview of visualization and visual analytics applications in water resources management, Environmental Modelling and Software. https://doi.org/10.1016/j.envsoft.2022.105396

Yang, C., Raskin, R., Goodchild, M., Gahegan, M., 2010. Geospatial Cyberinfrastructure: Past, present and future. Comput. Environ. Urban Syst. 34, 264–277. https://doi.org/10.1016/j.compenvurbsys.2010.04.001

Zaslavsky, I., Valentine, D., Whiteaker, T., 2007. CUAHSI WaterML. Open Geospatial Consort. Discuss. Pap. OGC 07-041r1 88.

FIGURES

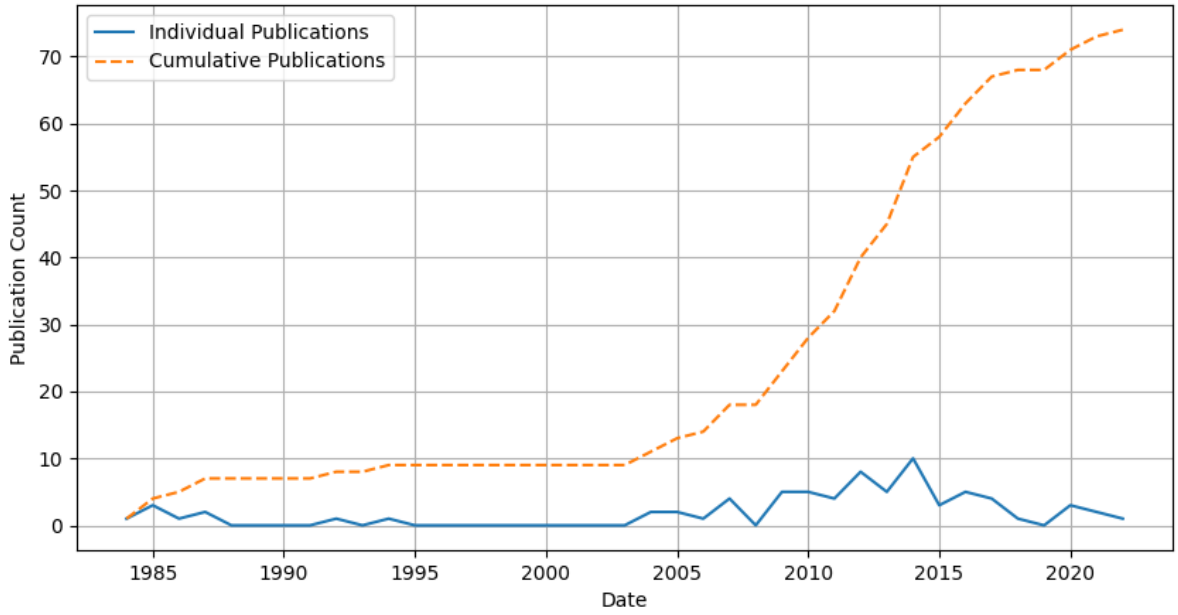

Figure 4.1 Count of publications identified by Scopus with the keyword “hydrologic information systems” from 1980 to 2023.

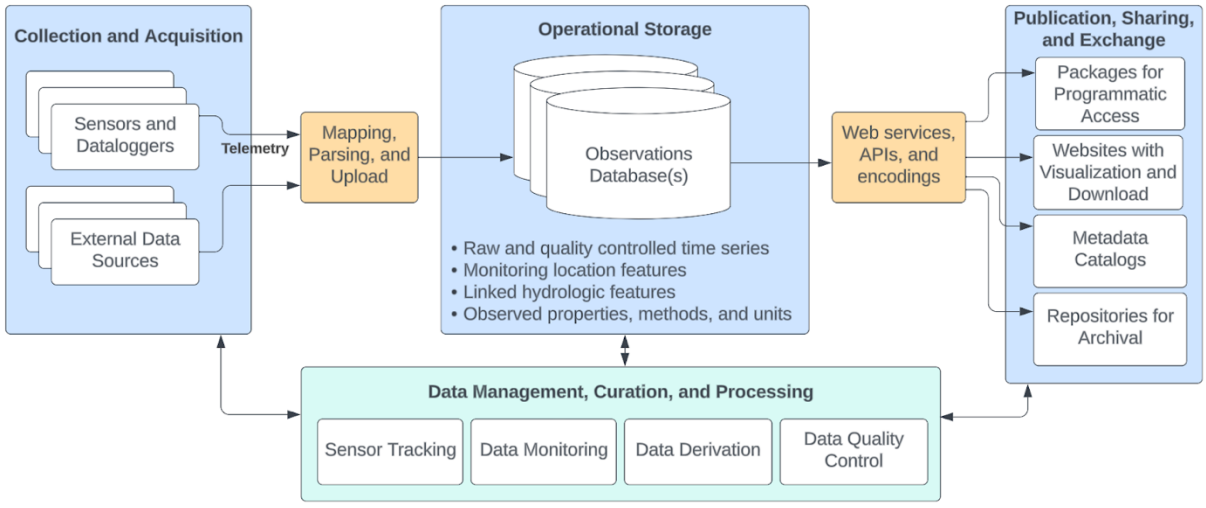

Figure 4.2 Diagram of a generalized HIS architecture. Major data pools are represented in blue with associated data fluxes in orange and operational steps that run parallel to the data flow are in green.

CHAPTER 5

SUMMARY AND CONCLUSIONS

As barriers for environmental monitoring continue to fall, there has been a proliferation in the generation of water observations data. In some cases, sensors and monitoring equipment have become more affordable, in other cases, challenges with available power or communication have been met by new technology, and in other cases, organizations have realized the operational benefits of high resolution monitoring data. Regardless of the driver, environmental monitoring data are now being produced in larger quantities, at higher spatial and temporal resolutions, and over extended periods and wider spatial coverage. We expect this trend in data collection growth to continue in the future. Although these factors make sensor data collection increasingly accessible and attractive for researchers and practitioners, managing data of increasing volume and complexity resulting from sensor observations remains a challenge. Software options, operational guidelines, and training are limited for scientists seeking approachable cyberinfrastructure to help them collect and manage sensor data. This work sought to advance tools and resources for management and use of high frequency data collected using *in situ* environmental sensors.

Specifically, this dissertation contributes new software tools and a workflow to improve assurance of the quality of high-frequency data, offers guidance on utilizing data-intensive techniques in water science education, and presents available systems for managing high frequency sensor data along with a generalized architecture that serves as a foundation for modernizing and advancing those systems. The work described in this dissertation incorporates data science methodologies to demonstrate tools and techniques

for making data fit for scientific analysis, water management, and educational purposes. The outcomes of this dissertation include reusable algorithms, open-source code packages, interactive notebook examples, and online educational resources aimed at enhancing the handling and utilization of data acquired from *in situ* sensors.

Chapter 2 presented a software package, pyhydroqc, for streamlining and potentially automating the process for reviewing and correcting time series collected by *in situ* sensors. This work is motivated by the challenge of performing quality control post processing on large quantities of sensor-observed data. Anomalous values related to adverse ambient conditions, sensor drift, and malfunction of sensors or peripherals result in data that must be reviewed and addressed with corrective action. The software package we developed incorporates both rules-based and data-driven techniques for reviewing data and identifying anomalies. We also explored techniques for correcting anomalies by automating the imputation of data values. The software package was designed to reduce the burden of manual quality control post processing, which is tedious for technicians and costly to monitoring networks. Furthermore, automating data processing has the potential to make high quality data available with greater consistency and immediacy.

Development and testing of the pyhydroqc algorithms was performed on case study datasets of high frequency water quality observations collected in the Logan River Observatory (LRO). Though there was no gold standard for comparison, we were able to assess performance by comparing results from pyhydroqc to technician-corrected and labeled datasets, and we generally observed high rates of true positives for anomaly detection while minimizing false positives and false negatives. We conclude that rules-based detection of anomalies in continuous water quality time series is a key component

of automated quality control for algorithm development and application, and in many cases, it may be adequate on its own compared to using more complex models. Model-based detection may also be effective in identifying anomalies missed by rigid rules. The regression models used by pyhydroqc are valuable in conjunction with a threshold to determine how much variability from the observation causes a data point to be labeled as "anomalous". Although there is often a push to find the "best" model, we found performance between multiple models to be similar, and simpler models may be preferred for computational efficiency. We also implemented a novel approach to correct data gaps or anomalous values with blended forecasts and backcasts of predicted data that may hold value for approximating diurnal patterns. While not completely removing the need for technician involvement, incorporating pyhydroqc functions into a technician's workflow can greatly reduce the number of points for review and present options for correction. The functions encapsulated in the pyhydroqc Python package can be called in scripts or code notebooks, and we developed and published examples of performing data quality control operations for the LRO water quality case study data.

Chapter 3 addressed the need for training on skills and tools for working with large and complex datasets. Water scientists and engineers are quantitatively savvy, but their general lack of proficiency with current programming, software, data management, and data visualization tools and techniques may limit their ability to analyze and work with growing volumes of water data. "Hydroinformatics" and "water data science" are topics for which educational resources (e.g., courses, modules, and/or online educational materials) can help bridge the gaps. In surveying and interviewing instructors who teach formal courses in these topics, we were able to gain a better understanding of the state of

hydroinformatics and water data science instruction and determine the key topics being taught, general approaches to instruction, and ideas regarding necessary components for successfully sharing educational resources via accessible platforms. Using those results, we developed and implemented a set of educational modules focused on demonstrating how the needs and gaps identified by study participants can be addressed through the sharing of instructional materials online.

Our results showed that shared online educational content can address instructors' needs for up-to-date and flexible resources, especially considering the transition to virtual platforms that occurred related to the COVID-19 pandemic. Instructors used a mix of online and in-person modalities for their courses and retained desirable aspects of online teaching post-pandemic. Many instructors were using custom materials with coding demonstrations in Python or R with a strong focus on teaching students new data visualization and analysis tools, how to troubleshoot code, and how to find and understand documentation. To address the unanimous interest among instructors in exchanging instructional materials, we implemented several educational modules designed to cover participant-identified topics of interest including programmatic access to public agency data, databases and structured query language (SQL), sensor data quality control, and machine learning classification. These modules demonstrated how to address identified gaps in available educational resources by incorporating online educational content, portable programming examples, accessible slide decks, and example assignments. Based on the criteria for online sharing and content organization recommended by study participants, we implemented the modules on the HydroLearn platform, and we evaluated the effectiveness of HydroLearn for meeting community

needs for online content exchange. We conclude that with broad community buy in, a system like HydroLearn can help instructors keep pace with the rapid evolution of technology and topics in the field and maintain the value of their course.

Chapter 4 focused on synthesizing and evaluating the landscape of options for hydrologic information systems (HIS), the software and hardware components for managing time series data from fixed *in situ* sensors. As technology developed over the past two decades, options for HIS evolved to handle data from national and local agencies, research monitoring networks, and citizen science groups. HIS include functionality to support the full data life cycle from collection to sharing with data consumers, and, based on our review, we extracted the architecture for a generalized HIS (i.e., the common structure exhibited by all of the systems we reviewed) with the following key components as data "pools": 1) data collection and acquisition; 2) operational storage; and 3) publication, sharing, and exchange. Data "fluxes" occur as data are transferred and transmitted between each pool. Additional data management steps (i.e., curation, processing, and derivation of higher-level data products) are performed in parallel to the primary data pools. We found that, although past efforts advanced HIS by developing software applications, data models, and standards for data exchange, much of this work is now out-of-date. Further development is needed to support advancements in monitoring technology, the demand for timely data, and capabilities for data driven analyses.

We conclude that modern HIS need to address and work to overcome some challenges that have persisted from the early days of HIS and others that are new and emerging. These include: 1) non-interoperability of sensing and datalogging systems; 2)

heterogeneity in data models, metadata, and vocabularies; 3) modernizing existing systems; 4) multiple different web service interfaces and exchange standards; 5) mapping data to exchange standards; and 6) automating data processing and review. While not a comprehensive list, we identified these challenges as having greatest potential for advancing hydrologic data management in support of research and operational data collection and use. Although existing HIS development teams are working to address these challenges, barriers persist, and many software solutions are specific to a network, a project, or an agency. In considering how to make effective progress for HIS, we determined that renewed community efforts are needed to provide clear guidance on acceptable standards for data exchange and how to use them, to determine how communities can support and use shared vocabularies or other technologies to address semantic heterogeneity in data, and to develop translations from older, legacy information models and systems to modernized standards.

CHAPTER 6

RECOMMENDATIONS

While the work in this dissertation demonstrates progress in the development and application of data driven techniques, training, and tools for working with high frequency environmental sensor data, there are several opportunities for addressing remaining challenges. First, our work on automating quality control produced a Python package with functions that can be accessed in any Python environment, but which may not be approachable for users without Python programming experience. To improve accessibility for non-programmers, more user-friendly software could be developed (e.g., a graphical user interface (GUI) software that implements pyhydroqc functions). This may help more scientists and practitioners use the algorithms to improve their data review and processing. In addition to the simple rules deployed in the pyhydroqc package, we recommend developing additional capabilities that would enable users to implement rules of increased complexity such as rate of change and thresholds for anomaly detection that can vary over time.

Within the area of sensor data quality control, research efforts continue to explore techniques for streamlining anomaly detection, with several recent studies demonstrating algorithms and tools that are applicable in different contexts and for a variety of data. Quality control of continuous sensor data does not have a “one size fits all” tool or algorithm that is applicable for all cases. Instead, a flexible framework (e.g., a set of software tools) that that can incorporate a broad spectrum of existing methods/algorithms to aid in data processing would be valuable. This would allow technicians to choose an algorithm/technique from an existing library to perform quality control processing of

their data. To enable this, different algorithms could be structured to accept the same formats of input data for training and model application and to return labeled or corrected data in consistent formats. By using libraries of commonly-structured algorithms, multiple and new algorithms could be incorporated into commercial software systems (e.g., Aquarius, Kisters) and parallel open source software tools (e.g., Dendra, HydroServer) making data review and post processing more consistent between systems. Furthermore, there may be opportunities for inserting algorithms and automated steps into different points in the data management workflow. In short, algorithms in consistent wrappers could be incorporated into a variety of software tools, GUIs, or stages in user workflows.

Furthermore, there is opportunity in applying the algorithms in pyhydroqc to a greater body of training data as the case study datasets do not represent the full spectrum of hydrologic and water quality behavior. More directly examining the performance of model types related to physical characteristics of the data or the hydrologic or environmental system within which they were collected could help inform transferability of the techniques. Because some algorithms might be better suited to certain site or data characteristics, additional application and testing could make it easier to create recommendations on which algorithm to use for which situations. In the vein of developing a common library of algorithms for environmental sensor data anomaly detection, benchmark datasets with confirmed labels could also be produced to act as gold standards for training and testing algorithms to avoid concerns with technician subjectivity. Such benchmark datasets have driven innovation in algorithm development and have been instrumental in computational fields such as computer vision, image

classification, and other applications of machine learning.

Despite advances in automating quality control post processing with the pyhydroqc package or other similar software tools, technician review may still be required, and performance tradeoffs between false positives and false negatives must be evaluated to determine how well these tools will integrate with existing software, systems, and workflows used by scientists, data managers, and practitioners. Different monitoring networks may require different approaches based on the size of the network (i.e., the number of sites/sensors producing data), the number of technicians available to perform QAQC, existing or agency-mandated QAQC protocols, and requirements for data latency and turnaround. Operators of monitoring systems also need to consider whether data should be processed agnostic of other sites or variables, which may limit the algorithms/approaches that can be used.

Our work on hydroinformatics and water data science instruction illustrated that despite high interest from instructors in sharing educational resources online, barriers persist resulting in a lack of broadly available educational resources. As an online repository for water-related educational materials, HydroLearn is a natural fit for this purpose, and based on our evaluation, there are opportunities for using HydroLearn in this role and for the hydrology community to coalesce around HydroLearn as a community platform for sharing these types of materials. However, our experience in using HydroLearn resulted in recommendations for improvements to make deposit of educational materials more straightforward and valuable for instructors. Specific suggestions include enhancing discoverability of resources with additional search facets such as keywords, standardizing and facilitating metadata entry with webforms or

markdown, improving course/module navigability, and reducing enforced hierarchical organization that may not fit some materials. A major potential improvement for HydroLearn is support for more direct linkages to example code. We were able to launch code notebooks by creating resources in HydroShare, depositing code notebooks as part of those resources, and manually linking the HydroShare resources from HydroLearn to then open in the HydroShare JupyterHub server. By deploying a JupyterHub environment or a backend connection to the HydroShare JupyterHub, HydroLearn could reduce overhead for incorporating code into instructional materials. Launching a code notebook directly from an educational module in HydroLearn is functionality that would be a major benefit for integrating technical content in hydrologic learning and that would make HydroLearn attractive for hydroinformatics and water data science instructors.

We recommend that instructors share their own educational resources as well as make use of existing shared materials by adapting, reusing, and providing feedback. As articulated by study participants, problem sets that show application of data science and machine learning approaches using real water data and real hydrology/water resources/water management scenarios are especially needed. This type of experiential learning is important for students, but difficult for instructors to develop at scale, thus the ability to incorporate examples from other instructors would be highly valuable in developing course content. The structure of the educational modules we developed and shared as part of this work serves as a template for instructors (e.g., our examples include learning objectives, structured content narrative, integration of example code, and a technical assignment). Considering the evolving technological landscape, hydroinformatics and water data science are fields in continuous development. Access to

and collaboration on educational materials can help instructors adapt more quickly and effectively to ensure that their course offerings remain valuable, relevant, and up to date. Institutions of higher education should also consider ways to encourage community sourcing of course content and to incentivize instructors who make efforts to share their educational materials.

Our work reviewing and synthesizing HIS illustrates opportunities to advance existing HIS to better support collection, management, sharing, and use of fixed-point time series observations. This is a common and important data type in hydrology and water resources engineering for which there remains a shortage of effective and available software systems and tools. Overarching issues related to interoperability and heterogeneity (in tools, formats, semantics, etc.) could be addressed through authoritative guidance produced by community leaders who work to achieve broad buy in from community members. Objectives of a community effort might include guidance on appropriate use of web service interfaces and data encoding standards for exchange over the Internet, consensus mappings to align existing information models to generic/modernized standards (e.g., mapping the older ODM information model to the newer SensorThings information model), and the development of or recommendations for a more comprehensive system for curating and promoting the use of shared vocabularies and/or other technologies to minimize remaining semantic heterogeneity in observational data. We suggest that collaboration among representatives of academic monitoring networks, practitioners from national or other agencies that collect data, vendors of commercial HIS software, and governing bodies and consortiums will be required to comprehensively address these remaining challenges.

When the Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI) HIS was initially established, these groups collaborated and produced some of the first attempts at a standardized data model, a moderated vocabulary system, and data mapping and encoding for web services. Furthermore, the CUAHSI-led efforts also resulted in open-source software applications that enabled data providers to establish more robust HIS than they could have developed on their own. Over time, the software products that are part of the CUAHSI HIS are becoming deprecated and are losing relevancy. Without necessary updates, it has become increasingly difficult to operate the software on more modern computer systems. While the suite of available software has not kept up, the pace of data collection has only increased, leaving a major gap between our ability to produce data and our ability to manage and use data. Commercially available software has met some, but not all of this need. Software (both commercial and open-source), data systems, and practitioners must continue to adapt. Because many organizations that collect hydrologic time series data lack staff with software development expertise and cannot afford existing commercial software, there remains a growing need for open-source software applications that use standards to enable integration with each other, with existing systems, and with commercial products. A next generation community effort is needed to modernize HIS and to produce systems that can adapt to the growing volume of monitoring data and the needs of scientists, agencies, and practitioners who collect, share, and use the data. Through united and coordinated work, a next iteration of HIS can enable advancements in hydrologic science.

APPENDICES

Appendix A. Anomaly Detection Background

Manual post processing by a technician remains the most commonly implemented approach for correcting anomalies in environmental sensor data. Software tools have been developed to assist technicians in performing quality control, wherein anomalies are identified visually or using filters or rules that are implemented based on user-input (Horsburgh et al., 2015; Sheldon, 2008). While initially straightforward to implement, manual post processing is resource-intensive, requires significant expertise, and may be implemented unevenly within and between sensor networks. Additionally, manual approaches may not be reproducible making it difficult to track the provenance of data from raw measurements to quality controlled products. Data driven anomaly detection has the potential to address the deficiencies of manual post processing by streamlining and standardizing the workflow.

Numerous data driven approaches have been documented for anomaly detection (Chandola et al., 2009; Cook et al., 2020; Tan et al., 2019). Basic approaches use rules to test data plausibility - e.g., range and variability checks (Taylor and Loescher, 2013), and even studies with complex workflows initially implement rules based approaches (e.g., Leigh et al., 2018). Statistical approaches rely on the distribution of data to identify points outside of the expectation (Cook et al., 2020). Regression approaches estimate a value and compare it to the observation (Chandola et al., 2009). Feature based approaches apply numerous variables (or features) within one or more machine learning methods to determine if the data point should be grouped with valid or anomalous points (Talagala et al., 2019). In approaching data driven methods for anomaly detection, important considerations include:

- Data extent: What duration of data are available? Some methods require data partitioned into separate groups for training and testing models.
- Data labels: Do sufficient data exist in which anomalies have been identified by an expert? The availability of labeled data impacts which types of models can be used. Supervised model types require labeled data for training while unsupervised model types do not. For all model types, labeled data enable assessment of performance.
- Data quality: Do sufficient data exist in which anomalies have been corrected? Some methods require 'clean' data that are free from anomalies for training models.
- Variables: What variables are to be considered? Is a single variable/sensor observed or are multiple variables measured? Do sensors at nearby sites provide additional information?
- Anomaly types: What types of anomalies are of particular concern? Can rules based detection effectively detect some of these cases?
- Online/offline detection: Does detection need to occur in real time online, or is a retrospective, offline approach acceptable?

In the following sections, we provide a brief description of several approaches and methods for detecting and correcting anomalies in environmental sensor data. We also illustrate gaps in the current state of practice for anomaly detection and correction in the quality control process.

A.1 Data Redundancy Approaches

Various types of data redundancy, including sensors, people, and models, are used to detect anomalies in environmental sensor data. The gold standard (World Meteorological Organization, 2008, Mourad and Bertrand-Krajewski, 2002) compares

data from multiple sensors, requiring at least three sensors to determine which observation is erroneous. Increased cost, maintenance, power, and data storage requirements challenge observational networks to implement redundant sensors. Furthermore, multiple sensors may all exhibit the vagaries of environmental events, sensor malfunctions, and infrastructure failures, complicating assessment and correction of data quality. To improve the consistency of quality control, Jones et al. (2018) suggest another form of data redundancy in which multiple technicians collaborate to review and correct data. Finally, data redundancy may be achieved by modeling expected values for comparison with sensor measurements. A physically based model could be used; however, model availability and uncertainty are barriers (Moatar et al., 2001). Given the relative simplicity of implementation, ability to scale to large volumes of data, few input requirements, and potential for fast performance, statistical and data driven techniques may be more appropriate. Thus, we examined several classes of data driven techniques to model expected sensor behavior as data redundancy approaches.

A.2 Univariate or Multivariate Approaches

Some predictive time series models are based on data from a single sensor independent of the condition of other co-located sensors or data. Advantages of these univariate methods are that processing can be performed on multiple sensors independently and simultaneously, and gaps or errors in data from one sensor will not impact data from other sensors (Hill and Minsker, 2010). However, anomalies in one sensor stream may correspond to anomalies in a related sensor, so approaches that utilize the information from multiple sensors provide multiple lines of evidence toward anomaly detection (Li et al., 2017). Furthermore, when performing quality control post processing,

technicians regularly consult the record of other variables simultaneously recorded at the same site to check for ‘internal consistency’ (Campbell et al., 2013) and to inform corrective actions. There is no clear best approach, and even the same authors simultaneously promote a univariate detector (Hill and Minsker, 2010) and a multivariate approach (Hill et al., 2009). Either method may yield acceptable results, although Leigh et al. (2018) report poor performance for multivariate time series regression compared to univariate. The data in question will drive whether a univariate method is required or if additional power could be achieved with multiple variables. In our work, we considered both univariate and multivariate approaches and compared the benefits and drawbacks related to the data we examined.

A.3 Spatial Dependency

‘External consistency’ refers to comparison with data from other locations (Campbell et al., 2013), and some data driven approaches are based on relationships between sites. In particular, spatial dependencies between weather sensors have been used to identify anomalies (Galarus et al., 2012). In another application, data driven models used weighted data from neighboring stream monitoring sites to infill daily mean flow records (Giustarini et al., 2016). One study included data at an upstream site offset by estimated travel time to detect anomalies in aquatic data (Conde, 2011). Spatial methods assume high correlation for a particular variable at sites having similar characteristics, which may not be clearly established for the data of interest. In this work, we focused models on data at a single site of interest so that detection and correction could be applied to sites independently.

A.4 Regression Approaches

Regression models are a class of data driven anomaly detectors for time series that predict the next anticipated value based on previous data (either univariate or multivariate). To detect anomalies with regression, the modeled value is compared to the observed, and a range of acceptability is determined for the residuals such that points outside of that range are classed as anomalous (and vice versa). Constant acceptability thresholds may be based on a user defined range or determined as a prediction interval based on the model results (Leigh et al., 2018). Thresholds may also be dynamic, varying based on the range of the model residuals (Hundman et al., 2018). For example, in one study (Dereszynski and Dietterich, 2007), the threshold range for an observation varied based on the modeled state of the sensor (i.e., a narrower range when the sensor was classed as “Good” versus “Bad”).

Auto-regressive integrated moving average (ARIMA) is a regression technique that uses a combination of past data to forecast the next point. ARIMA has been successfully implemented to predict environmental data and subsequently detect anomalies (Hill and Minsker, 2010; Leigh et al., 2018; Papacharalampous et al., 2019). Another regression technique based on a previous sequence of data is Long Short-Term Memory (LSTM), a class of Artificial Neural Networks (ANNs). Though applications to environmental data anomalies to date are limited, LSTM models have been used to reconstruct time series to detect anomalies in other fields (Hundman et al., 2018; Lindemann et al., 2019; Malhotra et al., 2016; Yin et al., 2020), and other ANN model types have been used for environmental anomaly detection (Hill and Minsker, 2010; Russo et al., 2020). Other algorithms that show promise for time series regression include

Prophet, a time series forecasting method developed by Facebook with focus on business applications (Taylor and Letham, 2018), and Hierarchical Temporal Memory (HTM) (Ahmad et al., 2017). Another method that has been implemented for anomaly detection in environmental sensor data is Dynamic Bayesian Networks, which predict values in a time series based on assigned model states corresponding to temporal windows. Studies developed models based on a few previous points (Hill et al., 2009), thousands of previous points (Hill and Minsker, 2006), and multiple past years of data to give an output based on the day of year and hour of day (Dereszynski and Dietterich, 2007). These models assume that temporal states can be definitively assigned as well as consistently applied, and we did not attempt them due to complexity and obscurity of implementation.

Because regression models produce an estimate, they are well-suited for both detection and correction of anomalous data. The time series regression models we investigated were ARIMA, LSTM, and Facebook Prophet. While ARIMA has been commonly attempted for anomaly detection in time series data, other techniques are emergent in this field (e.g., LSTM), and there are few examples comparing multiple regression techniques for aquatic sensor data.

A.5 Feature Based Approaches

Feature based methods comprise another class of anomaly detectors commonly used for discrete data (Tan et al., 2019), which some authors have applied to environmental time series (Leigh et al., 2018; Russo et al., 2020; Talagala et al., 2019). Unlike regression methods, feature based methods do not make a prediction of the observation. Anomalies are detected either based on a supervised model trained to data

labels (anomalous or valid) (Russo et al., 2020), or an unsupervised model that determines the likelihood of the point being anomalous based on distance to neighboring points. These methods rely on multiple variables as model input (features), which, in the case of aquatic sensor time series, may correspond to variables measured concurrently by adjacent sensors, past values of the variable of interest, or transformations of the relationships between these variables. Particularly for data with temporal correlation, it is not obvious which features should be selected, and complex feature engineering may be required (Christ et al., 2018). Another challenge is selecting an appropriate data transformation, a preprocessing step (e.g., taking the first derivative of the data) to highlight outlying points (Leigh et al., 2018; Talagala et al., 2019).

Almost any feature based machine learning method may be applied to anomaly detection problems, and approaches described in the literature include principal components analysis, support vector machines (Tran et al., 2019), HDOutliers (Leigh et al., 2018), k-nearest neighbor (Russo et al., 2020; Talagala et al., 2019), clustering (Hill and Minsker, 2010), random forest (Russo et al., 2020), xgboost, and isolated forest (Smolyakov et al., 2019). The success of feature based techniques in detecting anomalies from environmental sensor data is mixed (Hill and Minsker, 2010; Leigh et al., 2018; Russo et al., 2020). As they do not make predictions, feature based approaches are not well-suited to performing corrections. Given that our objectives were to both detect and correct anomalies, we did not pursue feature based approaches in the work reported here.

A.6 Anomaly Types

In most of the studies cited here, the emphasis is on anomalies that are outliers where the value of the variable is outside of expected ranges or rates of change. Detection

of gradual bias that may occur due to drift in the sensor or ongoing fouling has not been successfully reported. The models implemented by Dereszynski and Dietterich (2007) identify some biases resulting from abrupt shifts in conditions; however, the authors acknowledge that complex anomalies are outside of the performance of their detector. Conde (2011) was unable to identify labeled anomalies with relatively small variation from the measured baseline. Leigh et al. (2018) intentionally prioritized outliers in development of anomaly detection techniques for aquatic sensors. Given that existing methods have not addressed anomalies caused by drift and fouling, there is significant room for improvement in methods for detecting these types of anomalies. We examined both outliers and more subtle anomaly types in our methods and software implementation.

A.7 Reproducibility

Although effectively implemented for specific case studies in the research realm, none of the techniques described in the cited studies have been packaged as easily accessible software for broad application and dissemination. Without reusable code, the specifics of the algorithms as implemented with environmental data cannot be examined, further tested, or applied to other datasets. Recent work in outlier detection was encapsulated in an R package (Talagala et al., 2019); however, a lack of documentation made it difficult to know how to install the package and apply the methods to our datasets. Provenance of data from raw field observations to quality controlled data products is vitally important yet rarely described in sufficient detail that the process used to arrive at final data products could be repeated (Horsburgh et al., 2015). Applying more automated techniques can help, and reusable software tools can overcome barriers related

to understanding and implementing complex algorithms for practical application. Rather than a model calibrated to a specific variable/site combination, practitioners need tools that can be applied to a broad suite of variables and/or monitoring locations documented in a reusable and reproducible way. Thus, we sought to package the tools we developed as open source software that could easily be deployed in a commonly available analytical environment.

A.8 Anomaly Correction

Various techniques and past studies developed functionality for detecting anomalies, but few applied corrective actions, which is an important and time consuming step in quality control post processing. A handful of studies used modeled ARIMA forecasts to directly replace anomalies that were detected by the same ARIMA model, termed ‘anomaly detection and mitigation’ (ADAM) (Hill and Minsker, 2010; Leigh et al., 2018). However, the objective of ADAM was to improve detection by ensuring that model input data did not include detected anomalies, not to generate a corrected version of the dataset. Furthermore, the success of ADAM was mixed and resulted in high rates of false positives (Leigh et al., 2018). Given the general lack of available methods for automated correction, we explored new approaches for inclusion in the software package we developed.

REFERENCES

Ahmad, S., Lavin, A., Purdy, S., Agha, Z., 2017. Unsupervised real-time anomaly detection for streaming data. Neurocomputing 262, 134–147. https://doi.org/10.1016/j.neucom.2017.04.070

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, Wade, M., Boose, E.R., 2013. Quantity is Nothing without Quality. Bioscience 63, 574–585. https://doi.org/10.1525/bio.2013.63.7.10

Chandola, V., Banerjee, A., Kumar, V., 2009. Survey of Anomaly Detection. ACM Comput. Surv. 41, 1–72. https://doi.org/10.1145/1541880.1541882

Christ, M., Braun, N., Neuffer, J., Kempa-Liehr, A.W., 2018. Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package). Neurocomputing 307, 72–77. https://doi.org/10.1016/j.neucom.2018.03.067

Conde, E.F., 2011. Environmental Sensor Anomaly Detection Using Learning Machines. Learning. Utah State University.

Cook, A., Misirli, G., Fan, Z., 2020. Anomaly Detection for IoT Time-Series Data: A Survey. IEEE Internet Things J. 1–1. https://doi.org/10.1109/jiot.2019.2958185

Dereszynski, E.W., Dietterich, T.G., 2007. Probabilistic Models for Anomaly Detection in Remote Sensor Data Streams, in: Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence (UAI2007). pp. 75–82.

Galarus, D., Angryk, R., Sheppard, J., 2012. Automated weather sensor quality control. Proc. 25th Int. Florida Artif. Intell. Res. Soc. Conf. FLAIRS-25 388–393.

Giustarini, L., Parisot, O., Ghoniem, M., Hostache, R., Trebs, I., Otjacques, B., 2016. A user-driven case-based reasoning tool for infilling missing values in daily mean river flow records. Environ. Model. Softw. 82, 308–320. https://doi.org/10.1016/j.envsoft.2016.04.013

Hill, D.J., Minsker, B.S., 2010. Anomaly detection in streaming environmental sensor data: A data-driven modeling approach. Environ. Model. Softw. 25, 1014–1022. https://doi.org/10.1016/j.envsoft.2009.08.010

Hill, D.J., Minsker, B.S., 2006. Automated Fault Detection for In-Situ Environmental Sensors, in: 7th International Conference on Hydroinformatics.

Hill, D.J., Minsker, B.S., Amir, E., 2009. Real-time Bayesian anomaly detection in streaming environmental data. Water Resour. Res. 45, 1–16. https://doi.org/10.1029/2008WR006956

Horsburgh, J.S., Reeder, S.L., Jones, A.S., Meline, J., 2015. Open source software for visualization and quality control of continuous hydrologic and water quality sensor data. Environ. Model. Softw. 70, 32–44. https://doi.org/10.1016/j.envsoft.2015.04.002

Hundman, K., Constantinou, V., Laporte, C., Colwell, I., Soderstrom, T., 2018. Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding. Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. 387–395. https://doi.org/10.1145/3219819.3219845

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

Leigh, C., Alsibai, O., Hyndman, R.J., Kandanaarachchi, S., King, O.C., McGree, J.M., Neelamraju, C., Strauss, J., Talagala, P.D., Turner, R.S., Mengersen, K., Peterson, E.E., 2018. A framework for automated anomaly detection in high frequency wate r-quality data from in situ sensors. Sci. Total Environ. 664, 885–898. https://doi.org/10.1016/j.scitotenv.2019.02.085

Li, J., Pedrycz, W., Jamal, I., 2017. Multivariate time series anomaly detection: A framework of Hidden Markov Models. Appl. Soft Comput. J. 60, 229–240. https://doi.org/10.1016/j.asoc.2017.06.035

Lindemann, B., Fesenmayr, F., Jazdi, N., Weyrich, M., 2019. Anomaly detection in discrete manufacturing using self-learning approaches. Procedia CIRP 79, 313–318. https://doi.org/10.1016/j.procir.2019.02.073

Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., Shroff, G., 2016. LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection.

Moatar, F., Miquel, J., Poirel, A., 2001. A quality-control method for physical and chemical monitoring data. Application to dissolved oxygen levels in the river Loire (France). J. Hydrol. 252, 25–36. https://doi.org/10.1016/S0022-1694(01)00439-5

Mourad, M., Bertrand-Krajewski, J.-L., 2002. A method for automatic validation of long time series of data in urban hydrology. Water Sci. Technol. 45, 263 LP – 270.

Papacharalampous, G., Tyralis, H., Koutsoyiannis, D., 2019. Comparison of stochastic and machine learning methods for multi-step ahead forecasting of hydrological processes, Stochastic Environmental Research and Risk Assessment. Springer Berlin Heidelberg. https://doi.org/10.1007/s00477-018-1638-6

Russo, S., Lürig, M., Hao, W., Matthews, B., Villez, K., 2020. Active Learning for Anomaly Detection in Environmental Data. Environ. Model. Softw. https://doi.org/10.1016/j.envsoft.2020.104869

Sheldon, W.M., 2008. Dynamic, Rule-based Quality Control Framework for Real-time Sensor Data, in: Gries, C., Jones, M.B. (Eds.), Proceedings of the Environmental Information Management Conference. Albuquerque, NM, pp. 145–150.

Smolyakov, D., Sviridenko, N., Ishimtsev, V., Burikov, E., Burnaev, E., 2019. Learning Ensembles of Anomaly Detectors on Synthetic Data. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) 11555 LNCS, 292–306. https://doi.org/10.1007/978-3-030-22808-8_30

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 Resour. Res. 55, 8547–8568. https://doi.org/10.1029/2019WR024906

Tan, P.-N., Steinback, M., Karpatne, A., Kumar, V., 2019. Introduction to Data Mining, second. ed. Pearson, New York.

Taylor, J.R., Loescher, H.L., 2013. Automated quality control methods for sensor data: a novel observatory approach. Biogeosciences 9, 18175–18210. https://doi.org/10.5194/bg-10-4957-2013

Taylor, S.J., Letham, B., 2018. Forecasting at Scale. Am. Stat. 72, 37–45. https://doi.org/10.1080/00031305.2017.1380080

Tran, L., Fan, L., Shahabi, C., 2019. Outlier Detection in Non-stationary Data Streams 25–36. https://doi.org/10.1145/3335783.3335788

World Meteorological Organization, 2008. Guide to meteorological instruments and methods of observation. WMO-No 8. Geneva.

Yin, C., Zhang, S., Wang, J., Xiong, N.N., 2020. Anomaly Detection Based on Convolutional Recurrent Autoencoder for IoT Time Series. IEEE Trans. Syst. Man, Cybern. Syst. 1–11. https://doi.org/10.1109/tsmc.2020.2968516

Appendix B. List of pyhydroqc Files and Functions

This appendix provides a listing of each of the Python files in the pyhydroqc package and describes the functionality that each provides. More detailed documentation is found in the GitHub repository and package documentation (see the Software Availability Section).

parameters.py: This file contains assignments of parameters for all steps of the anomaly detection workflow. Parameters are defined specific to each site and observed variable that are referenced in the detect script. LSTM parameters are consistent across sites and variables. ARIMA hyper parameters are specific to each site/variable combination, other parameters are used for rules based anomaly detection, determining dynamic thresholds, and for widening anomalous events.

anomaly_utilities.py: Contains functions for performing anomaly detection and correction:

- **get_data:** Retrieves and formats data. Retrieval is based on site, observed variable, and year. To pass through subsequent steps, the required format is a Pandas data frame with columns corresponding to datetime (as the index), raw data, corrected data, and data labels (anomalies identified by technicians).
- **anomaly_events:** Widens anomalies and indexes events or groups of anomalous data.
- **assign_cm:** A helper function for resizing anomaly events to the original size for determining metrics.
- **compare_events:** Compares anomaly events detected by an algorithm to events labeled by a technician.
- **metrics:** Determines performance metrics of the detections relative to labeled data.

- **event_metrics:** Determines performance metrics based on number of events rather than the number of data points.
- **print_metrics:** Prints the metrics to the console.
- **group_bools:** Indexes contiguous groups of anomalous and valid data to facilitate correction.
- **xfade:** Uses a cross-fade to blend forecasted and backcasted data over anomaly events for generating data correction.
- **set_dynamic_threshold:** Creates a threshold that varies dynamically based on the model residuals.
- **set_cons_threshold:** Creates a threshold of constant value.
- **detect_anomalies:** Uses model residuals and threshold values to classify anomalous data.
- **aggregate_results:** Combines the detections from multiple models to give a single output of anomaly detections.
- **plt_threshold:** Plots thresholds and model residuals.
- **plt_results:** Plots raw data, model predictions, detected and labeled anomalies.
- modeling_utilities.py
- Contains functions for building and training models:
- **pdq:** Automatically determines the (p, d, q) hyperparameters of a time series for ARIMA modeling.
- **build_arima_model, LSTM_univar, LSTM_multivar, LSTM_univar_bidir, LSTM_multivar_bidir:** wrappers that call other functions in the file to scale and reshape data (for LSTM models only), create and train a model, and output model

predictions and residuals.

- **create_scaler:** Creates a scaler object for scaling and unscaling data.
- **create_training_dataset**, **create_bidir_training_dataset:** Creates a training dataset based on a random selection of points from the dataset. Reshapes data to include the desired time_steps for input to the LSTM model - the number of past data points to examine or past and future points (bidirectional). Ensures that data already identified as anomalous (i.e., by rules based detection) are not used.
- **create_sequenced_dataset**, **create_bidir_sequenced_dataset:** Reshapes all inputs into sequences that include time_steps for input to the LSTM model - using either only past data points or past and future data points (bidirectional). Used for testing or for applying the model to a full dataset.
- **create_vanilla_model, create_bidir_model:** Helper functions used to create single layer LSTM models.
- **train_model:** Fits the model to training data. Uses a validation subset to monitor for improvements to ensure that training is not too long.

rules_detect.py: Contains functions for rules based anomaly detection and preprocessing. Depends on anomaly_utilities.py. Functions include:

- **range_check:** Scans for data points outside of user defined limits and marks the points as anomalous.
- **persistence:** Scans for repeated values in the data and marks them as anomalous if the duration exceeds a user defined length.
- **group_size:** Determines the maximum length of anomalous groups identified by the previous steps.

- **interpolate:** Corrects data points with linear interpolation, a typical approach for short anomalous events.
- **add_labels:** Enables the addition of anomaly labels (referring to anomalies previously identified by an expert) in the case that labels may have been missed for corrected data that are NaN or have been set to a no data value (e.g., -9999).

calibration.py: Contains functions for identifying and correcting calibration events. Functions include:

- **calib_edge_detect:** Identifies possible calibration event candidates by using edge filtering.
- **calib_persist_detect:** Identifies possible calibration event candidates based on persistence of a user defined length.
- **calib_overlap:** Identifies possible calibration event candidates by finding concurrent events of multiple sensors from the calib_persist_detect function.
- **find_gap:** Determines a gap value for a calibration event based on the largest data difference within a time window around a datetime.
- **lin_drift_cor:** Performs linear drift correction to address sensor drift given calibration dates and a gap value.

model_workflow.py: Contains functionality to build and train ARIMA and LSTM models, apply the models to make predictions, set thresholds, detect anomalies, widen anomalous events, and determine metrics. Depends on anomaly_utilities.py, modeling_utilities.py, and rules_detect.py. Wrapper function names are: **ARIMA_detect, LSTM_detect_univar,** and **LSTM_detect_multivar**. LSTM model workflows include options for vanilla or bidirectional. Within each wrapper function, the full detection

workflow is followed. Options allow for output of plots, summaries, and metrics.

ARIMA_correct.py: Contains functionality to perform corrections and plot results using ARIMA models. Depends on anomaly_utilities.py.

- **ARIMA_group:** Ensures that the valid data surrounding anomalous data points and groups of data points are sufficient forecasting/backcasting.
- **ARIMA_forecast:** Creates predictions of data where anomalies occur.
- **generate_corrections:** The primary function for determining corrections. Passes through data with anomalies and determines corrections using piecewise ARIMA models. Corrections are determined by averaging together (cross fade) both a forecast and a backcast.

Appendix C. Anomaly Detection and Correction Examples

This appendix includes additional examples of anomaly detection and correction performed by the pyhydroqc workflow on LRO case study data.

Figure C1 illustrates anomaly detection false positives and true positives. Peaks and troughs in the data were considered anomalies by the model (ARIMA), but only two of them (2017-12-18 and 2017-12-26) were labeled by the technician. It is unclear why certain peaks were labeled by the technician while others were not. Although this example includes several false positives, the algorithm behaved as expected.

In some cases, the apparent success of the model results may be an artifact of both the generalization of detections in the 'compare_events' function and the liberal application of labels by technicians. Some time series contain extensive periods of data labeled as anomalous that correspond to concerns with sensor validity or site conditions (e.g., Figure C2). When comparing events to determine confusion matrix categories, any overlap in model detections results in all points of the anomalous period being identified as true positives. This is an example where large events may bias the metrics toward true positives if any point in the event is detected or toward false negatives if the event goes undetected (less likely). This particular event contributes to the 13,000+ true positives for this time series (pH at Main Street).

We were interested in whether the models could detect calibration events. For one time series (pH at Main Street), one model type (LSTM multivariate bidirectional) detected approximately 20% of labeled calibration events. We found that the master list of calibrations recorded in the field notes differs from what technicians labeled in the data. Some calibrations recorded in the field notes were not labeled by technicians in the

data, and other events labeled by technicians appeared to be calibrations but were not part of the master list derived from the field notes. These discrepancies point to deficiencies in the labeled data. The model predictions are erratic and do not track the observations at most calibration events (Figure C3a), even if the threshold was not sensitive enough to result in detections. In some cases, calibration events were detected as anomalous by the model (Figure C3b), but there was no mechanism to distinguish from other anomalies. These examples illustrate the challenge of using the model based approach for detecting and correcting calibration events.

A direct comparison of results from each model type illustrates model behaviors and associated detections. For specific conductance at Tony Grove, where there was variability in performance between model types (see Section 2.3.4), the ARIMA and LSTM multivariate vanilla models detected points at the edges of long duration labeled events, improving their performance metrics relative to the other model types. Figure C4 further illustrates differences between model estimates and resulting detections. For the first date range, the estimates of both multivariate models deviate from the original data because they use other variables as input. In the absence of this information, only one univariate model detects an anomaly. In the second date range, models responded to the localized event in distinct ways, and none resulted in a detection. In the third date range, estimates from the multivariate models exhibit spikes around the detections illustrating that information is coming from other variables. It is likely that some of these labeled anomalies correspond to calibration events for which other variables exhibited greater shifts than did specific conductance.

Although the correction algorithm was capable of capturing diurnal oscillations,

in some cases, data patterns did not translate and propagate through the corrections (e.g., Figure C5). Because each correction is based on individual, independent models trained for data immediately prior to and following an anomalous event, the number of data points considered can vary. Even though the adjacent data used for input is limited by the maximum duration parameter, some models may still overgeneralize (i.e., a straight line). Other models may use so little data that a pattern is missed, while still others are focused on a single dominant feature (i.e., an oscillation or a curve). Furthermore, a pattern may be damped over an extended time period. Explicitly incorporating seasonality into development of the ARIMA models may result in more consistent output of oscillations. However, developing seasonal ARIMA models is computationally demanding, and the correction algorithm already requires significant computational resources.

The correction algorithm is directly dependent on identified anomalies. In Figure C5c, an anomalous event (2018-06-19 – 2018-06-20) was detected by the model, but even with widening, the initial abrupt decrease was not labeled anomalous, so it was considered valid data, and it directly influenced the forecast. For the correction algorithm to be effective, anomalies should be reviewed and may need adjustment (e.g., further widening).

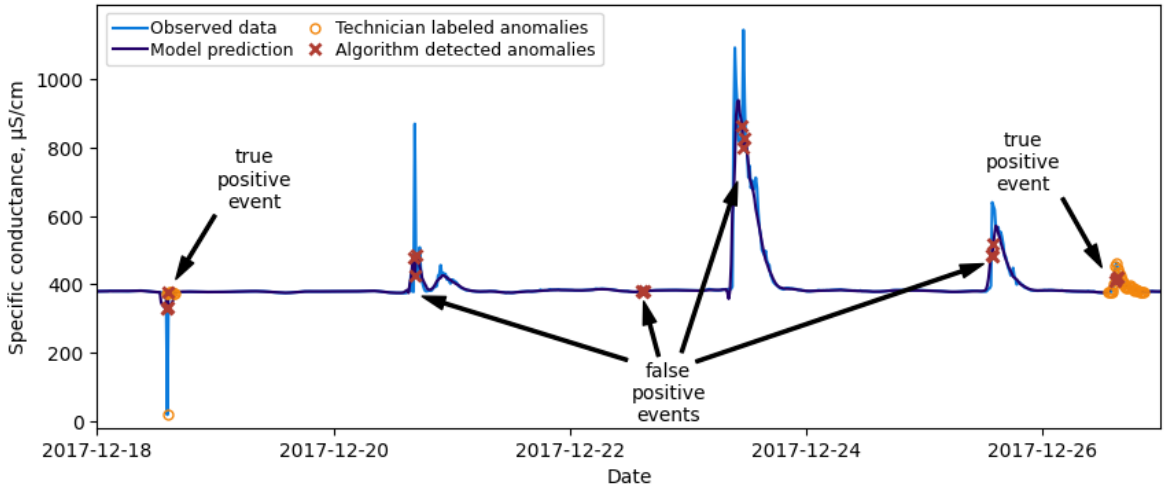

Figure C1. Examples of anomalies detected using an ARIMA model for specific conductance at Main Street.

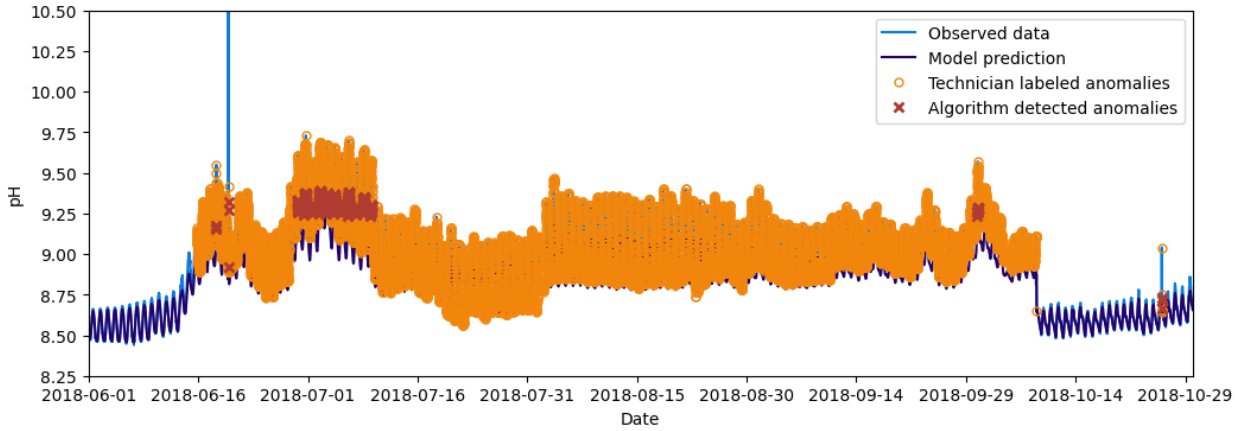

Figure C2. Examples of anomalies detected using an LSTM multivariate bidirectional model for pH at Main Street for of an extended period of data labeled as a sensor malfunction.

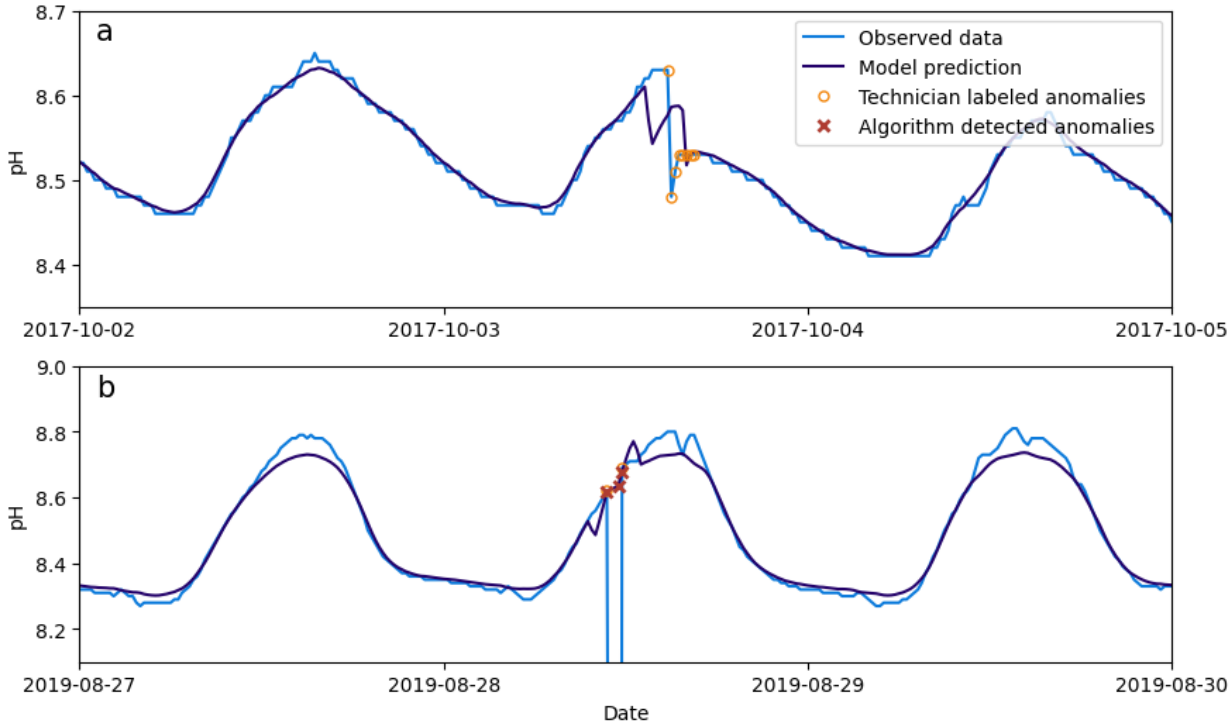

Figure C3. Examples of anomalies detected using an LSTM multivariate bidirectional model on a pH sensor at Main Street with calibration events.

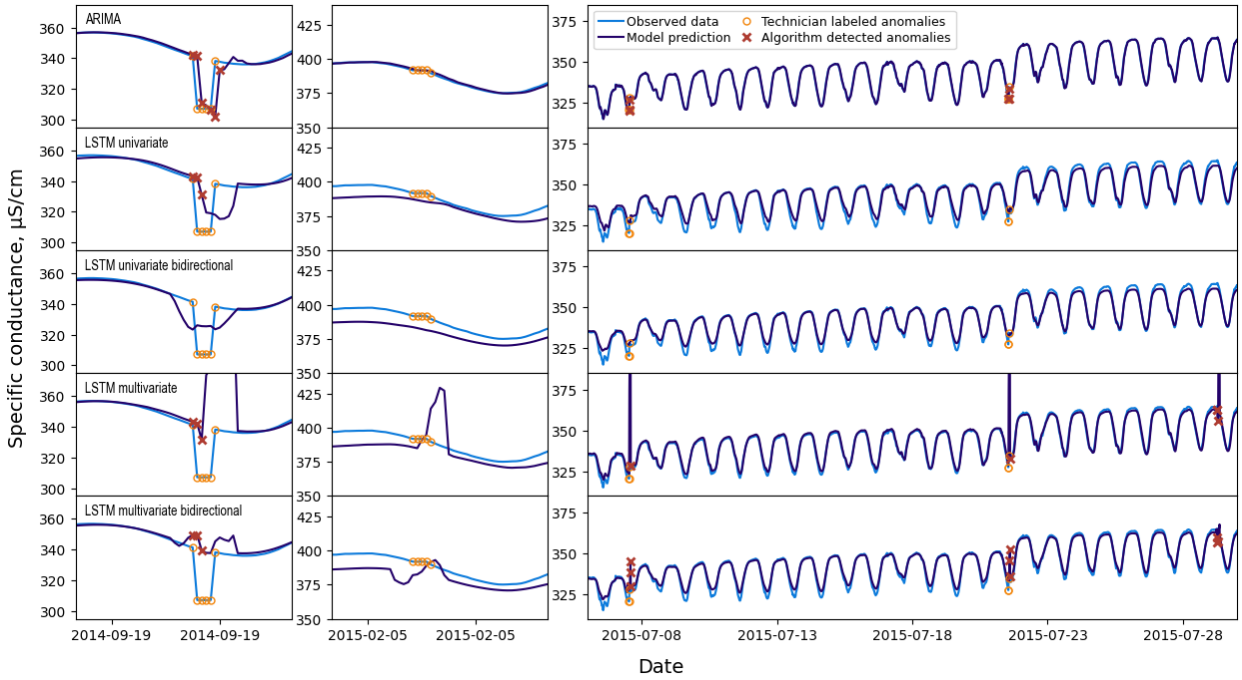

Figure C4. Examples comparing model estimates and detected anomalies for all model types for specific conductance at Tony Grove

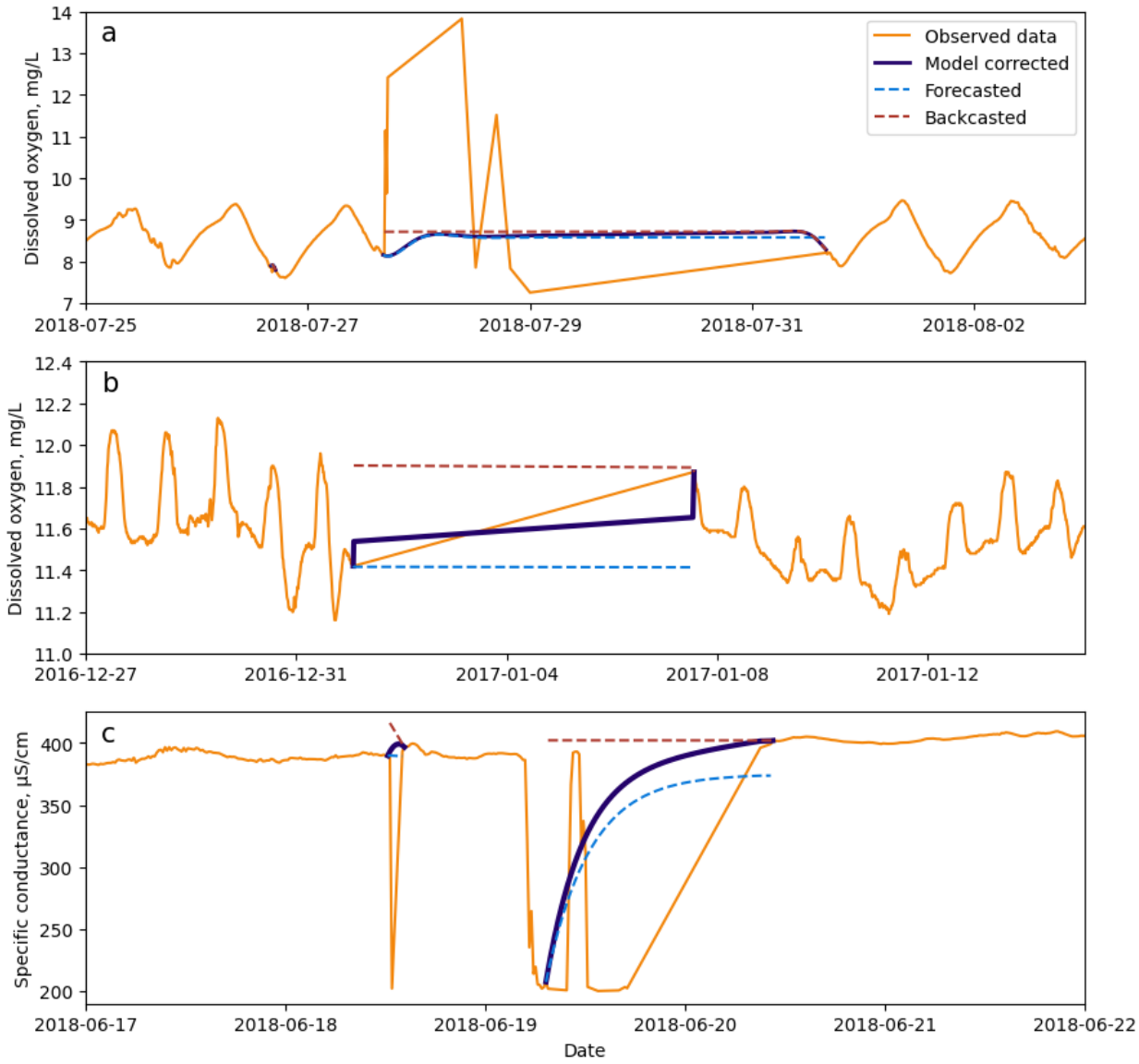

Figure C5: Examples of problematic algorithm correction. a and b: dissolved oxygen at Tony Grove, c: specific conductance at Mendon.

CURRICULUM VITAE

Amber Spackman Jones
Department of Civil and Environmental Engineering
Utah Water Research Laboratory
Utah State University, 8200 Old Main Hill, Logan, UT 84322-8200 USA
Phone: (435) 512-2503
Email: amber.jones@usu.edu

Education

Ph.D. Civil and Environmental Engineering, Utah State University, Logan, UT, 2023. GPA: 4.0
Dissertation: Water Data Science: Data Driven Techniques, Training, and Tools for Improved Management of High Frequency Water Resources Data. Advisor: Jeffery S. Horsburgh.

M.S. Civil and Environmental Engineering, Utah State University, Logan, UT, 2008. GPA: 4.0
Thesis: Estimating total phosphorus and total suspended solids loads from high frequency data. Advisor: David K. Stevens.

B.S. Environmental Engineering, Utah State University, Logan, UT, 2006. GPA: 3.89
Magna Cum Laude. Additional coursework in Spanish. Passed Fundamentals of Engineering Exam 2003.

Professional Experience

Physical Scientist: 2022 – present
United States Geological Survey, Water Mission Area.

Graduate Research and Teaching Assistant: 2019 – 2022
Civil and Environmental Engineering, Logan UT

Research Engineer: 2011 – 2018
Utah Water Research Laboratory, Utah State University, Logan, UT

Environmental Scientist II: 2010 – 2011
Public Works Department, Hillsborough County, FL

Research Assistant: 2006 – 2009
Utah Water Research Laboratory, Logan, UT

Board Member: 2006 – 2009
Engineers Without Borders, Utah State University, Logan, UT

Ambassador: 2002 – 2008
Utah State University College of Engineering, Logan, UT

Teaching Fellow: 2006
Water Engineering, USU Civil and Environmental Engineering, Logan, UT

Research Technician: 2001 – 2006
Environmental Management Research Group, Logan, UT

Expertise

My research and training are in watershed hydrology, surface water quality, data science and machine learning, hydroinformatics, and environmental engineering. As a data manager, I worked with interdisciplinary teams to facilitate reproducible research for water related data by

developing policy, providing training, and curating datasets. As a software product manager, I defined requirements, tested, and implemented environmental cyberinfrastructure. My dissertation research focuses on data science approaches for managing and interpreting high frequency water data.

Awards

USU College of Engineering Outstanding PhD Scholar Award, 2023.

USU Civil and Environmental Engineering Doctoral Student of the Year, 2023.

W.C. Swanson Non-traditional Scholarship, Utah State University, 2022.

FAIR Cyber Training Fellow for Climate and Water, Purdue University, 2021.

USU Cyberinfrastructure for Intelligent Water Supply, Data Visualization Challenge, First Place, 2020.

USU College of Engineering Technical Writing Competition Award, Graduate Division, 2019 and 2021.

American Water Works Association Intermountain Section, Graduate Student Scholarship, 2007.

Air and Waste Management Association Rocky Mountain Section, Graduate Student Scholarship, 2007.

USU Civil and Environmental Engineering Senior Student of the Year, 2006.

USU College of Engineering Teaching Fellow of the Year, 2006.

USU Civil and Environmental Engineering Sophomore Student of the Year, 2003.

Presentations and Publications

Journal Papers in Print or Press

1. Jones, A.S., Horsburgh, J.S., Bastidas Pacheco, C.J., Flint, C.G., Lane, B.A. (2022). Advancing Hydroinformatics and Water Data Science Instruction: Community Perspectives and Online Learning Resources. *Frontiers in Water.* https://doi.org/10.3389/frwa.2022.901393
2. Jones, A.S., Jones, T.L., Horsburgh, J.S. (2022). Toward automating post processing of aquatic sensor data. *Environmental Modelling and Software*, 151. https://doi.org/10.1016/j.envsoft.2022.105364
3. Rosenberg, D., Jones, A.S., Filion, Y., Teasley, R., Sandoval-Solis, S., Stagge, J., Abdallah, A., Castronova, A., Ostfeld, A., & Watkins, D. (2021). Reproducible Results Policy. *Journal of Water Resources Planning and Management, 147*(2). http://doi.org/10.1061/(ASCE)WR.1943-5452.0001368
4. 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, doi: 10.1016/j.ecoinf.2018.05.001.
5. Jones, A.S., Aanderud, Z.T., Horsburgh, J.S., Eiriksson, D.P., Dastrup, D., Cox, C., Jones, S.B., Bowling, D.R., Carlisle, J., Carling, G.T., Baker, M.A. (2017) Designing and Implementing a Network for Sensing Water Quality and Hydrology Across Mountain to Urban Transitions. *Journal of the American Water Resources Association,* 53:5, 1095-1120, doi: 10.1111/1752-1688.12557.

6. Flint, C.G., Jones, A.S., Horsburgh, J.S. (2017). Data Management Dimensions of Social Water Science: The iUTAH Experience. *Journal of the American Water Resources Association*, 53:5, 988-996, doi: 10.1111/1752-1688.12568.
7. Jones, A.S., Horsburgh, J.S., Jackson-Smith, D., Ramirez, M., Flint, C.G., and Caraballo, J. (2016). A Web-based, interactive visualization tool for social environmental survey data, *Environmental Modelling & Software*, 84, 412-426, doi:10.1016/j.envsoft.2016.07.013.
8. Horsburgh, J. S., Aufdenkampe, A. K., Mayorga, E., Lehnert, K. A., Hsu, L., Song, L., Jones, A.S., Damiano, S. G., Tarboton, D. G., Valentine, D., Zaslavsky, I., Whitenack, T. (2016). Observations Data Model 2: A community information model for spatially discrete Earth observations, *Environmental Modelling & Software*, 79, 55-74, doi:10.1016/j.envsoft.2016.01.010.
9. Jones, A.S., J. S. Horsburgh, S.L. Reeder, M. Ramirez, J. Caraballo (2015). A data management and publication workflow for a large-scale, heterogeneous sensor network, *Environmental Monitoring and Assessment*, 187:348, doi:10.1007/s10661-015-4594-3.
10. Horsburgh, J. S., S. L. Reeder, A. S. Jones, J. Meline (2015). Open source software for visualization and quality control of continuous hydrologic and water quality sensor data, *Environmental Modelling & Software*, 70, 32-44, doi:10.1016/j.envsoft.2015.04.002.
11. Hale, R. L., A. Armstrong, M. A. Baker, S. Bedingfield, D. Betts, C. Buahin, M. Buchert, T. Crowl, R. R. Dupont, J. R. Ehleringer, J. Endter-Wada, C. Flint, J. Grant, S. Hinners, J. S. Horsburgh, D. Jackson-Smith, A. S. Jones, C. Licon, S. E. Null, A. Odame, D. E. Pataki, D. Rosenberg, M. Runberg, P. Stoker, and C. Strong (2015), iSAW: Integrating structure, actors, and water to study socio-hydro-ecological systems, *Earth's Future*, 3, doi:10.1002/2014EF000295.
12. Jones, A. S., J. S. Horsburgh, N. O. Mesner, R. J. Ryel, and D. K. Stevens (2012), Influence of Sampling Frequency on Estimation of Annual Total Phosphorus and Total Suspended Solids Loads. *Journal of the American Water Resources Association,* 1-18, doi:10.1111/j.1752-1688.2012.00684.x.
13. Jones, A.S., D. K. Stevens, J. S. Horsburgh, and N. O. Mesner (2011), Surrogate measures for providing high frequency estimates of total suspended solids and total phosphorus concentrations, *Journal of the American Water Resources Association*, 47(2), 239-253, doi:10.1111/j.1752-1688.2010.00505.x.
14. Horsburgh, J. S., A. S. Jones, D. K. Stevens, D. G. Tarboton, and N. O. Mesner (2010), A sensor network for high frequency estimation of water quality constituent fluxes using surrogates, *Environmental Modelling & Software*, 25, 1031-1044, doi:10.1016/j.envsoft.2009.10.012.

Selected Datasets, Online Resources, and Code Repositories

1. Jones, A.S., J.S. Horsburgh, C.J. Bastidas Pacheco. (2022). Hydroinformatics and Water Data Science. HydroLearn. https://edx.hydrolearn.org/courses/course-v1:USU+CEE6110+2022/about
2. Jones, A.S., J. S. Horsburgh, C. J. Bastidas Pacheco (2022). Hydroinformatics Instruction Modules Example Code, HydroShare, http://www.hydroshare.org/resource/761d75df3eee4037b4ff656a02256d67 (collection consisting of 4 resources)
3. Jones, A. S., J. S. Horsburgh, C. G. Flint (2022). Hydroinformatics and Water Data Science Instructor Interviews and Surveys, HydroShare, https://doi.org/10.4211/hs.15b1a61f47724a6e8deb100789353df2

4. Jones, A.S., T.L. Jones, J.S. Horsburgh (2022). pyhydroqc v0.0.4 Zenodo, 10.5281/zenodo.6336536. More available: https://ambersjones.github.io/pyhydroqc/ and https://pypi.org/project/pyhydroqc/
5. Jones, A. S. (2022). pyhydroqc Sensor Data QC: Single Site Example, HydroShare, https://doi.org/10.4211/hs.92f393cbd06b47c398bdd2bbb86887ac
6. Jones, A. S., J. S. Horsburgh, T. Jones (2021). Techniques for Increased Automation of Aquatic Sensor Data Post Processing in Python: Video Presentation, HydroShare, http://www.hydroshare.org/resource/bc5c616426214b60b068352ae028d963
7. iUTAH Data Manager, iUTAH GAMUT Working Group (2021). iUTAH GAMUT Environmental Observatory Collected Datasets, HydroShare, (collection consisting of 73 resources). http://www.hydroshare.org/resource/02b155615b794671bfc4c48870b3ce6f
8. Logan River Observatory (2021). Logan River Observatory Datasets, HydroShare, (collection consisting of 51 resources). http://www.hydroshare.org/resource/def147fa06de4c67810586d45337b413
9. Jones, A. S., S. M. Alger, H. Salehabadi, A. Repko (2019). Elasticity in the Colorado River Basin Using the Budyko Method, HydroShare, http://www.hydroshare.org/resource/692cd36ffac24978b13b7352f62532ff
10. Jones, A. S., S. M. Alger, H. Salehabadi (2019). Postponing Equalization to Retain Accumulated Sediment in the Grand Canyon Ecosystem, HydroShare, http://www.hydroshare.org/resource/8860d75de30747e2a06e06f2d9783a8e
11. Jones, A.S., W. Rhoads, J. S. Horsburgh (2019). Water Quality Data - RAPID Maria Project, HydroShare, http://www.hydroshare.org/resource/ddccfc3133034c43b04ebedac2822a23
12. Tennant, H., A. S. Jones (2019). Development and Implementation of Database and Analyses for High Frequency Data, HydroShare, http://www.hydroshare.org/resource/bf57045c30054383a6df9bb8cab381d3
13. Jones, A.S., D. Eiriksson, J. S. Horsburgh (2018). Quality Control Experiment, HydroShare, https://doi.org/10.4211/hs.31f30d14c88748d986842d278d125a5c

Theses

1. Jones, Amber Spackman (2008), Estimating Total Phosphorus and Total Suspended Solids Loads from High Frequency Data, M.S. Thesis, Utah State University, Logan UT. (Available at: http://digitalcommons.usu.edu/etd/205)

Conference Proceedings Papers

1. Cox, S., Smith, T., Jones, T.L., Droge, G., Jones, A.S. (2020) Power-Optimal Slew Maneuvers in Support of Small Satellite Earth Imaging Missions. SSC20-P3015. *34th Annual Small Satellite Conference*, Utah State University, July 2020, Logan Utah.
2. Horsburgh, J.S., A.S. Jones, M. Ramirez, Caraballo, J. (2016). Time Series Analyst: Interactive online visualization of environmental time series data, In Sauvage, S., Sanchez-Perez, J.M., Rizzoli, A. (Eds.), Proceedings of the 8th International Congress on Environmental Modeling and Software, July 11-15, Toulouse France.
3. Horsburgh, J. S., A. S. Jones, S. Reeder (2014). ODM Tools Python: Open source software for managing continuous sensor data, In: Proceedings of the 11th International Conference on Hydroinformatics, 17-21 August, New York City, NY.
4. Horsburgh, J. S., A. S. Jones, S. Reeder (2014). Automating data management and sharing within a large-scale, heterogeneous sensor network, In: Ames, D.P., Quinn, N.W.T., Rizzoli,

A.E. (Eds.), Proceedings of the 7th International Congress on Environmental Modelling and Software, June 15-19, San Diego, California, USA. ISBN: 978-88-9035-744-2.

Technical Reports and White Papers

1. Jones, A.S., Brazil, L. (2018). HydroShare Guide for Data Authors and Publishers. Consortium of Universities for the Advancement of Hydrologic Science, Inc. https://help.hydroshare.org/static/media/uploads/hydroshare_guide_for_data_authors_and_publishers.pdf
2. Horsburgh, J.S., A.S. Jones (Eds.) (2016). iUTAH Research Data Policy Version 1.7, innovative Urban Transitions and Aridregion Hydrosustainability, Logan, UT.
3. Jones, A.S., Eiriksson, D., Cox, C., Crawford, J. (2014). iUTAH GAMUT Network Quality Assurance and Quality Control Plan Version 1.3, innovative Urban Transitions and Aridregion Hydrosustainability, Logan, UT.
4. Neilson, B. T., J. S. Horsburgh, D. K. Stevens, M. R. Matassa, J. N. Brogdon, and A. Spackman (2004), Comparison of Complex Watershed Models' Predictive Capabilities: EPRI's Watershed Analysis Risk Management Framework (WARMF) vs. USEPA's Better Assessment Science Integrating Point and Nonpoint Sources (BASINS/WinHSPF), Utah Water Research Laboratory, Utah State University, Logan, UT.

Conference Presentations, Posters, and Abstracts

1. Jones, A.S. (2023). NuGo2: A Centralized and Standardized Program for Data Monitoring and Alerts., Presented at USGS National Water Data Training Workshop, Phoenix, AZ, 24 August.
2. Jones, A.S., Walker, W.J. (2023). Progress Toward Automating Aquatic Time Series Records Processing at the USGS. Presented at National Monitoring Conference, Virginia Beach, VA. 26 April.
3. Jones, A.S. (2022). Progress Toward Automated QAQC – Research and Applications, Presented at Nordic Hydrometry Workshop, 29 September.
4. Jones, A.S., Horsburgh, J.S., Flint, C.G., Lane, B.A., Bastidas Pacheco, C.J. (2022). Water Data Science and Hydroinformatics Instruction: Community Perspectives and Online Learning Resources, Presented at Data Science and Open Science Virtual Summit, 29 July.
5. Jones, A.S., Horsburgh, J.S., Flint, C.G., Lane, B.A., Bastidas Pacheco, C.J. (2022). Community Perspectives and Online Learning Resources for Advancing Instruction for Hydroinformatics and Water Data Instruction, Abstract 401-03, Presented at the 2022 AGU Frontiers in Hydrology Meeting, 23 June.
6. Jones, A.S., Horsburgh, J.S., Jones, T.L. (2021). pyhydroqc: A Python Package for Automating and Streamlining Aquatic Sensor Data Post Processing. Abstract H21B-07, Presented at American Geophysical Union Fall Meeting. 14 December.
7. Jones, A.S., Jones, T.L., Horsburgh, J.S. (2021). Techniques for Increased Automation of Aquatic Sensor Data Post Processing in Python, Presented at National Monitoring Conference, 21 April
8. Jones, A.S., Alger, S.M., Salehabadi, H., Repko, A, Lane, B. (2019). Sensitivity to Climate Change in the Colorado Basin Using the Budyko Method, Presented at Universities Council on Water Research Annual Meeting, Snowbird, UT, 11 June.
9. Jones A.S., Horsburgh, J.S., Eiriksson, D. (2017). Assessing Subjectivity in Sensor Data Post Processing via a Controlled Experiment, Abstract IN41C-0050, Presented at the 2017 AGU Fall Meeting, New Orleans, LA, 11-15. December.
10. Eiriksson, D., Jones A.S., Horsburgh, J.S., Cox, C., Dastrup, D. (2017). Data Quality Control: Challenges, Methods, and Solutions from an Eco-Hydrologic Instrumentation Network,

Abstract IN41C-0056, Presented at the 2017 AGU Fall Meeting, New Orleans, LA, 11-15. December.

11. Bandaragoda, C., Phuong, J., Mooney, S., Stephens, K., Istanbulluoglu, E., Pieper, K., Rhoads, W., Edwards, M., Pruden, A., Bales, J., Clark, E., Brazil, L., Leon, M., Horsburgh, J.S., Tarboton, D.G., Jones, A.S., Hutton, E., Tucker, G.E., McCready, L., Peckham, S.D., Lenhardt, W.C., Idaszak, R. (2017). Building infrastructure to prevent disasters like Hurricane Maria, Abstract NH23E-2888, Presented at the 2017 AGU Fall Meeting, New Orleans, LA, 11-15. December.
12. Aanderud, Z.T., Jones, A.S., Horsburgh, J.S., Eiriksson, D., Dastrup, D., Cox, C., Jones, S., Bowling, D.R., Gabel, A.B., Call, A.M., Carlisle, J., Carling, G., Baker, M.A. (2017). Capturing rapid changes in water quality with high frequency networks across mountain to urban transitions, Abstract PS 39-111, Presented at the 2017 Ecological Society of America, Portland, OR, 6-11 August.
13. Jones, A.S., Horsburgh, J.S., Flint, C.G., Jackson-Smith, D. (2017). Social Water Science Data in iUTAH: Dimensions, Data Management, and Visualization. Presented at iUTAH Annual Symposium and Summer Meeting, Logan, UT, 13-14 July.
14. Jones, A.S., Horsburgh, J.S., Flint, C.G., Jackson-Smith, D. (2016). Social Water Science Data: Dimensions, Data Management, and Visualization, Abstract H34F-03, Presented at the 2016 AGU Fall Meeting, San Francisco, CA, 12-16 December.
15. Horsburgh, J.S., Jones, A.S. (2016). HydroShare for iUTAH: Collaborative Publication, Interoperability, and Reuse of Hydrologic Data and Models for a Large, Interdisciplinary Wtare Research Project, Abstract H43P-03, Presented at the 2016 AGU Fall Meeting, San Francisco, CA, 12-16 December.
16. Jones, A.S., J.S. Horsburgh (2016). Water quality surrogates: Development of surrogate relationships, review of recent advances, and applications, Presented at: National Non-Point Source Monitoring Conference, Salt Lake City, UT, 23-25 August.
17. Horsburgh, J.S., A.S. Jones, M. Ramirez, Caraballo, J. (2016). Time Series Analyst: Interactive online visualization of environmental time series data, Presented at: 8th International Congress on Environmental Modeling and Software, Toulouse France, 11-15 July.
18. Jones, A.S., J.S. Horsburgh, S.L. Reeder, J. Caraballo, D. Smith, Z. Yoshikawa, M. Matos (2016). Streaming Sensor Data: Tools for acquisition, management, and visualization, Presented at: National Water Quality Monitoring Council 10th National Monitoring Conference. Tampa, FL. 5 May.
19. Suiter, P., A. S. Jones, J. S. Horsburgh, B. Mihalevich (2016). Development of a water quality mobile monitoring platform and techniques for managing resulting data, Presented at the Utah State University Spring Runoff Conference, Logan, UT, 5 April.
20. Jones, A.S., Horsburgh, J. S., Matos, M., Caraballo, J. (2015). Equipment management for sensor networks: Linking physical infrastructure and actions to observational data, Abstract IN11D-1793 Presented at the 2015 AGU Fall Meeting, San Francisco, CA, 14-18 December.
21. Castronova, A., J. S. Horsburgh, A. S. Jones (2014). A relational model for simulation data to promote interdisciplinary collaboration, Abstract H13H-1212 Presented at the 2014 AGU Fall Meeting, San Francisco, CA, 15-19 December.
22. Jones, A. S., J. S. Horsburgh (2014). Implementation of cyberinfrastructure and data management workflow for a large-scale sensor network, Abstract IN41A-3645 Presented at the 2014 AGU Fall Meeting, San Francisco, CA, 15-19 December.
23. Jones, A.S., J.S. Horsburgh, S.L. Reeder, J. Meline (2014) ODM Tools Python: Open source software for managing environmental sensor data, Presented at: CUAHSI Field Data Management Solutions Virtual Workshop, Online, 30 October.

24. Horsburgh, J. S., A. S. Jones, S. Reeder (2014). ODM Tools Python: Open source software for managing continuous sensor data, Presented at: 11th International Conference on Hydroinformatics, New York City, NY, 17-21 August.
25. Jones, A.S., J. S. Horsburgh, S. Reeder (2014). Cyberinfrastructure for data management and sharing within a large-scale, heterogeneous sensor network, Presented at: Global Fair and Workshop on Long-Term Observing Systems of Mountain Social-Ecological Systems, Reno NV, July.
26. Horsburgh, J. S., A. S. Jones, S. Reeder (2014). Automating data management and sharing within a large-scale, heterogeneous sensor network, Presented at: 7th International Congress on Environmental Modelling and Software, June 15-19, San Diego, California, USA.
27. Horsburgh, J. S., S. Reeder, J. Patton, A. S. Jones (2014). ODM Tools Python: Open source software for managing hydrologic and water quality time series data, Presented at: National Water Quality Monitoring Council 9th National Monitoring Conference, Cincinnati, OH, 30 April.
28. Jones, A.S., J. S. Horsburgh, M. Ramirez, J. Caraballo (2014). Managing monitoring equipment: A sensor extension for the CUAHSI Observations Data Model, Presented at: National Water Quality Monitoring Council 9th National Monitoring Conference, Cincinnati, OH, 29 April.
29. Jones, A.S., J. S. Horsburgh, S. Reeder (2014). Implementation of a workflow for streaming sensor data for a large-scale hydrologic monitoring network, Presented at: Utah State University Spring Runoff Conference, Logan, UT, 2 April.
30. Horsburgh, J. S., S. Reeder, J. Patton, A. S. Jones (2013). ODM Tools Python: Data management software for hydrologic time series, Presented at: CUAHSI Conference on Hydroinformatics and Modeling, Logan, UT, July.
31. Jones, A.S., J. S. Horsburgh, J. Caraballo, M. Rarmirez (2013). Managing sensor infrastructure using a sensor extension for the ODM2 data model, Presented at: CUAHSI Conference on Hydroinformatics and Modeling, Logan, UT, July.
32. Horsburgh, J. S., A. S. Jones, S. Reeder, J. Patton, J. Caraballo, M. Ramirez, and N. Mouzon (2013). Using CUAHSI HIS to support large scale collaborative research in Utah, CUAHSI HIS Cyberseminar, 1 May.
33. Jones, A.S., J.S. Horsburgh, S.L. Reeder, J. Caraballo (2013). iUTAH cyberinfrastructure to support data collection and management for the GAMUT monitoring network, Presented at: Utah State University Spring Runoff Conference, Logan, UT, 10 April.
34. Horsburgh, J. S., A. S. Jones, J. Caraballo (2013). Cyberinfrastructure to support large scale collaborative water research, Presented at: Utah State University Spring Runoff Conference, Logan, UT, 10 April.
35. Jones, A. S., J. S. Horsburgh (2012), Adventures in monitoring: Maintaining sensor networks and managing the flood of data, presented at the Utah State University Spring Runoff Conference, Logan, UT, 3-4 April.
36. Whiting B., J. S. Horsburgh, A. S. Jones (2012), Improving estimates of suspended sediment concentration and flux in the Little Bear River watershed, presented at the Utah State University Spring Runoff Conference, Logan, UT, 3-4 April.
37. Jones, A. S., J. S. Horsburgh, D. K. Stevens, D. G. Tarboton, N. O. Mesner (2011), Sensors, networks, and tools: Communicating with sensors and managing the flood of data (Invited), presented at the at the Global Lake Ecological Observatory Network Freshwater Advanced Aquatic Sensor Workshop, Douglas Lake, MI, 11-13 September.

38. Jones, A. S. (2011), Demonstration of CUAHSI tools (Invited), presented at the Global Lake Ecological Observatory Network Freshwater Advanced Aquatic Sensor Workshop, Douglas Lake, MI, 11-13 September.
39. Horsburgh, J. S., A. S. Jones, D. K. Stevens, D. G. Tarboton, and N. O. Mesner (2011), Sensors, cyberinfrastructure, and water quality monitoring in the Little Bear River: Adventures in continuous monitoring, presented at the USGS/CUAHSI Workshop on Optical Sensors, Shepherdstown, WV, 8-10 June.
40. Horsburgh, J. S., A. S. Jones, D. K. Stevens, D. G. Tarboton, and N. O. Mesner (2010), A Study of high frequency water quality observations in the Little Bear River Utah, USA, Abstract B42C-07 presented at the 2010 Fall Meeting, AGU, San Francisco, Calif., 13-17 December.
41. Jones, A. S., N. O. Mesner, J. S. Horsburgh, R. J. Ryel, D. K. Stevens (2009), Impact of sampling frequency on annual load estimation, presented at the Utah State University Water Initiative Spring Runoff Conference. Logan, UT, 2-3 April.
42. Jones, A. S., N. O. Mesner, J. S. Horsburgh, R. J. Ryel, and D. K. Stevens, (2009), Impact of sampling frequency on annual load estimation, Presented at the USDA CSREES National Water Conference, St. Louis, MO, 8-12 February.
43. Horsburgh, J. S., D. K. Stevens, D. G. Tarboton, N. O. Mesner, A. S. Jones (2008), Sensors, cyberinfrastructure, and examination of hydrologic and hydrochemical response in the Little Bear River Observatory Test Bed, Eos Trans. AGU, 89(53), Fall Meet. Suppl., Abstract H43K-04.
44. Horsburgh, J. S., A. Spackman, D. K. Stevens, D. G. Tarboton, and N. O. Mesner (2008), Using GIS in creating an end-to-end system for publishing environmental observations data, Presented at the AWRA Spring Specialty Conference on GIS and Water Resources V, San Mateo, CA, 17-19 March.
45. Horsburgh, J. S., A. Spackman, D. K. Stevens, D. G. Tarboton, and N. O. Mesner (2008), An end-to-end system for publishing environmental observations data, Presented at the Utah State University Water Initiative Spring Runoff Conference, Logan, UT, 31 March-1 April.
46. Spackman A., D. K. Stevens, J. S. Horsburgh, D. G. Tarboton, N. O. Mesner (2008), Surrogate measures for providing high frequency estimates of total suspended solids and phosphorus concentrations, Presented at the Utah State University Water Initiative Spring Runoff Conference, Logan, UT, 31 March 31-1 April.
47. Spackman, A., D. K. Stevens, D. G. Tarboton, N. O. Mesner, and J. S. Horsburgh (2007), Surrogate measures for providing high frequency estimates of total suspended solids and phosphorus concentrations in the Little Bear River, Presented at the Bear River Symposium, Utah State University, Logan, UT, 5-7 September.
48. Spackman, A., R. Winters, G. Sullivan (2006), Phosphorus removal for Logan City wastewater treatment facility, presented at the Water Environment Association of Utah Fall Meeting, Salt Lake City, UT, November.
49. Stevens, D. K., J. S. Horsburgh, N. O. Mesner, T. Glover, A. Caplan, and A. Spackman (2006), Integrating historical and realtime monitoring data into an internet based watershed information system, Presented at the 2006 National Water Quality Monitoring Council National Monitoring Conference, San Jose, CA, 7-11 May.
50. Spackman, A. and B. T. Neilson (2003), Comparison of complex watershed models' predictive capabilities: USEPA's BASINS vs. WARMF, presented at the National Council for Undergraduate Research Posters on the Hill. Washington D.C., April.

Teaching, Training, and Workshops

Instructor, CUASHI Biennial Meeting, Using Python Packages and HydroShare to Advance Open Data Science and Analytics for Water. June 13, 2023.

Invited Guest Speaker, Utah State University Libraries Open Access Week, Open for Climate Justice: Open Learning Resources for Water Data Analysis. October 28, 2022.

Panelist, Utah State University Libraries Datapolooza Data Management Event. March 30, 2022.

Invited Guest Lecturer, Utah State University Climate Adaptation Science Studio: Managing and Sharing Scientific Data. 2018, 2019, 2020, 2021.

Teaching Assistant, Geographic Information Systems, Civil and Environmental Engineering, Utah State University, 2020.

Invited Guest Lecturer, Hydrologic Field Methods Course, Civil and Environmental Engineering, Utah State University: Sensor Data Quality Control. 2019.

Invited Guest Speaker, CUAHSI 2020 Cyberseminar: Publishing Data and Research. Jan 30, 2020.

Instructor, Water Data Services Workshop, Training presented by the Consortium of Universities for the Advancement of Hydrologic Science, Inc. at the Universities Council on Water Resources Annual Meeting, Snowbird, UT, June 11, 2019.

Organizer and Instructor, Sensor Data Quality Control and Post Processing, Training presented by iUTAH at the Utah Water Research Laboratory, Logan, UT, May 15, 2015.

Instructor and Developer, Tutorials for Data Publication for iUTAH. Training presented online by iUTAH, Logan, UT. 2015.

Teaching Fellow, CEE 3640 Water Engineering Undergraduate Course. Civil and Environmental Engineering, Utah State University, 2006.

Trainer, Better Assessment Science Incorporating Point and Nonpoint Sources (BASINS) Software Training, week long training course sponsored by USEPA, Presented at Utah State University, 2000 – 2002.

Reviewer for Journals and Organization

- Journal of Water Resources Planning and Management *(Associate Editor for Reproducibility)*
- Water Resources Research
- Ecological Informatics
- Environmental Modeling and Software
- Ecological Indicators
- Hydrological Sciences
- Environmental Monitoring and Assessment
- Applied Geochemistry
- Environmental Research
- Hydrology Research
- Journal of the American Water Resources Association
- New York Sea Grant
- Water Research
- Water