Machine Learning Protocols for Water and Environmental Modeling

October 2024

Modeling Support Office

Protocols for Water and Environmental Modeling using Machine Learning

Table of Contents

Executive Summary

The recent surge in popularity of generative artificial intelligence (GenAI) tools like ChatGPT has reignited global interest in AI, a technology with a well-established history spanning several decades. The California Department of Water Resources (DWR) has been at the forefront of this field, leveraging Artificial Neural Networks (ANNs), a core technique in machine learning which is a subfield of AI, for water and environmental modeling since the early 1990s. This document draws upon DWR's extensive experience to establish protocols for the development and implementation of machine learning models specifically tailored to water and environmental modeling efforts within California using quantitative and spatial water-related data including meteorological, hydrological, water quality, biological, and geospatial data. These protocols are not applicable to machine learning models designed for processing text, code, image, or video data. These protocols can help achieve standardization, quality assurance, interoperability, transparency, and compliance, among other benefits. Given the dynamic nature of the machine learning field, the protocols are anticipated to evolve alongside advancements in the field, thus making this document a living resource that will be updated to reflect emerging needs and technical progressions on an as-needed basis.

Water managers in California heavily rely on computer models to inform their decision-making processes. These models, encompassing empirical, mechanical, and statistical approaches, play a vital role in various aspects of water management. They facilitate real-time water operations and emergency responses, regulatory compliance, water rights management, climate change adaptation and mitigation planning, infrastructure investment and design, and ecosystem restoration, among others. Given the interconnected nature of water systems, ecosystems, and human activities, computer simulations are indispensable for comprehensively understanding and predicting their interactions. Consequently, the majority of significant water projects in California rely on analyses conducted through modeling. This widespread adoption underscores the importance of quality control of modeling practices for stakeholders and decision-makers. In light of this, the Bay-Delta Modeling Forum (now California Water and Environmental Modeling Forum (CWEMF)) developed a landmark document in 2000 outlining protocols for water and environmental modeling in California. CWEMF updated this document in 2021 to reflect advancements in the field, driven by new questions, technologies, and increased stakeholder engagement. The updated document provides current best practices and highlights emerging techniques like machine learning, which could significantly enhance modeling capabilities. However, it does not offer specific guidance on developing and applying machine learning models.

Machine learning techniques have made tremendous progress in recent years, leading to a surge in their application across California, including the water sector. Recognizing the vast potential of machine learning, especially GenAI, for the state's well-being and economy, the Governor issued Executive Order N-12-23 in 2023. This initiative seeks to develop a comprehensive and responsible approach to AI adoption in state government, acknowledging its significant impact and implications. The Executive Order has yielded several key outputs, including a report on GenAI's benefits and risks,

guidelines for procurement and deployment, and a GenAI toolkit to facilitate responsible adoption by state entities.

GenAI and conventional machine learning differ fundamentally in their objectives and approaches. Conventional machine learning focuses on making predictions or classifications based on existing data, aiming to identify patterns and relationships within the data. In contrast, GenAI seeks to create entirely new, original content, such as images, text, or music, that resembles existing data but is not simply a copy or extrapolation. GenAI uses much more complex algorithms to learn the underlying distribution of the data and generate novel samples that align with that distribution.

The current work seeks to supplement CWEMF's 2021 update by presenting the protocols for developing and applying machine learning models rather than traditional computer models in water and environmental modeling. These protocols are based on prior experience with conventional machine learning models only, as GenAI models have not yet been widely adopted in this field in California, to our knowledge. For GenAI model development, procurement, or application, please refer to the guidelines established under Executive Order N-12-23.

This document provides an overview of protocols for the machine learning modeling process. The process can be broadly divided into four stages: problem definition, data preparation, model development, and model deployment and communication.

Problem Definition involves identifying and framing the specific challenges or questions that need to be addressed through modeling. For instance, this could entail understanding water resource allocation, assessing the impact of environmental changes on ecosystems, or predicting future trends in water availability. Clear and precise articulation of the problem ensures that subsequent modeling efforts are focused and aligned with the objectives of stakeholders and policymakers.

Once the problem is defined, the next phase is data preparation. This involves gathering, cleaning, and organizing relevant datasets that will be used to train and validate the machine learning models. This could include diverse sources such as hydroclimatic records, satellite imagery, land use maps, and biological surveys. Ensuring the quality, completeness, and compatibility of data is essential for the accuracy and reliability of subsequent model development efforts.

With the data prepared, the focus shifts to model development. This phase entails selecting appropriate machine learning algorithms, designing model architectures, and training, testing, and evaluating models using the prepared datasets. This could involve developing predictive models for drought monitoring, hydrological forecasting, species distribution modeling, or water quality prediction. Iterative refinement and validation of models against historical data and real-world observations are key aspects of this phase to ensure their effectiveness and robustness.

The final phase involves deploying the developed models for practical use and communicating their findings and insights to relevant stakeholders. This could mean integrating the models into decision support systems used by water management agencies, stakeholders, or policymakers. Effective communication of model outputs, including uncertainty and limitations, is crucial for informing decision-making processes and facilitating public engagement. This can be accomplished through various means such as stakeholder gatherings, detailed documentation, practical training sessions,

user community forums, and similar avenues. Additionally, ongoing monitoring and evaluation of model performance post-deployment enable adaptive management strategies in response to changing environmental conditions and stakeholder needs.

Effective machine learning in water and environmental modeling requires rigorous protocols at each step. The first process, problem definition, requires a clear articulation of the problem, establishing specific objectives, and understanding the context and constraints. In the data preparation phase, standardized methodologies for data collection, quality assurance, and quality control are crucial. This phase also involves selecting representative data, improving data accessibility, and thoroughly documenting data sources and data processing methods. During model development, starting with simple models and progressively incorporating hybrid and ensemble methods is recommended. Integrating domain knowledge enhances model accuracy, while hyperparameter optimization and cross-validation are essential to avoid overfitting and ensure robust model performance. For model deployment, providing comprehensive documentation is vital to facilitate understanding and reproducibility. Training sessions for end-users, community engagement, and making models open source are best practices to promote transparency, collaboration, and further innovation in the field.

These protocols aim to support both machine learning practitioners, including technical experts involved in the development and implementation of models, and the wider community of stakeholders interested in the accuracy and reliability of such models.

1. Introduction

1.1 Motivation and Scope

The water sector in California boasts a rich tradition of crafting and utilizing mathematical models to guide decision-making. Relevant modeling protocols have been developed and implemented to guide the development and application of these models (BDMF, 2000; CWEMF, 2021). Unlike traditional mathematical models that are grounded in physical laws or empirical evidence, machine learning (ML) models are largely data-driven. As a result, the protocols governing these two types of models may not always be applicable or interchangeable.

This document aims to supplement existing modeling best practices (e.g., BDMF, 2000; CWEMF, 2021; Black et al., 2014; Wu et al., 2014) by providing protocols for the development and implementation of ML models specifically. These protocols stem from practical insights gained from using conventional ML techniques in water and environmental modeling. Although Generative Artificial Intelligence (GenAI) represents the latest ML advancement, its application in the water sector remains limited. As such, the guidelines provided in this document are not specifically tailored for GenAI models.

This document is designed to evolve over time and will be updated as required. All opinions expressed in this document belong solely to the authors and do not reflect those of the Department of Water Resources (DWR).

1.2 Water and Environmental Modeling

Water and environmental systems comprise surface water bodies like rivers and lakes, along with groundwater aquifers, wetlands, and other components. Currently, these systems confront pressing challenges. These challenges include, but are not limited to, intensifying weather and hydrological extremes, habitat degradation, climate change, and unsustainable use of natural resources. California, in particular, is highly susceptible to these challenges due to its intricate and aging water infrastructure. Addressing these challenges requires a comprehensive understanding of complex environmental processes and their interactions, which is where water and environmental modeling plays a pivotal role.

Specifically, water and environmental modeling offers a multifaceted approach to comprehending complex systems by integrating diverse data sources, representing complex processes, and facilitating prediction and scenario analysis. By assimilating field measurements, remote sensing observations, and laboratory experiments, these models provide a holistic view of our water and environmental systems. They can simulate physical, chemical, and biological processes, revealing the underlying mechanisms driving system behavior and identifying feedback loops and emergent properties. Through quantitative analysis and visualization, these models enable model users to interpret spatial and temporal patterns, assess model performance, communicate findings effectively, and to identify and fill data gaps.

Water and environmental modeling also serves as a valuable tool for informing decision-making across various sectors by providing insights into the current and projected conditions of the systems and their responses to different scenarios. These

models enable decision-makers to assess the potential impacts of alternative scenarios, and to identify and prioritize effective solutions to environmental challenges, optimize resource allocation, and mitigate risks. Moreover, these models facilitate communication and collaboration among stakeholders by providing a common platform for analyzing data, evaluating options, and fostering productive dialogue, ultimately contributing to more informed decision-making.

California has a rich history of developing and employing models to plan and manage the state's intricate water and environmental systems. For example, as far back as 1930, the California Division of Water Resources (now known as the California Department of Water Resources (DWR)) issued Bulletin 120, which provides forecasts for April to July runoff volume based on various variables such as precipitation, snowfall, and runoff indices (Hannaford, 1956; Hart and Gehrke, 1990). In 1931, DWR devised the Bulletin 27 Salinity Intrusion Model to estimate the required flows for managing seawater intrusion in the Delta (DWR, 1931). In 1978, DWR introduced DAYFLOW, a straightforward mass balance model intended for determining the daily historical hydrology along the boundaries of the Delta. Subsequently, in 1986, DWR developed DWRSIM, a model for water project operations, to simulate the joint operation of the State Water Project (SWP) and Central Valley Project (CVP) (Barnes and Chung, 1986). Later on, DWR and the U.S. Bureau of Reclamation collaborated to develop CalSim as a replacement for DWRSIM (Draper et al., 2004). In the early 1990s, DWR implemented the hydrodynamics and water quality model DSM2, designed to simulate variables such as flow, stage, salinity, water temperature, and other water quality variables in the Delta (DWR, 1991).

Generally speaking, these models include mechanistic models (based on physical principles), empirical models (based on empirical observations), or hybrid mechanistic and empirical models. They serve several purposes in the state’s water community including: 1) supporting scientific research (such as using individual-based models to replicate the movement and survival of juvenile salmon in the Delta); 2) facilitating real-time operations (such as flood forecasting to inform reservoir releases and Delta operations); 3) aiding in planning and decision-making (such as evaluating the potential impacts of structural, operational, regulatory, and climate changes on the State Water Project's delivery capacity); and 4) assisting in dispute resolution (for example, settling disputes over water rights or allocating water for different beneficial uses).

As models become more central to managing California's water systems, scrutiny and controversy surrounding their use and development are also rising. It’s widely understood among the state's water community that the adoption of standardized modeling protocols is imperative. These protocols can help enhance the quality of models and modeling studies, while also bolstering the trust and confidence of decision-makers and stakeholders who rely on model outcomes to inform their decisions. In 1997, the Bay-Delta Modeling Forum (BDMF), the predecessor organization of the California Water and Environmental Modeling Forum (CWEMF), established an Ad hoc committee on modeling protocols. The committee’s purpose was to develop principles and guidelines (protocols) aimed at offering direction to water stakeholders, decision-makers, and their technical personnel in the development and utilization of models to address water and environmental challenges in California. In 2000, the committee

documented their efforts in a report titled “Modeling Protocols for Water and Environmental Modeling” (BDMF, 2000). In 2021, CWEMF updated this report to account for changes in the water and environmental modeling practices over the past two decades since the initial release (CWEMF, 2021).

This document extends the achievements of the two earlier protocols, with a particular emphasis on machine learning modeling, which represents a relatively recent addition to the broader modeling practices within California's water community.

1.3 Artificial Intelligence and Machine Learning

Artificial intelligence (AI) is a specific field within the broader discipline of computer science (CS) (Figure 1). It is an umbrella term encompassing the development of intelligent systems capable of mimicking human cognitive functions like learning, problem-solving, and decision-making (Barr et al., 1981). The history of AI dates back to the mid-20th century, with early pioneers like Alan Turing laying the foundation for the field with his seminal work on the Turing test (Saygin et al., 2000), a theoretical test of a machine's ability to exhibit intelligent behavior equivalent to, or indistinguishable from, that of a human. AI has undergone a remarkable evolution since its inception, showcasing diverse applications across various fields. For example, in engineering, AI optimizes designs, streamlines production processes, and enhances product performance through techniques like predictive maintenance and generative design. As another example, in scientific research, AI facilitates data analysis, accelerates discoveries, and models complex systems across disciplines such as physics, chemistry, and biology.

Machine learning (ML) is a specific subfield of AI that focuses on developing algorithms and techniques that allow computers to learn from data without explicit programming (Carbonell et al., 1983) (Figure 1). This learning process typically involves exposing the algorithm to a large dataset, enabling it to identify patterns and relationships within the data. Over time, the algorithm improves its performance on specific tasks, such as image recognition, speech recognition, or predicting future outcomes. The history of ML is deeply intertwined with the development of AI, with key advancements in computing power and statistical methods facilitating the creation of powerful ML algorithms.

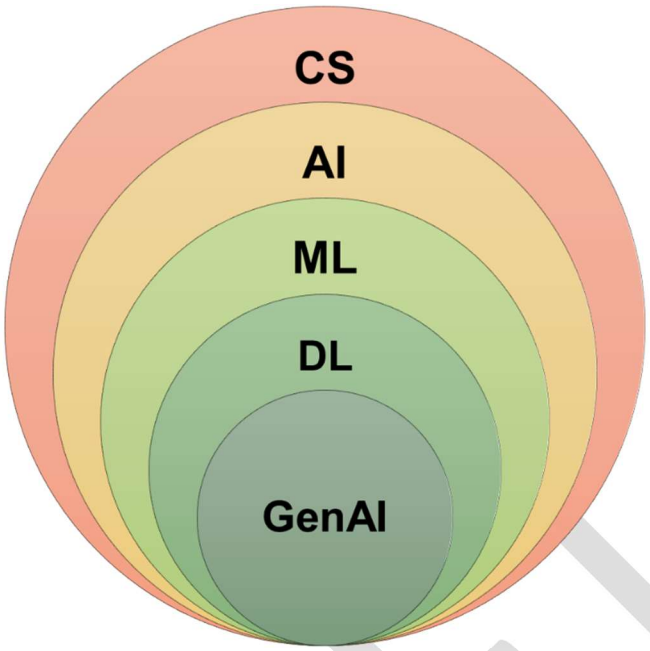

Figure 1. Schematic illustrating the relationship among computer science (CS), artificial intelligence (AI), machine learning (ML), deep learning (DL), and generative AI (GenAI).

While AI encompasses a broader range of techniques and goals, including symbolic reasoning and expert systems, ML serves as a crucial engine for enabling machines to learn and improve their performance on specific tasks (Jordan and Mitchell, 2015). It's important to understand that not all AI applications rely on ML, but the vast majority of modern AI advancements leverage the power of machine learning algorithms to achieve their goals. In essence, AI defines the "what" – the broad objective of creating intelligent machines – while ML provides the "how" – the specific techniques and algorithms that enable machines to learn and improve.

Deep learning (DL) is a subset of ML (LeCun et al., 2015; Razavi, 2021) (Figure 1). While ML encompasses a broad spectrum of algorithms and techniques aimed at enabling computers to learn from data and make predictions or decisions, DL specifically focuses on using artificial neural networks (ANNs) (Hassoun, 1995) with multiple layers to model and understand complex patterns. Unlike traditional ML methods that rely on human experts to define input features, DL algorithms autonomously learn hierarchical representations of data directly from raw input. This ability to automatically extract intricate features from large datasets has led to breakthroughs in various domains such as image recognition, natural language processing, and speech recognition. DL models, such as convolutional neural networks (CNNs) (LeCun et al., 1989) for images and recurrent neural networks (RNNs) (Elman, 1990) for sequential data, have demonstrated unparalleled performance in tasks ranging from image classification to language translation, revolutionizing industries and driving advancements in AI research.

Generative AI (GenAI) (Epstein and Hertzmann, 2023) is a subfield of DL (Figure 1) focused on creating entirely new data, rather than just making predictions based on existing information. DL plays a pivotal role in generative AI by providing powerful frameworks such as generative adversarial networks (GANs) (Goodfellow et al., 2014)

and variational autoencoders (VAEs). These models are trained on massive amounts of data, allowing them to learn the underlying structure and relationships within that data. Once trained, GenAI can then use this knowledge to produce entirely new outputs that closely resemble, but aren't copies of, the data it studied. This opens doors for applications like generating text, images, speeches, and videos. Essentially, GenAI utilizes the power of DL to move beyond analysis and toward content creation.

The rapid advancement of AI has unlocked numerous opportunities while also prompting concerns and ethical considerations. In response, various policies and guidelines have emerged worldwide to both foster and oversee the development and deployment of AI technologies. For example, in 2020, the U.S. Federal Government issued Executive Order (EO) 13960, aimed at promoting AI adoption across federal agencies. Subsequently, in 2023, Executive Order 14110 followed, focusing on the Safe, Secure, and Trustworthy Development and Use of Artificial Intelligence, extending its scope beyond the federal government to encompass all industries. Additionally, in 2022, UNESCO issued a global recommendation on AI ethics. In early 2024, the European Union (EU) introduced "The EU AI Act" to regulate AI usage within the EU.

In California, due to the broad potential of GenAI for Californians and the state's economy, the Governor issued an executive order (EO N-12-23) in 2023 to examine the development, utilization, and associated risks of AI technology across the state. This initiative aims to establish a careful and responsible approach to assessing and implementing AI within the state government, recognizing its significant impact and implications. The first deliverable of this EO is a report on the benefits and risks of GenAI for the State (GovOps,2023). The report provides an initial examination of the potential benefits and risks of GenAI, focusing on enhancing access to vital goods and services while also addressing concerns such as security vulnerabilities and potential impacts on public health, safety, and the economy of the state. The latest deliverable of the EO is a report summarizing guidelines on procurement, usage, and training for integrating GenAI into the California state government (GovOps, 2024). These guidelines provide the best practices and criteria to safely and effectively utilize this innovative technology. The final procurement and training policy stemming from this EO is expected to be released in 2025.

The current document aligns with EO N-12-23 but specifically concentrates on the conventional ML models including DL models, with a scope confined to water and environmental modeling only.

1.4 Life Cycle of a Machine Learning Model

The life cycle of a typical machine learning model encompasses several key stages including problem definition, data preparation, model development, mode deployment and communication (Figure 2). Data preparation can be divided into data collection and data pre-processing. Model development can further be divided into model selection, model training and testing, and model evaluation. During the problem definition phase, the specific problem or task that the model aims to address is clearly defined, along with the objectives and desired outcomes. Following problem definition, the process moves to data collection, where relevant datasets are gathered from various sources, ensuring they are comprehensive and representative of the problem

domain. Subsequently, data pre-processing is carried out to clean, transform, and prepare the data for analysis, addressing issues such as missing values, outliers, and noise.

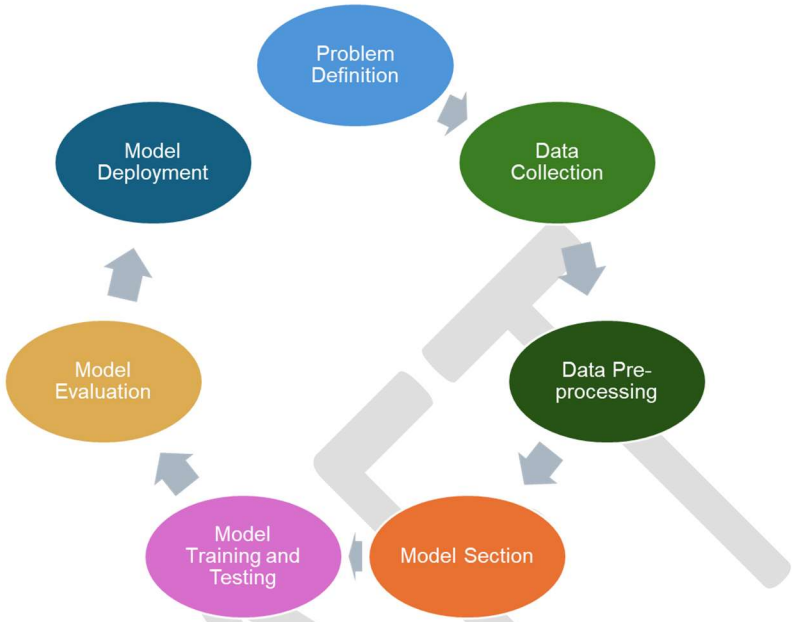

Figure 2. The life cycle of a typical machine learning model.

Once the data is pre-processed, the next step is model selection, where the most appropriate machine learning algorithms are chosen based on the problem type, data characteristics, and desired outcomes. With data pre-processed and models selected, the models are trained and tested, where they learn patterns and relationships between features (model input) and the target variable (model output). The model demonstrating the best performance in testing may be chosen for implementation, or alternatively, multiple models showing similar performance levels could be considered.

Following model training and testing, thorough model evaluation is conducted to assess its performance, often employing statistical metrics and/or visual inspection. Based on the evaluation results, adjustments may be made to the model or its parameters to optimize performance further. Finally, upon satisfactory evaluation outcomes, the model is deployed into production, making it available for real-world use, where it can provide insights, make predictions, or automate decision-making processes, thus completing the life cycle of a machine learning model. Throughout this cycle, iteration and continuous improvement are key, as models may require updating or retraining to maintain relevance and effectiveness over time. Further elaboration on each of these steps will be provided in later Sections.

1.5 Roles of Machine Learning Models in Water and Environmental Modeling

The complexities of water and environmental systems along with a changing climate pose challenges for traditional modeling approaches. ML models have emerged

as powerful tools, offering unique capabilities that complement and enhance existing methods (e.g., Huang et al., 2021; Sit et al., 2020; Tripathy and Mishra, 2023; Wai et al., 2022). A number of roles that ML models play in water and environmental modeling are described as follows.

1) Enhanced simulation and prediction

ML models excel at identifying patterns and relationships within vast datasets, allowing them to make accurate simulations and predictions about future environmental conditions. This includes tasks like:

a) Hydrological forecasting: Forecasting precipitation (e.g., Ghazvinian et al., 2021), reservoir inflows (e.g., Zarei et al., 2021), water level (e.g., Rajaee et al., 2018; Wee et al., 2021), flood events and flood risks (e.g., Jones et al., 2023; Kumar et al., 2023;Mosavi et al., 2018; Nearing et al., 2024; Ng et al., 2023), flood inundation area (e.g., Karim et al., 2023), and drought events (e.g., Prodhan et al., 2022) with improved accuracy and lead time.

b) Water quality modeling: Forecasting the spatial and temporal variations of water quality variables (Chen et al., 2020; Zhi et al., 2024) including salinity (e.g., He et al., 2020), water temperature (e.g., Feigl et al., 2021), ion concentrations (e.g., Namadi et al., 2022, 2023), sediment (e.g., Kim et al., 2022), Algal bloom (Nguyen et al., 2023; Park et al., 2022)among others; identifying potential contamination sources; and assessing the effectiveness of mitigation strategies.

c) Climate change analysis: Understanding the complex dynamics of climate systems and predicting future trends in temperature and precipitation patterns (e.g., Chen et al., 2023; de Burgh-Day and Leeuwenburg, 2023; Kaack et al., 2022; Rolnick et al., 2022; Slater et al., 2023).

2) Uncovering hidden insights

ML algorithms can analyze vast amounts of water and environmental data, including satellite imagery, sensor readings, and historical records. This ability to process complex data allows them to uncover hidden patterns and relationships that might be missed or even misrepresented by traditional methods. These insights can be crucial for:

a) Identifying emerging environmental threats: Early detection of drought (e.g., snow drought) events and changes in water quality (e.g., Cui et al., 2023; Lei et al., 2023; Zhu et al., 2023).

b) Developing targeted conservation strategies: Identifying crucial habitats (e.g., Wicaksono et al., 2019), predicting species (e.g., fish, birds, etc.) distribution (e.g., Pichler et al., 2020), and informing conservation efforts (e.g., Mosebo Fernandes et al., 2020; Tuia et al., 2022).

c) Optimizing resource management: Understanding the factors influencing water availability and demand, leading to more sustainable water use practices (e.g., Anjum et al., 2023).

3) Efficiency and scalability

ML models offer significant advantages in terms of efficiency and scalability. They can process large datasets rapidly, enabling model users to conduct simulations and explore various scenarios in a shorter timeframe. This allows for:

a) Real-time monitoring and decision-making: Providing near-real-time insights into water and environmental conditions, facilitating proactive responses to emerging issues (e.g., Pollard et al., 2018; Sun and Scanlon, 2019).

b) Simulating large-scale water and environmental processes: Modeling complex interactions between different components of the environment, such as the impact of climate change on water resources or the spatial and temporal variations of critical water quality variables in an ecosystem (e.g., Tahmasebi et al., 2020, Zheng et al., 2023).

4) Integration with traditional models

ML models are not intended to replace traditional models used in water and environmental modeling. Instead, they offer a complementary approach. ML models can be integrated with existing models to:

a) Improve the accuracy, speed, and efficiency of simulations (e.g., He et al., 2020; Doh et al., 2023; Qi et al., 2023). For example, Qi et al. (2023) developed a machine learning-based dashboard to mimic the DSM2 model for salinity simulation at critical spots in the Sacramento-San Joaquin Delta. This dashboard operates notably quicker than DSM2, and it's capable of running on a mobile phone with internet connectivity. The outcomes can be swiftly generated and displayed within seconds. Additional specifics can be found in the Appendices.

b) Address the limitations of traditional models, such as their reliance on complete and accurate data (e.g., Shen, 2018; Zhong et al., 2021; Muñoz-Carpena et al., 2023).

c) Provide a deeper understanding of the complex interactions within environmental systems (e.g., Reichstein et al., 2019).

In brief, ML models play a transformative role in water and environmental modeling. Their ability to learn from data, make skillful predictions, and uncover hidden insights empowers scientists, engineers, managers, and policymakers to address the complex challenges facing our water and environmental systems.

1.6 Reasons for Developing Machine Learning Protocols

As machine learning garners increasing attention and finds more applications within the California water community of the State, the development of machine learning protocols becomes imperative for several distinct reasons:

Standardization: Protocols provide standardized guidelines and procedures for data collection, preprocessing, model development, evaluation, and deployment. Standardization ensures consistency and reproducibility across different projects and research efforts, facilitating comparison and collaboration among model developers, users, and stakeholders.

Quality Assurance: Protocols help ensure the quality and reliability of ML models by specifying rigorous criteria for data quality, model performance, and validation methods. By adhering to established protocols, model developers can mitigate common pitfalls such as overfitting, data biases, and model uncertainties, resulting in more robust and trustworthy modeling outcomes.

Interoperability: Protocols promote interoperability by defining common data formats, interfaces, and protocols for model integration and exchange. Interoperable models enable seamless integration with existing modeling frameworks, databases, and decision support systems, enhancing the usability and scalability of ML-based solutions for water and environmental modeling.

Transparency and Reproducibility: Protocols promote transparency and reproducibility by documenting the entire modeling process, including data sources, preprocessing steps, model architectures, hyperparameters, and evaluation metrics. Transparent and reproducible models allow stakeholders to understand and validate modeling decisions, ensuring accountability and trustworthiness in decision-making processes.

Compliance and Regulation: Protocols help ensure compliance with regulatory requirements and ethical standards governing data privacy, environmental protection, and public safety. Adhering to established protocols minimizes legal risks and regulatory hurdles associated with ML modeling in sensitive environmental regions, such as the Sacramento-San Joaquin Delta.

Capacity Building: Protocols facilitate capacity building by providing training materials, best practices, and guidelines for model developers, users, and stakeholders involved in ML-based water and environmental modeling. Training programs and workshops on protocol implementation empower stakeholders with the knowledge and skills needed to effectively apply ML techniques in real-world water and environmental applications.

Continual Improvement: Protocols support continual improvement by fostering feedback mechanisms, community engagement, and peer review processes. Iterative refinement of protocols based on feedback from stakeholders and advances in ML research ensures that protocols remain relevant, adaptable, and responsive to evolving challenges and opportunities in water and environmental modeling.

2. Pre-modeling Considerations

2.1 Problem Definition

Problem definition is the cornerstone of the successful application of ML in all fields including water and environmental modeling. It acts as a compass, guiding the entire process and ensuring efficient resource allocation, optimal model selection, and ultimately, the development of appropriate tools for addressing critical water and environmental challenges. Problem defining aims to address the question of "who, what, when, where, and why": clearly articulate who the problem affects, what to predict or analyze, the timeframe and location of interest, and the rationale behind tackling this problem. The typical process to craft a clear and well-defined problem statement involves:

1) Understand the Context

a) Identify the problem: Begin by clearly identifying the specific water-related or environmental issues you aim to address. Is it water data anomaly detection, water quality deterioration, flood, drought, and water supply forecasting, or others?

b) Stakeholder needs: Consider the needs and interests of stakeholders who might benefit from the model's insights, such as policymakers, water managers, or the public.

2) Define the Scope

a) Specificity: Avoid broad and ambiguous statements like "improve water quality understanding." Instead, focus on specific and measurable objectives.

b) Temporal and spatial considerations: Specify the desired prediction timeframe (e.g., hourly, daily, seasonal, annual) and the spatial scope (e.g., a specific river, channel, an entire watershed or groundwater basin).

3) Define the Desirable Outcome

a) Measurable goals: State what you want the model to achieve in a quantifiable manner. This could involve predicting specific water quality parameters, identifying areas of water quality degradation, or forecasting the probability of specific events.

b) Performance metrics: Define the metrics you will use to evaluate the model's success in achieving the desired outcome. This could involve accuracy, precision, or recall, depending on the specific task.

4) Refine and Iterate

a) Seek feedback: Share the draft problem statement with colleagues or experts in the field and stakeholders to gather feedback and ensure clarity and feasibility.

b) Iterative process: Be prepared to make adjustments based on feedback and emerging information throughout the project.

Adhering to these steps can lay the groundwork for successful ML applications in water and environmental modeling.

2.2 Data Collection

Just like in other areas, developing skillful simulation and forecasting models, particularly data-driven ML models in water and environmental modeling relies heavily on gathering best available high-quality data. This section dives into why data collection is so important and offers tips on the best ways to get and prepare datasets for developing ML models in this field.

2.2.1 Importance of Data Collection

Water and environmental systems are characterized by complex interactions between various climatic, hydrological, and biophysical factors. Understanding and simulating these dynamics require comprehensive datasets that capture the spatiotemporal variability of them. Data collection serves several crucial purposes:

1) Informing Model Development: High-quality data provide the foundation for building ML models that accurately represent the underlying dynamics of water and environmental systems.

2) Validation and Evaluation: Collected data serve as a benchmark for assessing the performance of ML models, enabling model developers and users to validate model outcomes against the corresponding observations.

3) Informing Decision-Making: Reliable data empower model users to make informed decisions regarding water resource management, environmental conservation, and climate adaptation strategies, among others.

2.2.2 Types of Data to Collect

In the water and environmental field, a wide range of data are utilized to comprehend, simulate, and predict water and environmental processes and variables. In general, these data encompass meteorological, hydrological, water quality, biological, and geospatial data. Each category of data offers distinct perspectives into various facets of water and environmental phenomena and relationships. The following table provides concise explanations of these data categories alongside several illustrative examples.

Table 1. Types of data typically utilized in water and environmental modeling.

Data Type	Description	Examples
Meteorological Data	Data related to atmospheric conditions and processes	Air temperature, precipitation, humidity, wind speed, wind direction, solar radiation
Hydrological Data	Data related to the movement and distribution of water within the hydrological cycle	Streamflow, water stage, evapotranspiration, groundwater level, snow depth, infiltration
Water Quality Data	Data related to chemical, physical, and biological properties of water	Water temperature, pH, dissolved oxygen, electrical conductivity, total dissolved solids, turbidity, nutrient levels, chlorophyll-a
Biological Data	Data related to living organisms within ecosystems	Species diversity, population, abundance and distribution, functional traits
Geospatial Data	Data that are spatially referenced	Digital elevation models, remote sensing imagery on land cover, land use, vegetation indices, and water features

These data can be obtained through both measured and simulated sources. Firstly, data can be measured at in-situ stations or via remote-sensing techniques. In-

situ stations are physical locations where instruments are placed to directly measure water and environmental parameters such as water stage, water temperature, dissolved oxygen levels, turbidity, and flow rates. Data collected from in-situ stations provide direct observations of real-world conditions. Remote sensing involves the use of satellites, aircraft, drones, or other platforms to capture data about the Earth's surface and atmosphere without direct physical contact. Remote sensing techniques can provide spatially extensive and temporally frequent measurements of various water and environmental variables, including land cover, water quality, vegetation indices, and atmospheric conditions. Sensors onboard these platforms collect data using various wavelengths of electromagnetic radiation, such as visible, infrared, and microwave, allowing for the monitoring of diverse water and environmental parameters. Secondly, data can also be generated via existing process-based numerical models that simulate the physical, chemical, and biological processes governing water and environmental systems. For instance, hydrological models simulate the movement of water through the landscape, while water quality models simulate the behavior of various constituents and parameters that determine the quality of water in water bodies. These models rely on mathematical equations representing fundamental principles of physics, chemistry, and biology to simulate the behavior of environmental systems over time and space. Simulated data from these models provide insights into the dynamics and interactions within environmental systems under different scenarios and conditions.

By leveraging both measured and simulated data, machine learning models in water and environmental modeling can improve predictive accuracy, enhance our understanding of complex processes, and support informed decision-making for sustainable water resources management and environmental protection. Measured data are essential for training and validating ML models, ensuring that they accurately capture the relationships between input variables and the outcomes, while simulated data offer insights into the underlying processes governing water and environmental systems. Simulated data can complement measured data by providing additional samples, especially in regions or time periods with sparse observational data. Simulated data also allow for the exploration of various scenarios and what-if analyses, enabling decision-makers to evaluate the potential impacts of different management and operational practices.

3. Model Development

ML model development in water and environmental modeling encompasses three key aspects. Firstly, data pre-processing plays a vital role, involving the cleaning and integration of diverse datasets from various sources. This step ensures the quality and consistency of input data. Furthermore, model selection, the process of identifying the most suitable algorithm or architecture for a given task, is pivotal in ML model development as it directly impacts the performance and effectiveness of the model. Additionally, model training and testing are essential for fine-tuning model (hyper)parameters and assessing their generalization ability to unseen data. This section provides an overview of these three components. Additionally, this section outlines the cloud computing utilities and ML development frameworks commonly

utilized in ML model development across various domains, including water and environmental modeling.

3.1 Data pre-processing

In ML model development, various data processing methods are utilized to prepare raw data for analysis. Some typical data processing methods include:

1) Data Cleaning: This involves identifying and handling missing values, outliers, and erroneous data points. Techniques such as imputation (e.g., filling missing values with mean or median) (e.g., Osman et al., 2018), anomaly detection (e.g., Dogo et al., 2019), and data validation (e.g., Lamrini et al., 2011) are commonly used to ensure the quality and integrity of the dataset.

2) Normalization and Standardization: Normalizing or standardizing features helps in scaling the data to a similar range, which prevents certain features from dominating others during model training. For example, when considering inputs such as tide stage level (measured in feet) and flow rate (measured in cubic feet per second) to model salinity level (measured in microSiemens per centimeter) in the Sacramento-San Joaquin Delta (Delta), the magnitudes of flow rate and salinity may significantly outweigh that of the tide stage. Without scaling these data to a comparable range, machine learning models and other data-driven approaches are prone to favoring one input over the others. Techniques like Min-Max scaling or Z-score normalization (e.g., Patro and Sahu, 2015) are commonly used for this purpose.

3) Feature Selection: Feature selection involves identifying the most relevant predictors to improve model efficiency and interpretability while reducing overfitting (Guyon and Elisseeff, 2003). For instance, in hydrological modeling, features such as precipitation, temperature, soil moisture content, and land cover types are often selected based on their significance in predicting streamflow or groundwater levels. Techniques like recursive feature elimination (e.g., Chen and Jeong, 2007) or tree-based feature importance analysis (e.g., Zhou et al., 2021) are commonly applied to identify the most informative features for model training, ensuring that the resulting models accurately capture the underlying dynamics of water and environmental processes while minimizing computational complexity.

4) Dimensionality Reduction: High-dimensional datasets are common in water and environmental modeling, which can lead to computational challenges and overfitting. Dimensionality reduction techniques such as Principal Component Analysis (PCA) or feature selection methods help reduce the number of features while preserving important information, thus improving model efficiency and generalization (e.g., Fodor, 2002).

5) Spatial Interpolation: Spatial interpolation methods are used to estimate values at unmeasured locations within a study area based on observed data from neighboring locations. Techniques such as kriging (e.g., Erdogan Erten et al., 2022; Li et al., 2011), inverse distance weighting (e.g., Pereira et al., 2022), or spline interpolation (e.g., Tait and Woods, 2007) are employed to interpolate environmental variables, such as precipitation or soil moisture, which are often spatially heterogeneous.

6) Temporal Aggregation: Temporal aggregation involves aggregating high-frequency time-series data into coarser time intervals (e.g., hourly to daily or monthly) to

reduce noise and capture long-term trends. Aggregation methods such as averaging, summing, or max/min aggregation are used to create aggregated features that are more suitable for modeling.

7) Spatial Downsampling or Resampling: In cases where high-resolution spatial data are available, downsampling or resampling techniques may be applied to reduce computational complexity and memory requirements (e.g., Niroumand-Jadidi et al., 2022). This involves reducing the spatial resolution of raster datasets while preserving essential spatial patterns and characteristics.

8) Temporal Smoothing: Temporal smoothing techniques such as moving averages or exponential smoothing are applied to time-series data to remove noise and highlight underlying trends, making the data more suitable for modeling purposes.

3.2 Model Selection

Selecting the appropriate ML model architecture for specific water and environmental problems is crucial to achieving accurate and reliable results. This section discusses the types of widely used ML architectures, why this selection process is essential, and some common considerations for selecting ML architectures.

3.2.1 Why Selecting the Right Machine Learning Architectures is Essential

Selecting the appropriate ML architectures ensures accurate and reliable predictions. It also helps optimize computational resources in terms of reducing complexities and training time, particularly crucial in resource-constrained environments. Moreover, the chosen architectures impact interpretability and transparency, fostering user trust and facilitating collaboration. Additionally, the adaptability of ML models to changing environmental conditions enables proactive decision-making and resilience to emerging challenges, essential for long-term planning efforts. These aspects are further elaborated as follows.

1) Accuracy and Reliability: Choosing the appropriate ML architecture ensures accurate and reliable predictions, aiding decision-makers in effectively managing water resources and addressing environmental challenges. Models that accurately capture the underlying patterns in the data can provide valuable insights for informed decision-making.

2) Resource Optimization: Selecting the right ML architecture helps optimize computational resources and time, reducing unnecessary complexities and computational burdens. This allows for efficient model training and deployment, particularly in resource-constrained environments or real-time applications.

3) Interpretability and Transparency: The choice of ML architecture influences the interpretability and transparency of model simulations and predictions, which are essential for gaining users' trust and understanding the reasoning behind model outputs. Transparent models facilitate communication and collaboration among stakeholders, leading to more effective decision-making.

4) Adaptability to Changing Conditions: ML architectures that can adapt to changing environmental conditions and evolving datasets are crucial for addressing long-term water and environmental challenges. Models that can robustly generalize to

unseen conditions and incorporate new data seamlessly enable proactive decision-making and adaptation to emerging threats or opportunities.

3.2.2 Commonly Used Machine Learning Architectures

Based on learning paradigms, machine learning architectures can be broadly classified into three categories: supervised learning, unsupervised learning, and reinforcement learning. Simply speaking, supervised learning uses labeled data, unsupervised learning uses unlabeled data, and reinforcement learning learns by interacting with an environment.

The history of ML architectures is marked by a progression of key milestones across various paradigms. In the realm of supervised learning, the introduction of the perceptron algorithm (Rosenblatt, 1958) in the mid-20th century laid the groundwork. The algorithm was inspired by the structure and function of biological neurons in the brain. Concurrently, unsupervised learning approaches began emerging (e.g., clustering algorithms in the 1950s; Lloyd, 1982; Xu and Wunsch, 2005). The inception of reinforcement learning can be traced back to the 1950s as well, with the development of early techniques like dynamic programming (Bellman, 1966). The theoretical underpinnings of dynamic programming were subsequently leveraged in the development of reinforcement learning algorithms and applications. However, it wasn't until the 1980s that reinforcement learning gained prominence, particularly with the introduction of temporal difference learning (Sutton, 1988) and Q-learning (Watkins and Dayan,1992). The advent of deep learning in the 21st century revolutionized ML, leveraging neural networks with multiple layers to tackle increasingly complex tasks in supervised, unsupervised, and reinforcement learning domains. Today, the landscape of ML architectures continues to evolve rapidly, with ongoing research pushing the boundaries of what is possible in terms of model performance, scalability, and interpretability.

The following table showcases widely used machine learning architectures in those three categories, which have been investigated and employed within the realm of water and environmental modeling. Although this compilation is thorough, it does not encompass every existing architecture and is subject to updates in subsequent revisions of this document.

Table 2. Commonly used machine learning architectures.

Learning Paradigm	Example Architecture	Description
Supervised Learning	Support Vector Machine (SVM)	Finds the optimal hyperplane to separate data points for classification tasks.
	Decision Tree	Creates a tree-like model for classification and regression tasks, helpful in fraud detection and medical diagnosis.
	Random Forest	An ensemble learning method that constructs multiple decision trees using random subsets of data and features to improve predictive accuracy and reduce overfitting.
	Gradient Boosting	An ensemble learning method that sequentially builds a series of weak learners to improve predictive accuracy by focusing on the mistakes made by previous models.
	K-Nearest Neighbors (KNN)	predicts by finding the k closest data points and mimicking their label (classification) or averaging their value (regression).
	Multi-Layer Perceptron (MLP)	A basic artificial neural network architecture with multiple hidden layers, useful for various classification and regression problems.
	Convolutional Neural Network (CNN)	Analyzes spatial data to identify patterns, excelling in image recognition, object detection, and segmentation.
	Recurrent Neural Network (RNN)	Processes sequential data, useful for tasks like machine translation, speech recognition, and text generation.
	Long Short-Term Memory (LSTM)	A special type of RNN capable of learning long-term dependencies, valuable for sentiment analysis, time series forecasting, and anomaly detection.
	Transformer	Powerful neural network architecture for various sequence-to-sequence tasks, excelling in machine translation and text summarization.

Unsupervised Learning	Principal Component Analysis (PCA)	Reduces data dimensionality while preserving variance, valuable for anomaly detection and data visualization.
	Self-Organizing Maps (SOMs)	Projects high-dimensional data onto a lower-dimensional grid while preserving relationships, useful for data visualization and dimensionality reduction.
	Autoencoders	Neural networks that learn compressed representations of data by reconstructing the input from a lower-dimensional encoding
	Variational Autoencoders (VAEs)	Probabilistic generative models that learn to encode and decode data samples, enabling the generation of new data instances while capturing complex distributions in the latent space.
	Generative Adversarial Networks (GANs)	Consisting of two neural networks, a generator and a discriminator, competing against each other to generate realistic data samples
	K-Means Clustering	Groups data points into predefined clusters based on similarities, used for customer segmentation and image compression.
Reinforcement Learning	Temporal Difference Learning	Learning value functions in reinforcement learning, where agents learn to make decisions by interacting with an environment and receiving feedback in the form of rewards
	Q-Learning	Learns optimal actions for an agent in an environment through trial and error, used in robot control and game playing.
	Deep Q-Networks (DQNs)	Combines Q-learning with deep neural networks for handling complex environments like complex games and autonomous driving.
	Policy Gradients	Optimizes an agent's policy directly by evaluating its actions, applicable in robotics control and resource allocation.

It's worth noting that transformers, GANs, and VAEs stand out as widely recognized architectures in the field of generative AI. For example, both OpenAI's ChatGPT, Google's Gemini, and Meta's Llama leverage transformer-based technology.

3.2.3 Considerations for Selecting Machine Learning Architectures

Selecting ML architectures for water and environmental modeling involves several key considerations. Firstly, the complexity of the problem at hand determines the level of sophistication needed in the ML architecture. Secondly, the availability and quality of data significantly influence the choice of architecture. Balancing interpretability and accuracy is crucial, especially considering the trade-offs between simpler and more complex models. Additionally, scalability and computational resources must be considered to ensure efficient model training and deployment. Moreover, domain expertise plays a vital role in selecting relevant features and designing appropriate models. Lastly, ensuring the generalization and robustness of the model is essential for addressing dynamic environmental conditions effectively. By understanding these considerations and their implications, ML model users can make informed decisions to leverage the power of ML for addressing pressing water and environmental challenges effectively. These considerations are expanded upon as follows:

1) Problem Complexity: The complexity of the water and environmental problem at hand plays a significant role in determining the appropriate ML architecture. For instance, simple linear regression models may suffice for straightforward problems with few variables, while complex non-linear problems such as salinity modeling under a changing climate may require more advanced architectures like deep learning neural networks.

2) Data Availability and Quality: The quality and availability of data greatly influence the choice of ML architecture. If the dataset is limited or contains missing values, simpler models with regularization techniques may be preferred to avoid overfitting. Conversely, if ample high-quality data is available, more complex architectures can be explored to capture intricate patterns within the data.

3) Interpretability vs. Accuracy Trade-off: ML architectures vary in terms of interpretability and accuracy. While simpler models like decision trees or linear regression are often more interpretable, they may sacrifice predictive accuracy compared to complex models like ensemble methods or deep learning. Understanding the trade-off between interpretability and accuracy is essential when selecting the appropriate architecture for a given problem.

4) Scalability and Computational Resources: Consideration must be given to the scalability of the chosen ML architecture, especially for large-scale water and environmental modeling applications. Deep learning architectures, for example, may require substantial computational resources and training time compared to simpler models. Assessing the available computational resources and scalability requirements is crucial for selecting an architecture that meets the project's needs.

5) Domain Expertise: Domain knowledge and expertise play a critical role in selecting ML architectures for water and environmental problems. Understanding the underlying processes and variables influencing the problem allows for the identification of relevant features and the design of appropriate ML models. Collaboration between

domain experts and data scientists is essential to ensure the chosen architecture aligns with the problem requirements and domain-specific knowledge.

6) Generalization and Robustness: Ensuring the generalization and robustness of ML models is paramount, particularly in dynamic water and environmental systems where conditions may change over time. Techniques such as cross-validation, regularization, and model ensemble methods can enhance the model's ability to generalize well to unseen data and withstand variations in evolving conditions.

3.3 Model Training and Testing

Training and testing of ML models are critical steps in the development and application of ML models for water and environmental modeling. Training of ML models involves using a portion of the available data to teach the model to recognize patterns and relationships between input variables (features) and the target variable (e.g., precipitation, streamflow, water temperature, etc.). This process aims to optimize the model's parameters to minimize the difference between predicted and actual outcomes. Testing, on the other hand, evaluates the trained model's performance on unseen data to assess its ability to generalize to new observations. The goal is to ensure that the model can make accurate predictions on data it has not encountered during training. This sub-section provides an overview of these processes, their importance, and typical methods employed. Specific examples will be provided in Section 6.

Training and testing of ML models are essential for several reasons including performance evaluation, generalization, overfitting prevention, and model improvement. They are detailed as follows:

1) Training and testing allow for the assessment of the model's performance in predicting water and environmental variables. Evaluating model performance helps gauge its accuracy, reliability, and suitability for real-world applications. This is crucial in ensuring that the model's predictions align with observed data and can be trusted for decision-making purposes in water resource management and environmental planning.

2) ML models trained on historical data are expected to generalize well to new, unseen data. By testing the model on a separate dataset not used during training, researchers can assess its ability to generalize patterns learned from historical data to new observations. Ensuring that the model can make accurate predictions on unseen data is vital for its practical utility and reliability in water and environmental modeling applications.

3) Overfitting occurs when a model learns to memorize the training data instead of capturing underlying patterns. Testing the model on an independent dataset helps identify and prevent overfitting by evaluating its performance on data it has not seen before. This ensures that the model captures relevant patterns and relationships without being overly influenced by noise or random fluctuations present in the training data.

4) Training and testing are iterative processes that allow for continuous improvement of the ML model. By evaluating different algorithms, feature sets, and model configurations, model developers can iteratively refine the model to enhance its predictive performance, robustness, and interpretability. This iterative improvement process is essential for developing ML models that accurately capture the complex

dynamics of environmental systems and provide actionable insights for decision-makers.

Typical training and testing methods employed in the development of ML models for water and environmental modeling include handout, temporal splitting and spatial splitting (e.g.,Stone,1974; Raschka, 2018; Abraham et al., 2021). Understanding the strengths and limitations of each method is essential for making informed decisions in environmental modeling applications and ensuring the effective use of ML techniques for addressing key challenges in water resource management and environmental conservation. These methods are briefly described as follows.

The holdout method is one of the simplest and most commonly used techniques for training and testing ML models. In this approach, the available dataset is divided into two subsets: a training set and a testing set. The model is trained on the training set, which typically comprises a larger proportion of the data, and then evaluated on the testing set. The holdout method is straightforward to implement and computationally efficient. However, the results may vary depending on the random partitioning of the data, and the performance estimate may be biased, especially for small datasets.

Temporal splitting is particularly important in environmental modeling, where time-dependent relationships are prevalent. In this approach, the dataset is divided into training and testing sets based on time, ensuring that the model is trained on historical data and tested on unseen data. This allows for the evaluation of the model's ability to generalize to new temporal patterns and trends. Temporal splitting is essential for applications such as hydrological forecasting, where accurate predictions of future water levels or streamflow are critical for decision-making.

Spatial splitting is employed when the dataset exhibits spatial heterogeneity, such as variations in environmental variables across geographical regions. In this approach, the dataset is divided into training and testing sets based on spatial characteristics, ensuring that the model is evaluated on spatially diverse samples. Spatial splitting allows for the assessment of the model's performance across different geographical areas and can be particularly useful for applications such as land cover classification or spatial interpolation of environmental variables.

3.4 Cloud Computing Services

Training and testing machine learning (ML) models can often require significant computational resources, especially when dealing with large datasets or complex model architectures. Cloud resources offer several advantages for ML model training and testing, including scalability, flexibility, cost-effectiveness, and accessibility. There are a number of typical cloud services that are used for ML model training and testing in DWR. These cloud services include Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Azure.

AWS offers a wide range of services for ML model training and testing, including Amazon SageMaker for building, training, and deploying ML models, Amazon EC2 for scalable compute resources, Amazon S3 for storage, and AWS Lambda for serverless computing. GCP provides various ML services such as Google Cloud AI Platform for ML model development, Google Compute Engine for virtual machines, Google Cloud Storage for data storage, and Google Kubernetes Engine for containerized applications.

Azure offers Azure Machine Learning for ML model development, Azure Virtual Machines for scalable compute resources, Azure Blob Storage for data storage, and Azure Kubernetes Service for container orchestration.

There are also other cloud services such as IBM cloud, Alibaba Cloud, and others available. These cloud services provide a comprehensive ecosystem of tools, resources, and managed services for ML model development, training, and testing, enabling organizations to leverage the benefits of cloud computing for their ML projects.

3.5 Model Development Frameworks

ML model developers normally use ML development frameworks to aid in their development efforts. ML model development frameworks play a crucial role in simplifying and accelerating the process of building, training, and deploying machine learning models. By providing a standardized interface and implementation for common machine learning tasks, these frameworks empower model developers to focus on solving complex problems without getting bogged down by implementation details. Understanding the capabilities and features of different machine learning frameworks is essential for choosing the right tool for specific tasks and ensuring the success of machine learning projects. A number of widely used ML frameworks (e.g., Scikit-learn, TensorFlow, and PyTroch) are described below.

Scikit-learn (https://scikit-learn.org/stable/) is a popular open-source machine learning library in Python that provides a comprehensive set of tools for data preprocessing, model building, and evaluation. It offers a wide range of supervised and unsupervised learning algorithms, including classification, regression, clustering, and dimensionality reduction. Scikit-learn is known for its user-friendly API, extensive documentation, and ease of integration with other Python libraries.

TensorFlow (https://www.tensorflow.org/) is developed by Google, TensorFlow is an open-source deep learning framework that provides a flexible and scalable platform for building and training neural network models. TensorFlow offers a high-level API called Keras, which simplifies the process of building and training deep learning models. TensorFlow is widely used for various deep learning tasks, including image recognition, natural language processing, and reinforcement learning.

PyTorch (https://pytorch.org/) is another popular deep learning framework known for its dynamic computational graph and intuitive Pythonic interface. Developed by Facebook's AI Research lab, PyTorch offers a flexible platform for building and training deep neural networks. It provides support for dynamic graph computation, allowing users to define and modify computational graphs on-the-fly, making it particularly well-suited for research settings.

4. Model Evaluation and Interpretation

4.1 Model Performance Evaluation

The efficacy and reliability of ML models hinge upon rigorous validation procedures. In the domain of water and environmental modeling, where decisions can have profound socio-economic and ecological implications, the accuracy and reliability of ML models are paramount. Validating ML models ensures that they accurately capture underlying patterns and dynamics, enabling ML model users to make informed decisions with confidence. This sub-section discusses widely used ML model evaluation methods, metrics, challenges, and best practices.

4.1.1 Evaluation Methods

Cross-validation is a crucial technique in machine learning for assessing the performance and generalization ability of predictive models (e.g., Bates et al., 2023; Roberts et al., 2017; Stone, 1974). It involves dividing the available dataset into multiple subsets, commonly referred to as folds. The model is trained on a portion of the data and then validated on the remaining data. This process is repeated multiple times, with different subsets used for training and validation in each iteration. Cross-validation helps to mitigate the risk of overfitting by ensuring that the model's performance is evaluated on multiple independent subsets of the data. This technique is especially important in fields such as water and environmental modeling, where accurate predictions for a range of conditions are essential for effective decision-making regarding water and environmental management and planning.

In the context of water and environmental modeling, various cross-validation techniques are employed to evaluate the performance of machine learning models. These techniques are tailored to address the specific challenges and characteristics of environmental datasets. One widely used approach is **k-fold cross-validation** (e.g. Rodriguez et al., 2009), where the dataset is divided into k subsets of approximately equal size. The model is trained k times, each time using a different fold for validation and the remaining folds for training. This technique ensures that every data point is used for both training and validation, providing a comprehensive assessment of the model's performance across different subsets of the data.

Another common cross-validation technique is **stratified k-fold cross-validation** (e.g., Diamantidis et al., 2000; Zeng and Martinez, 2000), which is particularly useful for datasets with imbalanced class distributions. In this approach, the class distribution is preserved in each fold to ensure that all classes are represented in both the training and validation sets. This helps to prevent biases in the evaluation process and ensures that the model's performance is assessed accurately for each class.

Nested cross-validation (e.g., Bates et al., 2023) is another variant of k-fold cross-validation. This method is particularly useful for hyperparameter tuning and model selection. It involves an outer k-fold cross-validation loop for model evaluation and an inner k-fold cross-validation loop for hyperparameter tuning. Nested cross-validation provides a robust estimate of model performance while avoiding data leakage.

Bootstrap resampling (e.g., Fortin et al., 1997; Dixon, 2006) is another approach that can be utilized for cross-validation in environmental modeling. In this

technique, multiple random samples are drawn with replacement from the dataset, and the model is trained and validated on each sample. This process is repeated multiple times to assess the variability and uncertainty associated with the model's performance metrics. Bootstrap resampling helps to provide a more comprehensive understanding of the model's robustness and reliability in different scenarios.

Monte Carlo Cross-Validation (e.g., Xu and Liang, 2001) is another widely used cross-validation technique. In this approach, the dataset is randomly partitioned into training and validation sets multiple times. The model is trained and evaluated on each partition, and performance metrics are averaged across iterations. Monte Carlo cross-validation helps to assess the stability of model performance across different random partitions of the data.

Overall, cross-validation techniques play a critical role in evaluating machine learning models in water and environmental modeling. By assessing the model's performance on multiple independent subsets of the data, these techniques help to identify potential issues such as overfitting, underfitting, or biases in the model's predictions. This ensures that the resulting models are accurate, reliable, and well-suited for making informed decisions in water and environmental management and planning.

4.1.2 Evaluation Metrics

Evaluating machine learning models requires careful consideration of various metrics tailored to the specific characteristics of the data and the objectives of the modeling task. In this section, we describe typical evaluation metrics for regression problems, classification problems, and visualization commonly used in the field of water and environmental modeling.

Regression problems in water and environmental modeling often involve predicting continuous numerical values, such as salinity level, streamflow rate, water stage, water temperature, among others. Common evaluation metrics for regression tasks are summarized in Table 3 below.

Table 3. Commonly used evaluation metrics for regression models.

Evaluation Metric	Description	Example Reference
Coefficient of Determination (R^2)	Reflects the proportion of variation in a dependent variable that can be explained by the changes in an independent variable	Chen et al., 2018; Namadi et al., 2022, 2023
Percent Bias	Shows how much on average a model's predictions tend to be over or under the actual values	Moriasi et al., 2007; Roh et al., 2023
Mean Square Error (MSE)	Tells how far a model's predictions are from the actual values by taking the average of the squared differences on average	Rath et al., 2017; Chen et al., 2018
Mean Absolute Error (MAE)	Captures the average absolute amount of error a model's predictions have from the actual values	Moriasi et al., 2007; Namadi et al., 2022, 2023
Root Mean Square Error (RMSE)	Quantifies the average magnitude of the difference between predicted and actual values	Moriasi et al., 2007; Chen et al., 2018
RMSE-Observation Standard Deviation Ratio (RSR)	Indicates how much error a model has relative to the natural variability of the data itself	Moriasi et al., 2007; Roh et al., 2023
Nash-Sutcliffe Efficiency (NSE)	Evaluates how well a model captures the variability and magnitude of the target variable	Nash and Sutcliffe, 1970; Qi et al., 2023
Kling-Gupta Efficiency (KGE)	Assesses how well a model replicates observed data by considering correlation, variability, and mean bias in a single score	Gupta et al., 2009; Tongal and Booij, 2018

Classification problems in water and environmental modeling involve predicting categorical outcomes, such as water quality classes, land cover types, or habitat suitability. By employing a combination of these metrics and visualizations tailored to the specific problem and dataset, model developers and users can comprehensively evaluate the performance of machine learning models in water and environmental applications. This facilitates informed decision-making regarding model selection,

optimization, and real-world deployment, ensuring reliable predictions for critical environmental issues. Common evaluation metrics for classification tasks include (e.g., Vujović, 2021):

Accuracy: Measures the proportion of correctly classified instances out of the total number of instances. While simple to interpret, accuracy may not be suitable for imbalanced datasets.

Precision and Recall: Precision measures the proportion of true positive predictions among all positive predictions, while recall measures the proportion of true positive predictions among all actual positives. These metrics are particularly relevant for imbalanced datasets.

F1 Score: The harmonic mean of precision and recall, providing a balanced measure of a model's performance that considers both false positives and false negatives.

Visualization plays a crucial role in understanding model performance and interpreting model predictions. In the context of water and environmental modeling, visualization techniques can include:

Scatter Plots: Visualizing predicted versus actual values for regression tasks to assess the model's accuracy and identify any systematic errors or trends.

Time Series Plots: When dealing with temporal data (e.g., streamflow), time series plots can visually compare predicted and observed values over time, revealing potential issues like seasonality or sudden fluctuations that the model may not be capturing accurately.

Error Histograms: Histograms of prediction errors can shed light on the distribution of errors and identify potential outliers. A skewed error distribution might indicate that the model performs better for a certain range of values and needs further calibration.

Confusion Matrices: Providing a visual representation of the model's performance in classification tasks by illustrating the distribution of true positive, true negative, false positive, and false negative predictions.

Receiver Operating Characteristic (ROC) Curves: Plotting the true positive rate against the false positive rate across different threshold values to visualize the trade-off between sensitivity and specificity in binary classification tasks.

Feature Importance Plots: Illustrating the importance of different input features in predicting the target variable, providing insights into the underlying relationships between variables and the model's decision-making process.

4.1.3 Evaluation Challenges

Evaluating machine learning models in the field of water and environmental modeling presents unique challenges due to the complexity of environmental systems, the diversity of data sources, and the multifaceted nature of environmental processes. In this sub-section, we explore some of the key challenges encountered when assessing the performance of machine learning models in water and environmental modeling:

1) Data Quality and Availability:

Limited Data: Water and environmental data collection can be expensive and time-consuming, often resulting in limited datasets. This restricts the amount of data available for training and evaluating ML models.

Heterogeneity: Water and environmental data can be highly heterogeneous, with spatial and temporal variations. Models trained on data from one location or time period might not generalize well to different conditions.

Missing Values and Noise: Water and environmental data often contains missing values due to sensor malfunctions or limitations. Additionally, noise can be introduced during data collection and processing. These issues can significantly impact the accuracy and reliability of ML models.

Imbalanced Datasets: Many water and environmental problems involve imbalanced datasets. For instance, water quality classifications might have significantly more "safe" samples compared to "polluted" ones. This can lead to models that excel at predicting the majority class but struggle with identifying less frequent but critical events (e.g., pollution outbreaks).

Non-Stationary Data: Water and environmental systems are inherently dynamic, with constant changes in factors including sea level, climate, land use, and land subsidence. Models trained on historical data might not accurately predict future behavior if these underlying conditions shift significantly.

2) Model Complexity and Interpretability:

Complexity-Interpretability Trade-off: Advanced machine learning models, such as deep learning algorithms, may achieve high predictive performance but lack interpretability, making it challenging to understand the underlying mechanisms driving model predictions. This point will be elaborated in Section 4.2.

Model Overfitting: Complex models trained on limited data may overfit to the training set, capturing noise instead of true underlying patterns, leading to poor generalization performance on unseen data.

Model Uncertainty: Assessing uncertainty in model predictions is crucial for decision-making in environmental management, but quantifying uncertainty in machine learning models remains a challenging task.

Despite these challenges, a number of techniques have been developed to enhance the evaluation process for ML models. These techniques include data augmentation (e.g., Wen et al., 2020) and explainable AI (XAI) (e.g., Arrieta et al., 2020; Linardatos et al. 2020).

Data augmentation is a technique widely used in machine learning to artificially expand the size of a dataset by applying various transformations to existing data points. These transformations include but are not limited to rotation, scaling, translation, flipping, cropping, and adding noise. In the realm of water and environmental modeling, data augmentation plays a crucial role in overcoming challenges associated with model evaluation. Data augmentation helps alleviate the limited data issue by generating synthetic data points, thus increasing the diversity and quantity of available data for model training and evaluation. In addition, data augmentation techniques such as oversampling of minority classes or generating synthetic samples for underrepresented classes can help mitigate the imbalance issue in available data, leading to more robust model evaluation. Furthermore, environmental data often exhibit inherent variability and noise due to natural processes, measurement errors, or sensor inaccuracies.

Augmenting the training data with noise injection or variations in input features can help the model learn to be more robust to such variations, leading to more reliable performance evaluation under real-world conditions.

Explainable AI (XAI) refers to the set of techniques and methodologies aimed at providing human-understandable explanations for the decisions made by machine learning models (Gilpin et al., 2018). In the context of water and environmental modeling, XAI plays a crucial role in addressing challenges associated with model evaluation, particularly concerning model complexity and interpretability. XAI techniques, such as feature importance analysis, partial dependence plots, and SHAP (SHapley Additive exPlanations) (e.g., Lundberg and Lee, 2017) values, can help reveal how different input variables influence model predictions, thereby providing insights into the underlying mechanisms captured by the model. Additionally, XAI methods can aid in simplifying ML models by identifying redundant or irrelevant features and highlighting the most influential factors in the model's decision-making process. By understanding which variables drive the model's predictions, model developers can build simpler, more interpretable models without sacrificing predictive accuracy. XAI will be further discussed in Section 4.2.

Acknowledging and addressing the challenges associated with evaluating ML models in water and environmental modeling is critical. By employing appropriate evaluation methods, data augmentation techniques, and interpretability tools, model developers can build robust and reliable models that contribute to sustainable water and environmental management.

4.2 Model Interpretation

ML models have revolutionized water and environmental modeling, offering powerful tools for tasks like predicting water quality, forecasting floods, and assessing climate change impacts. However, these models often operate as complex "black boxes", generating skillful simulations and predictions without revealing the underlying logic behind their decisions. Model interpretation aims to shed light on the inner workings of ML models. It seeks to answer questions like which factors most influence the model's predictions? how do different inputs interact to produce an output? are there potential biases present in the model's decision-making process?

By addressing these questions, ML model interpretation can help enhance:

1) trust and transparency: stakeholders can understand the rationale behind a model's predictions, fostering trust in its reliability and reducing the risk of relying on opaque algorithms.

2) model debugging and improvement: explanations can reveal potential biases or weaknesses in the model, enabling model developers to refine and improve its performance.

3) decision-making: by examining model interpretations, model developers and users to identify important variables and relationships in environmental systems and critical environmental factors driving specific outcomes, leading to more informed decisions for water and environmental management.

Traditionally, interpreting ML models relied on basic techniques like correlation analysis and feature importance scores. These methods offered limited insights, often

highlighting correlations without uncovering causal relationships. The emergence of XAI has brought a new wave of powerful interpretation tools. These techniques can be broadly categorized into:

1) Model-Agnostic Methods

Model-Agnostic Methods are a category of techniques used to explain the predictions made by any machine learning model, regardless of its underlying architecture. This is particularly valuable because many complex models used in environmental science can be difficult to interpret directly. Model-agnostic methods provide explanations for individual predictions, helping us understand which factors played the most significant role in a specific model's output. Here are some common model-agnostic methods used in XAI for water and environmental modeling (Başağaoğlu et al., 2022):

Permutation Feature Importance: This method assesses the importance of input features by permuting their values and observing the resulting change in model performance. Features with larger decreases in performance upon permutation are deemed more important. This method is simple yet effective in assessing feature importance across various models. For instance, Ramirez et al. (2022) utilized the PFI method within a conventional multi-layer perceptron model to pinpoint the crucial input features essential for estimating the Palmer Drought Severity Index on a global scale.

SHAP: SHAP draws on concepts from game theory to explain how each feature in the data contributes to a model's prediction. It assigns a "SHAP value" to each feature, representing its marginal contribution to the model's output. By analyzing these SHAP values, we can understand which features were most important for pushing the model's prediction in a particular direction. As an example, Park and colleagues (2022) employed SHAP within XB boosting models to determine the most significant input features for simulating chlorophyll-a concentration.

LIME (Local Interpretable Model-agnostic Explanations): LIME (Ribeiro et al., 2016) generates locally interpretable models around specific instances by perturbing the input data and observing the resulting changes in model predictions. These locally fitted models provide insights into how the global model behaves in the vicinity of a particular data point. For instance, Wikle et al. (2023) showcased the utilization of three model-agnostic explainability techniques, including LIME, across various machine learning models applied in an environmental prediction context.

2) Model-Specific Methods

Model-Specific Methods are a category of techniques used to interpret the predictions made by machine learning models. Unlike model-agnostic methods which work for any model, model-specific methods leverage the internal structure and workings of a particular model architecture to explain its behavior. This allows for a deeper understanding of how the model arrives at its predictions, tailored to its specific design. Here are some common Model-Specific Methods used in XAI:

Decision Tree Interpretability: This method is particularly useful for interpreting decision tree models, which are relatively simple and transparent. It works by analyzing the decision rules used in the tree. Each node in a decision tree represents a split on a particular feature, and the branches represent the possible outcomes of these splits. By tracing the path a data point takes through the tree based on its feature values, we can understand the sequence of decisions the model made to arrive at a prediction.

Gradient Boosting Model Interpretation: Gradient boosting models are ensembles of weaker decision trees. While the final model can be complex, interpretation techniques can focus on individual trees within the ensemble. We can analyze the contribution of each tree to the final prediction and identify the features that were most influential within each individual tree. This provides a more granular understanding of how the ensemble model arrives at its predictions.

Attention Mechanisms: This method is commonly used for interpreting complex models like deep neural networks, especially in computer vision tasks. Attention mechanisms assign weights to different parts of the input data, indicating which parts the model focused on most for making its prediction. By visualizing these attention weights, we can understand which features (e.g., specific pixels in an image) were most critical for the model's decision.

5. Model Deployment and Communication

5.1 Model Deployment

ML model deployment refers to the process of making trained machine learning models accessible and operational for use in real-world applications in the fields such as climate, hydrology, environmental science and engineering. It involves taking the model developed during the training phase and deploying it in a production environment where it can generate predictions or insights based on new, unseen data. ML model deployment is a critical step in the lifecycle of any ML project, as it enables ML model users to derive value from the model's capabilities and integrate it into decision-making processes or operational workflows. This sub-section explores the considerations, processes, platforms, and tools involved in deploying ML models for water and environmental applications.

5.1.1 Deployment Considerations

Before deploying an ML model, several factors need careful consideration:

Use Case: Clearly define the model's purpose and intended outcomes (e.g., flood prediction, water quality assessment). This influences the deployment environment and user interface (UI) design. For instance, a flood prediction model for reservoir operators would require a real-time, low-latency interface with visualizations of predicted reservoir level, inflow rate, and inundation zones (if applicable). In contrast, a water quality assessment model might prioritize detailed model outputs and data exploration capabilities.

Target Audience: Who will be using the model? Are they technical experts or non-specialists? This determines the level of user interaction required in the deployed system. A model for environmental scientists might involve a programmatic interface for integration with existing workflows. On the other hand, a public-facing water quality forecasting tool would necessitate a user-friendly graphical interface with minimal technical jargon.

Data Availability and Management: Consider ongoing data access needs for model updates and performance monitoring. A robust data pipeline for feeding real-time

or historical data is essential. For instance, a model predicting Delta salinity levels based on flow and gate operations would require uninterrupted access to the real-time flow measurements and gate operation data.

Computational Resources: Evaluate the computational power required for model inference. Cloud platforms offer scalability, while on-premise deployments necessitate sufficient hardware resources. Complex deep learning models for environmental simulations might necessitate powerful GPUs or specialized hardware accelerators, while simpler regression models for water quality simulation and prediction could run efficiently on standard servers.

Explainability and Interpretability: Environmental models often deal with complex relationships. Consider techniques like LIME (Local Interpretable Model-agnostic Explanations) to make model predictions understandable for users. Transparency in model decision-making is crucial for building trust with stakeholders and regulators, especially when models are used for critical environmental decisions.

Security and Privacy: Ensure data security and user privacy, especially when dealing with sensitive environmental data. Implement access controls and anonymization techniques where necessary. For instance, user authentication and authorization mechanisms would be essential for a flood forecasting tool used by multiple government agencies.

5.1.2 Deployment Process

The deployment process typically follows these steps:

1). Model Packaging: Package the trained model along with its dependencies (libraries, frameworks) into a format suitable for the chosen deployment platform. Common formats include PMML (Predictive Model Markup Language) or containerized environments (Docker).

2). API Development: Develop an Application Programming Interface (API) to expose the model's prediction capabilities to external applications or user interfaces. This allows users to interact with the model without directly accessing the code. A well-designed API with clear documentation facilitates integration with other systems and streamlines model adoption by a wider audience.

3). Environment Setup: Configure the deployment environment based on the chosen platform. This could involve setting up cloud instances, installing containerization software, or configuring web servers. Cloud platforms often provide pre-built environments and tools to simplify this process.

4). Model Deployment: Deploy the packaged model and API to the chosen environment. This may involve pushing code to a version control system (e.g., Git on GitHub) or deploying a container image to a cloud platform. Version control systems ensure proper tracking of deployed models and facilitate rollbacks if necessary.

5). Monitoring and Maintenance: Continuously monitor model performance and data quality. Regularly retrain the model with new data to maintain accuracy. Implement logging and error handling mechanisms for troubleshooting. Monitoring tools and dashboards help identify issues with data quality or model degradation over time, allowing for proactive maintenance and improvement.

5.1.3 Deployment Platforms

Several platforms offer suitable environments for deploying ML models for water and environmental applications:

Cloud Platforms: These offer scalable, pay-as-you-go resources for model deployment and management. Cloud providers like Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Azure offer pre-built infrastructure, tools for model training and deployment, and integration with other cloud services. They are particularly advantageous for models requiring significant computational resources or real-time predictions. For instance, a cloud-based flood prediction model could leverage on-demand scaling of virtual machines to handle surges in data during extreme weather events.

Containerization Platform (Docker): Docker is a popular containerization tool. It allows packaging a model with all its dependencies (libraries, frameworks) into a lightweight, self-contained unit. This containerized model can then be easily deployed across different environments, whether on cloud platforms or on-premise servers. For instance, a water quality prediction model trained on historical sensor data could be packaged as a Docker container. This container would include the model itself, along with any necessary libraries and Python code for data processing and prediction. The containerized model could then be deployed to a server near a water treatment plant, enabling real-time water quality predictions based on live sensor data streams. This approach simplifies deployment, ensures consistent model behavior across environments, and minimizes the need for specialized technical knowledge at the deployment site.

In addition, version control systems like **GitHub** can be used in conjunction with the cloud and containerization platforms. GitHub excels at version control, allowing developers to track changes made to the model code, data pipelines, and deployment scripts. This facilitates collaboration, rollback to previous versions if necessary, and ensures everyone involved is working on the same codebase. Code from GitHub might be pushed to a cloud platform for deployment, or container images built from code in GitHub could be deployed on cloud platforms or on-premise servers.

5.2 Model Communication

Effective deployment of an ML model for water or environmental applications is only the first step. To ensure its ongoing success and user adoption, clear communication strategies are essential. Water and environmental models often involve complex algorithms and data. Without clear communication, users may struggle to understand the model's capabilities and limitations. By implementing effective communication strategies, deployed ML models for water and environmental applications can reach their full potential. Clear documentation, user training, and ongoing engagement empower users to leverage the model's capabilities and contribute to a more informed and sustainable future for our water resources and environment. This chapter explores the importance of documentation, training, and ongoing user engagement for deployed ML models in these critical fields.

5.2.1 The Importance of Clear Communication

Effective communication fosters trust and transparency, allowing users to make informed decisions based on model outputs. Here's why communication is crucial:

Building User Confidence: Comprehensive documentation helps users understand the model's purpose, strengths, and weaknesses. This builds user confidence and encourages them to integrate the model into their workflows.

Ensuring Model Adoption: Clear communication strategies increase user understanding and promote model adoption across various stakeholder groups, from water resource managers to environmental scientists.

Facilitating Feedback and Improvement: Transparent communication channels allow users to provide feedback on model performance and identify potential areas for improvement. This iterative process leads to continuous refinement and ensures the model remains relevant and accurate.

5.2.2 Communication Strategies for Deployed Models

Several communication strategies can enhance understanding and engagement with deployed ML models:

Comprehensive Documentation: Develop user manuals, FAQs, and online resources that explain the model's purpose, algorithms used, data sources, and limitations. Visual aids like flowcharts and diagrams can further enhance clarity.

Interactive Training Materials: Provide online tutorials, webinars, or in-person workshops to train users on how to interact with the model and interpret its outputs. This can include training on data formatting, API usage, and visualization tools. For instance, the Modeling Support Office of DWR provided a hands-on training on an ML-based Delta salinity simulation dashboard tool in January 2023 to stakeholders and partners. The training slides, input and output data, step-by-step guidance, and the source code were made publicly available at the following GitHub link: https://github.com/CADWRDeltaModeling/SalinityMLWorkshop_DMS_UCD. The training session was recorded, and the videos were likewise made accessible in the public domain: https://www.youtube.com/playlist?list=PL33EJkVWqEIV7Eo03RGnHabMoMiF7IPBW.

Interactive User Interface Design: Where applicable, design user interfaces (UIs) that are intuitive and user-friendly. Consider features like interactive visualizations, data exploration tools, and clear explanations of model outputs. The Modeling Support Office developed two ML-based interactive dashboard tools for salinity simulation and ion concentration simulation, respectively. These dashboards are further discussed in Section 6.1.

Case Studies and Success Stories: Showcase real-world examples of how the model is being used to address water and environmental challenges. This demonstrates the model's value proposition and encourages broader adoption.

Open Communication Channels: Establish clear communication channels for users to provide feedback, report issues, and ask questions. This could include email support, online forums, or designated personnel for technical inquiries.

5.2.3 Fostering a Culture of Continuous Learning

Communication is an ongoing process. Regularly update documentation and training materials as the model evolves and new features are added. Encourage user feedback and actively engage with the user community to ensure the model continues to meet their needs. Here are a few additional strategies that can be considered:

Knowledge Base and FAQs: Develop a searchable knowledge base populated with frequently asked questions and troubleshooting tips. This empowers users to find answers independently.

User Group Meetings: Organize regular user group meetings to foster knowledge sharing, address common challenges, and gather user feedback for future model improvements. As an instance, the Modeling Support Office initiated the Delta Modeling User Group (DMUG) two decades ago. The DMUG convenes quarterly gatherings to exchange advancements within the Delta modeling community.

Brownbag Meetings: Brownbag meetings serve as an excellent communication strategy for disseminating insights and updates regarding deployed machine learning models for water and environmental modeling. These gatherings provide a casual and interactive platform where researchers, stakeholders, and practitioners can share their findings, discuss challenges, and explore potential solutions in a collaborative manner. By convening regularly over lunch or coffee breaks, participants have the opportunity to exchange knowledge, best practices, and real-world experiences related to the deployed models. Furthermore, brownbag meetings foster a sense of community and engagement among team members, encouraging open dialogue and the sharing of diverse perspectives. This informal setting not only enhances communication efficiency but also facilitates the identification of emerging issues and the implementation of timely adjustments to improve model performance. Ultimately, brownbag meetings serve as a vital conduit for promoting transparency, fostering innovation, and ensuring the continued relevance and effectiveness of machine learning applications in water and environmental modeling.

The Modeling Support Office of DWR recently organized an ML brownbag meeting to daylight several ML applications in the Delta. The documentation and recording of the brownbag are available in the public domain.

Model Explainability Tools: Integrate explainability tools (e.g., LIME) that help users understand the rationale behind model predictions. This builds trust and transparency in the model's decision-making process.

6. Case Study and Best Practices

This section presents a case study that showcases the full life cycle of a developed and deployed ML model (Figure 2). Two additional case studies are provided in Appendix A and Appendix B, respectively. Additionally, this section outlines the best practices for developing ML models related to water and environmental issues.

6.1 Example Case Study

This case study illustrates the development of ML models for ion concentration simulation as well as a dashboard tool centered on the developed ML models.

6.1.1 Problem Definition

In the Sacramento-San Joaquin Delta (Delta), the water hub of the complex water systems of California, the composition of water is significantly influenced by various salts and minerals such as chloride, sulfate, sodium, magnesium, and potassium. Collectively identified as ions, these salts and minerals are pivotal for maintaining the ecological balance, agricultural viability, and ensuring the safety of drinking water in the Delta.

The traditional approach to measuring these ions involves infrequent (e.g., roughly monthly) onsite sampling (i.e., grab samples) at limited locations followed by laboratory analysis. This method, while accurate, is time-consuming, expensive, and provides results that are not immediately available and lack spatial comprehensiveness. The necessity for an advanced approach is underscored by the challenges inherent in sampling methods, prompting the exploration of more efficient alternatives. Measurements of Electrical Conductivity (EC) show great promise because they correlate with ions and are easily accessible from automatic sensors installed at key locations throughout the Delta. Historically, DWR has utilized parametric regression equations for converting EC into ion levels (Guivetchi, 1986). However, these linear conversion equations are limited for ions having strong nonlinear relationships with EC.

The objective of this case study is twofold. Firstly, it seeks to create machine learning models that can mimic and possibly enhance the current regression equations for EC-ion conversion. Secondly, it provides a user-friendly interactive dashboard based on these ML models. Through its interactive features, the dashboard allows users to easily modify input parameters and quickly visualize ion concentrations for various hydrological scenarios, without needing any programming skills.

6.1.2 Data Collection

The data used in this study were compiled from three distinct sources. This comprehensive dataset spans a significant temporal range, covering measurements from 1959 to 2022, and includes data collected from 30 positioned monitoring locations throughout the Delta. Figure 3 illustrates the geographical distribution of the 30 locations across the Delta. This figure visually conveys the scope of data collection efforts, emphasizing the spatial coverage that supports the robustness of analysis. The locations are grouped into three main sub-regions to reflect their unique hydrological and environmental characteristics: the Old-Middle River (OMR), the San Joaquin River Corridor, and the South Delta. These sub-regions were identified to account for the diverse sources of water and the varying quality impacts, thereby facilitating a nuanced understanding of the Delta's water quality dynamics.

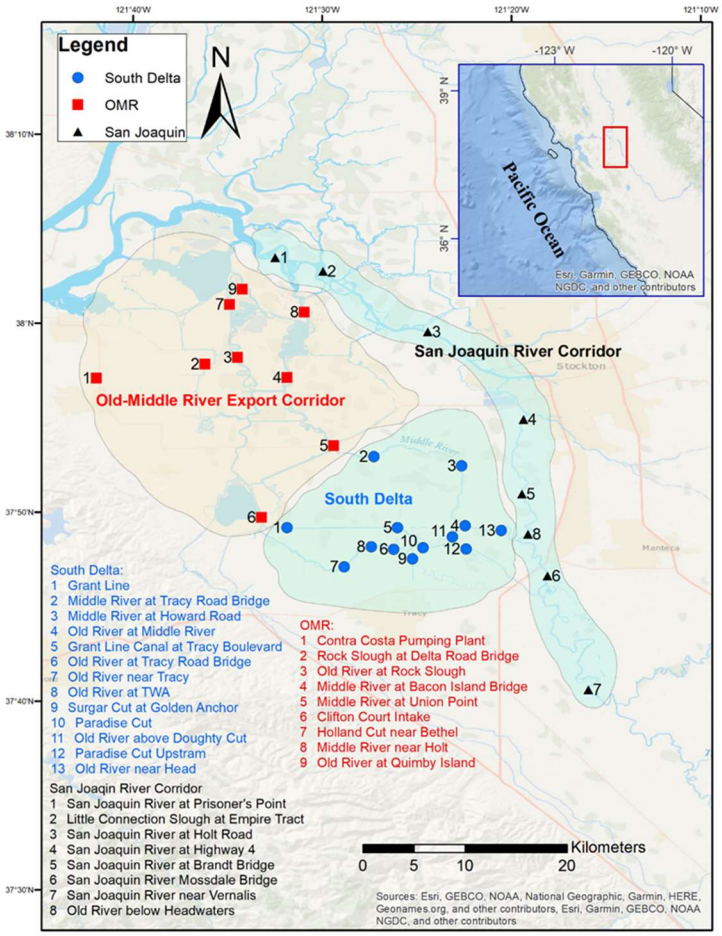

Figure 3: Study locations of the case study.

The primary dataset originates from historical records maintained by Hutton et al. (2022), encompassing ion grab samples, EC, and X2 position data collected between 1959 and 2018 at 19 stations within the study area. The second dataset includes samples collected by DWR between 2018 and 2020, focusing on seven stations in the South Delta sub-regions. Finally, the third dataset, downloaded from the California Data Exchange Center (CDEC) website (https://cdec.water.ca.gov/), extends the geographical and temporal coverage by including samples from 13 stations within the interior Delta, collected from 2018 to 2022.

The integration of these datasets provides a holistic view of the water quality dynamics within the Delta, capturing a broad spectrum of hydrologic conditions and environmental changes over time. This approach ensures a robust analytical foundation, enabling the development of models to accurately simulate ion levels under a variety of conditions.

Table 4 presents a detailed overview of the ion constituents analyzed in this study, highlighting the sample size, range of data (Data), and the standard deviation (SD) of the data for each ion constituent. This table illustrates the dataset's composition and variability, offering insights into the distribution and concentration of ions across the Delta. The ions featured in the table include chloride, sulfate, sodium, magnesium,

potassium, and others, each playing a unique role in influencing the water quality and ecological health of the Delta.

Table 4. Overview of ion sample data.

Ion	Sample size	Data range	SD	Period	Units
TDS	1466	49-2120	204	1968 - 2022	mg/L
Mg^{2+}	1336	2-102	8.6	1959 - 2022	mg/L
Na^{+}	1575	6-343	44	1959 - 2022	mg/L
Ca^{2+}	1335	5.8-244	18	1959 - 2022	mg/L
Cl^{-}	1972	4-775	77	1959 - 2022	mg/L
$SO4^{2-}$	1066	5-350	46.5	1959 - 2022	mg/L
Br^{-}	1239	0.01-2.3	0.22	1990 - 2022	mg/L
Alkalinity	1036	26-198	27.6	1959 - 2020	mg/L as CaCO3
K^{+}	1148	0.87-11	1.35	1959 - 2022	mg/L

6.1.3 Data Pre-processing

The data pre-processing stage is crucial in preparing the dataset for machine learning analysis, ensuring that the input data is clean, normalized, and suitable for the algorithms to process efficiently. This section outlines the key pre-processing steps undertaken, including visualization, normalization, and the treatment of categorical data. Visualization plays a pivotal role in understanding the underlying patterns and relationships within the dataset. The initial visualization step was instrumental in segmenting the ions into three groups, guided by their relationship patterns with EC.

Figures 4, 5, and 6 illustrate the relationship between EC and the ion constituents across the different sub-regions of the Sacramento-San Joaquin Delta. These visual representations helped identify the linear, bifurcated, and non-linear relationships between EC and ion levels, facilitating the categorization of ions into three distinct groups based on their response patterns to changes in EC. For example, Figure 4 depicts a strong linear relationship for certain ions, suggesting a direct proportionality with EC levels. However, Figures 5 and 6 reveal more complex patterns, including bifurcation and non-linear responses, indicating that the relationship between EC and ion concentrations can vary significantly depending on the ion type and environmental factors. The groups are defined as follows:

Group 1: Ions with a linear relationship to EC, indicating a direct and proportional increase in ion concentration with EC. (magnesium (Mg2+) and total dissolved solids (TDS))

Group 2: Ions exhibiting bifurcation or branching patterns in response to EC, suggesting varying ion concentration pathways under different conditions. (sodium (Na+), chloride (Cl-), and calcium (Ca2+))

Group 3: Ions with non-linear relationships to EC, reflecting complex dynamics that cannot be captured by a straightforward linear model. (sulfate (SO42-), alkalinity, potassium (K+), and bromide (Br-))

The division into these groups allows for a more nuanced analysis, acknowledging the diverse behaviors of ion constituents in the Delta's waters and ensuring that the machine learning models are optimally configured to reflect these complexities.

To accommodate the diverse range observed across the dataset, particularly in the numerical predictors such as Electrical Conductivity (EC) and X2 position, a normalization process was helpful. This normalization adjusted the dataset to a uniform scale, specifically between 0 and 1, for these predictors. The procedure ensures equal contribution of each feature to the analytical model, crucial for mitigating potential biases. High-value ions, or any disproportionately large measurements, could otherwise skew the machine learning models' performance. By normalizing EC and X2 data, we enhanced the dataset's comparability and interpretability, facilitating an objective analysis.

The incorporation of categorical variables, such as Water Year Type (WYT), month, and sub-region, posed a distinct challenge due to the inherent numerical nature of machine learning algorithms. For instance, WYT, denoting the hydrological condition of a year with classifications ranging from 'wet' to 'critical,' required conversion into a computationally understandable format. To achieve this, one-hot encoding was utilized, transforming these categorical variables into binary columns. Each column represents a potential category value, converting qualitative attributes into a quantitative framework suitable for analysis. For example, a 'wet' year is encoded as [1, 0, 0, 0, 0], while a 'dry' year as [0, 0, 0, 1, 0]. This method allows the seamless integration of essential hydrological, temporal, and spatial data into the models without compromising the numerical requirements of the machine learning algorithms.

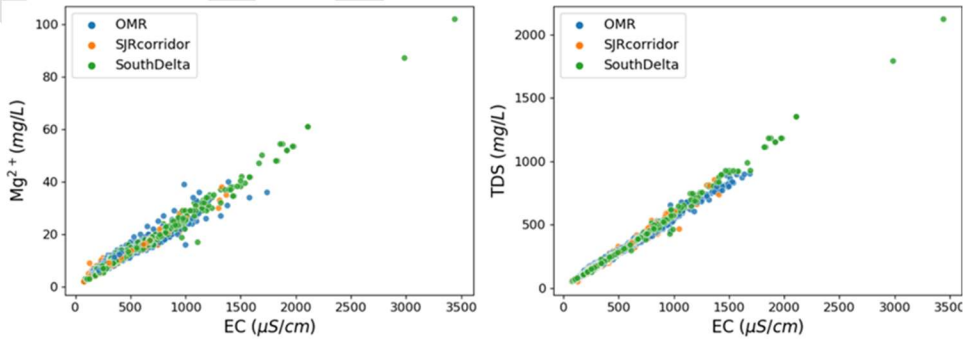

Figure 4. Scatter Plots showing the relationship between salinity (represented by EC) and ion constituents with linear relationship with EC (group 1).

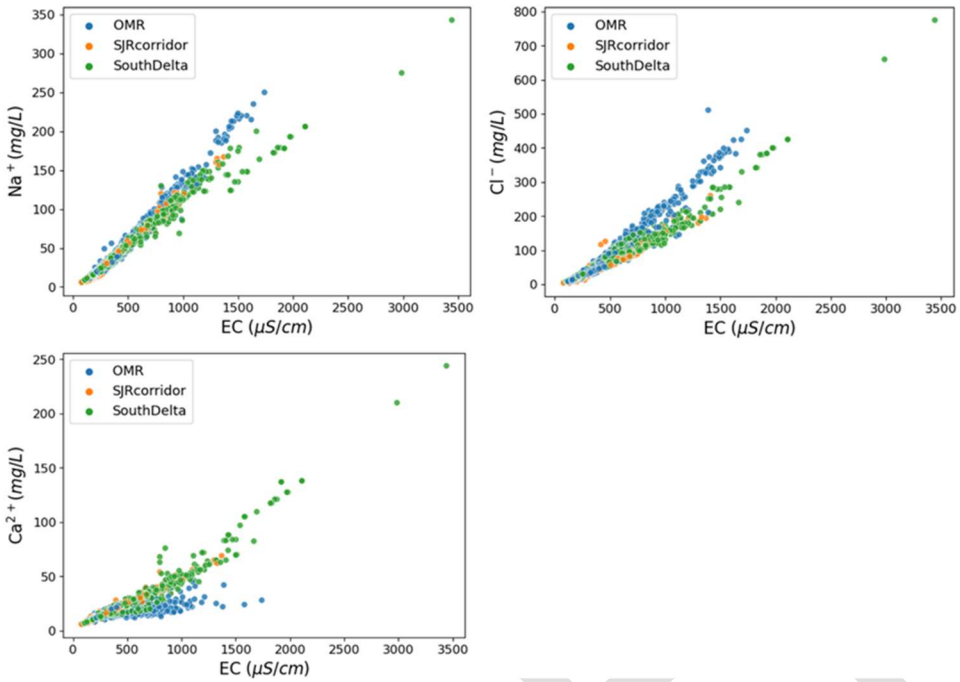

Figure 5. Scatter Plots showing the relationship between salinity (represented by EC) and ion constituents with bifurcation relationship with EC (group 2).

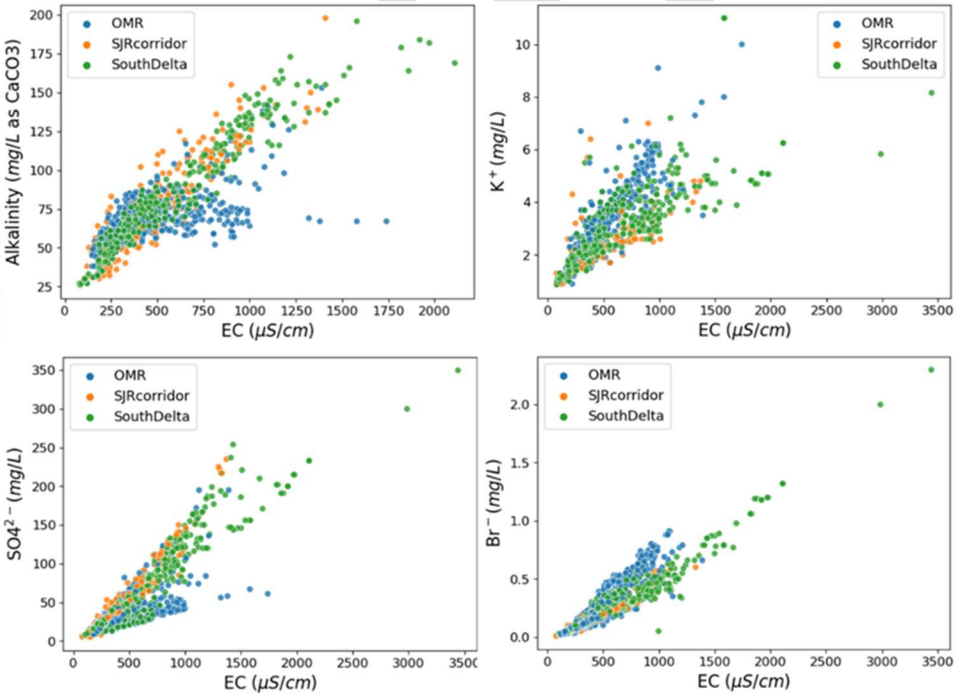

Figure 6. Scatter Plots showing the relationship between salinity (represented by EC) and ion constituents with non-linear relationship with EC (group 3).

6.1.4 Model Selection

The primary objective of this study was to predict ion concentrations in the Sacramento-San Joaquin Delta, a problem that inherently falls within the domain of regression analysis in machine learning. Regression problems require models that can predict continuous outcomes, such as ion concentrations, based on one or more predictor variables, such as Electrical Conductivity (EC) and X2 position. Given this context, the model selection process was guided by the necessity to choose machine learning models adept at solving regression problems and capable of handling the complexity and non-linearity of environmental data.

Moreover, it's important to note that our dataset, derived from grab samples, does not constitute a time series. Grab samples provide a snapshot of water quality at specific points in time and locations but lack the sequential, time-dependent structure that characterizes time series data. As a result, models that are good at time series forecasting, such as Long Short-Term Memory (LSTM) networks, are not the most appropriate models for our dataset. LSTM and similar time series models excel in capturing temporal dependencies and patterns over time, which our grab sample dataset does not present.

Given these considerations, the following machine learning models were selected for their proven effectiveness in regression problems and their suitability for the dataset's characteristics:

Decision Trees (DT): Decision Trees offer a straightforward, interpretable approach to regression, making them a useful starting point for understanding the relationships within our data (Loh, 2011). They allow us to visualize how decisions are made based on the input features, although they may not capture complex relationships as effectively as more sophisticated models.

Random Forest (RF): As an ensemble of Decision Trees, Random Forests mitigate some of the overfitting issues associated with individual trees and are more capable of handling complex, non-linear relationships in regression problems (Breiman, 2001). Their robustness and ability to deal with high-dimensional data make them well-suited for predicting ion concentrations.

Gradient Boosting Machines (GBM): GBMs build models sequentially to correct the residuals of prior models, gradually improving prediction accuracy (Freund and Schapire, 1997). This technique is highly effective for regression, especially when dealing with non-linearity and interactions between predictors, as is common in environmental datasets.

Artificial Neural Networks (ANN): The Artificial Neural Network (ANN), specifically a Multilayer Perceptron (MLP), serves as a powerful tool in machine learning, capable of capturing complex and non-linear relationships within datasets (Carbonell et al., 1983). At its core, an MLP consists of interconnected units called neurons, which are organized into layers: an input layer, several hidden layers, and an output layer. Neurons in these layers interact through connections that are weighted, with the network's learning process revolving around adjusting these weights based on the input data.

Hidden layers are the MLP's workhorse, allowing the model to learn features at different levels of abstraction. The activation function within each neuron determines how signals are transformed as they pass through, introducing non-linearity into the

model and enabling it to learn complex patterns. Common activation functions include ReLU, sigmoid, and tanh, each bringing different properties to the model's learning capability. Additionally, the learning rate is a crucial parameter that governs the step size during the weight update process, influencing the model's convergence speed and accuracy. For optimization, the Adam optimizer is often used due to its efficiency in handling sparse gradients and adapting the learning rate for each network weight, making it highly effective for training deep neural networks.

In this study, the ANN architecture incorporates four hidden layers, designed to provide a deep learning structure capable of accurately predicting ion concentrations from Electrical Conductivity and other predictors. This setup aims to leverage the MLP's general abilities to process and analyze the data, utilizing neurons, activation functions, and an optimized learning rate to enhance model performance. The specific architecture, including the arrangement of neurons and the use of the Adam optimizer, is detailed in Figure 7. A comparative summary among the machine learning models were used in this study was presented Table 5.

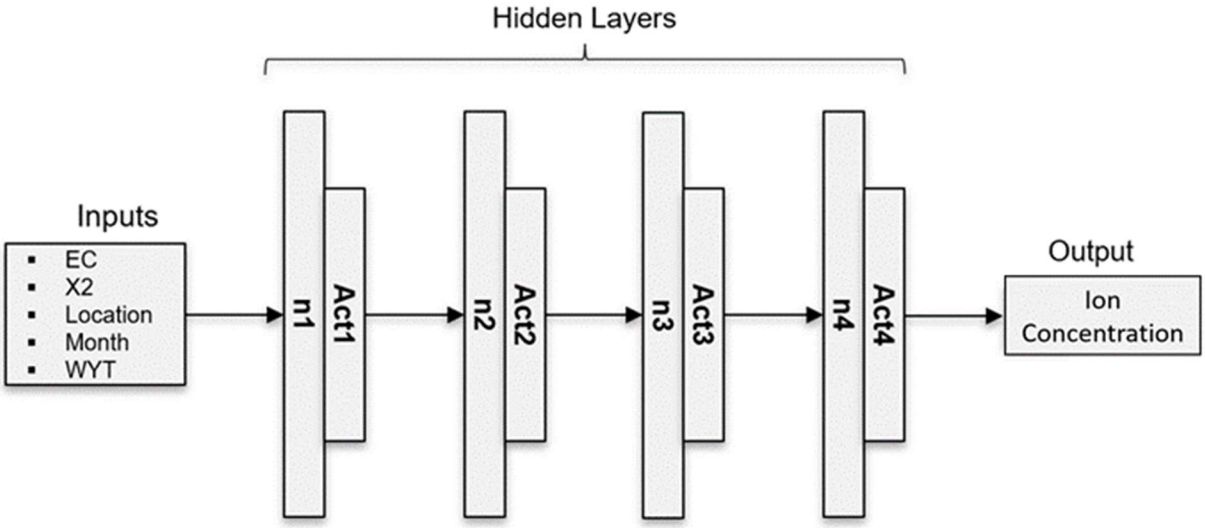

Figure 7. Artificial Neural Network architecture.

Table 5. Comparative Overview of Key Features Across Selected Machine Learning Models.

Feature/Model	Decision Trees	Random Forests	Gradient Boosting	Artificial Neural Networks
Model Type	Tree-based	Ensemble	Ensemble	Neural Network
Basic Unit	Decision Tree	Decision Trees	Weak learners	Neurons
Hidden Layers	None	None	None	One or more
Loss Function	Gini/Entropy	Gini/Entropy	Various	MSE, Cross-Entropy, etc.
Learning Algorithm	ID3, CART, etc.	Bagging	Boosting	Gradient Descent, Adam, etc.
Regularization	Pruning	Voting/Averaging	Shrinkage	Dropout, Weight Decay, etc.
Scalability	Moderate	High	Moderate to High	High
Robustness	Moderate	High	High	Varies
Interpretability	High	Moderate	Low	Low
Speed/Efficiency (Training)	Fast	Moderate	Moderate	Varies
Speed/Efficiency (Inference)	Fast	Fast	Fast	Fast
Applications	Classification, Regression	Classification, Regression, Anomaly Detection	Classification, Regression, Ranking	Classification, Regression, NLP, Image Processing

6.1.5 Model Training and Testing

The strategy for partitioning data for model training and evaluation in this study involves splitting the dataset into training and testing segments, ensuring a robust approach to model validation. Specifically, the input-output datasets are randomly divided, allocating 80% for training purposes and the remaining 20% for testing. This partitioning facilitates the assessment of each machine learning model's predictive accuracy and generalization capability, providing a clear indication of performance on unseen data.

To prevent overfitting—a scenario where a model learns the training data too well, compromising its performance on new data—an early stopping function is incorporated into the training process, with a threshold set at 50 epochs. Early stopping

acts as a form of regularization, effectively halting the training when no improvement in the model's performance on a validation set is observed for a consecutive number of epochs specified by the threshold. This technique ensures that the model maintains its ability to generalize from the training data to unseen data, thus safeguarding against overfitting.

The evaluation and comparison of the machine learning models are grounded in their performance metrics on the test dataset. The models are assessed based on their R-squared (R^2) and Mean Absolute Error (MAE), with a close examination of the convergence between their performance on training and testing datasets to further guard against overfitting. Such scrutiny ensures that the selected models not only exhibit strong predictive accuracy but also demonstrate robust generalization capabilities across different data samples.

This structured approach to data division, model evaluation, and the strategic incorporation of early stopping underscores the meticulousness of the study's methodology. It aims to ensure the development of reliable and generalizable machine learning models for predicting ion concentrations in the Sacramento-San Joaquin Delta, leveraging the strengths of TensorFlow and other advanced analytical techniques to enhance water quality modeling and management decision.

Among the machine learning models evaluated, the Artificial Neural Network (ANN) model demonstrates superior performance in predicting ion concentrations, particularly for ions exhibiting non-linear relationships with Electrical Conductivity (EC). This finding led to the selection of the ANN model for comparison against classical methods.

6.1.6 Model Evaluation

1) Hyperparameter Search

Integrating the selection and optimization of hyperparameters for the Artificial Neural Network (ANN) models is a computationally demanding task, particularly when models incorporate multiple hidden layers. The study's ANN models, which include four hidden layers, required a robust hyperparameter search to fine-tune each model's settings for predicting various ion concentrations accurately.

Hyperparameter search is a pivotal step in machine learning that involves experimenting with different combinations of model settings before training begins. These settings, or hyperparameters, such as the number of hidden layers, neurons in each layer, activation functions, and learning rate, are crucial for defining the model's capacity to learn from data. The optimal hyperparameters are those that contribute to the lowest validation error, indicating the model's effectiveness in making accurate predictions.

Given the complexity of searching hyperparameters for ANN models with four hidden layers, the computational workload exceeds the capabilities of standard desktop computers. To address this challenge, the study leveraged Microsoft Azure cloud computing resources, utilizing an environment equipped with 96 CPU cores. This powerful computing infrastructure significantly accelerated the hyperparameter search process, allowing for a more extensive and rapid exploration of the hyperparameter

space. This expeditious approach was instrumental in identifying the configurations that most effectively minimized prediction error.

The culmination of this search is documented in Table 6, which details the optimal number of neurons and activation functions for each ion constituent model. This optimization enhances the precision of the ANN models, making them reliable tools for water quality assessment in the Sacramento-San Joaquin Delta. The application of cloud computing for hyperparameter optimization exemplifies the integration of advanced computational techniques in environmental modeling, furthering the study's contribution to the field by expediting the development of high-performing ANN models.

The results of the hyperparameter search, including the selected number of neurons (N) and activation functions (Act) for each ion constituent model, are documented. This information is crucial for understanding how the model's architecture is specifically tailored to capture the complex relationships between EC and the various ion constituents.

Incorporating the findings from the hyperparameter search into the ANN models ensures that each model is finely tuned for its specific prediction task. This optimization step is vital for enhancing the model's accuracy, thereby contributing to more precise and reliable water quality predictions in the Sacramento-San Joaquin Delta.

The detailed outcomes of the hyperparameter search, particularly the optimal number of neurons and activation functions chosen for each ion constituent model, are presented in Table 6. This table serves as an essential reference, outlining the tailored configurations that enable the ANN models to achieve their best performance.

Table 6. Optimal number of neurons and activation functions for each ion constituent model.

Hidden Layer	TDS		Mg^{2+}		Na^{+}	
	N	Act	N	Act	N	Act
1	30	elu	30	relu	30	tanh
2	30	sigmoid	30	elu	30	elu
3	30	elu	30	tanh	30	sigmoid
4	30	relu	30	relu	30	elu
Hidden Layer	Ca^{2+}		Cl^{-}		$SO4^{2-}$	
	N	Act	N	Act	N	Act
1	40	elu	30	relu	44	relu
2	40	sigmoid	30	elu	44	relu
3	40	relu	30	simoid	44	relu
4	30	tanh	30	elu	22	relu
Hidden Layer	Br^{-}		Alkalinity		K^{+}	
	N	Act	N	Act	N	Act
1	44	elu	30	tanh	44	relu
2	44	sigmoid	30	relu	44	relu
3	30	elu	30	tanh	44	relu
4	30	tanh	30	elu	22	relu

2) K-fold Cross-validation

The K-fold cross-validation analysis was conducted to rigorously assess and validate the generalization capability of the developed Artificial Neural Network (ANN) models. This method is instrumental in ensuring that the models can accurately predict ion concentrations across different subsets of the dataset, thereby affirming their robustness and reliability.

K-fold cross-validation is a pivotal step in the model evaluation process because it provides a more comprehensive assessment of a model's performance. Unlike a simple train-test split, K-fold cross-validation systematically uses different portions of the data for training and validation. This approach helps mitigate the risk of overfitting and ensures that the model's performance is not dependent on a particular division of the data. It is particularly valuable in scenarios where the available dataset is not extensive, maximizing the use of data for both training and evaluation.

In this study, the dataset was divided into five equal-sized subsets, or folds. The model training and evaluation process was then repeated five times (K=5), with each fold used exactly once as the validation set while the remaining four folds were used for training. This method allowed every data point in the dataset to be used for both training and validation, providing a thorough evaluation of the model's predictive accuracy.

Using a fixed random seed for reproducibility, the subsets were created randomly to ensure each fold was a representative sample of the whole dataset. This randomized division is crucial for minimizing bias and variance in the model evaluation process.

The K-fold cross-validation analysis yielded insightful results, confirming the superior performance of the ANN models across multiple iterations. The models demonstrated consistent accuracy in predicting ion concentrations, as evidenced by the evaluation metrics, R^2 and MAE, calculated for each validation fold.

3) Sensitivity Analysis

Sensitivity analysis is a critical component in model validation, particularly for systems as complex and variable as ion constituent levels in the Delta. Figures 7 and 8 showcase the model’s predictions for Bromide (Br) and Chloride (Cl) concentrations, respectively, across three sub-regions: Old-Middle River (OMR), San Joaquin River Corridor, and South Delta, under specified conditions (April, Wet year, Sacramento X2= 75). These figures highlight the models' ability to adjust predictions in response to changing levels of Electrical Conductivity (EC), an easily-measured parameter in water quality analysis.

The graphs depict a smooth gradient in predicted ion concentrations as EC values vary, reflecting the models' nuanced response to input changes. This is indicative of an effective balance between sensitivity and stability; the models are responsive to EC changes without displaying erratic behavior for minor fluctuations. Such stability in the presence of varied input conditions is essential for reliable predictions in environmental contexts, where data can naturally exhibit variability.

The sensitivity analysis as demonstrated in these figures informs the practical application of these ANN models. For ML model users, the consistency and smoothness in predictions signify that the models can serve as dependable tools for water quality management. These characteristics ensure that the outputs are accurate and interpretable, enabling informed decision-making. The models' generalizability, indicated

by their performance under specific test conditions, is crucial for applications across the diverse ecological landscape of the Delta.

These findings underscore the value of ANNs in environmental modeling. By providing a clear visualization of how predicted ion levels correspond to variations in EC, Figures 8 and 9 affirm the models' capabilities in delivering robust and actionable insights into water quality dynamics across different sub-regions of the Delta.

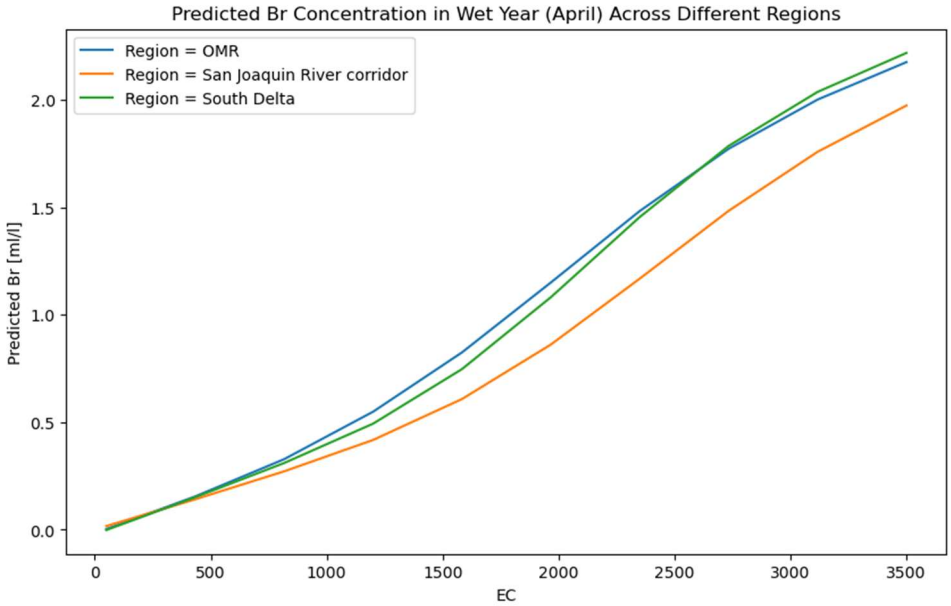

Figure 8. Sensitivity Analysis of Predicted Bromide (Br-) Concentration to EC in the Sacramento-San Joaquin Delta: A Wet Year Scenario in April.

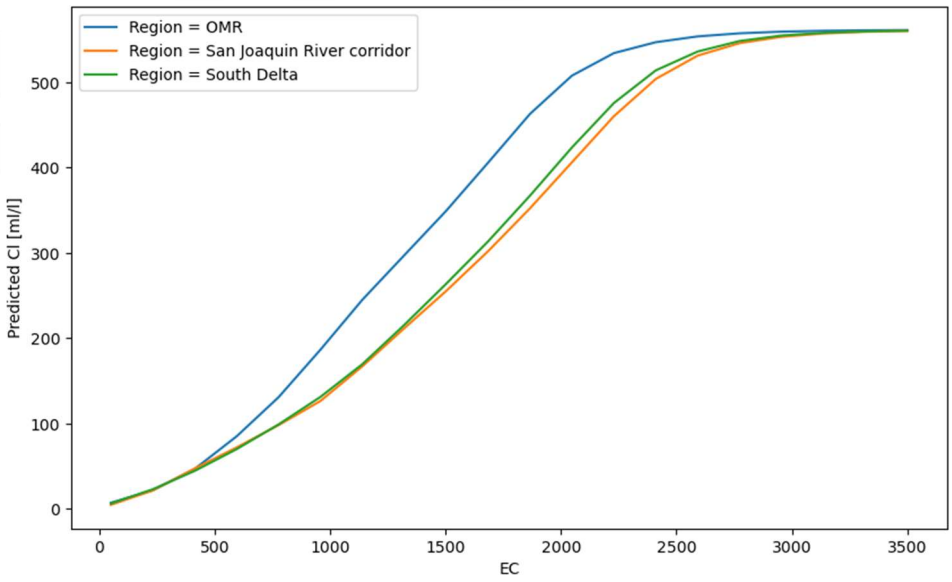

Figure 9. Sensitivity Analysis of Predicted Chloride (Cl-) Concentration to EC in the Sacramento-San Joaquin Delta: A Wet Year Scenario in April.

4) Performance Evaluation

The evaluation of Artificial Neural Network (ANN) models through K-fold cross-validation demonstrates their effectiveness in simulating ion concentrations in the Delta. The cross-validation process, particularly with a 5-fold configuration, reinforces the robust performance of ANNs and signifies an improvement over traditional parametric regression models. The results of this analysis are illustrated in Figure 10, where box plots detail the Mean Absolute Error (MAE) distribution across the ion constituents evaluated through the cross-validation process. A stark comparison is drawn between the ANN models and the benchmark, revealing that ANN models have consistently reduced the MAE, hence improving accuracy in simulating ion concentrations.

The performance of the ANN models is quantitatively superior to that of the traditional parametric regression models, especially notable in the Group 3 ion constituents, which exhibit more complex and non-linear relationships with Electrical Conductivity (EC). The improvements are highlighted in both R-squared (R^2) values and Mean Absolute Error (MAE), with some ions like Total Dissolved Solids (TDS) seeing a significant reduction in MAE by 24% without a corresponding increase in R^2. This demonstrates the ANN models' proficiency in reducing prediction error margins. Figure 11 effectively summarizes these findings, contrasting the performance gains of the ANN models over the benchmark across all ion groups, with Group 3 ions showing particularly marked enhancements in prediction accuracy.

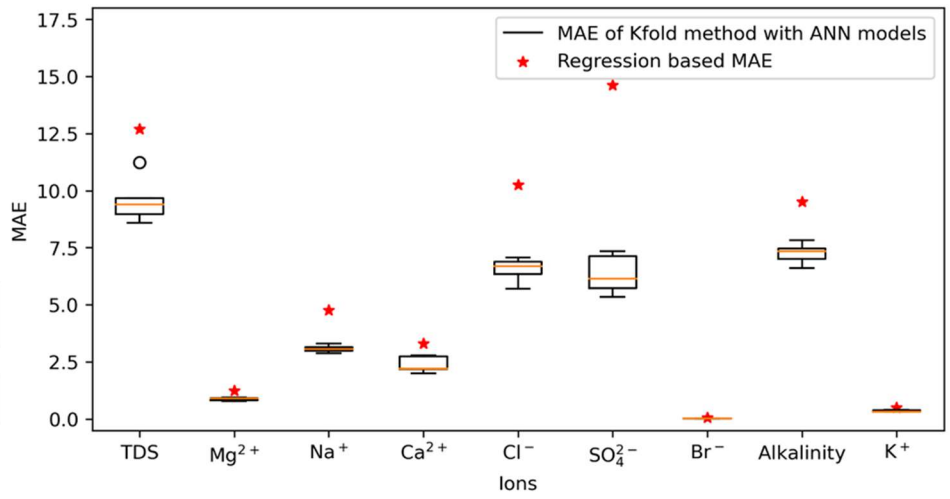

Figure 10. MAE values for the nine ion constituents across the 5-fold cross-validation using the selected ANN models vs MAE of benchmark model.

Figure 11. ANN model performance on simulating the concentrations of nine ion constituents based on percent improvement from the benchmark model represented by (a) R2 and (b) MAE.

6.1.7 Model Deployment and Communication

1) Case Study Product

For users with or without programming knowledge to simulate ion levels in the Delta using aforementioned four ML models, an interactive Delta ion concentration simulation dashboard tool (Figure 12) was developed as the end product of this case study. The dashboard allows users to explore the results of the four ML emulators (RT, RF, GB, and ANN) and the latest parametric regression equations for nine ion constituents based on the selected hydrological conditions. Users can interactively adjust the four predictor variables (EC, Sacramento X2, Month, and WYT) to generate simulations of ion levels from the four ML models and classical method for three sub-regions interior Delta. The pre-trained ML emulators for ion concentration simulation are stored on a GitHub repository, and their functionality is made available through Microsoft Azure. By connecting the Azure server to the GitHub repository, the models are hosted and executed on the Azure server. The dashboard can be assessed at: https://dwrdashion.azurewebsites.net/Dashboard.

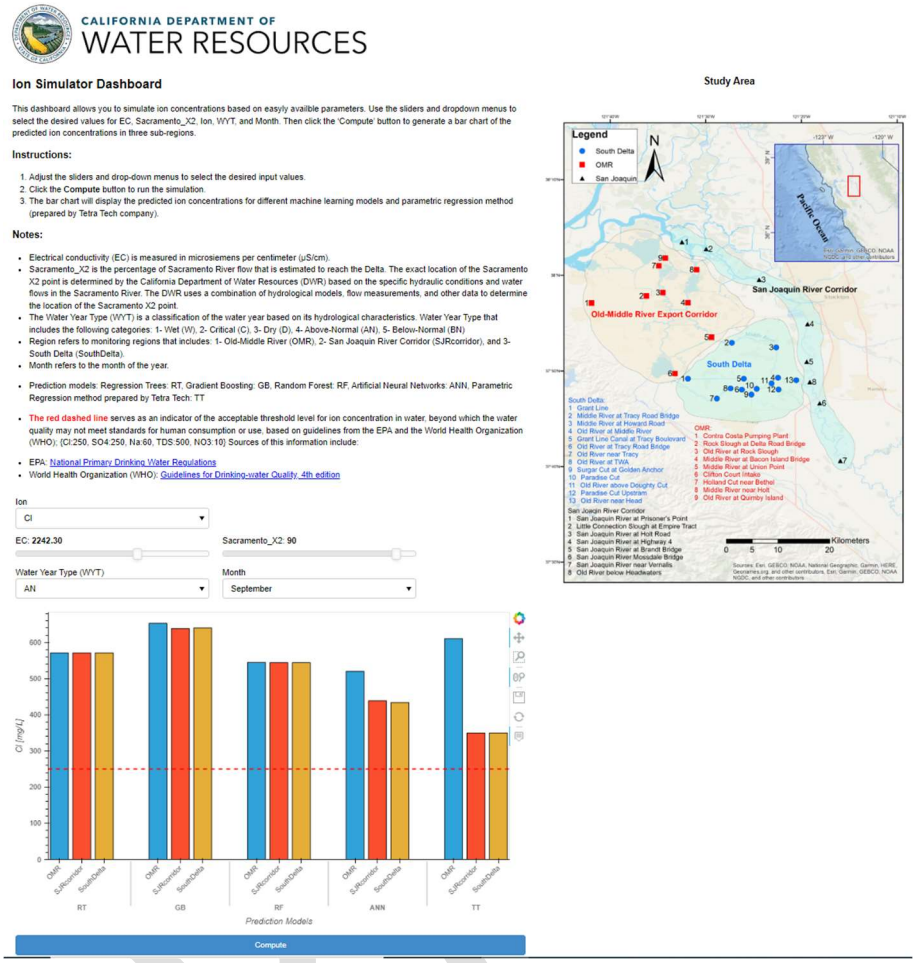

Figure 12. Screenshot of the interactive Delta ion concentration simulation dashboard interface, displaying the results of the four ML models (ANN, RT, RF, and GB) for simulating nine ion constituents in the Delta region.

The dashboard features dropdown menus, sliders, and other interactive elements that allow users to easily customize their queries. For instance, the EC value can be adjusted via a slider, while the location and WYT can be selected from dropdown lists. After the desired inputs are selected, the user can click a ‘Compute’ button, and the ion concentration predictions for each of the four models will be displayed in graphical form, as preferred by the user. The dashboard's interactive features also enable users to adjust input parameters and visualize the outcomes for different hypothetical hydrological conditions, comparing the performance of ANN, RT, RF, and GB models.

2) Case Study Documentation

The methodology and results of the case study are thoroughly documented in the Modeling Support Office’s annual reports submitted to the California State Water Resources Control Board (Namadi et al., 2022a, 2023a). They are also published in two peer-reviewed journal articles (Namadi et al., 2022b, 2023b). The data and trained ML models of the study are stored in a Github repository:

https://github.com/PeymanHNamadi/Ion_Study_Dashboard/tree/main. The repository is in the public domain.

3) Case Study Communication

The study was supported by the Municipal Water Quality Investigations (MWQI) program (Hutton et al., 2022). Both the study and the dashboard were presented during MWQI stakeholder gatherings. The dashboard was enhanced to integrate feedback from all stakeholders. Additionally, the study was presented shared at the annual meetings of the California Water and Environmental Modeling Forum (CWEMF). This organization is dedicated to enhancing the applicability of models in addressing water and environmental issues in California. We are planning to provide hands-on training for the dashboard in the near future.

6.2 Best Practices

6.2.1 Best Practices for Data Collection

Effective data collection is indispensable for advancing ML model development in water and environmental modeling. By collecting diverse, high-quality datasets and adhering to best practices for data collection, developers can build robust ML models that provide valuable insights into water and environmental processes and thus inform decision-making. To ensure the quality and reliability of collected data, it is essential to adhere to best practices throughout the data collection process. The following outlines some typical practices:

1) Use Standardized Protocols: Adhere to standardized protocols and methodologies for data collection, measurement, model simulation, and sampling to ensure consistency and comparability across datasets.

2) Implement Quality Assurance/Quality Control (QA/QC) Measures: Implement rigorous QA/QC procedures to identify and rectify errors, outliers, and inconsistencies in the collected data, including calibration of instruments, duplicate measurements, and cross-validation with reference data.

3) Select Representative Data: Choose sampling locations that are representative of the spatial and temporal variability of the water and environmental processes under study, taking into account factors such as study purpose, study period, and data availability, etc.

4) Data Accessibility and Documentation: Document metadata, data sources, and collection methods to facilitate data sharing, reproducibility, and transparency.

It is important to highlight that California's Open and Transparent Water Data Act (AB 1755) directs State agencies to set up, oversee, and maintain an extensive water and ecological data platform statewide (https://water.ca.gov/ab1755). The agencies in collaboration with the California Water Data Consortium are developing guidelines and standards for data sharing, ensuring proper documentation and quality control, and facilitating public access. To date, a federated AB 1755 open data has been developed as described in the Open and Transparent Water Data Act – Implementation Journal. Leveraging the existing and anticipated results of AB 1755 can greatly enhance and

streamline ML model development and applications in water and environmental modeling in California.

6.2.2 Best Practices for Model Selection

Selecting the appropriate ML architecture is crucial for developing effective models in water and environmental modeling. Based on previous ML model development experience in the Modeling Support Office, we outline several best practices for ML architecture selection, ranging from starting with simple models to utilizing ensemble and hybrid methods, tailored to the unique challenges and requirements of environmental applications.

1) Start Simple:

Begin with simple ML models, such as linear regression or decision trees, to establish a baseline understanding of the data and the problem at hand. Simple models are easier to interpret and can provide valuable insights into the relationships between input variables and the target variable.

2) Ensemble Methods:

Utilize ensemble methods, such as random forests or gradient boosting, to improve model performance and robustness. Ensemble methods combine multiple base models to produce a stronger learner, often achieving higher accuracy and better generalization. In water and environmental modeling, ensemble methods can help capture complex relationships and enhance predictive capabilities.

3) Hybrid Methods:

Explore hybrid methods that integrate multiple ML architectures or combine ML with traditional modeling approaches. Hybrid methods leverage the strengths of different techniques to address specific challenges in water and environmental modeling. For example, combining physics-based models with data-driven ML models can improve predictive accuracy while retaining interpretability and physical realism.

4) Consider Spatial and Temporal Dynamics:

Account for the spatial and temporal dynamics inherent in environmental data when selecting ML architectures. Convolutional neural networks (CNNs) are well-suited for spatial data analysis, capturing spatial patterns and relationships. Recurrent neural networks (RNNs) and long short-term memory (LSTM) networks excel at modeling temporal dependencies, making them suitable for time-series data in environmental modeling.

5) Incorporate Domain Knowledge:

Integrate domain knowledge and expert insights into ML architecture selection to ensure the models capture relevant environmental processes accurately. Domain-specific constraints, physical laws, and expert understanding can guide the choice of architectures and enhance the relevance and applicability of the models.

6.2.3 Best Practices for Model Training and Validation

Training and assessing ML models in water and environmental modeling necessitates meticulous attention to multiple factors to guarantee their resilience, dependability, and precision. Drawing from prior experience, we deem the subsequent practices indispensable:

1) Regularization Techniques:

Regularization methods, such as L1 and L2 regularization, help prevent overfitting by penalizing overly complex models. In water and environmental modeling, where datasets may be limited or noisy, regularization techniques ensure that models generalize well to unseen data. Implement regularization to balance model complexity and performance, enhancing the model's reliability and interpretability.

2) Ensemble Learning:

Ensemble learning techniques combine multiple models to improve predictive performance and robustness. In water and environmental modeling, where uncertainties and complexities are prevalent, ensemble methods offer enhanced predictive capabilities. Utilize ensemble techniques like bagging, boosting, or stacking to leverage the diversity of individual models and achieve more accurate predictions across different environmental scenarios.

3) Cross-Validation:

Cross-validation is essential for assessing model performance and generalizability, particularly in water and environmental modeling tasks where data may exhibit temporal or spatial dependencies. Employ techniques such as k-fold cross-validation or stratified cross-validation to partition the data into training and validation sets. Cross-validation helps mitigate overfitting and provides reliable estimates of model performance across diverse environmental conditions.

4) Hyperparameter Optimization:

Hyperparameter optimization involves systematically tuning model parameters to maximize performance while avoiding overfitting or underfitting. In water and environmental modeling, where model complexity may vary based on the intricacies of environmental processes, hyperparameter optimization is crucial for fine-tuning model performance. Experiment with techniques such as grid search, randomized search, or Bayesian optimization to identify the optimal hyperparameter settings for ML models. These best practices can contribute to the advancement of ML applications in environmental management and decision-making, facilitating sustainable solutions for water resource management and environmental protection.

6.2.4 Best Practices for Model Deployment

Deploying and effectively communicating ML models in water and environmental modeling are pivotal for their successful integration into decision-making processes. Numerous best practices exist to guarantee seamless ML model deployment and transparent communication. Deploying and communicating ML models effectively is imperative for their successful application in water and environmental modeling. By adhering to best practices such as comprehensive documentation, hands-on training, open-source collaboration, engagement with the ML community, and publication in peer-reviewed journals, model developers can ensure transparency, collaboration, and innovation throughout the model development and application process. These practices facilitate informed decision-making, enhance stakeholder engagement, and contribute to sustainable management of water resources and environmental conservation efforts.

The following delineates some of these practices we consider essential, drawing from past experiences.

1) Comprehensive Documentation:

Thorough documentation is paramount for understanding and replicating ML model deployment. Document the model architecture, data preprocessing steps, hyperparameters, and assumptions comprehensively. Include instructions for deployment, usage, and troubleshooting to aid users in effectively utilizing the ML model. Transparent documentation promotes collaboration and facilitates knowledge transfer among stakeholders.

2) Open Source Collaboration:

Consider open-sourcing the ML model code and associated documentation to encourage collaboration and transparency. Sharing code on open-source platforms like GitHub enables peer review, feedback, and contributions from the wider community. Open-source collaboration fosters innovation, accelerates model development, and ensures reproducibility in water and environmental modeling.

3) Hands-On Training:

Conduct hands-on training sessions to educate users on model deployment, interpretation, and application. These interactive sessions provide practical experience and guidance on utilizing the ML model in real-world scenarios. Customize training materials for different user groups, such as engineers, scientists, managers, and stakeholders, to ensure relevance and effectiveness.

4) ML Community Engagement:

Engage with the ML community through brown bag meetings and user group meetings dedicated to water and environmental modeling. These gatherings provide platforms for knowledge sharing, collaboration, and networking among stakeholders. Encourage participation from diverse backgrounds to foster interdisciplinary collaboration and address complex environmental challenges collectively.

5) Peer-Reviewed Journal Articles:

Publish research findings and methodologies in peer-reviewed journal articles to disseminate outcomes and contribute to the scientific community. Peer-reviewed publications lend credibility and validation to ML model development and application in water and environmental modeling. Share insights, challenges, and lessons learned to advance the field and promote transparency in model deployment and communication.

7. Summary and Future Directions

The protocols presented in this document outline a step-by-step guide for using machine learning in water and environmental modeling. The core of the protocols lies in careful pre-modeling considerations. This includes clearly defining the environmental problem one is trying to solve and collecting high-quality data. The protocols then delve into the model development process, from data pre-processing techniques to choosing the most suitable machine learning algorithm for the specific task.

Following development comes a crucial evaluation and interpretation phase. The protocols discuss methods to assess the model's performance and ensure its reliability. They also emphasize the importance of interpreting the model's results to gain insights into the environmental processes at play. Finally, the protocols cover the deployment and communication stages. They outline strategies for integrating the model into real-world applications and effectively communicating its results to ML model users. The

document further includes a case study demonstrating each of these steps, along with best practices. By following these protocols and best practices, water modelers and managers can leverage the power of machine learning to tackle complex water and environmental challenges.

This document intends to serve as a dynamic roadmap for harnessing the power of machine learning in California's water and environmental modeling endeavors. Conceived as a living document, it will be continuously adapted and expanded to address the state's evolving challenges in water resources and environmental protection.

First, as California grapples with new and ever-shifting water and environmental issues, the protocols will be tailored to tackle emerging threats and opportunities. This might involve incorporating novel data sources like real-time sensor networks or high-resolution satellite imagery which are not addressed specifically in the current document. The methodologies will be refined to address specific problems, such as developing more robust seasonal water supply forecasting models that account for the complexities of climate change. Additionally, communication strategies will be honed to effectively reach a wider range of stakeholders, encompassing policymakers, resource managers, and the public.

The second front for advancement lies within the burgeoning field of machine learning itself. The emergence of groundbreaking techniques like physics-informed neural networks (PINNs), digital twins (DT), and generative artificial intelligence (GenAI) presents immense potential for water and environmental modeling. Specifically, PINNs leverage domain-specific knowledge and physical principles to guide model training, enhancing the interpretability, accuracy, and generalizability of machine learning models in environmental applications (e.g., He and Tartakovsky, 2021; Karniadakis et al., 2021; Cuomo et al., 2022; Yu et al. 2022; Roh et al., 2023; Song et al., 2023). By integrating physics-based constraints into neural network architectures, we can develop more robust and reliable models capable of capturing the complex interplay between environmental variables and processes. DTs present an exciting opportunity to develop more holistic and dynamic representations of environmental systems (e.g., Li et al., 2023;Brocca et al., 2024). By coupling real-time sensor data with physics-based models and machine learning algorithms, DTs offer a comprehensive platform for simulating and optimizing environmental processes, thereby facilitating proactive and data-driven decision-making in water resource management and environmental conservation efforts. This document is expected to be expanded in the future to include guidance on the creation and utilization of PINNs and DTs specifically tailored for water and environmental modeling within California.

GenAI has the potential to revolutionize environmental modeling by enabling the synthesis of realistic and diverse environmental scenarios, thus enhancing the robustness and adaptability of our models to unforeseen circumstances and novel challenges. In 2023, the Governor of California issued Executive Order (EO) N-12-23 to explore the development, application, and associated risks of AI technology, particularly GenAI, across the state. This initiative aims to establish a cautious and responsible approach to assessing and implementing AI within state governance, recognizing its significant impact. The first milestone of this order is a report assessing the benefits and risks of GenAI for the state, focusing on improving access to essential services while

addressing concerns such as security vulnerabilities and potential impacts on public health, safety, and the economy. Recently, a report summarizing procurement, usage, and training guidelines for integrating GenAI into California state government operations was released. These guidelines offer best practices and criteria for safe and effective utilization of this innovative technology, with the final procurement and training policy expected in 2025. The protocols outlined in the current document are not tailored specifically for GenAI. However, the guidelines and best practices on GenAI resulting from the EO N-12-23 will be adapted and integrated into future iterations of this document.

Acknowledgements

This work was initiated and guided by Nicky Sandhu (Modeling Support Office (MSO)). Minxue (Kevin) He (MSO) drafted the document. Peyman Namadi (MSO) helped prepare the case study presented in the document. Francis Chung and Kamyar Guivetchi (Exectuvie Division) provided insightful comments on earlier versions of the document that largely helped improve the quality of the document. During the preparation of this document, the authors used ChatGPT to check grammar and improve language. After using the tool, the authors reviewed and edited the content as needed and took full responsibility for the content of the document. All opinions expressed in this document belong solely to the authors and do not reflect those of the Department of Water Resources.

References

Abraham, M. T., Satyam, N., Jain, P., Pradhan, B., & Alamri, A. (2021). Effect of spatial resolution and data splitting on landslide susceptibility mapping using different machine learning algorithms. Geomatics, Natural Hazards and Risk, 12(1), 3381-3408.

Anjum, R., Parvin, F., & Ali, S. A. (2023). Machine Learning Applications in Sustainable Water Resource Management: A Systematic Review. Emerging Technologies for Water Supply, Conservation and Management, 29-47.

Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., ... & Herrera, F. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information fusion, 58, 82-115.

Barnes Jr, G. W., & Chung, F. I. (1986). Operational planning for California water system. Journal of Water Resources Planning and Management, 112(1), 71-86.

Barr, A., Feigenbaum, E. A., & Cohen, P. R. (Eds.). (1981). The handbook of artificial intelligence (Vol. 1). HeurisTech Press.

Başağaoğlu, H., Chakraborty, D., Lago, C. D., Gutierrez, L., Şahinli, M. A., Giacomoni, M., ... & Şengör, S. S. (2022). A review on interpretable and explainable artificial intelligence in hydroclimatic applications. Water, 14(8), 1230.

Bay-Delta Modeling Forum (BDMF) (2000). Protocols for Water and Environmental Modeling. Sacramento, CA. P.47.

Bellman, R. (1966). Dynamic programming. Science, 153(3731), 34-37.

Black, D. C., Wallbrink, P. J., & Jordan, P. W. (2014). Towards best practice implementation and application of models for analysis of water resources management scenarios. Environmental Modelling & Software, 52, 136-148.

Breiman, L. (2001). Random forests. Machine learning, 45, 5-32.

Brocca, L., Barbetta, S., Camici, S., Ciabatta, L., Dari, J., Filippucci, P., ... & Fernandez, D. (2024). A Digital Twin of the terrestrial water cycle: a glimpse into the future through high-resolution Earth observations. Frontiers in Science, 1, 1190191.

California Department of Water Resources (DWR; formerly the Division of Water Resources) (1931), Variation and Control of Salinity, California Division of Water Resources Bulletin 27.

California Government Operations Agency (GovOps) (2023). State of California Benefits and Risks of Generative Artificial Intelligence Report, Sacramento, CA, pp. 33.

California Government Operations Agency (GovOps) (2024). State of California GenAI Guidelines for Public Sector Procurement, Uses and Training, Sacramento, CA, pp. 16.

California Department of Water Resources (DWR) (1991). Calibration and verification of DWRDSM. Methodology for Flow and Salinity Estimates in the Sacramento-San Joaquin Delta and Suisun Marsh: 12th Annual Progress Report.

California Water & Environmental Modeling Forum (CWEMF) (2021). Protocols for Water and Environmental Modeling. Sacramento, CA. P.119.

Carbonell, J. G., Michalski, R. S., & Mitchell, T. M. (1983). An overview of machine learning. Machine learning, 3-23.

Chen, L., Han, B., Wang, X., Zhao, J., Yang, W., & Yang, Z. (2023). Machine learning methods in weather and climate applications: A survey. Applied Sciences, 13(21), 12019.

Chen, L., Roy, S. B., & Hutton, P. H. (2018). Emulation of a process-based estuarine hydrodynamic model. Hydrological Sciences Journal, 63(5), 783-802.

Chen, X. W., & Jeong, J. C. (2007). Enhanced recursive feature elimination. In Sixth international conference on machine learning and applications (ICMLA 2007) (pp. 429-435). IEEE.

Chen, Y., Song, L., Liu, Y., Yang, L., & Li, D. (2020). A review of the artificial neural network models for water quality prediction. Applied Sciences, 10(17), 5776.

Cui, S., Gao, Y., Huang, Y., Shen, L., Zhao, Q., Pan, Y., & Zhuang, S. (2023). Advances and applications of machine learning and deep learning in environmental ecology and health. Environmental Pollution, 122358.

Cuomo, S., Di Cola, V. S., Giampaolo, F., Rozza, G., Raissi, M., & Piccialli, F. (2022). Scientific machine learning through physics–informed neural networks: Where we are and what's next. Journal of Scientific Computing, 92(3), 88.

de Burgh-Day, C. O., & Leeuwenburg, T. (2023). Machine learning for numerical weather and climate modelling: a review. Geoscientific Model Development, 16(22), 6433-6477.

Diamantidis, N. A., Karlis, D., & Giakoumakis, E. A. (2000). Unsupervised stratification of cross-validation for accuracy estimation. Artificial Intelligence, 116(1-2), 1-16.

Dixon, P. M. (2006). Bootstrap resampling. Encyclopedia of Environmetrics.

Dogo, E. M., Nwulu, N. I., Twala, B., & Aigbavboa, C. (2019). A survey of machine learning methods applied to anomaly detection on drinking-water quality data. Urban Water Journal, 16(3), 235-248.

Draper, A. J., Munévar, A., Arora, S. K., Reyes, E., Parker, N. L., Chung, F. I., & Peterson, L. E. (2004). CalSim: Generalized model for reservoir system analysis. Journal of Water Resources Planning and Management, 130(6), 480-489.

Elman, J. L. (1990). Finding structure in time. Cognitive science, 14(2), 179-211.

Epstein, Z. and Hertzmann A. (2023). Art and the science of generative AI. Science 380,1110-1111(2023). DOI:10.1126/science.adh4451

Erdogan Erten, G., Yavuz, M., & Deutsch, C. V. (2022). Combination of machine learning and kriging for spatial estimation of geological attributes. Natural Resources Research, 31(1), 191-213.

Feigl, M., Lebiedzinski, K., Herrnegger, M., & Schulz, K. (2021). Machine-learning methods for stream water temperature prediction. Hydrology and Earth System Sciences, 25(5), 2951-2951.

Fodor, I. K. (2002). A survey of dimension reduction techniques (No. UCRL-ID-148494). Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States).

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. Journal of computer and system sciences, 55(1), 119-139.

Ghazvinian, M., Zhang, Y., Seo, D. J., He, M., & Fernando, N. (2021). A novel hybrid artificial neural network-Parametric scheme for postprocessing medium-range precipitation forecasts. Advances in Water Resources, 151, 103907.

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., & Kagal, L. (2018, October). Explaining explanations: An overview of interpretability of machine learning. In 2018 IEEE 5th International Conference on data science and advanced analytics (DSAA) (pp. 80-89). IEEE.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. Advances in neural information processing systems, 27.

Gupta, H. V., Kling, H., Yilmaz, K. K., & Martinez, G. F. (2009). Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling. Journal of hydrology, 377(1-2), 80-91.

Gupta, H. V., Sorooshian, S., & Yapo, P. O. (1999). Status of automatic calibration for hydrologic models: Comparison with multilevel expert calibration. Journal of hydrologic engineering, 4(2), 135-143.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. Journal of machine learning research, 3(Mar), 1157-1182.

Hannaford, J. F. (1956). Multiple-graphical correlation for water supply forecasting, 24th Annual Western Snow Conference, April 1956, Penticton, British Columbia (https://westernsnowconference.org/node/1134)

Hart, D., and Gehrke F. (1990). Status of the California cooperative snow survey program, 58th Annual Western Snow Conference, April 1990, Sacramento, California (https://westernsnowconference.org/node/607)

Hassoun, M. H. (1995). Fundamentals of artificial neural networks. MIT press.

He, Q., & Tartakovsky, A. M. (2021). Physics-Informed neural network method for forward and backward advection-dispersion equations. Water Resources Research, 57(7).

He, M., Zhong, L., Sandhu, P., & Zhou, Y. (2020). Emulation of a Process-Based Salinity Generator for the Sacramento–San Joaquin Delta of California via Deep Learning. Water, 12(8), 2088.

Huang, R., Ma, C., Ma, J., Huangfu, X., & He, Q. (2021). Machine learning in natural and engineered water systems. Water Research, 205, 117666.

Jayasundara, N. C., Seneviratne, S. A., Reyes, E., & Chung, F. I. (2020). Artificial neural network for Sacramento–San Joaquin Delta flow–salinity relationship for CalSim 3.0. Journal of Water Resources Planning and Management, 146(4), 04020015.

Jones, A., Kuehnert, J., Fraccaro, P., Meuriot, O., Ishikawa, T., Edwards, B., ... & Assefa, S. (2023). AI for climate impacts: applications in flood risk. npj Climate and Atmospheric Science, 6(1), 63.

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. Science, 349(6245), 255-260.

Kaack, L. H., Donti, P. L., Strubell, E., Kamiya, G., Creutzig, F., & Rolnick, D. (2022). Aligning artificial intelligence with climate change mitigation. Nature Climate Change, 12(6), 518-527.

Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., & Yang, L. (2021). Physics-informed machine learning. Nature Reviews Physics, 3(6), 422-440.

Karim, F., Armin, M. A., Ahmedt-Aristizabal, D., Tychsen-Smith, L., & Petersson, L. (2023). A review of hydrodynamic and machine learning approaches for flood inundation modeling. Water, 15(3), 566.

Kim, H. S., He, M., & Sandhu, P. (2022). Suspended sediment concentration estimation in the Sacramento-San Joaquin Delta of California using long short-term memory networks. Hydrological Processes, 36(10), e14694.

Kumar, V., Azamathulla, H. M., Sharma, K. V., Mehta, D. J., & Maharaj, K. T. (2023). The state of the art in deep learning applications, challenges, and future prospects: A comprehensive review of flood forecasting and management. Sustainability, 15(13), 10543.

Lamrini, B., Lakhal, E. K., Le Lann, M. V., & Wehenkel, L. (2011). Data validation and missing data reconstruction using self-organizing map for water treatment. Neural Computing and Applications, 20, 575-588.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.

LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., & Jackel, L.

(1989). Handwritten digit recognition with a back-propagation network. Advances in neural information processing systems, 2.

Lei, L., Pang, R., Han, Z., Wu, D., Xie, B., & Su, Y. (2023). Current applications and future impact of machine learning in emerging contaminants: a review. Critical Reviews in Environmental Science and Technology, 53(20), 1817-1835.

Li, J., Heap, A. D., Potter, A., & Daniell, J. J. (2011). Application of machine learning methods to spatial interpolation of environmental variables. Environmental Modelling & Software, 26(12), 1647-1659.

Li, X., Feng, M., Ran, Y., Su, Y., Liu, F., Huang, C., ... & Guo, H. (2023). Big Data in Earth system science and progress towards a digital twin. Nature Reviews Earth & Environment, 4(5), 319-332.

Linardatos, P., Papastefanopoulos, V., & Kotsiantis, S. (2020). Explainable ai: A review of machine learning interpretability methods. Entropy, 23(1), 18.

Lloyd, S. (1982). Least squares quantization in PCM. IEEE transactions on information theory, 28(2), 129-137.

Loh, W. Y. (2011). Classification and regression trees. Wiley interdisciplinary reviews: data mining and knowledge discovery, 1(1), 14-23.

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. Advances in neural information processing systems, 30.

Maier, H. R., & Dandy, G. C. (2000). Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications. Environmental modelling & software, 15(1), 101-124.

Maier, H. R., Galelli, S., Razavi, S., Castelletti, A., Rizzoli, A., Athanasiadis, I. N., ... & Humphrey, G. B. (2023). Exploding the myths: An introduction to artificial neural networks for prediction and forecasting. Environmental modelling & software, 105776.

Moriasi, D. N., Arnold, J. G., Van Liew, M. W., Bingner, R. L., Harmel, R. D., & Veith, T. L. (2007). Model evaluation guidelines for systematic quantification of accuracy in watershed simulations. Transactions of the ASABE, 50(3), 885-900.

Mosavi, A., Ozturk, P., & Chau, K. W. (2018). Flood prediction using machine learning models: Literature review. Water, 10(11), 1536.

Mosebo Fernandes, A. C., Quintero Gonzalez, R., Lenihan-Clarke, M. A., Leslie Trotter, E. F., & Jokar Arsanjani, J. (2020). Machine learning for conservation planning in a changing climate. Sustainability, 12(18), 7657.

Muñoz-Carpena, R., Carmona-Cabrero, A., Yu, Z., Fox, G., & Batelaan, O. (2023). Convergence of mechanistic modeling and artificial intelligence in hydrologic science and engineering. PLOS Water, 2(8), e0000059.

Namadi, P., He, M., & Sandhu, P. (2022). Salinity-constituent conversion in South Sacramento-San Joaquin Delta of California via machine learning. Earth Science Informatics, 15(3), 1749-1764.

Namadi, P., He, M., & Sandhu, P. (2023). Modeling ion constituents in the Sacramento-San Joaquin Delta using multiple machine learning approaches. Journal of Hydroinformatics, 25(6), 2541-2560.

Nearing, G., Cohen, D., Dube, V., Gauch, M., Gilon, O., Harrigan, S., ... & Matias, Y. (2024). Global prediction of extreme floods in ungauged watersheds. Nature, 627(8004), 559-563.

Niroumand-Jadidi, M., Legleiter, C. J., & Bovolo, F. (2022). River bathymetry retrieval from Landsat-9 images based on neural networks and comparison to SuperDove and Sentinel-2. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 15, 5250-5260.

Ng, K. W., Huang, Y. F., Koo, C. H., Chong, K. L., El-Shafie, A., & Ahmed, A. N. (2023). A review of hybrid deep learning applications for streamflow forecasting. Journal of Hydrology, 130141.

Nguyen, X. C., Bui, V. K. H., Cho, K. H., & Hur, J. (2023). Practical application of machine learning for organic matter and harmful algal blooms in freshwater systems: A review. Critical Reviews in Environmental Science and Technology, 1-23.

Osman, M. S., Abu-Mahfouz, A. M., & Page, P. R. (2018). A survey on data imputation techniques: Water distribution system as a use case. IEEE Access, 6, 63279-63291.

Park, J., Lee, W. H., Kim, K. T., Park, C. Y., Lee, S., & Heo, T. Y. (2022). Interpretation of ensemble learning to predict water quality using explainable artificial intelligence. Science of the Total Environment, 832, 155070.

Patro, S. G. O. P. A. L., & Sahu, K. K. (2015). Normalization: A preprocessing stage. arXiv preprint arXiv:1503.06462.

Pereira, G. W., Valente, D. S. M., de Queiroz, D. M., Santos, N. T., & Fernandes-Filho, E. I. (2022). Soil mapping for precision agriculture using support vector machines combined with inverse distance weighting. Precision Agriculture, 23(4), 1189-1204.

Pichler, Maximilian, et al. "Machine learning algorithms to infer trait-matching and predict species interactions in ecological networks." Methods in Ecology and Evolution 11.2 (2020): 281-293.

Pollard, J. A., Spencer, T., & Jude, S. (2018). Big Data Approaches for coastal flood risk assessment and emergency response. Wiley Interdisciplinary Reviews: Climate Change, 9(5), e543.

Prodhan, F. A., Zhang, J., Hasan, S. S., Sharma, T. P. P., & Mohana, H. P. (2022). A review of machine learning methods for drought hazard monitoring and forecasting: Current research trends, challenges, and future research directions. Environmental modelling & software, 149, 105327.

Qi, S. Y., Bai, Z. J., Ding, Z., Jayasundara, N., He, M. X., Sandhu, P., ... & Kadir, T. (2021). Enhanced artificial neural networks for salinity estimation and forecasting in the Sacramento-San Joaquin delta of California.

Qi, S., He, M., Bai, Z., Ding, Z., Sandhu, P., Chung, F., ... & Roh, D. M. (2022a). Novel Salinity Modeling Using Deep Learning for the Sacramento–San Joaquin Delta of California. Water (20734441), 14(22).

Qi, S., He, M., Bai, Z., Ding, Z., Sandhu, P., Zhou, Y., ... & Anderson, J. (2022b). Multi-location emulation of a process-based salinity model using machine learning. Water, 14(13), 2030.

Qi, S., He, M., Hoang, R., Zhou, Y., Namadi, P., Bradley, T., ... & Huynh, V. (2023). Salinity Modeling Using Deep Learning with Data Augmentation and Transfer Learning. Water, 15(13), 2482.

Rajaee, T., Ebrahimi, H., & Nourani, V. (2019). A review of the artificial intelligence methods in groundwater level modeling. Journal of hydrology, 572, 336-351.

Ramirez, S. G., Hales, R. C., Williams, G. P., & Jones, N. L. (2022). Extending SC-PDSI-PM with neural network regression using GLDAS data and Permutation Feature Importance. Environmental Modelling & Software, 157, 105475.

Raschka, S. (2018). Model evaluation, model selection, and algorithm selection in machine learning. arXiv preprint arXiv:1811.12808.

Rath, J. S., Hutton, P. H., Chen, L., & Roy, S. B. (2017). A hybrid empirical-Bayesian artificial neural network model of salinity in the San Francisco Bay-Delta estuary. Environmental Modelling & Software, 93, 193-208.

Razavi, S. (2021). Deep learning, explained: Fundamentals, explainability, and bridgeability to process-based modelling. Environmental Modelling & Software, 144, 105159.

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., & Prabhat, F. (2019). Deep learning and process understanding for data-driven Earth system science. Nature, 566(7743), 195-204.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016, August). " Why should I trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1135-1144).

Roberts, D. R., Bahn, V., Ciuti, S., Boyce, M. S., Elith, J., Guillera-Arroita, G., ... & Dormann, C. F. (2017). Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. Ecography, 40(8), 913-929.

Rodriguez, J. D., Perez, A., & Lozano, J. A. (2009). Sensitivity analysis of k-fold cross validation in prediction error estimation. IEEE transactions on pattern analysis and machine intelligence, 32(3), 569-575.

Roh, D. M., He, M., Bai, Z., Sandhu, P., Chung, F., Ding, Z., ... & Anderson, J. (2023). Physics-Informed Neural Networks-Based Salinity Modeling in the Sacramento–San Joaquin Delta of California. Water, 15(13), 2320.

Rolnick, D., Donti, P. L., Kaack, L. H., Kochanski, K., Lacoste, A., Sankaran, K., ... & Bengio, Y. (2022). Tackling climate change with machine learning. ACM Computing Surveys (CSUR), 55(2), 1-96.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review, 65(6), 386.

Saygin, A.P., Cicekli, I., & Akman, V. (2000). Turing test: 50 years later. Minds and machines, 10(4), 463-518.

Shen, C. (2018). A transdisciplinary review of deep learning research and its relevance for water resources scientists. Water Resources Research, 54(11), 8558-8593.

Shen, C., Appling, A. P., Gentine, P., Bandai, T., Gupta, H., Tartakovsky, A., ... & Lawson, K. (2023). Differentiable modelling to unify machine learning and physical models for geosciences. Nature Reviews Earth & Environment, 4(8), 552-567.

Sit, M., Demiray, B. Z., Xiang, Z., Ewing, G. J., Sermet, Y., & Demir, I. (2020). A comprehensive review of deep learning applications in hydrology and water resources. Water Science and Technology, 82(12), 2635-2670.

Slater, L. J., Arnal, L., Boucher, M. A., Chang, A. Y. Y., Moulds, S., Murphy, C., ... & Zappa, M. (2023). Hybrid forecasting: blending climate predictions with AI models. Hydrology and Earth System Sciences, 27(9), 1865-1889.

Song, Y., Knoben, W. J., Clark, M. P., Feng, D., Lawson, K. E., & Shen, C. (2023). When ancient numerical demons meet physics-informed machine learning: adjoint-based gradients for implicit differentiable modeling. Hydrology and Earth System Sciences Discussions, 2023, 1-35.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. Journal of the royal statistical society: Series B (Methodological), 36(2), 111-133.

Sun, A. Y., & Scanlon, B. R. (2019). How can Big Data and machine learning benefit environment and water management: a survey of methods, applications, and future directions. Environmental Research Letters, 14(7), 073001.

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. Machine learning, 3, 9-44.

Tahmasebi, P., Kamrava, S., Bai, T., & Sahimi, M. (2020). Machine learning in geo-and environmental sciences: From small to large scale. Advances in Water Resources, 142, 103619.

Tait, A., & Woods, R. (2007). Spatial interpolation of daily potential evapotranspiration for New Zealand using a spline model. Journal of Hydrometeorology, 8(3), 430-438.

Tongal, H., & Booij, M. J. (2018). Simulation and forecasting of streamflows using machine learning models coupled with base flow separation. Journal of hydrology, 564, 266-282.

Tripathy, K. P., & Mishra, A. K. (2023). Deep learning in hydrology and water resources disciplines: Concepts, methods, applications, and research directions. Journal of Hydrology, 130458.

Tuia, D., Kellenberger, B., Beery, S., Costelloe, B. R., Zuffi, S., Risse, B., ... & Berger-Wolf, T. (2022). Perspectives in machine learning for wildlife conservation. Nature communications, 13(1), 1-15.

Vujović, Ž. (2021). Classification model evaluation metrics. International Journal of Advanced Computer Science and Applications, 12(6), 599-606.

Wai, K. P., Chia, M. Y., Koo, C. H., Huang, Y. F., & Chong, W. C. (2022). Applications of deep learning in water quality management: A state-of-the-art review. Journal of Hydrology, 613, 128332.

Watkins, C. J., & Dayan, P. (1992). Q-learning. Machine learning, 8, 279-292.

Wee, W. J., Zaini, N. A. B., Ahmed, A. N., & El-Shafie, A. (2021). A review of models for water level forecasting based on machine learning. Earth Science Informatics, 14, 1707-1728.

Wen, Q., Sun, L., Yang, F., Song, X., Gao, J., Wang, X., & Xu, H. (2020). Time series data augmentation for deep learning: A survey. arXiv preprint arXiv:2002.12478.

Wicaksono, P., Aryaguna, P. A., & Lazuardi, W. (2019). Benthic habitat mapping model and cross validation using machine-learning classification algorithms. Remote Sensing, 11(11), 1279.

Wikle, C. K., Datta, A., Hari, B. V., Boone, E. L., Sahoo, I., Kavila, I., ... & Chang, W. (2023). An illustration of model agnostic explainability methods applied to environmental data. Environmetrics, 34(1).

Wu, W., Dandy, G. C., & Maier, H. R. (2014). Protocol for developing ANN models and its application to the assessment of the quality of the ANN model development process in drinking water quality modelling. Environmental Modelling & Software, 54, 108-127.

Xu, Q. S., & Liang, Y. Z. (2001). Monte Carlo cross validation. Chemometrics and Intelligent Laboratory Systems, 56(1), 1-11.

Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. IEEE Transactions on neural networks, 16(3), 645-678.

Yu, J., Lu, L., Meng, X., & Karniadakis, G. E. (2022). Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems. Computer Methods in Applied Mechanics and Engineering, 393, 114823.

Zarei, M., Bozorg-Haddad, O., Baghban, S., Delpasand, M., Goharian, E., & Loáiciga, H. A. (2021). Machine-learning algorithms for forecast-informed reservoir operation (FIRO) to reduce flood damages. Scientific reports, 11(1), 24295.

Zeng, X., & Martinez, T. R. (2000). Distribution-balanced stratified cross-validation for accuracy estimation. Journal of Experimental & Theoretical Artificial Intelligence, 12(1), 1-12.

Zheng, H., Liu, Y., Wan, W., Zhao, J., & Xie, G. (2023). Large-scale prediction of stream water quality using an interpretable deep learning approach. Journal of environmental management, 331, 117309.

Zhi, W., Appling, A. P., Golden, H. E., Podgorski, J., & Li, L. (2024). Deep learning for water quality. Nature Water, 1-14.

Zhong, S., Zhang, K., Bagheri, M., Burken, J. G., Gu, A., Li, B., ... & Zhang, H. (2021). Machine learning: new ideas and tools in environmental science and engineering. Environmental Science & Technology, 55(19), 12741-12754.

Zhou, H., Zhang, J., Zhou, Y., Guo, X., & Ma, Y. (2021). A feature selection algorithm of decision tree based on feature weight. Expert Systems with Applications, 164, 113842.

Zhu, M., Wang, J., Yang, X., Zhang, Y., Zhang, L., Ren, H., ... & Ye, L. (2022). A review of the application of machine learning in water quality evaluation. Eco-Environment & Health, 1(2), 107-116.

Appendix A: Artificial Neural Networks in CalSim 3.0

The State Water Project (SWP) and the Central Valley Project (CVP), operating under various environmental regulations, serve as the backbone of the State's water storage and delivery system and collectively supply water to over two third of the State's population. A crucial aspect of these regulations is the stringent control of salinity intrusion into the Delta, a complex tidal estuary influenced by numerous factors affecting salinity levels. The nonlinear relationship among these factors poses challenges for system operations. While operational models like the California Water Resources Simulation Model (CalSim) offer guidelines for efficient planning and management, they do not directly simulate salinity. To address this, the hydrodynamic and water quality model Delta Simulation Model II (DSM2), is utilized. However, due to linking issues and longer simulation time, integrating DSM2 directly into CalSim3 is impractical. The Modeling Support Office created a machine learning (ML) model based on the multi-layer perceptron (MLP), an artificial neural network (ANN) variant, in the mid-1990s to replicate DSM2 for simulating Delta salinity within CalSim. This MLP has undergone recent enhancements and has been incorporated into the newest version of the operational model, CalSim 3.0.

The ANN requires seven input features (depicted in Figure 2) (Jayasundara et al. 2020; Qi et al., 2021). It comprises an input layer, two fully connected hidden layers, and an output layer. The first hidden layer comprises eight neurons, while the second hidden layer contains two neurons. Sigmoid serves as the activation function for the hidden layers, while leaky ReLU is utilized as the activation function for the output layer. The number of neurons in the hidden layers and the choice of activation functions were established through iterative experimentation.

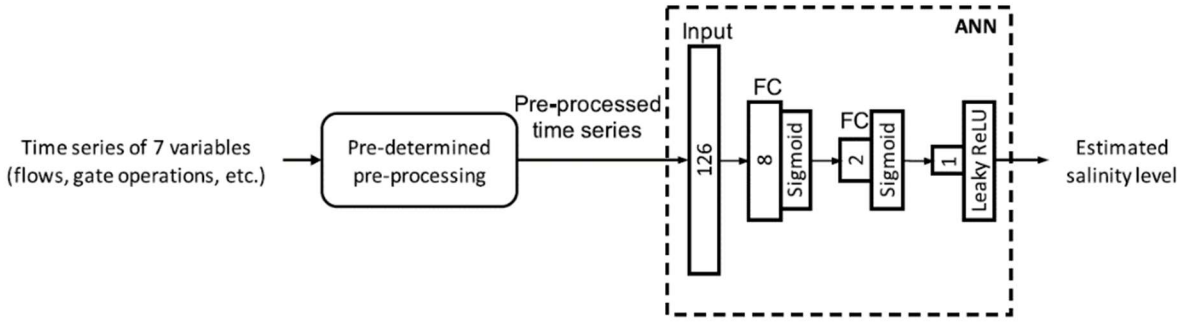

Figure A1. A schematic showing the ANNs implemented in CalSim 3.0.

The ANN necessitates data spanning both the current day and the preceding 117 days for each of the seven input features. This aligns with the widely recognized understanding that Delta salinity exhibits approximately four months of dependence on its predictors. Through an iterative approach, it was established that incorporating eight daily values (the current day plus the preceding seven days) along with 10 periods of 11-day average values for each feature resulted in favorable salinity simulations (Figure 3). Therefore, a total of 18 values for each of the seven input features are required, amounting to 126 input values in total (Figure 2).

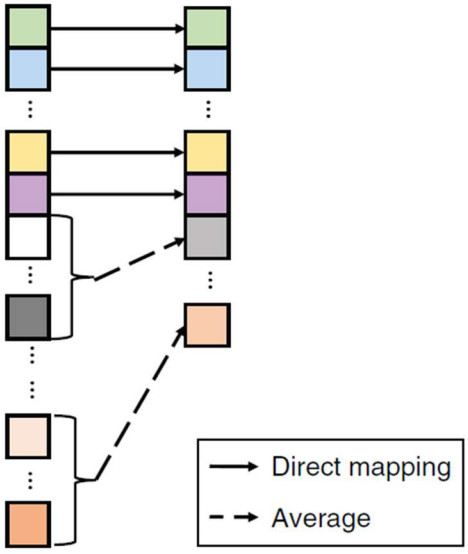

Figure A2. Preprocessing diagram of the ANN illustrated in Figure A1.

A comparison reveals that the salinity results produced by the improved ANN within CalSim 3.0 significantly outperformed those generated by the previous version of CalSim using the old ANN. Consequently, the enhanced ANN was integrated into CalSim 3.0 in 2020 (Jayasundara et al. 2020).

Appendix B: Delta Salinity Simulation Dashboard

The Model Support Office developed ML models to emulate DSM2 in simulating salinity at key locations across the Delta. Specifically, data augmentation techniques were applied to develop six DSM2 salinity emulators: a multi-layer perceptron (MLP), a Residual Long-Short-Term Memory (Res-LSTM) network. a LSTM network, a Residual Network (ResNet) network, a Gated Recurrent Unit (GRU) network, and a Residual GRU (Res-GRU) (Qi et al., 2022a, 2022b; Qi et al., 2023). A complementary browser-based Delta Salinity Dashboard (Dashboard) was developed to serve as the front-end user interface for these DSM2 salinity emulators. This Dashboard evaluates pre-trained models based on user-modified inputs with no need to re-train the models. In this dashboard, users can interactively explore hypothetical scenarios and view the corresponding salinity outputs at key compliance locations during user-defined simulation periods.

Dashboard Architecture

The Dashboard is written in Python and uses several Python libraries, including TensorFlow, Panel, Bokeh, and Pandas. The pre-trained deep learning models (GRU, LSTM, ResNet, Res-GRU, and Res-LSTM) are evaluated by the Dashboard and the results are presented as a Pandas DataFrame object. Bokeh, a Python library for creating interactive visualizations, is used to plot the evaluated models. The plots and widgets are assembled in a dashboard layout using Panel, a Python library used for creating interactive web apps and dashboards by connecting user-defined widgets to plots, images, tables, or text.

The Dashboard is hosted on Azure using Azure Web App Service, which hosts a Linux Docker Container with the required dependencies. Docker packages software into standardized units called containers which include everything needed for the Dashboard to run, such as Python and its packages. The Dashboard Python code, pre-trained models, and input data templates are stored in a GitHub repository. Azure servers can be scaled up to improve virtual machine processing performance or scaled out to increase the number of virtual machines. This scaling can be adjusted based on the application's resource needs and user traffic.

Dashboard Elements

The Dashboard is hosted on the cloud and accessible through a web link (dwrbdodash.azurewebsites.net). Upon opening the link, the Dashboard will be displayed, as shown in Figure B1. Dashboard elements are described in Table B1.

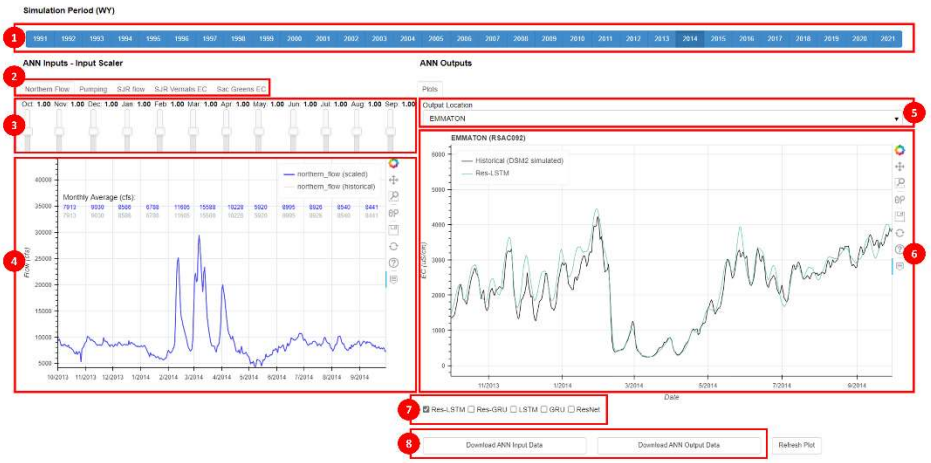

Figure B1. Screenshot of the Delta salinity simulation dashboard, highlighting the interactive elements.

Table B1. Description of the interactive Delta salinity simulation dashboard elements. Dashboard Element Description.

Dashboard Element	Description
1. Water Year Selector	Select the water year for simulation. This tool simulates one water year at a time, where antecedent inputs are assumed to be historical conditions. Note: the modified inputs are not saved in memory. For example, if a user modifies the inputs for 2014, and then changes the simulation to 2015, the scaling factors for 2014 are reverted to historical conditions and 2015 inputs will be modified to reflect the scaling configurations shown on the Dashboard.
2. Input Location Selector	Select the boundary input to modify. This dashboard supports the modification of major Delta boundary conditions, including Northern Flows, Pumping, San Joquin River Flow, and EC boundaries for the San Joaquin River and Sacramento River. Refer to Table 1 in Qi et al. (2022b) for descriptions of the input features.
3. Input Scaler	Sliders to scale the selected Delta boundary condition ± 20% from the baseline (historical) values. The values shown next to the slider for each month indicate the *scale factor*. The *scale factor* uniformly scales the daily values of the boundary condition for a given month. The default scale factor of 1 sets the inputs equal to historical conditions for that month. A scale factor of 1.2 uniformly scales the daily values of that month to 120% of historical values

	and a scale factor of 0.8 uniformly scales the daily values of that month to 80% of historical values.
4. Input Data Plot	Plot of the input features. The blue line indicates the user-modified (i.e., scaled) input feature and the light grey line shows the historical value.
5. Output Location Selector	Select the output location for which to display outputs. Key water quality locations (including compliance locations) are included. See Figure 1 for details of the available output locations.
6. Output Location Salinity Plot	Plot of the output locations. The black line represents the historical DSM2 simulation, and the various colored lines represent the outcomes from the salinity emulator, based on the user-modified inputs.
7. Machine Learning Architecture Selection	Select the architectures to use in the emulator. See Section 2.2 for details.
8. Data Export Options	Options to export the inputs and outputs of the simulation in .csv format for the selected water year.

Dashboard Example Use Case

This section demonstrates the Dashboard by simulating a critically dry year, 2015. The simulated scenario will use a version of 2015 hydrology that is modified to represent arbitrarily drier conditions (reduced winter peak inflows, spring runoff, and exports). The Dashboard configurations used to set up the scenario are shown as follows:

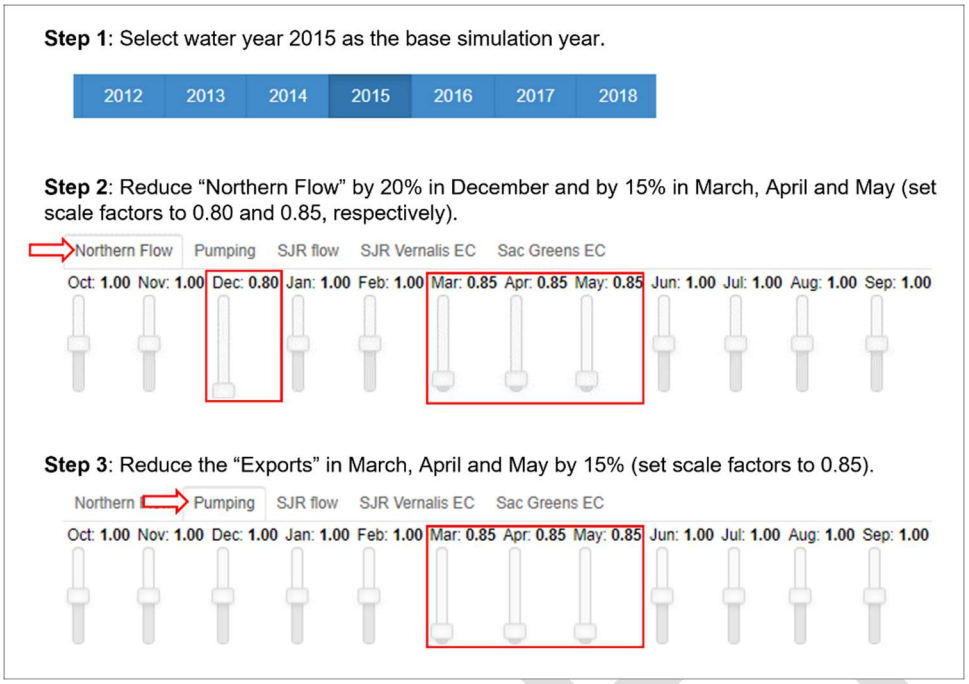

Figure B2. Delta salinity simulation dashboard configurations for example use case.

The outcomes of the example simulation are shown in Figure B3 as the solid lines. Salinity results are shown at Sacramento River near Emmaton and San Joaquin River at Jersey Point, two important compliance locations for Delta water quality-control standards. Visual inspection of the Dashboard outputs indicates higher spring salinity resulting from the reduced inflows. Figure 4 also shows the comparison between the emulator (Res-LSTM architecture) and a full-featured DSM2 run with the same hydrologic inputs from the example use case. Dashed lines represent the DSM2 simulations and solid lines represent the outcomes of the DSM2 emulator. Generally, the alignment between the emulator and DSM2 are close, supporting the concept that the emulator can be used as a screening tool.

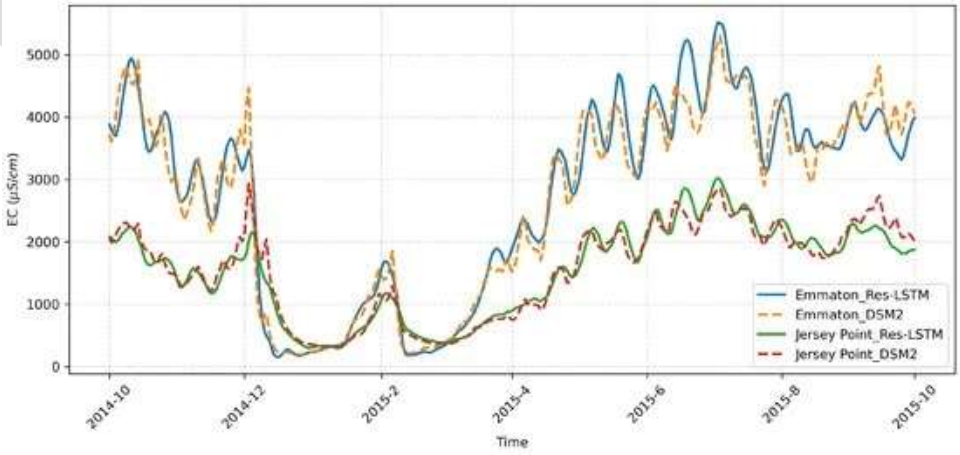

Figure B3. Outputs of the Res-LSTM model compared to DSM2 outputs for Sacramento River at Emmaton and San Joaquin River at Jersey Point.

Appendix C: Glossary

Term	Definition
Accuracy	Correctness of predictions made by a machine learning model.
Activation Function	A mathematical operation applied to a neuron's output to introduce non-linearity, enabling the model to learn complex relationships between inputs and outputs.
Adaptability	The capability of a model to adjust and perform effectively in new or changing environments or tasks.
Artificial Intelligence (AI)	The development of computer systems that can perform tasks that typically require human intelligence.
Artificial Neural Network (ANN)	A model inspired by the structure and function of biological neural networks, consisting of interconnected nodes (neurons) organized into layers for learning from data.
Attention Mechanisms	Mechanisms that allow models to focus on specific parts of input data, assigning different weights to different elements, enhancing performance in tasks.
Classification	Techniques used to categorize data into predefined classes or categories based on input features.
Cloud	A network of remote servers hosted on the internet for storing, managing, and processing data and algorithms, providing scalable resources and services.
Communication	The process of conveying updates, status reports, or instructions between team members, stakeholders, and deployed systems to ensure smooth operation and maintenance.
Computational Resources	The hardware and software components utilized for training, inference, and executing machine learning algorithms, including CPUs, GPUs, and memory.
Computer Science (CS)	The study of the theory, design, and implementation of computer systems and algorithms, including hardware, software, and networking, to solve problems and develop innovative technologies.
Containerization	The practice of encapsulating software, dependencies, and libraries into a standardized unit, called a container, to ensure consistent and portable deployment across different computing environments.
Cross-Validation	A technique used to assess the performance and generalization ability of a model by dividing the dataset into multiple subsets for training and testing.
Data Collection	The gathering of relevant information or samples, often from various sources, to build a dataset suitable for training and evaluating machine learning models.
Data Pre-processing	The processing of cleaning, transforming, and preparing raw data to make it suitable for analysis and machine learning model training.
Deep Learning (DL)	A subset of machine learning where artificial neural networks with multiple layers learn to represent data in increasingly abstract and complex ways.

Development Framework	A toolbox that streamlines building, training, and deploying models.
Dimensionality Reduction	Methods reducing the number of input variables in a dataset while retaining its key information.
Domain Expertise	Specialized knowledge and understanding of a particular subject area or industry that informs the development and application of machine learning models within that domain.
Ensemble	A technique where multiple models are combined to improve predictive performance by aggregating their individual predictions.
Evaluation	The process of assessing the performance, robustness, and effectiveness of a trained machine learning model using various metrics and techniques.
Explainability	The degree to which the inner workings and decisions of a model can be understood and interpreted by humans.
Explainable AI	The development of models whose internal logic and reasoning can be transparently understood by human experts.
Generalization	A model's capacity to effectively learn from training data and apply that knowledge to accurately predict outcomes for unseen data.
Generative AI (GenAI)	A subset of deep learning that generates new content based on patterns learnt from existing data.
Heterogeneity	The diversity or variation among data points, features, or distributions within a dataset or system.
Hidden Layer	An intermediate layer of neurons between the input and output layers in a neural network, responsible for extracting and transforming features from the input data.
Hyperparameter	A configuration parameter external to the model that influences its learning process and performance, typically set before training.
Imbalanced Dataset	A dataset where the distribution of classes or categories is heavily skewed, with one or more classes being significantly underrepresented compared to others.
Input Features	The data attributes or variables that are fed into a machine learning model to train or make predictions.
Interoperability	The ability of different models or systems to seamlessly exchange and utilize data or functionality, promoting integration and collaboration across diverse environments.
Interpretability	The degree to which a model's internal decision-making process can be understood by humans.
Learning Algorithm	A set of procedures and rules that a model follows to adjust its parameters based on input data, aiming to minimize a predefined loss or error function.
Learning Rate	A hyperparameter that controls how quickly a model adapts to new information, determining the step size for each iteration's weight updates, balancing convergence speed and accuracy.
Loss Function	A function quantifying the difference between predicted and actual values, guiding the optimization process during model training.
Machine Learning (ML)	The field of study that enables computers to learn data without being explicitly programmed.

Model Architecture	The high-level design of a model, specifying the organization and connection of its components for data processing and transformation.
Model Deployment	The process of integrating a trained, tested, and evaluated machine learning model into a production environment.
Model Selection	The process of choosing the most appropriate machine learning algorithm or architecture for a given task by comparing and evaluating multiple candidates.
Neuron	Artificial neural network component that processes inputs, applies weights, and produces outputs, mimicking biological neurons' behavior, to learn and make predictions or decisions.
Noise	Irrelevant or random fluctuations in data that can interfere with the learning process or affect the accuracy of a model's predictions.
Normalization	The process of rescaling input features to a predetermined range, to ensure consistent scales and improve convergence in training.
Open Source	Software made available under a license that allows users to freely access, modify, and distribute the source code, promoting collaboration, community-driven development, and transparent innovation.
Overfitting	A model's tendency to capture noise or random fluctuations in the training data, resulting in poor performance on unseen data.
Privacy	Safeguarding sensitive information and preserving the confidentiality of data used in models, preventing unauthorized access or disclosure.
Problem Definition	Articulating the task or objective that the machine learning model aims to solve.
Protocols	Standardized procedures and guidelines for building, training, and evaluating machine learning models.
Quality Assurance	Ensuring that products or services meet specified requirements and standards through systematic processes and testing procedures.
Regression	Techniques establish a relationship between independent variables and a dependent variable for prediction.
Regularization	Techniques applied during model training to prevent overfitting by penalizing overly complex models.
Reinforcement Learning	The paradigm where agents learn to make decisions through trial and error interactions with an environment, aiming to maximize cumulative rewards.
Reliability	A model's ability to produce consistent and accurate results over time.
Reproducibility	The ability to repeat and obtain consistent results in an experiment, study, or computation.
Robustness	A model's ability to maintain performance when faced with variations in the data or unexpected conditions.
Scalability	The ability of a machine learning model to handle increasing amounts of data, computation, or users while maintaining performance and efficiency.
Security	Protecting models, data, and systems from unauthorized access, manipulation, or adversarial attacks, ensuring confidentiality, integrity, and availability.

Sensitivity	A measure of how much a model's output changes in response to small changes in its inputs or parameters.
Spatial Downsampling	The process of reducing the spatial resolution or size of an image or data while maintaining its essential features.
Spatial Interpolation	The technique of estimating values at unmeasured locations within a spatial domain based on known measurements.
Supervised Learning	The paradigm where models are trained on labeled data, with input-output pairs provided, to learn the mapping between inputs and outputs.
Standardization	Establishing uniform processes, specifications, or data formats to ensure compatibility and efficient operation.
Temporal Aggregation	The process of summarizing time-series data over larger time intervals to reduce the granularity while retaining essential patterns.
Temporal Smoothing	Techniques reducing noise or variability in time-series data by averaging neighboring values over time.
Testing	Evaluating the performance and generalization ability of a trained machine learning model on unseen data to assess its accuracy and reliability.
Training	The process of teaching a machine learning model to recognize patterns and make predictions by exposing it to data and adjusting its parameters through iterative optimization algorithms.
Transparency	The quality of being clear, open, and understandable.
Uncertainty	The lack of confidence or variability in predictions made by a model.
Unsupervised Learning	The paradigm where models are trained on unlabeled data to discover patterns, structures, or relationships without explicit guidance on the desired output.