water

Article

Physics-Informed Neural Networks-Based Salinity Modeling in the Sacramento–San Joaquin Delta of California

Dong Min Roh [1,*], Minxue He [2,*], Zhaojun Bai [3], Prabhjot Sandhu [2], Francis Chung [2], Zhi Ding [4], Siyu Qi [4], Yu Zhou [2], Raymond Hoang [2], Peyman Namadi [2], Bradley Tom [2] and Jamie Anderson [2]

[1] Department of Mathematics, University of California, Davis, CA 95616, USA
[2] California Department of Water Resources, 1516 9th Street, Sacramento, CA 95814, USA
[3] Department of Computer Science, University of California, Davis, CA 95616, USA
[4] Department of Electrical and Computer Engineering, University of California, Davis, CA 95616, USA
* Correspondence: droh@ucdavis.edu (D.M.R.); kevin.he@water.ca.gov (M.H.)

check for updates

Citation: Roh, D.M.; He, M.; Bai, Z.; Sandhu, P.; Chung, F.; Ding, Z.; Qi, S.; Zhou, Y.; Hoang, R.; Namadi, P.; et al. Physics-Informed Neural Networks-Based Salinity Modeling in the Sacramento–San Joaquin Delta of California. *Water* **2023**, *15*, 2320. https://doi.org/10.3390/w15132320

Academic Editors: Jin Zhang, Pei Hua and Yun Bai

Received: 23 May 2023
Revised: 14 June 2023
Accepted: 18 June 2023
Published: 21 June 2023

Abstract: Salinity in estuarine environments has been traditionally simulated using process-based models. More recently, data-driven models including artificial neural networks (ANNs) have been developed for simulating salinity. Compared to process-based models, ANNs yield faster salinity simulations with comparable accuracy. However, ANNs are often purely data-driven and not constrained by physical laws, making it difficult to interpret the causality between input and output data. Physics-informed neural networks (PINNs) are emerging machine-learning models to integrate the benefits of both process-based models and data-driven ANNs. PINNs can embed the knowledge of physical laws in terms of the partial differential equations (PDE) that govern the dynamics of salinity transport into the training of the neural networks. This study explores the application of PINNs in salinity modeling by incorporating the one-dimensional advection–dispersion salinity transport equation into the neural networks. Two PINN models are explored in this study, namely PINNs and FoNets. PINNs are multilayer perceptrons (MLPs) that incorporate the advection–dispersion equation, while FoNets are an extension of PINNs with an additional encoding layer. The exploration is exemplified at four study locations in the Sacramento–San Joaquin Delta of California: Pittsburg, Chipps Island, Port Chicago, and Martinez. Both PINN models and benchmark ANNs are trained and tested using simulated daily salinity from 1991 to 2015 at study locations. Results indicate that PINNs and FoNets outperform the benchmark ANNs in simulating salinity at the study locations. Specifically, PINNs and FoNets have lower absolute biases and higher correlation coefficients and Nash–Sutcliffe efficiency values than ANNs. In addition, PINN models overcome some limitations of purely data-driven ANNs (e.g., neuron saturation) and generate more realistic salinity simulations. Overall, this study demonstrates the potential of PINNs to supplement existing process-based and ANN models in providing accurate and timely salinity estimation.

Keywords: flow–salinity relationship; machine learning; multilayer perceptron; physics-informed neural networks

1. Introduction

Salinity is a critical variable in estuarine environments as it impacts the quality of freshwater withdraws and affects fish migration patterns, spawning habitat, and survivability, among others [1–5]. Salinity management in estuarine environments is thus important to maintain desirable water quality, protect aquatic habitats, support economic (e.g., agriculture and industry) development, and mitigate climate change-induced impacts (e.g., sea water intrusion).

Salinity modeling reflects global causes such as the United Nations Sustainable Development Goals (SDGs) and the pursuit of net-zero emissions. By employing advanced modeling techniques, researchers and policymakers can better understand and manage salinity levels, contributing to achieving net-zero emissions goals (SDG 13 [6]) and ensuring clean water and sanitation (SDG 6 [7]). These modeling efforts are particularly important for estuarine environments with enormous environmental and economic significance, including the Sacramento–San Joaquin Delta (Delta) of California, United States (U.S.).

The Delta is the hub of the complex water system of California, a top-five economy in the world. It is bounded by the two largest river systems in the State: the Sacramento River on the North and the San Joaquin River on the South which collectively contribute freshwater to the Delta. Tides from the Pacific Ocean on the west bring salty sea water into the Delta. Freshwater is pumped from the Delta to support over 25 million people and 15,000 km^2 of farmlands [8]. Managing water resources in the Delta (e.g., by the Ghyben–Herzberg relationship [9,10] for the saltwater intrusion behavior) is crucial to maintaining the balance between freshwater and saltwater and ensuring the availability of safe drinking water and irrigation for agriculture. The Delta is also an important biodiversity hotspot that provides a habitat for over 750 species of plants and animals [11,12]. State and federal regulatory requirements on maximum allowable salinity levels have been imposed at compliance locations across the Delta to ensure the quality of water (suitable for drinking and agricultural use) and protect endangered species [13,14]. Understanding the spatial and temporal variations in the Delta is foremost to comply with these regulations. Models have been traditionally developed and applied to gain that understanding.

Salinity simulation models applied in the Delta can be categorized into three types: empirical models, process-based models, and data-driven models. Empirical models were among the earliest methods used for salinity simulation in the Delta. The Minimum Delta Outflow (MDO) procedure [15] translates the salinity values in the Delta to determine the Delta outflow required to meet the requirements of water management practices and to quickly test proposed new standards. The G-model [16,17] is another empirical model that captures salinity transport in the Delta. It incorporates the antecedent Delta outflow and the Delta salinity relationship, which is described by the 1-dimensional advection–dispersion equation. This equation is a partial differential equation that models the spatial and temporal variations of salinity with respect to the outflow. By incorporating antecedent outflow information, the G-model has shown improvements in Delta salinity estimation and improved the accuracy of water supply estimates to MDO.

Although similar to empirical models, process-based models are more comprehensive and simulate Delta salinity with detailed physical processes. A popular one-dimensional process-based model is the Delta Simulation Model II (DSM2) [18] which can calculate flow, stages, flow velocities, and various mass transport processes [19]. In particular, DSM2 has made a significant contribution to the salinity transport process by explicitly solving the advection–dispersion equation in its modeling procedures [20–24]. DSM2 solves the advection–dispersion equation to simulate the salinity transport process in the Delta. Although there exist multi-dimensional models (e.g., TRIM2D [25], RMA10 [26], UnTrim [27], and SCHISM [28]), DSM2 is arguably the most commonly used process-based model to simulate Delta salinity and inform water quality operations and decisions within the Delta due to its fewer input data preparation and computing resources requirements [29] and its best-understood performance over decades [30].

Data-driven machine-learning models are prevalent in numerous scientific domains, including forecasting the performance and energy yield of solar farms [31] and in predicting the remaining usable life of lithium-ion batteries [32]. In Delta salinity simulation, machine-learning models are increasingly used as well due to recent advancements in computing power, which enable superior computational efficiency compared to empirical and process-based models. These models rely mainly on data, with little to no physical process information used. In the Delta, data-driven models are often used to either implement empirical models or to emulate process-based models. For example, Multilayer perceptron (MLP) ANNs were introduced in [19] to emulate DSM2 within CalSim [33,34], an operational water resources planning and simulation model, to estimate salinity at 12 locations in the Delta. In [35], the ANN model in [19] was further enhanced using a multitask-designed approach, with a salinity output for each location. Various deep learning architectures were considered in [29] to emulate DSM2 in salinity modeling, estimating salinity at 28 locations using MLP, Long-Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Residual Network (ResNet). Additionally, two novel deep learning models, Res-LSTM and Res-GRU, were developed in [36] to model salinity in the Delta. All these ANN-based machine-learning models showed improvements in salinity estimation compared to DSM2, while achieving significant reductions in training and inference time.

However, despite their strong predictive skills and efficiency in training, ANN-based machine-learning models rely solely on data and do not take into account the underlying physical processes of the data. With enough input variables, these models can capture correlations and produce correct results, but the entire learning process is a black-box, making it difficult to interpret their results and determine the causality between input and output data [37]. Moreover, the quality of these models depends heavily on the quantity and quality of the training data, and inference or prediction of untrained variables is not possible due to the limitation of these models to only output the training targets. All these limitations hinder the ability of data-driven models to make hypotheses and interpretations of the physical processes, which are critical for water management practices and planning.

Different approaches to integrating knowledge of physical processes into machine-learning models have been studied to overcome these limitations in the field of water resources engineering. One approach is the use of physics-based models, where the physical constraints are enforced in the models' architecture. For example, depth-temperature-density physical constraints were hard-coded in the neural network architecture for quantifying uncertainty in lake temperature modeling [38]. Mass-conserving LSTM architecture has also been explored in predicting peak flows in rainfall-runoff modeling [39] and in predicting the salinity of navigable waterways in Belgium [40]. In [41], differentiable, learnable, process-based models with embedded ANN that respect mass balances were proposed to predict untrained hydrologic variables. One popular approach to enforcing the physical constraints is the physics-informed neural network (PINN), which embeds the governing equations of the underlying physical processes into the loss function of a machine-learning model. Specifically, PINNs are artificial neural networks that are designed to solve ordinary and partial differential equations [42,43]. In recent years, PINNs have made significant impacts in many scientific applications, with the number of PINN papers quintupling between 2019 and 2020 and doubling in 2021 compared to 2020 [44]. Particularly in water engineering, PINNs have been studied to predict water surface profiles in a river [45] and to analyze soil-water infiltration processes for different soil types [46].

The current study aims to address the above-mentioned limitations of ANN-based machine-learning models using PINN models. In our proposed PINN models, we impose the underlying physical law governing flow–salinity relationships, namely the advection–dispersion equation, into artificial neural networks. By enforcing the physical law via the advection–dispersion equation, our proposed PINN models respect the flow–salinity relationships and further enhance salinity estimation accuracy. Our PINN-based machine-learning models can be viewed as a pathway between process-based modeling and machine-learning modeling in the Delta salinity modeling, as they maintain predictive accuracy and training efficiency due to their neural network architecture while respecting underlying physical laws by incorporating the advection–dispersion equation. To the best of our knowledge, no applications of PINN for salinity modeling in the Delta have been explored in the literature.

The rest of the paper is organized as follows. In Section 2, we present the methodology of our study, including locations, dataset, machine-learning model architectures, and evaluation metrics. In Section 3, we present the performance results of the machine-learning models of our study, demonstrating salinity estimation improvements of PINN models to benchmark ANNs. Then, we discuss the scientific and practical implications of our study and future research directions in Section 4 and conclude our paper in Section 5.

2. Materials and Methods

2.1. Study Locations and Dataset

The performance of the proposed machine-learning models in simulating salinity is exemplified at locations in the western part of the Sacramento–San Joaquin Delta (Delta) of California (Figure 1). Spanning through hundreds of kilometers of waterways in Northern California, the Delta is formed by the confluence of the Sacramento River from the north and San Joaquin River from the south and serves as a transition zone between the freshwater of the rivers and saltwater of the Pacific Ocean. In particular, freshwater inflows from the rivers travel westward through the Suisun Bay and exit through San Francisco Bay. Due to their proximity to the Pacific Ocean on the west, the four study locations of interest—Martinez, Port Chicago, Chipps Island, and Pittsburg—are more influenced by seawater than freshwater, unlike the locations in the northern or interior parts of the Delta.

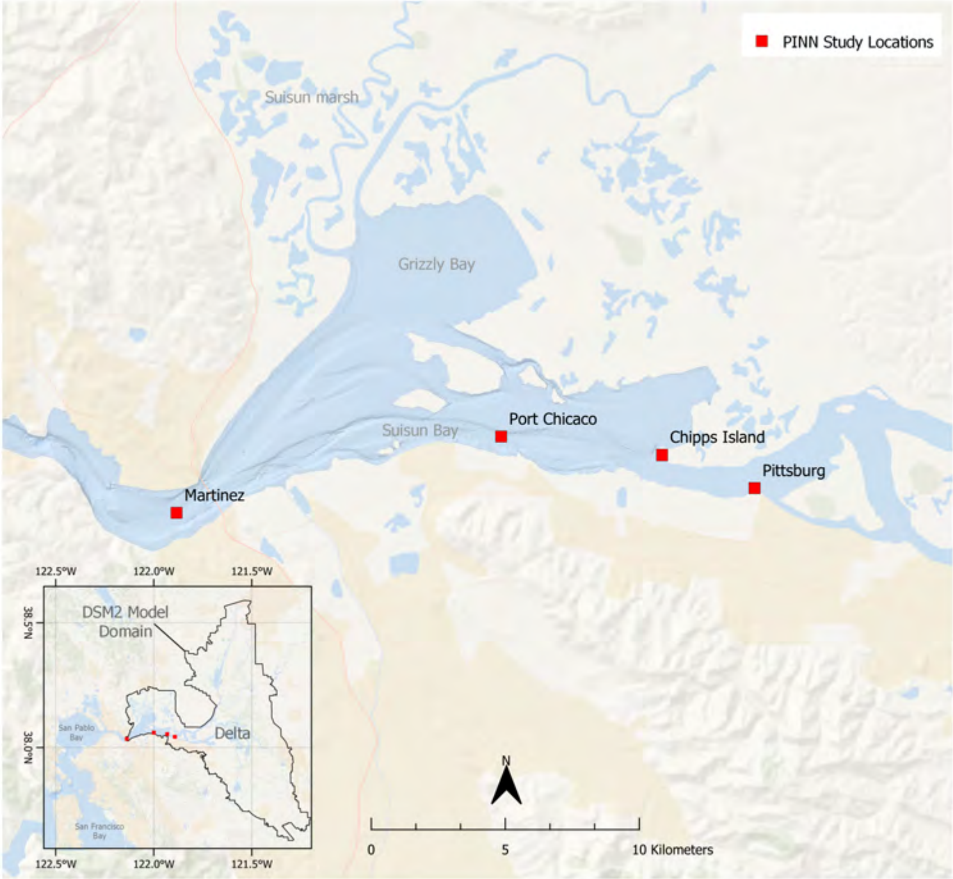


Figure 1. The four study salinity stations—Martinez, Port Chicago, Chipps Island, and Pittsburg—in the Sacramento–San Joaquin Delta Estuary. The insert map illustrates the DSM2 model domain in the Delta.

Historical daily outflow values and DSM2-simulated daily salinity values at four locations from 1 January 1991 to 31 December 2015 are used as input and output, respectively, to train and test the machine-learning models in this study. Outflow is measured in cubic feet per second, while salinity is measured in electrical conductivity (EC) and is represented by micro-Siemens per cm (µS/cm), which reflects the amount of salt dissolved in water.

2.2. Data Preprocessing

2.2.1. Normalization

We denote $Q_i^{(j)}$ as the i-th daily outflow value at the j-th location,

$$j \in \{\text{Martinez, Port Chicago, Chipps Island, Pittsburg}\},$$

and $S_i^{(j)}$ as the target salinity value on day i at location j. We normalize the outflow inputs and the salinity outputs to the range $[0, 1]$ by min-max normalization [47]. Specifically, denoting N_j as the total number of daily samples at location j, we normalize $Q_i^{(j)}$ to

$$Q_i^{(j)} \longleftarrow \frac{Q_i^{(j)} - \left(\min_j \min_{k=1,\dots,N_j} Q_k^{(j)}\right)}{\left(\max_j \max_{k=1,\dots,N_j} Q_k^{(j)}\right) - \left(\min_j \min_{k=1,\dots,N_j} Q_k^{(j)}\right)} \tag{1}$$

where we keep the notation $Q_i^{(j)}$ as the normalized output value of the i-th daily outflow value at the j-th location for simplicity. We apply a similar normalization procedure to the salinity output values as we do to the outflow input values in (1).

2.2.2. Input Memory

Previous study [48] on the DSM2 model showed that the daily salinity depends on the long-term memory of contributing parameters. Similarly, the G-model's [16,17] success in salinity estimation is attributed to its ability to account for the effects of antecedent outflow on salinity. To account for long-term input memory, recent studies on salinity modeling in the Delta using artificial neural networks [19,29,35,36] have followed the practice of aggregating 118 antecedent daily values into 18 values for the current day input variables. We adopt this same practice in our study for preprocessing the input outflow to account for antecedent outflows. Specifically, for each normalized daily outflow $Q_i^{(j)}$ we form an outflow data vector $\mathbf{Q}_i^{(j)} \in \mathbb{R}^{18}$ by keeping the outflow values of the current day plus the most recent 7 antecedent days as-is, along with 10 successive 11-day averages of the prior 110 days. In other words, we define $\mathbf{Q}_i^{(j)}$ as

$$\mathbf{Q}_i^{(j)} = [Q_i^{(j)}, \dots, Q_{i-7}^{(j)}, \overline{Q_{(i-8)\to(i-18)}^{(j)}}, \dots, \overline{Q_{(i-107)\to(i-117)}^{(j)}}]^T, \tag{2}$$

where $\overline{Q_{i_1\to i_2}^{(j)}}$ denotes the average of the normalized outflow values from day i_1 to i_2. By reducing the 118 antecedent daily outflow values into 18 values, we avoid unnecessary increases in the complexity of the machine-learning models proposed in our study while accounting for antecedent outflow memory.

2.3. Neural Network Architectures

In this work, we present/study three machine-learning models: a conventional multilayer perceptrons (MLP) network, referred to as 'ANN', which takes preprocessed outflow data vectors as inputs and outputs target salinity values; a PINN that incorporates the location variable x and the time variable t as additional inputs and embeds the advection–dispersion equation into the loss function; and an extension of the PINN, referred to as 'FoNet', which includes an additional input encoding layer to transform inputs to a higher-dimensional feature space via high-frequency functions. To demonstrate the improvement in salinity estimation by the proposed PINN and FoNet models, we keep the three models simple using fully connected networks. This way, we can directly demonstrate the impact of the inclusion of the causality information between input and output data, specifically the advection–dispersion equation for flow–salinity relationships, into the machine-learning models for salinity modeling in the Delta.

2.3.1. Artificial Neural Networks (ANN)

We consider a conventional MLP network, which has been extensively studied in Delta salinity modeling with varying input and output sizes and parameter choices [2,19,24,29,35,36,49]. In this study, we adopt an MLP network with four hidden layers. Its input variables are the daily outflow vector, represented as $\mathbf{Q} = [\mathbf{Q}_1, \mathbf{Q}_2, \ldots, \mathbf{Q}_{18}]^T$, and its output is the estimated salinity value $\hat{S}$. Concisely, the network estimated salinity value on the i-th day at the j-th location is $\hat{S}(\mathbf{Q}_i^{(j)}; \theta)$. The notation θ denotes the trainable parameters, weights, and biases of the hidden layers, in the neural network. The number of neurons in the hidden layers and the choice of activation functions are selected by random hyperparameter search, which will be discussed in Section 2.4.1. The architecture of the ANN is shown in Figure 2. The loss function of the ANN is defined as

$$L(\theta) = \sum_{i,j} (\hat{S}(\mathbf{Q}_i^{(j)}; \theta) - S_i^{(j)})^2, \tag{3}$$

i.e., the widely used mean squared error (MSE) loss function [47]. The ANN is trained by computing an optimal θ^* that minimizes $L(\theta)$.

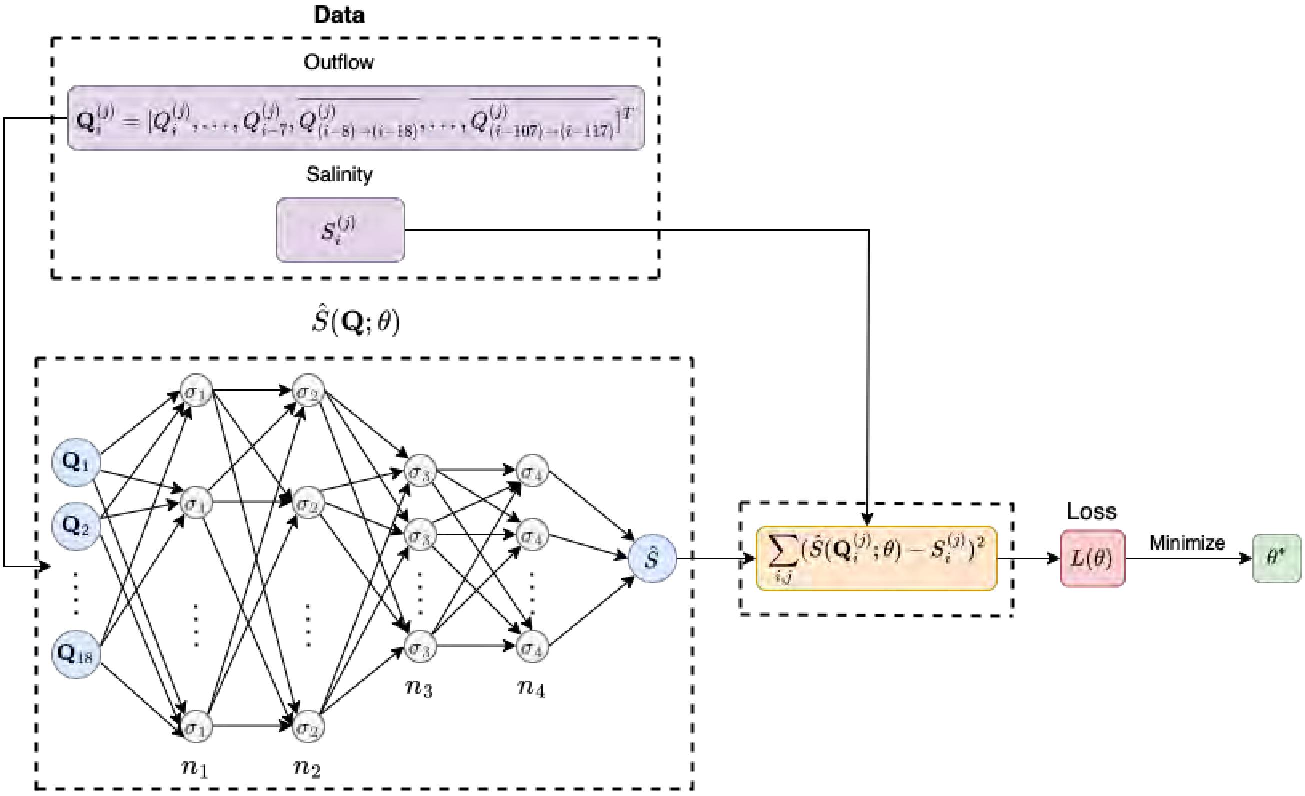

Data
Outflow
$\mathbf{Q}_i^{(j)} = [Q_i^{(j)}, \ldots, Q_{i-7}^{(j)}, \overline{Q_{(i-8)\to(i-18)}^{(j)}}, \ldots, \overline{Q_{(i-107)\to(i-117)}^{(j)}}]^T$
Salinity
$S_i^{(j)}$
$\hat{S}(\mathbf{Q}; \theta)$
$\mathbf{Q}_1$
$\mathbf{Q}_2$
$\mathbf{Q}_{18}$
σ_1
σ_2
σ_3
σ_4
n_1
n_2
n_3
n_4
$\hat{S}$
$\sum_{i,j}(\hat{S}(\mathbf{Q}_i^{(j)}; \theta) - S_i^{(j)})^2$
Loss
$L(\theta)$
Minimize
θ^*

Figure 2. Architecture of the ANN model. The ANN consists of an input layer of 18 variables corresponding to the outflow vector, four hidden layers with varying activation functions and numbers of neurons, and an output layer for salinity estimation. For $k \in \{1,2,3,4\}$, σ_k and n_k denote the activation function and the number of neurons at the k-th hidden layer, respectively. The choices of σ_k and n_k are achieved by random hyperparameter search, as discussed in Section 2.4.1. The model is trained using daily outflow data vectors $\mathbf{Q}_i^{(j)}$ and salinity values $S_i^{(j)}$ at training locations, and the training process involves minimizing the mean squared error between the model-estimated salinity values and the corresponding data salinity values using the loss function (3).

2.3.2. Physics-Informed Neural Networks (PINN)

We propose a novel machine-learning model in salinity estimation in the Delta, using PINN. Our PINN embeds the advection–dispersion equation into its loss function, therefore incorporating the flow–salinity relationships into the machine-learning model. The one-dimensional advection–dispersion equation for salinity transport is [16,17]:

$$A\frac{\partial S}{\partial t} - Q\frac{\partial S}{\partial x} = \frac{\partial}{\partial x}\left(KA\frac{\partial S}{\partial x}\right), \tag{4}$$

where $A(x)$ is the estuary cross-sectional area, $S(x,t)$ is the concentration of salt (i.e., salinity in this study), $Q(x,t)$ is the volumetric flowrate (i.e., outflow in this study), $K(x,t)$ is the longitudinal dispersion coefficient, x is the longitudinal distance (increasing in the upstream direction), and t is the time. Under various assumptions on the initial and boundary conditions and the spatial and temporal dependencies of the flowrate and the dispersion coefficient, different studies have developed techniques for deriving an analytical solution of the advection–dispersion Equation (4). We refer the readers to [50] and references there-within for these various techniques. In this study, the flowrate is spatiotemporally dependent, driven by the data, and the cross-sectional area A and the dispersion coefficient K are assumed to be constants for simplicity.

The PINN model, similar to the ANN, is an MLP network, but with two hidden layers instead of four. Two hidden layers are sufficient for the PINN to obtain adequate results, whereas the ANN requires a deeper network structure. The hyperparameters are selected by random hyperparameter search, which will be discussed in Section 2.4.1. The architecture of the PINN is shown in Figure 3. Besides the number of hidden layers, there are two major differences between the ANN and the PINN models. First, in addition to the outflow data vector, the PINN has two additional input variables: longitudinal distance x and time t. For each day i and each location j, data values $x_i^{(j)}$ and $t_i^{(j)}$ are created for the longitudinal distance and time variables, respectively. Both variables are normalized: $x_i^{(j)}$ with respect to the distance between the westernmost location, Martinez, and the easternmost location, Pittsburg, and $t_i^{(j)}$ with respect to the number of days between 1 January 1991 and 31 December 2015. The estimated salinity on the i-th day at the j-th location by the PINN is denoted as $\hat{S}(x_i^{(j)}, t_i^{(j)}, \mathbf{Q}_i^{(j)}; \theta)$. The other major difference is in the loss function. In addition to the MSE loss function, the PINN seeks to minimize the advection–dispersion loss function, which ensures that the data satisfy the advection–dispersion Equation (4) and consequently the flow–salinity relationships, i.e., the loss function of the PINN is defined as the sum of the two loss functions:

$$\begin{aligned} L(\theta) = &\sum_{i,j}(\hat{S}(x_i^{(j)}, t_i^{(j)}, \mathbf{Q}_i^{(j)}; \theta) - S_i^{(j)})^2 \\ &+ \sum_{i,j}\left(A\frac{\partial \hat{S}}{\partial t}\bigg|_{(x_i^{(j)}, t_i^{(j)}, \mathbf{Q}_i^{(j)}; \theta)} - Q_i^{(j)}\frac{\partial \hat{S}}{\partial x}\bigg|_{(x_i^{(j)}, t_i^{(j)}, \mathbf{Q}_i^{(j)}; \theta)} - KA\frac{\partial^2 \hat{S}}{\partial x^2}\bigg|_{(x_i^{(j)}, t_i^{(j)}, \mathbf{Q}_i^{(j)}; \theta)} \right)^2. \end{aligned} \tag{5}$$

In (5), $Q_i^{(j)}$ is the normalized daily outflow on the i-th day at the j-th location and corresponds to the first component of the outflow data vector $\mathbf{Q}_i^{(j)}$. The derivatives of the network outputs with respect to the network inputs can be computed efficiently using automatic differentiation with backpropagation [51].

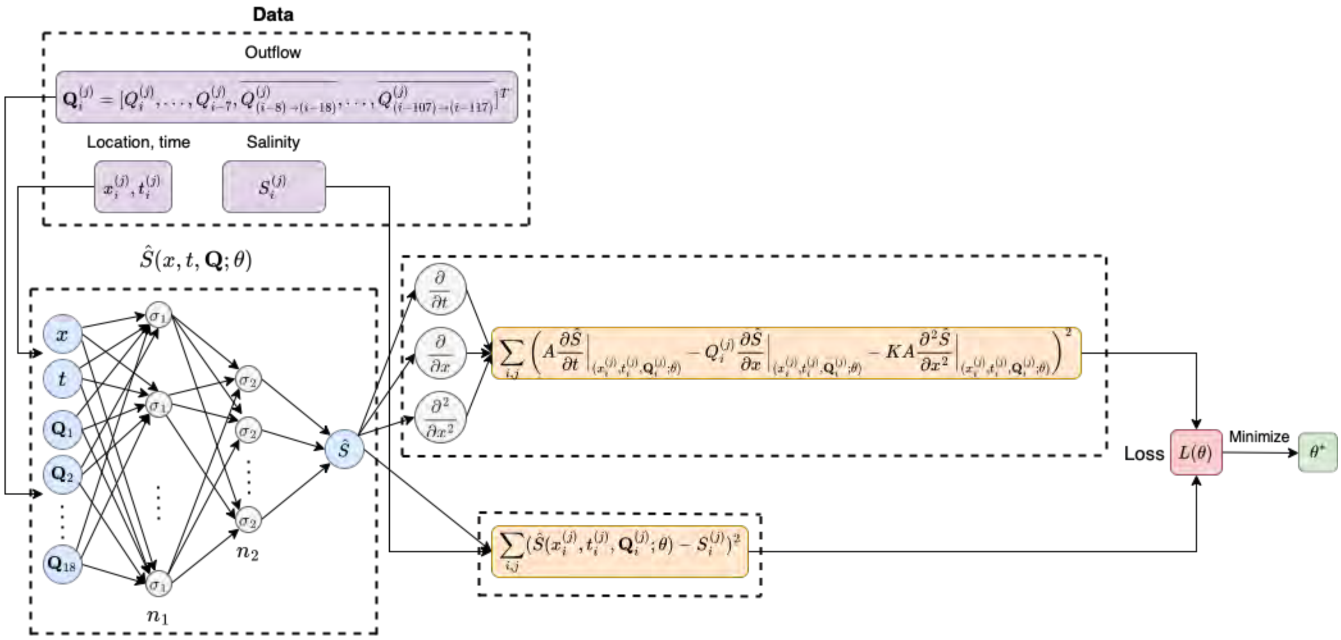

Data
Outflow
$\mathbf{Q}_i^{(j)} = [Q_i^{(j)}, \ldots, Q_{i-7}^{(j)}, \overline{Q_{(i-8)\to(i-18)}^{(j)}}, \ldots, \overline{Q_{(i-107)\to(i-117)}^{(j)}}]^T$
Location, time
Salinity
$x_i^{(j)}, t_i^{(j)}$
$S_i^{(j)}$
$\hat{S}(x, t, \mathbf{Q}; \theta)$
x
t
$\mathbf{Q}_1$
$\mathbf{Q}_2$
$\mathbf{Q}_{18}$
σ_1
σ_2
n_1
n_2
$\hat{S}$
$\frac{\partial}{\partial t}$
$\frac{\partial}{\partial x}$
$\frac{\partial^2}{\partial x^2}$
$\sum_{i,j} \left(A \frac{\partial \hat{S}}{\partial t}\Big|_{(x_i^{(j)}, t_i^{(j)}, \mathbf{Q}_i^{(j)}; \theta)} - Q_i^{(j)} \frac{\partial \hat{S}}{\partial x}\Big|_{(x_i^{(j)}, t_i^{(j)}, \mathbf{Q}_i^{(j)}; \theta)} - KA \frac{\partial^2 \hat{S}}{\partial x^2}\Big|_{(x_i^{(j)}, t_i^{(j)}, \mathbf{Q}_i^{(j)}; \theta)} \right)^2$
$\sum_{i,j} (\hat{S}(x_i^{(j)}, t_i^{(j)}, \mathbf{Q}_i^{(j)}; \theta) - S_i^{(j)})^2$
Loss
$L(\theta)$
Minimize
θ^*

Figure 3. Architecture of the PINN model. The PINN consists of an input layer of 20 variables corresponding to the location, the time, and the outflow vector, two hidden layers with varying activation functions and numbers of neurons, and an output layer for salinity estimation. For $k \in \{1, 2\}$, σ_k and n_k denote the activation function and the number of neurons at the k-th hidden layer, respectively. The choices of σ_k and n_k are achieved by random hyperparameter search, as discussed in Section 2.4.1. The model is trained using daily location $x_i^{(j)}$ and time $t_i^{(j)}$ values, daily outflow data vectors $\mathbf{Q}_i^{(j)}$, and salinity values $S_i^{(j)}$ at training locations, and the training process involves minimizing the sum of mean squared error between the model-estimated salinity values and the corresponding data salinity values and the advection–dispersion loss term using the loss function (5).

2.3.3. Physics-Informed Fourier Networks (FoNet)

Neural networks are known to favor low-frequency solutions, a phenomenon known as *spectral bias* [52]. One way to alleviate this issue is to add an encoding layer that transforms the inputs to a higher-dimensional feature space via high-frequency functions [52–54]. Our FoNet model includes an encoding layer in our PINN model, where the input encoding layer is trainable. Specifically, denoting the input variables concisely as $\mathbf{x} = [x, t, \mathbf{Q}]^T \in \mathbb{R}^{20}$, the encoding layer is a Fourier feature mapping [54] that encodes $\mathbf{x}$ as

$$\begin{bmatrix} \sin(2\pi W_f \mathbf{x}) \\ \cos(2\pi W_f \mathbf{x}) \end{bmatrix} \in \mathbb{R}^{2n_1}, \tag{6}$$

where $W_f \in \mathbb{R}^{n_1 \times 20}$ is a trainable frequency matrix. The output of the encoding layer (6) then passes through the rest of the network. The hyperparameters, including the frequency matrix's projection dimension size n_1 as well as the number of neurons and the choice of the activation functions, are selected by hyperparameter search that will be further discussed in Section 2.4.1. The architecture of the FoNet is summarized in Figure 4. The trainable parameters θ include the frequency matrix W_f and weights and biases of the hidden layers. The FoNet is trained by minimizing the same loss function (5) as the PINN.

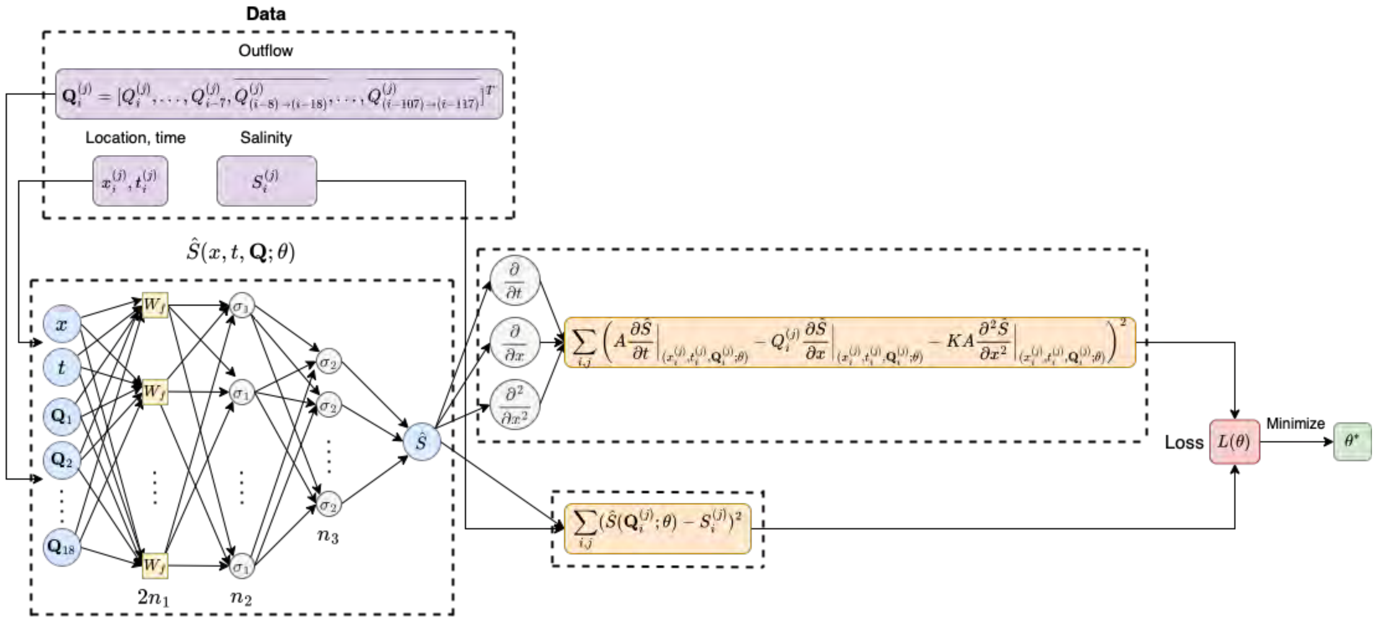

Data
Outflow
$\mathbf{Q}_i^{(j)} = [Q_i^{(j)}, \ldots, Q_{i-7}^{(j)}, \overline{Q_{(i-8)\to(i-18)}^{(j)}}, \ldots, \overline{Q_{(i-107)\to(i-117)}^{(j)}}]^T$
Location, time
$x_i^{(j)}, t_i^{(j)}$
Salinity
$S_i^{(j)}$
$\hat{S}(x, t, \mathbf{Q}; \theta)$
x
t
$\mathbf{Q}_1$
$\mathbf{Q}_2$
$\mathbf{Q}_{18}$
W_f
σ_1
σ_2
$2n_1$
n_2
n_3
$\hat{S}$
$\frac{\partial}{\partial t}$
$\frac{\partial}{\partial x}$
$\frac{\partial^2}{\partial x^2}$
$\sum_{i,j}\left(A\frac{\partial \hat{S}}{\partial t}\Big|_{(x_i^{(j)},t_i^{(j)},\mathbf{Q}_i^{(j)};\theta)} - Q_i^{(j)}\frac{\partial \hat{S}}{\partial x}\Big|_{(x_i^{(j)},t_i^{(j)},\mathbf{Q}_i^{(j)};\theta)} - KA\frac{\partial^2 \hat{S}}{\partial x^2}\Big|_{(x_i^{(j)},t_i^{(j)},\mathbf{Q}_i^{(j)};\theta)}\right)^2$
$\sum_{i,j}(\hat{S}(\mathbf{Q}_i^{(j)};\theta) - S_i^{(j)})^2$
Loss
$L(\theta)$
Minimize
θ^*

Figure 4. Architecture of the FoNet model. The FoNet consists of an input layer of 20 variables corresponding to the location, the time, and the outflow vector, an encoding layer with a frequency matrix, two hidden layers with varying activation functions and numbers of neurons, and an output layer for salinity estimation. W_f denotes a trainable frequency matrix in the encoding layer that maps the input variables to a higher-dimensional feature space. For $k \in \{1,2\}$, σ_k and n_k denote the activation function and the number of neurons at the k-th hidden layer, respectively. The choices of W_f, σ_k, and n_k are achieved by random hyperparameter search, as discussed in Section 2.4.1. The model is trained using daily location $x_i^{(j)}$ and time $t_i^{(j)}$ values, daily outflow data vectors $\mathbf{Q}_i^{(j)}$, and salinity values $S_i^{(j)}$ at training locations, and the training process involves minimizing the sum of mean squared error between the model-estimated salinity values and the corresponding data salinity values and the advection–dispersion loss term using the loss function (5).

It should be pointed out that we considered an ANN model with two hidden layers, the same number of layers as PINN and FoNet models, as well as an ANN model with three hidden layers. However, these ANN models did not perform well in our study. As a result, we opted for a deeper ANN model with four hidden layers, which exhibited improved performance. In Table A1 in Appendix A.1, we provide a performance comparison between the two-layered, three-layered, and four-layered ANN models on the Fold 1 test dataset at the untrained location Port Chicago as an example.

2.4. Hyperparameter Search and Data Split

2.4.1. Hyperparameter Search

Proper selection of hyperparameters (i.e., the number of neurons and the activation functions) is crucial to achieving optimal model performance. We perform random hyperparameter searches for the machine-learning models to obtain the optimal hyperparameters. The search space for the number of neurons in each hidden layer for each model is set as follows:

- ANN—the number of neurons in hidden layers 1 and 2, denoted as n_1 and n_2, respectively, lie in the range of $[4, 8, 12, \ldots, 32]$, and the number of neurons in hidden layer 3 and 4, denoted as n_3 and n_4, lie in the range of $[2, 4, 6, \ldots, 16]$.
- PINN—n_1 lies in the range of $[4, 8, 12, \ldots, 32]$ and n_2 lies in the range of $[2, 4, 6, \ldots, 16]$.
- FoNet—n_1, which represents the projection dimension size of the frequency matrix, lies in the range of $[4, 8, 12, \ldots, 32]$, and n_2 and n_3, the number of neurons in hidden

layer 1 and 2, respectively, lie in the ranges of $[4, 8, 12, \ldots, 32]$ and $[2, 4, 6, \ldots, 16]$, respectively.

The possible activation functions for all three models include ReLu, Tanh, ELU, and Sigmoid functions. Each model is subject to 50 random hyperparameter searches, and the combination of hyperparameters with the smallest loss function value is selected as the optimal hyperparameter. We list these optimal hyperparameter combinations in Appendix A.2.

2.4.2. Data Split

To measure the salinity estimation capabilities of the machine-learning models, we use the Blocked Cross-Validation [55] procedure. Similar to k-fold cross-validation, the Blocked Cross-Validation partitions the dataset into k blocks of equal size, and leaves one block for testing and the other $k - 1$ blocks for training. The difference is that there is no initial random shuffling of the data, so the natural temporal ordering is preserved within each block. We consider the Blocked Cross-Validation with 5 blocks for the DSM2-simulated 25-year salinity dataset in the range from 1 January 1991 to 31 December 2015, resulting in 80% of the dataset for training and 20% for testing. However, we do not use any of the data at the Port Chicago location for training. Instead, the salinity value at Port Chicago will only be used for testing to evaluate the models' performance at an untrained location. The procedure of the Blocked Cross-Validation is illustrated in Figure 5 where we numbered the folds to reference them later.

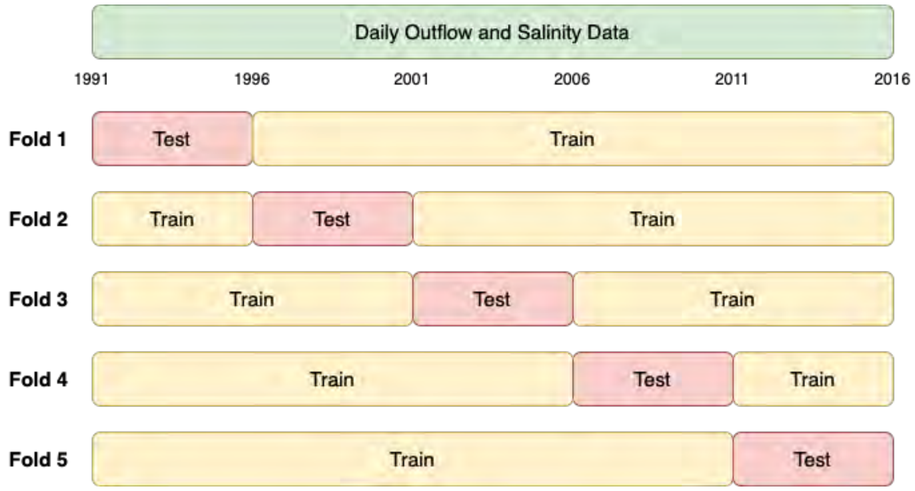


Figure 5. The Blocked Cross-Validation.

2.5. Evaluation Metrics

We use the following four statistical metrics to evaluate the performance of the machine-learning models: the square of the correlation coefficient r^2 [56], percentage bias [57], root mean standard deviation ratio (RSR) [58], and the Nash–Sutcliffe efficiency coefficient (NSE) [59]. Their formulas are shown in Table 1, where S stands for the salinity value and $\overline{S}$ stands for the average salinity value of the dataset. t represents an arbitrary day in the dataset, T is the total number of samples in the dataset, σ is the standard deviation, and the subscripts 'ref' and 'ML' designate the target values (DSM2-simulated) and the machine-learning model-estimated values, respectively. r^2 measures the linear-relation strength between the model-estimated salinity and the target salinity, percent bias quantifies how much the model under- or overestimates the salinity, RSR is a standardized representation of the root mean squared error (RMSE) between model outputs and targets, and NSE quantifies the predictive capacity of the models with the global mean of target sequences. For r^2 and NSE, a value closer to 1 indicates better performance, while for bias and RSR, a value close to 0 indicates better performance.

Table 1. Evaluation metrics.

Name	Definition	Formula
r^2	Squared Correlation Coefficient	$\left(\frac{\sum_t^T \lvert(S_{ref}^t - \overline{S_{ref}})(S_{ML}^t - \overline{S_{ML}})\rvert}{T\sigma_{ref}\sigma_{ML}}\right)^2$
Bias	Percent Bias	$\frac{\sum_t^T (S_{ML}^t - S_{ref}^t)}{\sum_t^T S_{ref}^t} \times 100\%$
RSR	RMSE-observations standard deviation ratio	$\frac{\sqrt{\sum_t^T (S_{ref}^t - S_{ML}^t)^2}}{\sqrt{\sum_t^T (S_{ref}^t - \overline{S_{ref}})^2}}$
NSE	Nash–Sutcliffe Efficiency coefficient	$1 - \frac{\sum_t^T (S_{ref}^t - S_{ML}^t)^2}{\sum_t^T (S_{ref}^t - \overline{S_{ref}})^2}$

2.6. Implementation Details

The experiments are conducted using Python 3.9.16 and the machine-learning models are built and trained using the TensorFlow 2.10.1 library. The programming codes are executed in Google Colaboratory, a hosted Jupyter notebook service that provides access to GPUs. For all three machine-learning models, Adam optimizer [60] with a constant learning rate of 0.01 is used for training. To prevent overfitting of the models, the batch size is set to 128 and the training process is limited to 5000 epochs. Furthermore, if the loss function on the test set does not decrease for consecutive 50 epochs, the training process stops.

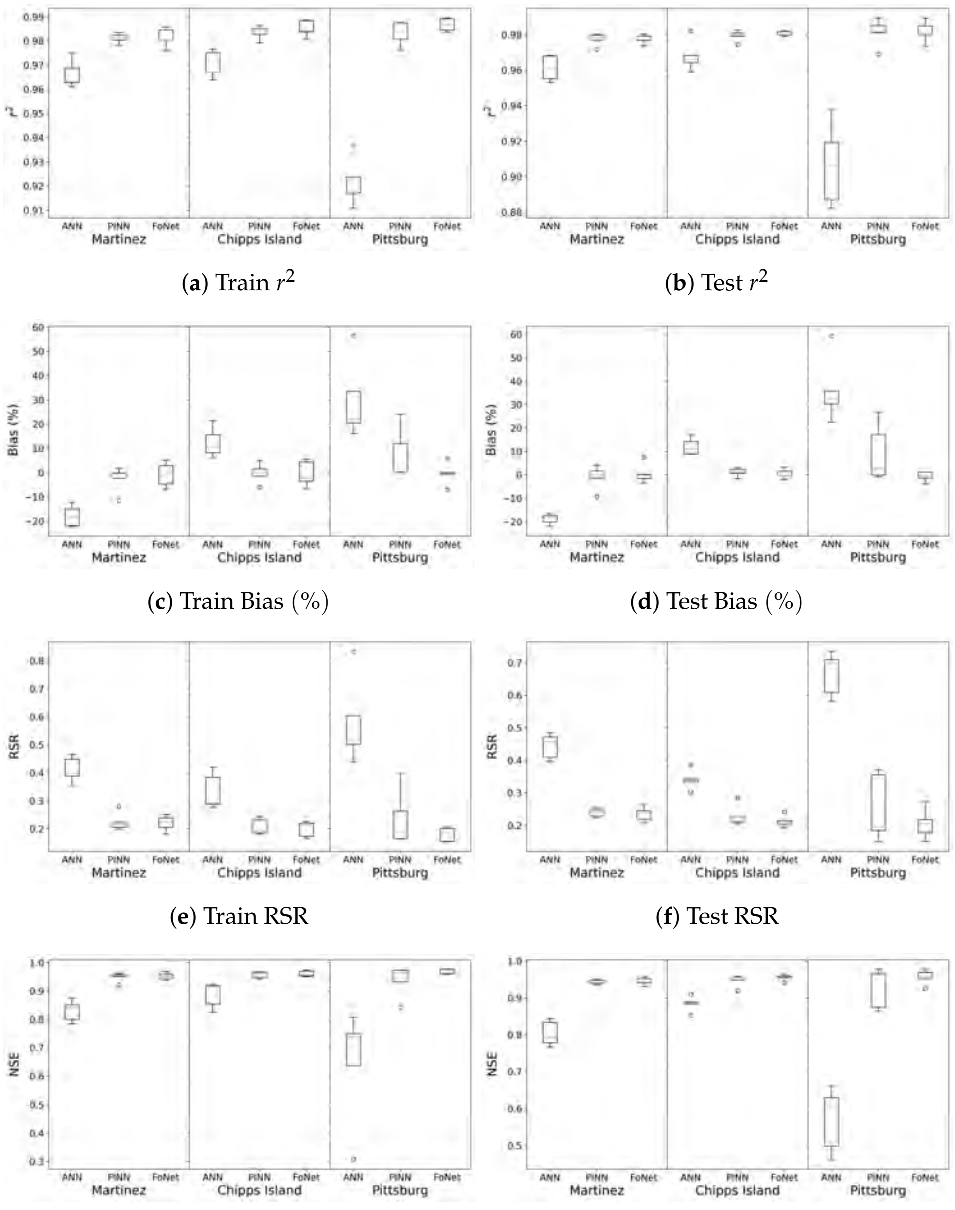
3. Results

In this section, we present the performance results of our three machine-learning models, namely the ANN, PINN, and FoNet models. We evaluate the performance of these models quantitatively using four statistical evaluation metrics described in Section 2.5, and qualitatively by visually inspecting the time series plots and the scatter plots. In the first subsection, we present the results on the three trained locations Martinez, Chipps Island, and Pittsburg, and in the second subsection, we present the results on the untrained (independent) location Port Chicago.

3.1. Performance Results on Trained Locations

To compare the performance of our three machine-learning models (ANN, PINN, and FoNet) and assess their generalizability, we use a 5-fold Blocked Cross-Validation and evaluate four statistical metrics (r^2, percentage bias, RSR, and NSE) for each fold and location. These metrics help us quantify the accuracy, precision, and goodness of fit of the models to the data. In Figure 6, we present the box and whisker plots of these metrics of the folds for each model on the training data (left column) and test data (right column) at the three locations (Martinez, Chipps Island, and Pittsburg) used for model training. The box and whisker plots show the distribution of the metric values for each model and provide a visual comparison of their performance.

The box and whisker plots in Figure 6 demonstrate a clear improvement in performance for the physics-informed models (PINN and FoNet) over the standard model (ANN). For all four considered metrics and at all three locations, the PINN and FoNet models outperform the ANN model by a significant margin. The PINN and FoNet models attain smaller percent bias and RSR values and larger correlation coefficient r^2 and NSE values than the ANN models. In particular, at Pittsburg, the improvement is prominent as the PINN and FoNet models achieve NSE values of around 0.93 on test data, compared to around 0.55 for the ANN model. The PINN and FoNet models perform similarly at locations Martinez and Chipps Island, but at Pittsburg, FoNet has a slight edge over PINN in estimating salinity, with slightly lower variance in its metric values. The exact values of the metrics for each fold are shown in the tables in Appendix A.3. Overall, there is no evidence of overfitting for any of the models, as they perform similarly on the training and test data.



(**a**) Train r^2

(**b**) Test r^2

(**c**) Train Bias (%)

(**d**) Test Bias (%)

(**e**) Train RSR

(**f**) Test RSR

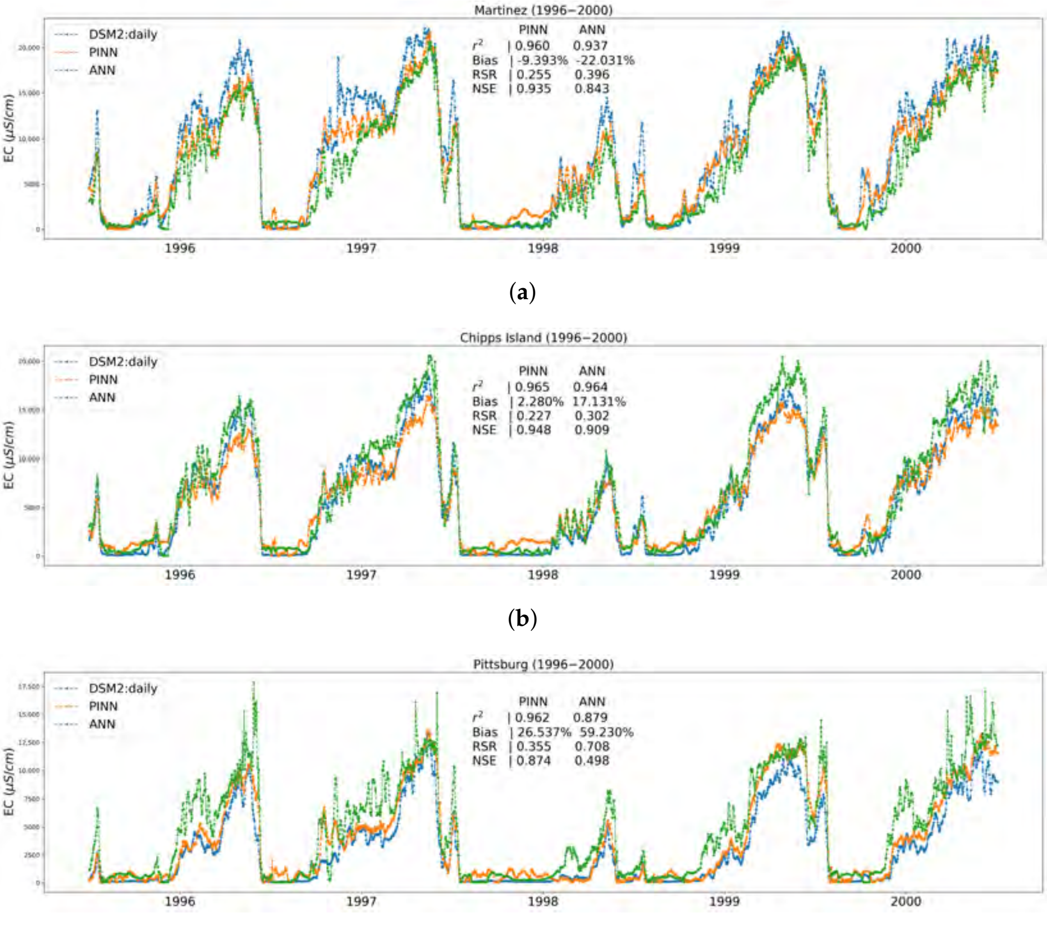
(**g**) Train NSE

(**h**) Test NSE

Figure 6. Box and whisker plots of evaluation metrics r^2, percentage bias, RSR, and NSE for the training (left column, panels (**a**,**c**,**e**,**g**)) and test (right column, panels (**b**,**d**,**f**,**h**)) data of the machine-learning models ANN, PINN, and FoNet at each of the three trained locations Martinez, Chipps Island, Pittsburg. For a box and whisker plot, the orange line represents the median value of the 5 metric values corresponding to 5 folds in the Blocked Cross-Validation; the box represents the interquartile range from the 25th percentile to the 75th percentile; the top bar represents the maximum metric value within 1.5 times the interquartile range above the 75th per percentile and the bottom bar represents the minimum metric values within 1.5 times the interquartile range below the 25th percentile; the open circles correspond to outliers.

We visually inspect the time series plots of the estimated salinity values of our three machine-learning models in comparison to the target salinity values (DSM2-simulated). For better visibility we consider two models at a time, along with DSM2-simulated salinity values, first comparing the estimated salinity values of ANN and PINN models and then those of PINN and FoNet models. Furthermore, we show the time series plots for test data only to more closely examine the performance of the models for a shorter time span of five years rather than 20 years for training data. Here, in this section, we display the time series plots for Fold 2, i.e., the test data being 1 January 1996 to 31 December 2000, as an exemplary time series plots of the Blocked Cross-Validation procedure. The rest of the time series plots of the folds from the procedure are available in Appendix A.4.

Figure 7 displays the daily time series plots of DSM2-simulated salinity values in blue along with the daily time series plots of estimated salinity values by the ANN and PINN models in green and orange, respectively, at the three locations (Martinez, Chipps Island, and Pittsburg) used for model training. Both machine-learning models perform well in estimating the overall seasonal salinity patterns throughout the five years, but PINN estimates the target salinity values more accurately than ANN overall. PINN outperforms ANN in estimating high peak salinity values. Specifically, at Martinez, PINN estimates the high salinity values more closely to target salinity than ANN, which tends to underestimate them. Similarly, at Chipps Island, PINN follows the temporal pattern of high salinity more closely than ANN, which tends to overestimate these peak values. At Pittsburg, ANN produces estimates that are quite noisy and overshoot the target salinity frequently, while PINN can estimate salinity more closely to the target in its estimations. Similar to the time series plots of Figure 7, Figure 8 illustrates the daily time series plots of PINN and FoNet models in orange and black, respectively, along with DSM2-simulated salinity values in blue. FoNet model performs quite similarly to PINN model but achieves slightly better evaluation metrics. Visual inspection of the time series plots in Figure 8 indicates that FoNet estimates low salinity values more accurately than PINN.

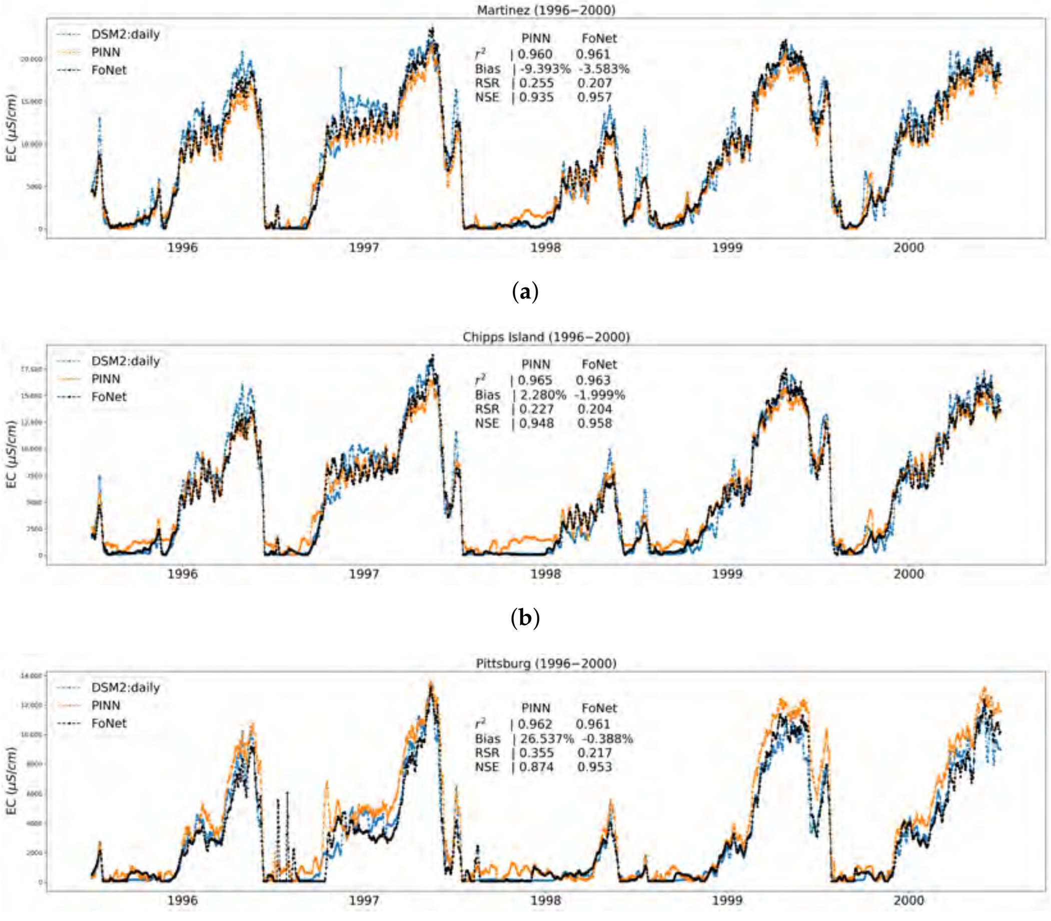


(**a**)

(**b**)

(**c**)

Figure 7. Time series plots of DSM2-simulated (in blue) versus ANN (in green) and PINN (in orange) estimated salinity at each of the three trained locations (**a**) Martinez, (**b**) Chipps Island, (**c**) Pittsburg on Fold 2 test data, which correspond to 1 January 1996 to 31 December 2000. Detailed values of four evaluation metrics of ANN and PINN are marked at each location.



(**a**)

(**b**)

(**c**)

Figure 8. Time series plots of DSM2-simulated (in blue) versus PINN (in orange) and FoNet (in black) estimated salinity at each of the three trained locations (**a**) Martinez, (**b**) Chipps Island, (**c**) Pittsburg on Fold 2 test data, which correspond to 1 January 1996 to 31 December 2000. Detailed values of four evaluation metrics of PINN and FoNet are marked at each location.

Figure 9 shows scatter plots comparing the estimated salinity values of the three models to the target salinity values (DSM2-simulated). The first, second, and third rows correspond to locations Martinez, Chipps Island, and Pittsburg, respectively, and the first, second, and third columns correspond to the ANN, PINN, and FoNet models, respectively. Each dot on a scatter plot represents a salinity data point in the testing dataset of one of the five folds, with its x-coordinate value being the model's estimated salinity value and its y-coordinate value being the target salinity value. The scatter plots reveal that the ANN models tend to either underestimate or overestimate the salinity values. Specifically, the ANN model underestimates salinity values at Martinez, overestimates salinity values at Chipps Island, and overestimates salinity values more significantly at Pittsburg. This suggests that the ANN model shifts from underestimation to overestimation from the westernmost to the easternmost location. In contrast, the physics-informed models, PINN and FoNet, do not exhibit such location-dependent underestimation/overestimation behaviors. Both models seem to perform similarly at all three locations. At Pittsburg, FoNet estimates the target salinity slightly more accurately than PINN; the latter overestimates some of the salinity values in the mid-range while the former does not. The averaged statistical metrics of the test datasets of the 5 folds, displayed in each scatter plot, confirm these observations.

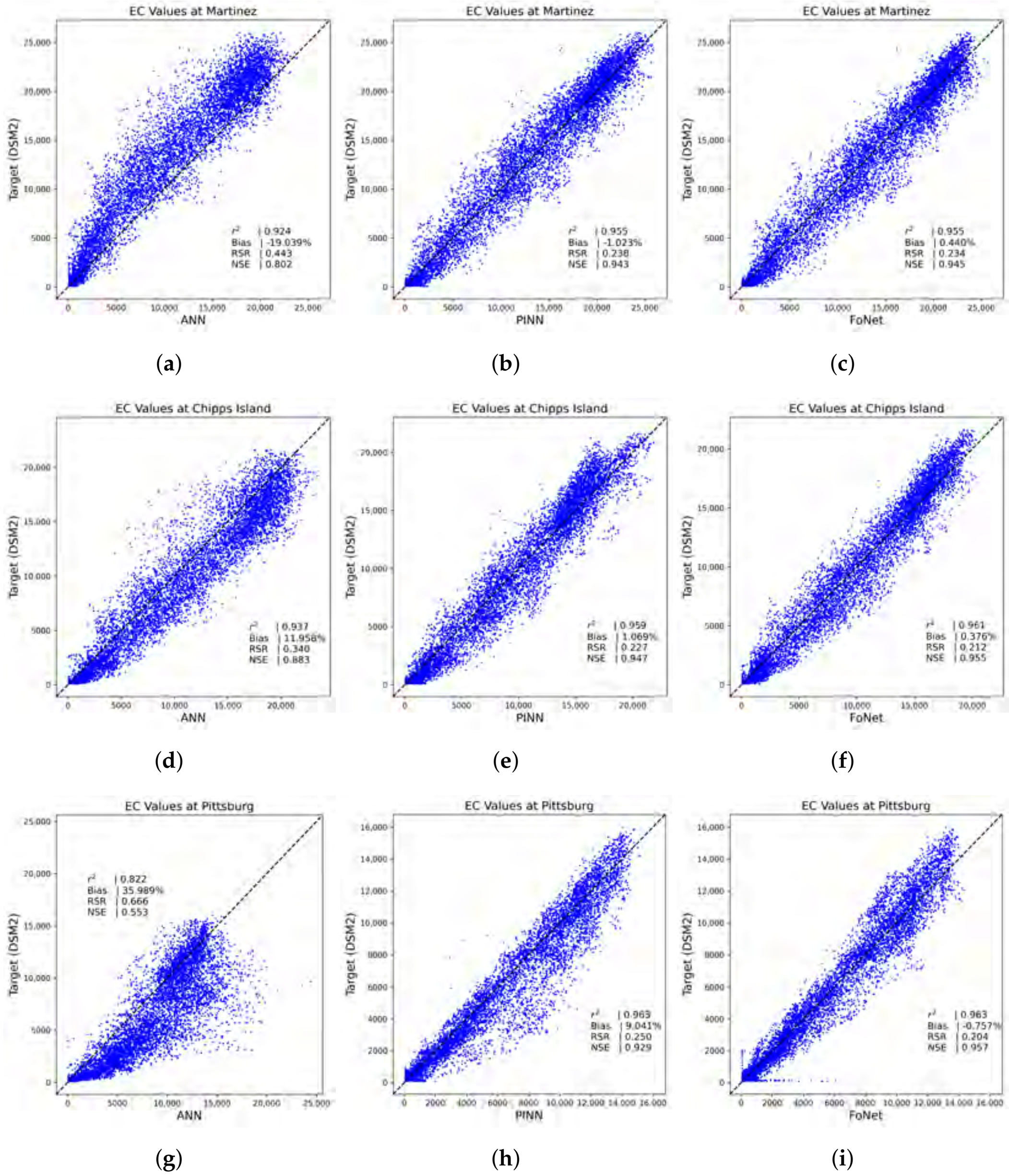
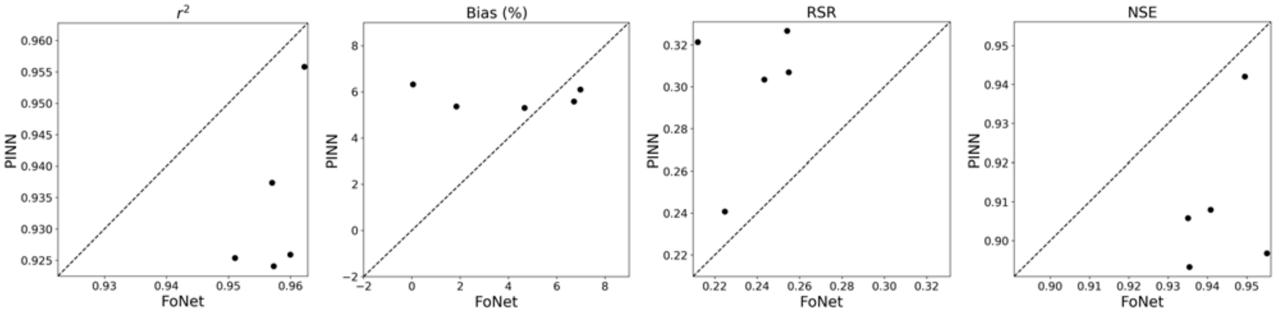


Figure 9. Scatter plots of target salinity (DSM2-simulated) vs. model-estimated salinity on the test datasets of all five folds. (**a**) ANN at Martinez (**b**) PINN at Martinez (**c**) FoNet at Martinez (**d**) ANN at Chipps Island (**e**) PINN at Chipps Island (**f**) FoNet at Chipps Island (**g**) ANN at Pittsburg (**h**) PINN at Pittsburg (**i**) FoNet at Pittsburg. Detailed 5-fold averaged values of four evaluation metrics are marked at each scatter plot.

3.2. Performance Results on Independent Untrained Location

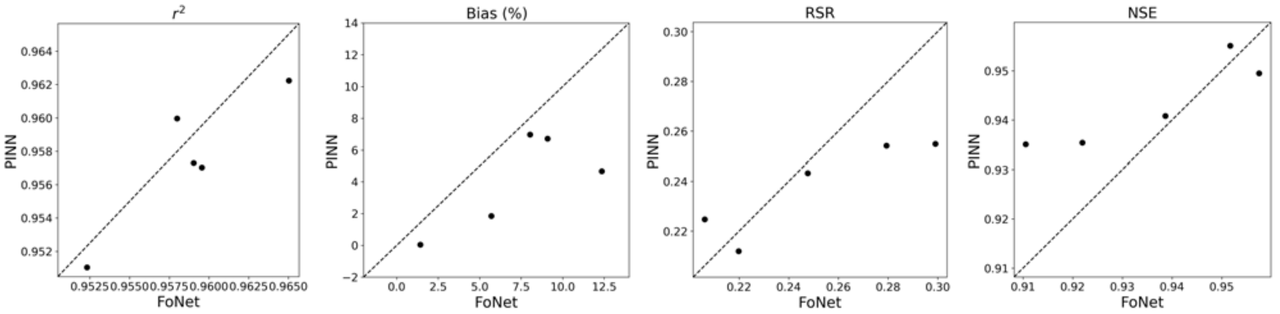
In this subsection, we evaluate the three machine-learning models at an untrained location, Port Chicago. By validating a model at an untrained location we can measure its predictive capacity at an unknown location for the testing time period at hand.

Figures 10 and 11 illustrate the comparison of the four evaluation metrics (r^2, percentage bias, RSR, NSE) for the three models on the Port Chicago test datasets of the five folds in scatter plots. Figure 10 compares the performance of ANN and PINN, while Figure 11 compares the performance of PINN and FoNet. Each dot in a scatter plot corresponds to one of the five folds. The scatter plots in Figure 10 show that PINN outperforms ANN, as it achieves higher r^2 and NSE values, and lower RSR values for all five folds. Although PINN obtains a slightly higher percent bias than ANN for a couple of folds, the percentage biases are smaller for PINN in the other three folds. The scatter plots in Figure 11 indicate that PINN performs slightly better than FoNet at Port Chicago, as PINN achieves smaller percent bias and RSR, and larger NSE than FoNet for most of the five folds. Appendix A.3 contains tables with the exact values of the metrics for each fold.



(**a**) Test r^2 (**b**) Test Bias (%) (**c**) Test RSR (**d**) Test NSE

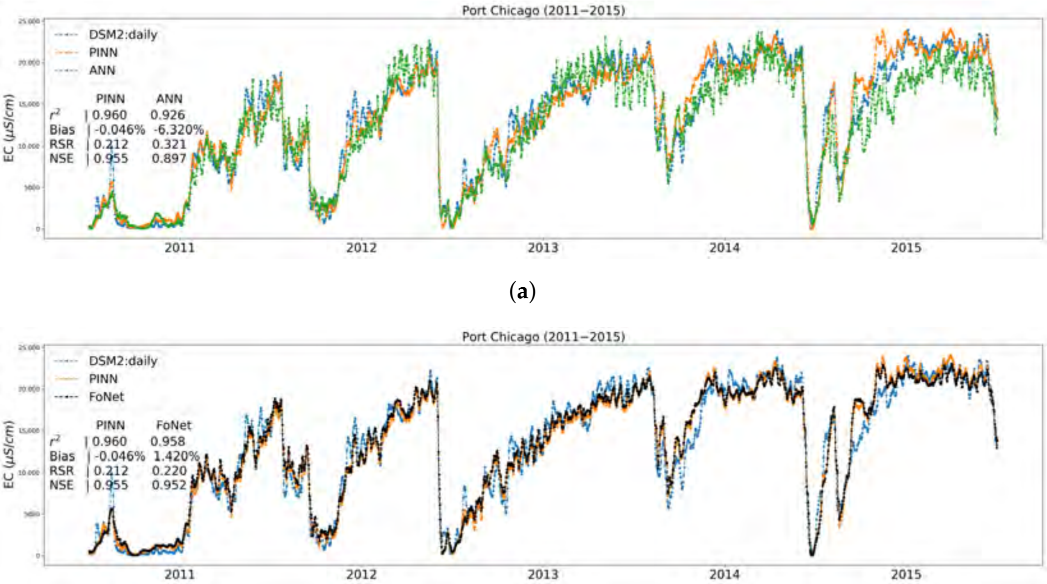
Figure 10. Scatter plots for each of the four evaluation metrics: (**a**) r^2 (**b**) percent bias (**c**) RSR (**d**) NSE for ANN and PINN models. A dot corresponds to a fold's test dataset results such that its x-coordinate value is the PINN evaluation result and y-coordinate value is the ANN evaluation result.



(**a**) Test r^2 (**b**) Test Bias (%) (**c**) Test RSR (**d**) Test NSE

Figure 11. Scatter plots for each of the four evaluation metrics: (**a**) r^2 (**b**) percent bias (**c**) RSR (**d**) NSE for PINN and FoNet models. A dot corresponds to a fold's test dataset results such that its x-coordinate value is the FoNet evaluation result and y-coordinate value is the PINN evaluation result.

The time series plots in Figure 12 compare the estimated salinity values of our three models to the target salinity values (DSM2-simulated). Once again, for better visibility, we consider two models at a time: the top time series plots show DSM2-simulated salinity values (in blue) with estimated salinity values of ANN (in green) and PINN (in orange), and the bottom time series plots show DSM2-simulated salinity values (in blue) with estimated salinity values of PINN (in orange) and FoNet (in black). These time series correspond to the test dataset of Fold 5, spanning from 1 January 2011 to 31 December 2015, at Port Chicago. Visual inspection indicates that ANN is less accurate than PINN and FoNet in estimating salinity at Port Chicago. Particularly for high salinity values in 2015, the performance of ANN drops as it noticeably underestimates the salinity. In contrast, both PINN and FoNet outperform ANN, and show similar levels of accuracy in salinity estimation. Appendix A.5 provides additional time series plots for the other folds.

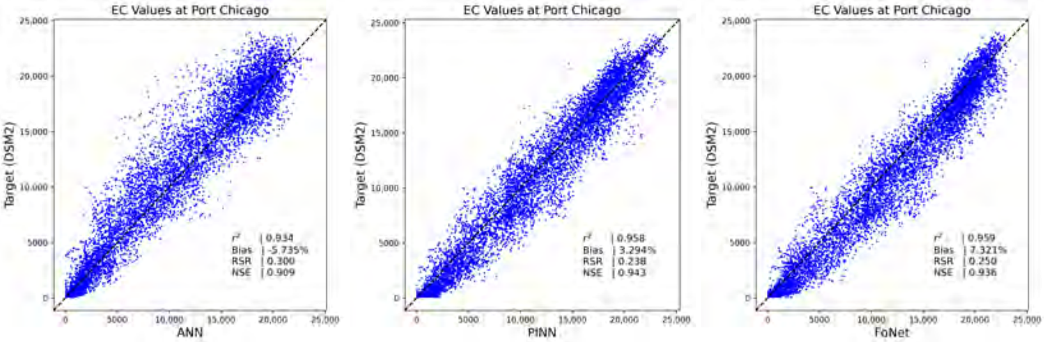


(**a**)

(**b**)

Figure 12. Time series plots at Port Chicago on Fold 5 test data, which correspond to 1 January 2011 to 31 December 2015. (**a**) Time series plots of DSM2-simulated (in blue) versus ANN (in green) and PINN (in orange) estimated salinity (**b**) Time series plots of DSM2-simulated (in blue) versus PINN (in orange) and FoNet (in black) estimated salinity. Detailed values of four evaluation metrics of the models are marked on the plots.

Figure 13 displays a comparison of the estimated salinity values of the three models (ANN, PINN, FoNet) to the target salinity values (DSM2-simulated) at Port Chicago in scatter plots. Each dot on a scatter plot represents a salinity data point in the testing dataset of one of the five folds, with its x-coordinate value being the model's estimated salinity value and its y-coordinate value being the target salinity value. All three models show no clear bias towards underestimation or overestimation at Port Chicago. However, ANN deviates from the target salinity more frequently than PINN and FoNet, especially for salinity values in the mid-to-high range. The salinity estimations of PINN and FoNet are quite similar. The averaged statistical metrics of the test datasets of the 5 folds are displayed in each scatter plot and confirm these observations. In particular, smaller NSE and larger RSR values for ANN indicate its under-performance in comparison to PINN and FoNet. PINN and FoNet have similar values in their four evaluation metrics.



(**a**) (**b**) (**c**)

Figure 13. Scatter plots of target salinity (DSM2-simulated) vs. model-estimated salinity on the Port Chicago test datasets of all five folds. (**a**) ANN (**b**) PINN (**c**) FoNet. Detailed 5-fold averaged values of four evaluation metrics are marked at each scatter plot.

4. Discussion

4.1. Implications

Our study has significant scientific implications. For the first time, we introduce flow–salinity modeling in the Delta using physics-informed neural network (PINN) models and demonstrate their potential to bridge the gap between process-based models and data-driven models. As machine-learning models with neural network structures, PINN models can efficiently simulate salinity, such as data-driven artificial neural network (ANN) models, while incorporating the advection–dispersion equation that governs the underlying physical laws of the flow–salinity relationship, such as process-based models. Furthermore, the PINN models we develop can be applied to other estuarine environments where flow–salinity modeling is of interest.

Our study also demonstrates the practical advantages of using PINN models, which offer comparable efficiency to data-driven ANN models while improving the accuracy of salinity simulations in the Delta. In terms of statistical metrics r^2, bias, RSR, and NSE, the PINN models outperform the ANN model across the board. This is particularly appealing to real-time operations and long-term planning which both require desirable estimates on the salinity levels.

4.2. Limitations and Future Work

Although this study demonstrates the improved salinity estimation capabilities of physics-informed neural networks compared to conventional neural networks, it is limited to only four locations which all lie within the flat estuary waterway in the western part of the Delta. There are other important locations of ecological significance in the interior parts of the Delta, where salinity values are less significant due to less impact of seawater and waterways between locations are irregularly shaped, making it difficult to apply the 1-dimensional advection–dispersion equation. In future studies, we plan to explore different methodologies of PINN models to model salinity at these locations. We acknowledge that our PINN models are simplified by assuming constant values for the estuary cross-sectional area A and longitudinal dispersion coefficient K in the advection–dispersion equation. In reality, these parameters can vary both spatially and temporally. However, we emphasize that they were treated as constants in our PINN model for the sake of simplicity. Our main focus was to conduct a proof-of-concept study of the PINN model and compare it with a simpler ANN model. Moreover, we currently lack advection–dispersion coefficient data which are not available from field observations, as it varies with the cross-sectional area. Obtaining the advection–dispersion coefficient data from modeling is an ongoing research effort for us. In future studies, we plan to incorporate space-variant cross-sectional area and advection–dispersion coefficients as additional variables in our PINN models. Furthermore, while the DSM2-simulated salinity values are suitable for validating our proof-of-concept models, they are generally less noisy than historical observed data, which can make it easier for the models to train on. To further strengthen the salinity estimation capabilities of our PINN models, we plan to evaluate them on historically observed salinity data.

In this study, we explore a machine-learning model called FoNet, a variation of a simple PINN model with an MLP architecture. However, there are other variations of PINN models with different neural network architectures. For instance, the long short-term memory (LSTM) architecture has been explored in predicting salinity in navigable waterways in Belgium [40]. There are extensions of FoNet, such as the Spatio-temporal Fourier Feature Network [61], which aims to tackle differential equations exhibiting multi-scale behaviors by applying multiple Fourier feature encodings initialized with different frequencies to input variables. We plan to explore these and other variations of PINN models in the future.

5. Conclusions

Salinity modeling in the Sacramento–San Joaquin Delta of California has traditionally relied on two separate approaches: process-based models and data-driven models. In this study, novel machine-learning models based on the framework of physics-informed neural networks are developed and applied in estimating salinity in the study locations in the Delta. These models integrate the flow–salinity relationships of process-based models with the computational efficiency of data-driven artificial neural network models. Specifically, the advection–dispersion equation, which describes the flow–salinity relationship, is embedded into the loss function of a multilayer perceptron. The findings of the study show that these new models outperform a benchmark artificial neural network in accurately estimating salinity levels in the study locations in the Delta. The efficiency and improved accuracy in this proof-of-concept study indicate the promising potential of these proposed models for salinity estimation in the Delta.

Author Contributions: Conceptualization, P.S., Z.B., Z.D. and M.H.; methodology, D.M.R., Z.B., M.H., Z.D. and P.S.; software, D.M.R.; validation, M.H. and Z.B.; formal analysis, D.M.R., Z.B. and M.H.; investigation, D.M.R. and M.H.; resources, M.R., Y.Z., R.H. and P.N.; data curation, B.T., R.H., P.N. and Y.Z.; writing—original draft, D.M.R. and M.H.; writing—review and editing, Z.B., P.S., F.C., Z.D., S.Q., Y.Z., R.H., P.N., B.D. and J.A.; visualization, D.M.R., M.H. and R.H.; supervision, M.H., Z.B., P.S., F.C., Z.D. and J.A.; project administration, P.S.; funding acquisition, P.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the California Department of Water Resources and the University of California, Davis grant number 4600014165-01.

Data Availability Statement: The data presented in this study are openly available at the following link: https://data.cnra.ca.gov/dataset/dsm2-v8-2-1, accessed on 1 April 2022. The source code of the developed models was uploaded to a GitHub repository and is available from the lead author upon request.

Acknowledgments: The authors would like to thank the editors and three anonymous reviewers for providing thoughtful and insightful comments that helped to improve the quality of this study. The views expressed in this paper are those of the authors, and not of the State of California.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Appendix A.1. ANN: Number of Layer Choices

Table A1. ANN performance for 2, 3, 4 layers on Fold 1 test dataset at Port Chicago. The best performing values are highlighted in bold fonts.

	ANN		
Evaluation Metrics	**2 Layers**	**3 Layers**	**4 Layers**
r^2	0.958	0.965	**0.968**
Bias	−6.491	−6.242	**−5.302**
RSR	0.338	0.313	**0.307**
NSE	0.886	0.902	**0.906**

Appendix A.2. Hyperparameter Choices

Table A2. ANN hyperparameters choices.

	Fold1		**Fold2**		**Fold3**		**Fold4**		**Fold5**	
Hidden Layer	**# Neuron**	**Activation**	**# Neuron**	**Activation**	**# Neuron**	**Activation**	**# Neuron**	**Activation**	**# Neuron**	**Activation**
hidden 1	32	elu	32	relu	32	relu	32	tanh	32	relu
hidden 2	32	elu	8	relu	24	relu	24	relu	4	tanh
hidden 3	8	tanh	14	relu	16	elu	2	elu	6	elu
hidden 4	14	sigmoid	6	tanh	4	sigmoid	14	relu	12	elu

Table A3. PINN hyperparameters choices.

	Fold1		Fold2		Fold3		Fold4		Fold5	
Hidden Layer	**# Neuron**	**Activation**	**# Neuron**	**Activation**	**# Neuron**	**Activation**	**# Neuron**	**Activation**	**# Neuron**	**Activation**
hidden 1	24	relu	32	relu	24	elu	32	tanh	28	tanh
hidden 2	12	tanh	16	tanh	12	sigmoid	16	tanh	8	sigmoid

Table A4. FoNet hyperparameters choices.

	Fold1		Fold2		Fold3		Fold4		Fold5	
Hidden Layer	**# Neuron**	**Activation**	**# Neuron**	**Activation**	**# Neuron**	**Activation**	**# Neuron**	**Activation**	**# Neuron**	**Activation**
encoding	24		28		32		16		8	
hidden 1	32	tanh	16	tanh	12	tanh	28	elu	16	tanh
hidden 2	10	elu	4	sigmoid	16	relu	8	tanh	10	tanh

Appendix A.3. Detailed Values for Box and Whisker Plots

Table A5. r^2 values of three machine-learning models on training datasets. The best performing values are highlighted in bold fonts.

	Fold1			Fold2			Fold3			Fold4			Fold5			Average		
Station Name	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**
Martinez	0.938	**0.961**	0.961	0.927	**0.957**	0.952	0.951	0.967	**0.972**	0.923	0.963	**0.969**	0.928	**0.965**	0.961	0.933	0.963	**0.963**
Port Chicago	0.948	**0.966**	0.965	0.932	**0.960**	0.956	0.956	0.970	**0.974**	0.933	0.966	**0.974**	0.935	**0.968**	0.965	0.941	0.966	**0.967**
Chipps Island	0.951	0.966	**0.968**	0.929	0.959	**0.962**	0.954	0.973	**0.977**	0.935	0.969	**0.977**	0.935	**0.970**	0.968	0.941	0.967	**0.970**
Pittsburg	0.853	0.962	**0.974**	0.852	0.953	**0.969**	0.878	0.975	**0.978**	0.829	0.968	**0.979**	0.841	**0.975**	0.967	0.851	0.967	**0.974**

Table A6. r^2 values of three machine-learning models on test datasets. The best performing values are highlighted in bold fonts.

	Fold1			Fold2			Fold3			Fold4			Fold5			Average		
Station Name	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**
Martinez	0.937	**0.961**	0.959	0.937	0.960	**0.961**	0.908	0.944	**0.948**	0.912	0.954	**0.954**	0.924	**0.957**	0.954	0.924	0.955	**0.955**
Port Chicago	0.937	0.957	**0.960**	0.956	0.962	**0.965**	0.925	0.951	**0.952**	0.924	0.957	**0.959**	0.926	**0.960**	0.958	0.934	0.958	**0.959**
Chipps Island	0.933	0.950	**0.960**	0.964	**0.965**	0.963	0.937	0.958	**0.959**	0.930	0.960	**0.965**	0.920	**0.962**	0.960	0.937	0.959	**0.961**
Pittsburg	0.820	0.939	**0.948**	0.879	**0.962**	0.961	0.845	0.964	**0.970**	0.778	**0.979**	0.979	0.787	**0.970**	0.960	0.822	0.963	**0.963**

Table A7. Bias values of three machine-learning models on training datasets. The best performing values are highlighted in bold fonts.

	Fold1			Fold2			Fold3			Fold4			Fold5			Average		
Station Name	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**
Martinez	−21.874	−11.528	**−6.950**	−14.911	**−0.411**	5.117	−18.398	**−0.163**	−0.957	−22.276	**−2.386**	−4.792	−12.369	**1.730**	2.825	−17.966	−2.552	**−0.952**
Port Chicago	−9.017	−7.348	**−1.886**	**−1.727**	3.670	13.094	−5.759	**5.394**	6.369	−9.957	**2.978**	3.245	**2.156**	4.105	6.845	−4.861	**1.760**	5.534
Chipps Island	8.201	**−6.066**	−6.773	15.562	4.887	**4.238**	10.801	**−0.191**	−2.297	6.214	**−1.454**	−3.619	21.264	**1.520**	5.597	12.408	**−0.261**	−0.571
Pittsburg	22.011	12.007	**−6.960**	33.481	23.983	**−0.474**	16.016	0.540	**0.009**	20.395	**0.451**	−0.707	56.357	**0.077**	5.697	29.652	7.412	**−0.487**

Table A8. Bias values of three machine-learning models on test datasets. The best performing values are highlighted in bold fonts.

	Fold1			Fold2			Fold3			Fold4			Fold5			Average		
Station Name	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**
Martinez	−16.656	**4.032**	7.369	−22.031	−9.393	**−3.583**	−20.185	**0.082**	0.136	−18.697	1.616	**−0.127**	−17.627	**−1.453**	−1.595	−19.039	−1.023	**0.440**
Port Chicago	−5.302	**4.666**	12.357	−5.369	**−1.848**	5.702	**−5.585**	6.715	9.091	**−6.101**	6.982	8.034	−6.320	**−0.046**	1.420	−5.735	**3.294**	7.321
Chipps Island	8.857	**0.543**	3.183	17.131	2.280	**−1.999**	14.103	1.209	**−0.441**	10.938	3.023	**1.459**	8.761	−1.709	**−0.323**	11.958	1.069	**0.376**
Pittsburg	22.409	17.181	**−4.021**	59.230	26.537	**−0.388**	30.237	**−0.126**	1.059	35.788	**−0.841**	−1.473	32.283	2.453	**1.040**	35.989	9.041	**−0.757**

Table A9. RSR values of three machine-learning models on training datasets. The best performing values are highlighted in bold fonts.

	Fold1			Fold2			Fold3			Fold4			Fold5			Average		
Station Name	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**
Martinez	0.448	0.279	**0.239**	0.399	**0.222**	0.251	0.388	0.197	**0.181**	0.465	0.214	**0.205**	0.353	**0.205**	0.223	0.411	0.224	**0.220**
Port Chicago	0.280	0.229	**0.199**	0.288	**0.224**	0.312	0.246	0.204	**0.201**	0.314	0.204	**0.186**	0.285	**0.202**	0.233	0.283	**0.213**	0.226
Chipps Island	0.276	0.231	**0.225**	0.383	0.244	**0.219**	0.286	0.181	**0.165**	0.291	0.194	**0.173**	0.419	**0.185**	0.213	0.331	0.207	**0.199**
Pittsburg	0.501	0.262	**0.200**	0.603	0.398	**0.187**	0.439	0.165	**0.153**	0.513	0.188	**0.154**	0.832	**0.163**	0.207	0.578	0.235	**0.180**

Table A10. RSR values of three machine-learning models on test datasets. The best performing values are highlighted in bold fonts.

	Fold1			Fold2			Fold3			Fold4			Fold5			Average		
Station Name	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**
Martinez	0.409	**0.239**	0.263	0.396	0.255	**0.207**	0.484	0.247	**0.244**	0.457	0.222	**0.219**	0.471	**0.228**	0.236	0.443	0.238	**0.234**
Port Chicago	0.307	**0.255**	0.299	0.241	0.225	**0.206**	0.327	**0.254**	0.279	0.303	**0.243**	0.248	0.321	**0.212**	0.220	0.300	**0.238**	0.250
Chipps Island	0.334	0.284	**0.241**	0.302	0.227	**0.204**	0.385	**0.212**	0.214	0.342	0.210	**0.192**	0.340	**0.204**	0.211	0.340	0.227	**0.212**
Pittsburg	0.608	0.370	**0.273**	0.708	0.355	**0.217**	0.581	0.194	**0.176**	0.734	**0.149**	0.150	0.698	**0.183**	0.204	0.666	0.250	**0.204**

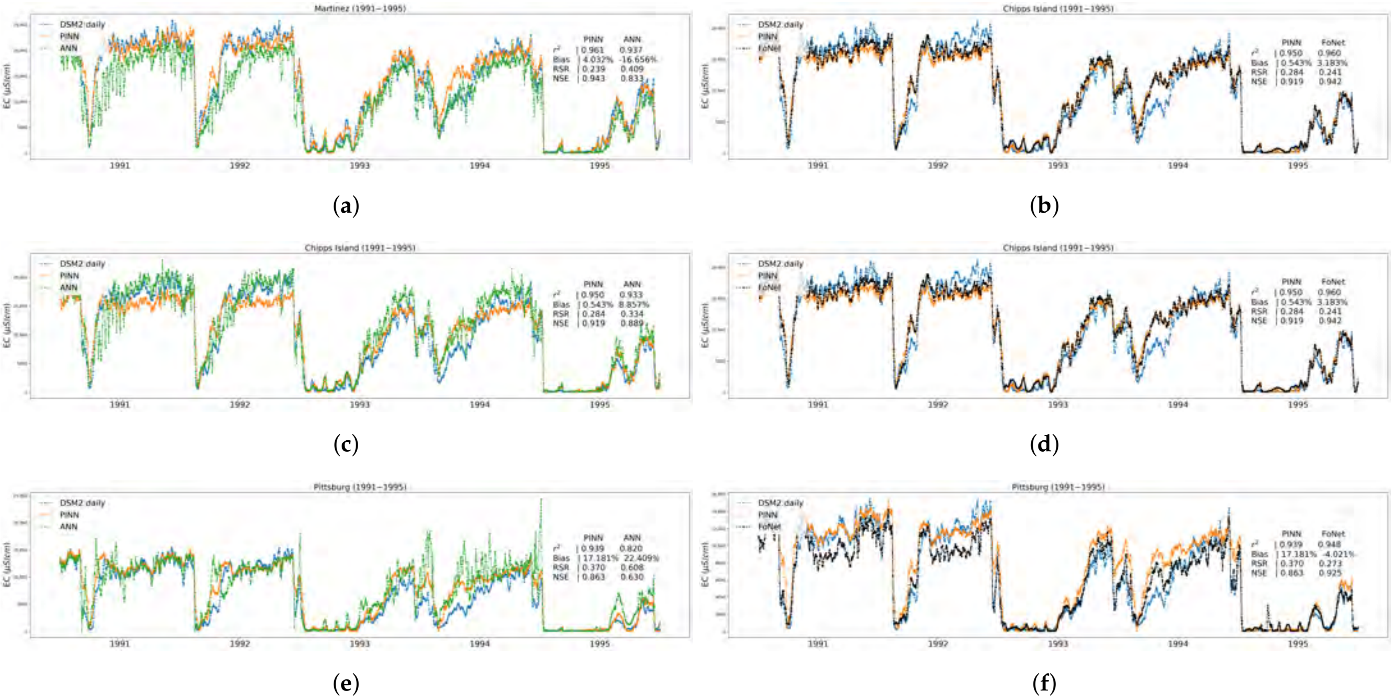
Table A11. NSE values of three machine-learning models on training datasets. The best performing values are highlighted in bold fonts.

	Fold1			Fold2			Fold3			Fold4			Fold5			Average		
Station Name	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**
Martinez	0.799	0.922	**0.943**	0.841	**0.951**	0.937	0.850	0.961	**0.967**	0.784	0.954	**0.958**	0.875	**0.958**	0.950	0.830	0.949	**0.951**
Port Chicago	0.921	0.948	**0.960**	0.917	**0.950**	0.902	0.940	0.958	**0.960**	0.901	0.958	**0.966**	0.919	**0.959**	0.946	0.920	**0.955**	0.947
Chipps Island	0.924	0.947	**0.949**	0.853	0.940	**0.952**	0.918	0.967	**0.973**	0.916	0.962	**0.970**	0.824	**0.966**	0.955	0.887	0.957	**0.960**
Pittsburg	0.749	0.931	**0.960**	0.636	0.842	**0.965**	0.807	0.973	**0.977**	0.737	0.965	**0.976**	0.307	**0.973**	0.957	0.647	0.937	**0.967**

Table A12. NSE values of three machine-learning models on test datasets. The best performing values are highlighted in bold fonts.

	Fold1			Fold2			Fold3			Fold4			Fold5			Average		
Station Name	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**	**ANN**	**PINN**	**FoNet**
Martinez	0.833	**0.943**	0.931	0.843	0.935	**0.957**	0.766	0.939	**0.941**	0.792	0.951	**0.952**	0.778	**0.948**	0.944	0.802	0.943	**0.945**
Port Chicago	0.906	**0.935**	0.911	0.942	0.950	**0.958**	0.893	**0.935**	0.922	0.908	**0.941**	0.939	0.897	**0.955**	0.952	0.909	**0.943**	0.936
Chipps Island	0.889	0.919	**0.942**	0.909	0.948	**0.958**	0.851	**0.955**	0.954	0.883	0.956	**0.963**	0.885	**0.958**	0.956	0.883	0.947	**0.955**
Pittsburg	0.630	0.863	**0.925**	0.498	0.874	**0.953**	0.663	0.962	**0.969**	0.461	**0.978**	0.978	0.512	**0.966**	0.958	0.553	0.929	**0.957**

Appendix A.4. Time Series Plots at Three Trained Locations

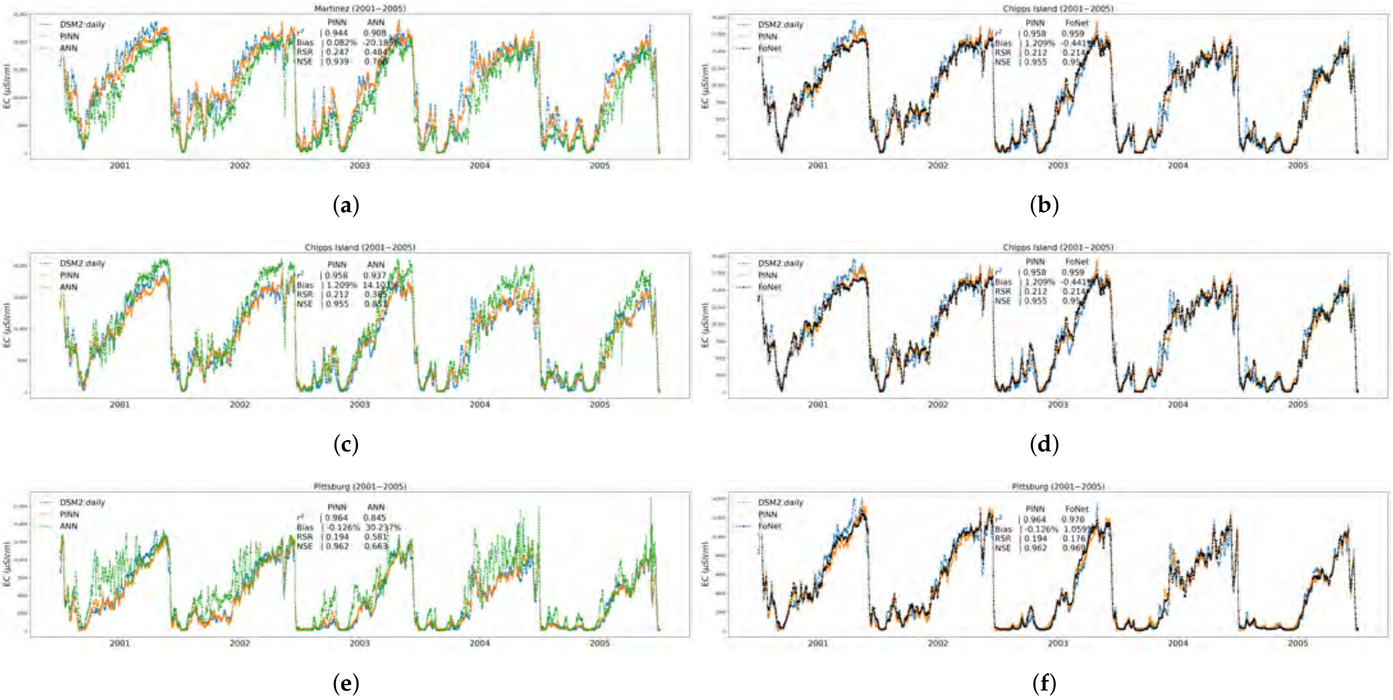


(**a**) (**b**)

(**c**) (**d**)

(**e**) (**f**)

Figure A1. Time series plots of DSM2-simulated (in blue) versus ANN (in green) and PINN (in orange) and FoNet (in black) estimated salinity at each of the three trained locations Martinez, Chipps Island, Pittsburg on Fold 1 test data, which correspond to 1 January 1991 to 31 December 1995. DSM2, ANN, PINN: (**a**) Martinez (**c**) Chipps Island (**e**) Pittsburg; DSM2, PINN, FoNet: (**b**) Martinez (**d**) Chipps Island (**f**) Pittsburg. Detailed values of four evaluation metrics of the models are marked at each location.

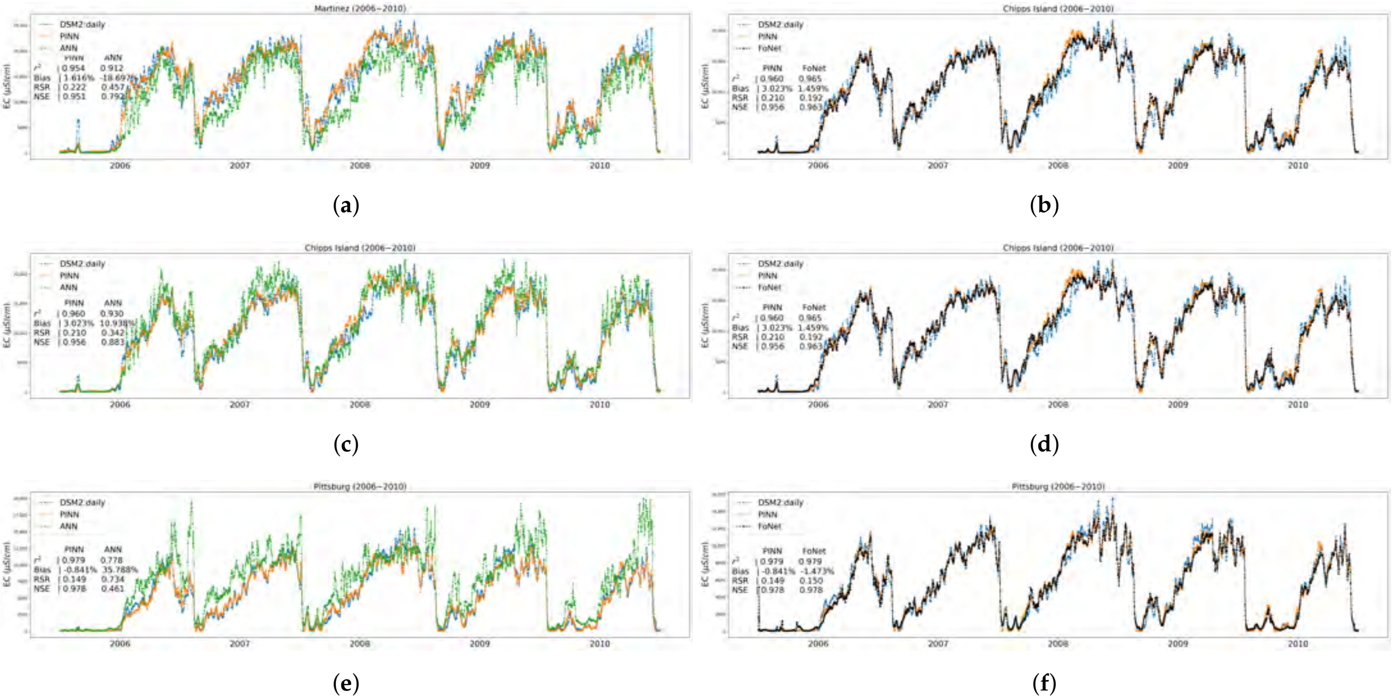


(**a**) (**b**)

(**c**) (**d**)

(**e**) (**f**)

Figure A2. Time series plots of DSM2-simulated (in blue) versus ANN (in green) and PINN (in orange) and FoNet (in black) estimated salinity at each of the three trained locations Martinez, Chipps Island, Pittsburg on Fold 3 test data, which correspond to 1 January 2001 to 31 December 2005. DSM2, ANN, PINN: (**a**) Martinez (**c**) Chipps Island (**e**) Pittsburg; DSM2, PINN, FoNet: (**b**) Martinez (**d**) Chipps Island (**f**) Pittsburg. Detailed values of four evaluation metrics of the models are marked at each location.

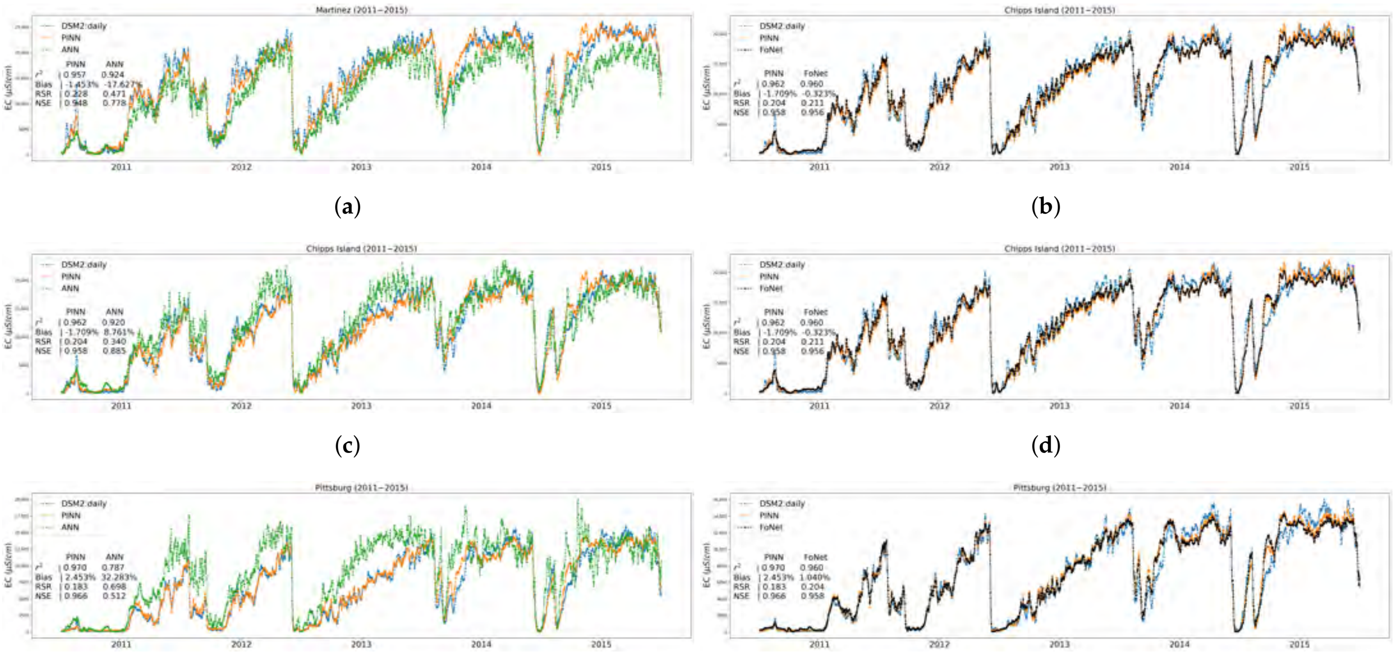


(**a**) (**b**)

(**c**) (**d**)

(**e**) (**f**)

Figure A3. Time series plots of DSM2-simulated (in blue) versus ANN (in green) and PINN (in orange) and FoNet (in black) estimated salinity at each of the three trained locations Martinez, Chipps Island, Pittsburg on Fold 4 test data, which correspond to 1 January 2006 to 31 December 2010. DSM2, ANN, PINN: (**a**) Martinez (**c**) Chipps Island (**e**) Pittsburg; DSM2, PINN, FoNet: (**b**) Martinez (**d**) Chipps Island (**f**) Pittsburg. Detailed values of four evaluation metrics of the models are marked at each location.



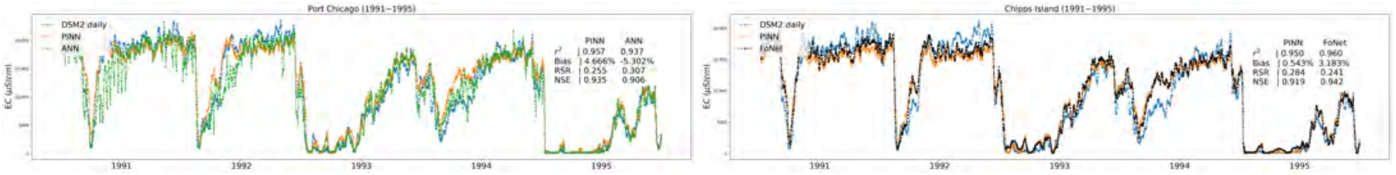
(**a**) (**b**)

(**c**) (**d**)

(**e**) (**f**)

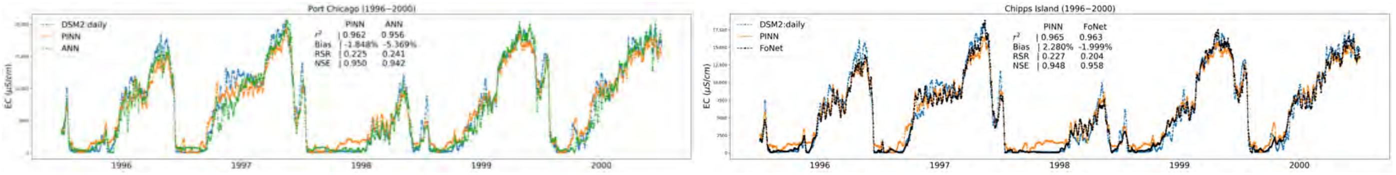
Figure A4. Time series plots of DSM2-simulated (in blue) versus ANN (in green) and PINN (in orange) and FoNet (in black) estimated salinity at each of the three trained locations Martinez, Chipps Island, Pittsburg on Fold 5 test data, which correspond to 1 January 2011 to 31 December 2015. DSM2, ANN, PINN: (**a**) Martinez (**c**) Chipps Island (**e**) Pittsburg; DSM2, PINN, FoNet: (**b**) Martinez (**d**) Chipps Island (**f**) Pittsburg. Detailed values of four evaluation metrics of the models are marked at each location.

Appendix A.5. Time Series Plots at Port Chicago, an Independent Test Location



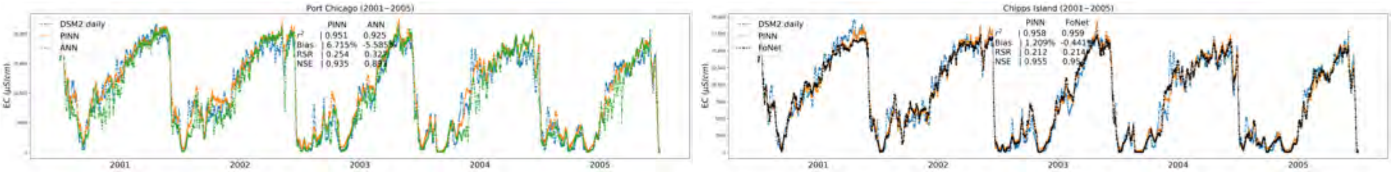
(**a**) (**b**)

Figure A5. Time series plots of DSM2-simulated (in blue) versus ANN (in green) and PINN (in orange) and FoNet (in black) estimated salinity at untrained location Port Chicago on Fold 1 test data, which correspond to 1 January 1991 to 31 December 1995. (**a**) DSM2, ANN, PINN (**b**) DSM2, PINN, FoNet. Detailed values of four evaluation metrics of the models are marked at each location.



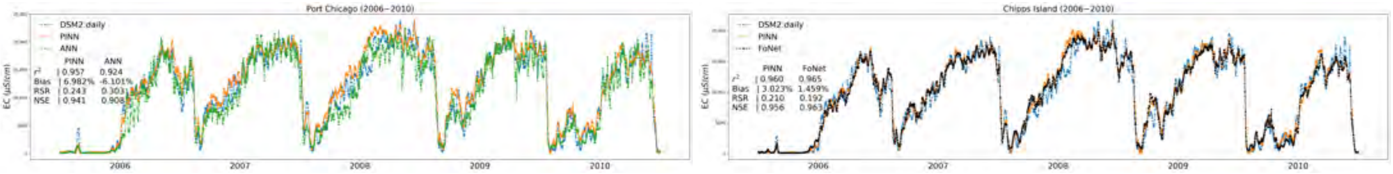
(**a**) (**b**)

Figure A6. Time series plots of DSM2-simulated (in blue) versus ANN (in green) and PINN (in orange) and FoNet (in black) estimated salinity at untrained location Port Chicago on Fold 2 test data, which correspond to 1 January 1996 to 31 December 2000. (**a**) DSM2, ANN, PINN (**b**) DSM2, PINN, FoNet. Detailed values of four evaluation metrics of the models are marked at each location.



(**a**) (**b**)

Figure A7. Time series plots of DSM2-simulated (in blue) versus ANN (in green) and PINN (in orange) and FoNet (in black) estimated salinity at untrained location Port Chicago on Fold 3 test data, which correspond to 1 January 2001 to 31 December 2005. (**a**) DSM2, ANN, PINN (**b**) DSM2, PINN, FoNet. Detailed values of four evaluation metrics of the models are marked at each location.



(**a**) (**b**)

Figure A8. Time series plots of DSM2-simulated (in blue) versus ANN (in green) and PINN (in orange) and FoNet (in black) estimated salinity at untrained location Port Chicago on Fold 4 test data, which correspond to 1 January 2006 to 31 December 2010. (**a**) DSM2, ANN, PINN (**b**) DSM2, PINN, FoNet. Detailed values of four evaluation metrics of the models are marked at each location.

References

1. Alber, M. A conceptual model of estuarine freshwater inflow management. *Estuaries* **2002**, *25*, 1246–1261. [CrossRef]
2. Rath, J.S.; Hutton, P.H.; Chen, L.; Roy, S.B. A hybrid empirical-Bayesian artificial neural network model of salinity in the San Francisco Bay-Delta estuary. *Environ. Model. Softw.* **2017**, *93*, 193–208. [CrossRef]
3. Xu, J.; Long, W.; Wiggert, J.D.; Lanerolle, L.W.; Brown, C.W.; Murtugudde, R.; Hood, R.R. Climate forcing and salinity variability in Chesapeake Bay, USA. *Estuaries Coasts* **2012**, *35*, 237–261. [CrossRef]
4. Tran Anh, D.; Hoang, L.P.; Bui, M.D.; Rutschmann, P. Simulating future flows and salinity intrusion using combined one-and two-dimensional hydrodynamic modelling—The case of Hau River, Vietnamese Mekong delta. *Water* **2018**, *10*, 897. [CrossRef]
5. Mulamba, T.; Bacopoulos, P.; Kubatko, E.J.; Pinto, G.F. Sea-level rise impacts on longitudinal salinity for a low-gradient estuarine system. *Clim. Chang.* **2019**, *152*, 533–550. [CrossRef]
6. Doni, F.; Gasperini, A.; Soares, J.T. What is the SDG 13? In *SDG13–Climate Action: Combating Climate Change and Its Impacts*; Emerald Publishing Limited: Bingley, UK , 2020.
7. Sadoff, C.W.; Borgomeo, E.; Uhlenbrook, S. Rethinking water for SDG 6. *Nat. Sustain.* **2020**, *3*, 346–347. [CrossRef]
8. He, M.; Zhong, L.; Sandhu, P.; Zhou, Y. Emulation of a process-based salinity generator for the Sacramento–San Joaquin Delta of California via deep learning. *Water* **2020**, *12*, 2088. [CrossRef]
9. Verruijt, A. A note on the Ghyben-Herzberg formula. *Hydrol. Sci. J.* **1968**, *13*, 43–46. [CrossRef]
10. Todd, D.K.; Mays, L.W. *Groundwater Hydrology*; John Wiley & Sons: Hoboken, NJ, USA, 2004.
11. Myers, N.; Mittermeier, R.A.; Mittermeier, C.G.; Da Fonseca, G.A.; Kent, J. Biodiversity hotspots for conservation priorities. *Nature* **2000**, *403*, 853–858. [CrossRef]
12. Moyle, P.B.; Brown, L.R.; Durand, J.R.; Hobbs, J.A. Delta smelt: Life history and decline of a once-abundant species in the San Francisco Estuary. *San Fr. Estuary Watershed Sci.* **2016**, *14* . [CrossRef]
13. CSWRCB. *Water Right Decision 1641*; CSWRCB: Sacramento, CA, USA, 1999; p. 225 .
14. USFWS. *Formal Endangered Species Act Consultation on the Proposed Coordinated Operations of the Central Valley Project (CVP) and State Water Project (SWP)*; USFWS: Sacramento, CA, USA, 2008; p. 410 .
15. CDWR. Minimum Delta Outflow Program. In *Methodology for Flow and Salinity Estimates in the Sacramento-San Joaquin Delta and Suisun Marsh: 11th Annual Progress Report*; California Department of Water Resources: Sacramento, CA, USA, 1990 .
16. Denton, R.; Sullivan, G. Antecedent Flow-Salinity Relations: Application to Delta Planning Models. Contra Costa Water District Report 1993; p. 20. Available online: http://www.waterboards.ca.gov/waterrights/water_issues/programs/bay_delta/deltaflow/docs/exhibits/ccwd/spprt_docs/ccwd_denton_sullivan_1993.pdf (accessed on 1 April 2023).
17. Denton, R.A. Accounting for antecedent conditions in seawater intrusion modeling—Applications for the San Francisco Bay-Delta. In *Hydraulic Engineering*; ASCE: Reston, VA, USA, 1993; pp. 448–453.
18. CDWR. DSM2: Model Development. In *Methodology for Flow and Salinity Estimates in the Sacramento-San Joaquin Delta and Suisun Marsh: 18th Annual Progress Report*; California Department of Water Resources: Sacramento, CA, USA, 1997 .

19. Jayasundara, N.C.; Seneviratne, S.A.; Reyes, E.; Chung, F.I. Artificial neural network for Sacramento–San Joaquin Delta flow–salinity relationship for CalSim 3.0. *J. Water Resour. Plan. Manag.* **2020**, *146*, 04020015. [CrossRef]
20. Wilbur, R.; Munevar, A. Integration of CALSIM and Artificial Neural Networks Models for Sacramento-San Joaquin Delta Flow-Salinity Relationships. In *Methodology for Flow and Salinity Estimates in the Sacramento-San Joaquin Delta and Suisun Marsh: 22nd Annual Progress Report*; California Department of Water Resources: Sacramento, CA, USA, 2001 .
21. Mierzwa, M. CALSIM versus DSM2 ANN and G-model Comparisons. In *Methodology for Flow and Salinity Estimates in the Sacramento-San Joaquin Delta and Suisun Marsh: 23rd Annual Progress Report*; California Department of Water Resources: Sacramento, CA, USA, 2002 .
22. Seneviratne, S.; Wu, S. Enhanced Development of Flow-Salinity Relationships in the Delta Using Artificial Neural Networks: Incorporating Tidal Influence. In *Methodology for Flow and Salinity Estimates in the Sacramento-San Joaquin Delta and Suisun Marsh: 28th Annual Progress Report*; California Department of Water Resources: Sacramento, CA, USA, 2007 .
23. CDWR. Calibration and verification of DWRDSM. In *Methodology for Flow and Salinity Estimates in the Sacramento-San Joaquin Delta and Suisun Marsh: 12th Annual Progress Report*; California Department of Water Resources: Sacramento, CA, USA, 1991 .
24. Chen, L.; Roy, S.B.; Hutton, P.H. Emulation of a process-based estuarine hydrodynamic model. *Hydrol. Sci. J.* **2018**, *63*, 783–802. [CrossRef]
25. Cheng, R.T.; Casulli, V.; Gartner, J.W. Tidal, residual, intertidal mudflat (TRIM) model and its applications to San Francisco Bay, California. *Estuarine, Coast. Shelf Sci.* **1993**, *36*, 235–280. [CrossRef]
26. DeGeorge, J.F. *A Multi-Dimensional Finite Element Transport Model Utilizing a Characteristic-Galerkin Algorithm*; University of California: Davis, CA, USA, 1996.
27. MacWilliams, M.; Bever, A.J.; Foresman, E. 3-D simulations of the San Francisco Estuary with subgrid bathymetry to explore long-term trends in salinity distribution and fish abundance. *San Fr. Estuary Watershed Sci.* **2016**, *14*. [CrossRef]
28. Chao, Y.; Farrara, J.D.; Zhang, H.; Zhang, Y.J.; Ateljevich, E.; Chai, F.; Davis, C.O.; Dugdale, R.; Wilkerson, F. Development, implementation, and validation of a modeling system for the San Francisco Bay and Estuary. *Estuarine, Coast. Shelf Sci.* **2017**, *194*, 40–56. [CrossRef]
29. Qi, S.; He, M.; Bai, Z.; Ding, Z.; Sandhu, P.; Zhou, Y.; Namadi, P.; Tom, B.; Hoang, R.; Anderson, J. Multi-Location Emulation of a Process-Based Salinity Model Using Machine Learning. *Water* **2022**, *14*, 2030. [CrossRef]
30. MacWilliams, M.L.; Ateljevich, E.S.; Monismith, S.G.; Enright, C. An overview of multi-dimensional models of the Sacramento–San Joaquin Delta. *San Fr. Estuary Watershed Sci.* **2016**, *14*. [CrossRef]
31. Gopi, A.; Sharma, P.; Sudhakar, K.; Ngui, W.K.; Kirpichnikova, I.; Cuce, E. Weather Impact on Solar Farm Performance: A Comparative Analysis of Machine Learning Techniques. *Sustainability* **2023**, *15*, 439. [CrossRef]
32. Sharma, P.; Bora, B.J. A Review of Modern Machine Learning Techniques in the Prediction of Remaining Useful Life of Lithium-Ion Batteries. *Batteries* **2022**, *9*, 13. [CrossRef]
33. Barnes, G.W., Jr.; Chung, F.I. Operational planning for California water system. *J. Water Resour. Plan. Manag.* **1986**, *112*, 71–86. [CrossRef]
34. Draper, A.J.; Munévar, A.; Arora, S.K.; Reyes, E.; Parker, N.L.; Chung, F.I.; Peterson, L.E. CalSim: Generalized model for reservoir system analysis. *J. Water Resour. Plan. Manag.* **2004**, *130*, 480–489. [CrossRef]
35. Qi, S.; Bai, Z.; Ding, Z.; Jayasundara, N.; He, M.; Sandhu, P.; Seneviratne, S.; Kadir, T. Enhanced Artificial Neural Networks for Salinity Estimation and Forecasting in the Sacramento-San Joaquin Delta of California. *J. Water Resour. Plan. Manag.* **2021**, *147*, 04021069. [CrossRef]
36. Qi, S.; He, M.; Bai, Z.; Ding, Z.; Sandhu, P.; Chung, F.; Namadi, P.; Zhou, Y.; Hoang, R.; Tom, B.; et al. Novel Salinity Modeling Using Deep Learning for the Sacramento–San Joaquin Delta of California. *Water* **2022**, *14*, 3628. [CrossRef]
37. Shen, C.; Appling, A.P.; Gentine, P.; Bandai, T.; Gupta, H.; Tartakovsky, A.; Baity-Jesi, M.; Fenicia, F.; Kifer, D.; Li, L.; et al. Differentiable modeling to unify machine learning and physical models and advance Geosciences. *arXiv* **2023**, arXiv:2301.04027.
38. Daw, A.; Thomas, R.Q.; Carey, C.C.; Read, J.S.; Appling, A.P.; Karpatne, A. Physics-guided architecture (pga) of neural networks for quantifying uncertainty in lake temperature modeling. In Proceedings of the 2020 Siam International Conference on Data Mining, Cincinnati, OH, USA, 7–9 May 2020; pp. 532–540.
39. Hoedt, P.J.; Kratzert, F.; Klotz, D.; Halmich, C.; Holzleitner, M.; Nearing, G.S.; Hochreiter, S.; Klambauer, G. Mc-lstm: Mass-conserving lstm. In Proceedings of the International Conference on Machine Learning, Virtual Event, 18–24 July 2021; pp. 4275–4286.
40. Bertels, D.; Willems, P. Physics-informed machine learning method for modelling transport of a conservative pollutant in surface water systems. *J. Hydrol.* **2023**, *619*, 129354. [CrossRef]
41. Feng, D.; Liu, J.; Lawson, K.; Shen, C. Differentiable, Learnable, Regionalized Process-Based Models With Multiphysical Outputs can Approach State-Of-The-Art Hydrologic Prediction Accuracy. *Water Resour. Res.* **2022**, *58*, e2022WR032404. [CrossRef]
42. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [CrossRef]
43. Psichogios, D.C.; Ungar, L.H. A hybrid neural network-first principles approach to process modeling. *AIChE J.* **1992**, *38*, 1499–1511. [CrossRef]
44. Cuomo, S.; Di Cola, V.S.; Giampaolo, F.; Rozza, G.; Raissi, M.; Piccialli, F. Scientific machine learning through physics–informed neural networks: Where we are and what's next. *J. Sci. Comput.* **2022**, *92*, 88. [CrossRef]

45. Cedillo, S.; Núñez, A.G.; Sánchez-Cordero, E.; Timbe, L.; Samaniego, E.; Alvarado, A. Physics-Informed Neural Network water surface predictability for 1D steady-state open channel cases with different flow types and complex bed profile shapes. *Adv. Model. Simul. Eng. Sci.* **2022**, *9*, 1–23. [CrossRef]
46. Yang, Y.; Mei, G. A Deep Learning-Based Approach for a Numerical Investigation of Soil–Water Vertical Infiltration with Physics-Informed Neural Networks. *Mathematics* **2022**, *10*, 2945. [CrossRef]
47. Bishop, C.M.; Nasrabadi, N.M. *Pattern Recognition and Machine Learning*; Springer: Berlin, Germany, 2006; Volume 4.
48. Anderson, J. DSM2 fingerprinting technology. In *Methodology for Flow and Salinity Estimates in the Sacramento-San Joaquin Delta and Suisun Marsh: 23rd Annual Progress Report*; California Department of Water Resources: Sacramento, CA, USA, 2002 .
49. Sandhu, N.; Finch, R. Application of artificial neural networks to the Sacramento-San Joaquin Delta. In *Proceedings of the Estuarine and Coastal Modeling*; ASCE: Reston, VA, USA, 1995; pp. 490–504.
50. Sanskrityayn, A.; Suk, H.; Chen, J.S.; Park, E. Generalized analytical solutions of the advection-dispersion equation with variable flow and transport coefficients. *Sustainability* **2021**, *13*, 7796. [CrossRef]
51. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [CrossRef]
52. Rahaman, N.; Baratin, A.; Arpit, D.; Draxler, F.; Lin, M.; Hamprecht, F.; Bengio, Y.; Courville, A. On the spectral bias of neural networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 10–15 June 2019; pp. 5301–5310.
53. Mildenhall, B.; Srinivasan, P.P.; Tancik, M.; Barron, J.T.; Ramamoorthi, R.; Ng, R. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM* **2021**, *65*, 99–106. [CrossRef]
54. Tancik, M.; Srinivasan, P.; Mildenhall, B.; Fridovich-Keil, S.; Raghavan, N.; Singhal, U.; Ramamoorthi, R.; Barron, J.; Ng, R. Fourier features let networks learn high frequency functions in low dimensional domains. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 7537–7547.
55. Snijders, T.A. On cross-validation for predictor evaluation in time series. In Proceedings of the On Model Uncertainty and its Statistical Implications: Proceedings of a Workshop, Groningen, The Netherlands, 25–26 September 1986; Springer: Berlin, Germany, 1988; pp. 56–69.
56. Montgomery, D.C.; Peck, E.A.; Vining, G.G. *Introduction to Linear Regression Analysis*; John Wiley & Sons: Hoboken, NJ, USA, 2021.
57. Legates, D.R.; McCabe, G.J., Jr. Evaluating the use of "goodness-of-fit" measures in hydrologic and hydroclimatic model validation. *Water Resour. Res.* **1999**, *35*, 233–241. [CrossRef]
58. Gupta, H.V.; Kling, H.; Yilmaz, K.K.; Martinez, G.F. Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling. *J. Hydrol.* **2009**, *377*, 80–91. [CrossRef]
59. Nash, J.E.; Sutcliffe, J.V. River flow forecasting through conceptual models part I—A discussion of principles. *J. Hydrol.* **1970**, *10*, 282–290. [CrossRef]
60. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
61. Wang, S.; Wang, H.; Perdikaris, P. On the eigenvector bias of Fourier feature networks: From regression to solving multi-scale PDEs with physics-informed neural networks. *Comput. Methods Appl. Mech. Eng.* **2021**, *384*, 113938. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.