water

MDPI

Article

Novel Salinity Modeling Using Deep Learning for the Sacramento–San Joaquin Delta of California

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

[1] Department of Electrical and Computer Engineering, 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 Mathematics, University of California, Davis, CA 95616, USA
[*] Correspondence: syqi@ucdavis.edu (S.Q.); kevin.he@water.ca.gov (M.H.)

Abstract: Water resources management in estuarine environments for water supply and environmental protection typically requires estimates of salinity for various flow and operational conditions. This study develops and applies two novel deep learning (DL) models, a residual long short-term memory (Res-LSTM) network, and a residual gated recurrent unit (Res-GRU) model, in estimating the spatial and temporal variations of salinity. Four other machine learning (ML) models, previously developed and reported, consisting of multi-layer perceptron (MLP), residual network (ResNet), LSTM, and GRU are utilized as the baseline models to benchmark the performance of the two novel models. All six models are applied at 23 study locations in the Sacramento–San Joaquin Delta (Delta), the hub of California's water supply system. Model input features include observed or calculated tidal stage (water level), flow and salinity at model upstream boundaries, salinity control gate operations, crop consumptive use, and pumping for the period of 2001–2019. Meanwhile, field observations of salinity at the study locations during the same period are also utilized for the development of the predictive use of the models. Results indicate that the proposed DL models generally outperform the baseline models in simulating and predicting salinity on both daily and hourly scales at the study locations. The absolute bias is generally less than 5%. The correlation coefficients and Nash–Sutcliffe efficiency values are close to 1. Particularly, Res-LSTM has slightly superior performance over Res-GRU. Moreover, the study investigates the overfitting issues of both the DL and baseline models. The investigation indicates that overfitting is not notable. Finally, the study compares the performance of Res-LSTM against that of an operational process-based salinity model. It is shown Res-LSTM outperforms the process-based model consistently across all study locations. Overall, the study demonstrates the feasibility of DL-based models in supplementing the existing operational models in providing accurate and real-time estimates of salinity to inform water management decision making.

Keywords: Sacramento–San Joaquin Delta; deep learning; salinity estimation and forecasting; Res-LSTM; Res-GRU; explainable artificial intelligence

check for updates

Citation: Qi, S.; He M.; Bai Z.; Ding Z.; Sandhu, P.; Chung, F.; Namadi, P.; Zhou, Y.; Hoang, R.; Tom, B.; Anderson, J.; Roh, D.M. Novel Salinity Modeling Using Deep Learning for the Sacramento–San Joaquin Delta of California. *Water* **2022**, *14*, 3628. https://doi.org/10.3390/w14223628

Academic Editor: Kun Shi

Received: 1 September 2022
Accepted: 4 November 2022
Published: 11 November 2022

1. Introduction

1.1. Background

This study develops novel machine learning (ML) approaches in salinity modeling in an estuarine environment. The innovations include (a) proposing inventive ML models never explored in the literature before and (b) a novel application of the proposed ML models at a finer hourly time scale in both simulation and prediction.

Salinity management in estuarine environments can impact the region's water supply and ecology. Estuarine salinity is linked to changes in migration patterns, spawning habitat, fish distribution, and survivability, and affects the water quality of freshwater withdrawals [1,2]. Globally, many examples of estuarine systems sensitive to salinization exist. In Chesapeake Bay, the largest estuary in the United States, salinity impacts the habitat characteristics and influences the distribution, survival, and growth of a variety of species [1,3]. The Mekong Delta in Vietnam is critically important to the country's economy and is vulnerable to salinity intrusion due to sea level rise and upstream flow regulation [4]. Mulamba et al. [5] studied the sea-level rise impacts on the lower St. Johns River in Florida, where salinity change due to sea-level rise can potentially alter its estuarine ecosystem and threaten groundwater and surface water supplies. The Murray–Darling Basin Authority is engaged in strategic salinity management in Australia's Murray–Darling River Basin, where salinization affects municipal, industrial, and agricultural water supply, soil salinity, and ecosystem health [6,7]. Salinity management is actively pursued worldwide and is of particular importance for estuarine environments with significant social, economic, and environmental values, including the Sacramento–San Joaquin Delta in California, United States (U.S.).

The Sacramento–San Joaquin Delta (Delta) is the largest estuary on the west coast of the U.S. and is formed by the confluence of the two largest river systems in California, the Sacramento River and the San Joaquin River. The river flows across the Delta and through the Delta's downstream boundary at Martinez and into San Francisco Bay. Tides from the Pacific Ocean bring salty water upstream into the Delta. The Delta provides a habitat for 750 species of animals and plants and is a globally important biodiversity hot spot [8,9]. The Delta also provides drinking and irrigation water for California through the state, federal, and local water distribution systems, including the State Water Project (SWP) and the Central Valley Project (CVP). The SWP and CVP projects provide water to over 25 million people and 15,000 km^2 of farmlands [10]. The Delta is also used by millions for recreation and transportation [11]. Upstream riverine runoff (typically controlled by reservoirs) provides water to meet the water supply needs of the projects, and to meet Delta salinity requirements for both agriculture and wildlife. Within the Delta, consumptive uses of water include evaporation, seepage, and crop evapotranspiration. Salinity levels across the Delta depend upon the complex interactions between fresh water and seawater, which vary by location and are affected by river channel geometry, physical structures such as gates and barriers, diversions, and upstream reservoir releases.

1.2. Literature Review

Optimizing the operation of the SWP and CVP requires estimating salinity for a wide range of climate and operational scenarios. Process-based models have been traditionally developed and utilized for this purpose [12–20]. However, applying these models can be time-consuming, particularly in studies that require numerous model runs with long simulation periods. There is a need for fast simulation models with reasonable accuracy. Data-driven ML models have been explored to that end.

The earliest attempt was probably using the multi-layer perceptron (MLP), a type of artificial neural network (ANN), to simulate flow–salinity relationships in the Delta [21]. The study developed MLP models with one hidden layer or two hidden layers and showed that they could outperform empirical models significantly. The MLP model was refined later in terms of (a) identifying the most effective input features and training strategy; (b) increasing their robustness by training them using a variety of hydrology and operational conditions; and (c) simplifying their implementation into operational water planning models [22–25]. The resulting MLP model with seven input variables and two-hidden layers (with eight and two neurons, respectively) was implemented into California's latest water resources planning model, CalSim3 [26]. CalSim3 simulates the operations of SWP and CVP under different planning scenarios constrained by regulatory requirements, including allowable salinity levels at various locations across the Delta [26]. Ref. [27] further enhanced the MLP models of [26] in the context of (a) changing the learning paradigm from single-task learning (one MLP per study location) to multitask learning (MTL; one MLP model for all study locations together); and (b) adding a convolution layer before the hidden layers to pre-process input data. These enhancements helped reduce training time and increase the accuracy of the salinity estimates.

In addition to simulating the flow–salinity relationships, ANNs have also been developed to emulate process-based models directly in the Delta. Ref. [2] incorporated the Bayesian ANNs with the delta salinity gradient (DSG) model [17] in a hybrid manner for salinity simulation in the Delta. Ref. [28] utilized ANNs to emulate DSM2 in simulating volumetric fingerprints of flow sources for several study locations across the Delta. Salinity levels at these locations were then derived by multiplying the fingerprints by their corresponding salinity levels at flow source locations. Ref. [10] developed deep learning models, including the long short-term memory (LSTM) networks and convolutional neural networks, to emulate a salinity generation model at the downstream boundary of the Delta. Ref. [29] explored the use of both conventional ML models and deep learning models in emulating DSM2 for salinity estimation at 28 locations across the Delta. However, Ref. [29] did not explore the forecasting capability of the ML models and focused only on the daily scale using simulated salinity data.

Despite their scientific advances and practical values, those Delta salinity ML studies generally have four limitations in common [29]. Firstly, the ML models developed in these studies were mostly applied in simulating salinity under different planning scenarios. The forecasting capability of the ML models was largely unexplored. Reliable and intelligent forecasting is one major practical application for water-resource studies. Over the past decades, machine learning (ML) methods have gained more and more popularity in this area [30,31], due to their ability to handle big data at different scales as well as their flexible structure for identifying non-linear and complex relationships between input and output data. Researchers worldwide have applied novel ML methods in forecasting various variables that are important to water resources management, including streamflow [32–36]; groundwater level [37], groundwater quality [38], groundwater storage change [39], and sediment [40,41], among others. Popular ML algorithms explored include random forest [42–44], artificial neural network [45,46], support vector regression [45,47], LSTM [48], regression trees [49], extreme learning machine [45,50], wavelet transform [46,50,51], and adaptive neuro-fuzzy inference system [50,51].

Secondly, those Delta salinity ML studies focused on daily or coarser temporal scales probably due to the prohibitively expensive computing requirement associated with finer time scales. However, sub-daily scales (e.g., tidal scale, hourly scale) are also meaningful for water resources planning and management practices in the Delta. For instance, farmers may need to make water diversion schedules on when to pump water from Delta channels to irrigate their crop lands during a day. Understanding the sub-daily variations of salinity would help inform their relevant decision making to avoid diverting salty water that may have a detrimental effect on crops.

Thirdly, ML models in those studies were typically trained using salinity simulations from process-based models. Simulated data are generally “noise-free”, as they follow the physical laws embedded in the advection–dispersion governing equations hardwired in process-based models. This characteristic makes it straightforward for ML models to learn the underlying patterns or signals in simulated data. This limits the application of those ML models for certain applications, including forecasting. To forecast the spatial and temporal variations of salinity in the near future, it would be ideal for the ML models to be trained and tested using field observations directly so that they can be utilized to predict what would happen in the field. These field observations reflect the real-world salinity conditions containing information not captured by process-based models, which are, at most, simplified representations of reality.

1.3. Scope of the Current Work

The current study attempts to tackle these limitations highlighted above. Specifically, built upon the success of previous studies, particularly the study of [29], which developed four ML models (MLP, LSTM, GRU, and ResNet) to simulate salinity at multiple locations in the Delta, this study proposes two novel ML models, Res-LSTM and Res-GRU, which are less complex but more efficient compared to their vanilla versions (i.e., LSTM and GRU). This study utilizes salinity observations as the target to train these six ML models and assesses their performance on both daily and hourly time scales. Moreover, this study explores the forecasting capability of the two proposed novel ML models. Furthermore, the study discusses the overfitting potential of the proposed models and evaluates model performance against that of a process-based model.

The paper is organized as follows. In Section 2, we describe the methodology applied, including the study ML models proposed and their setup, study locations, study dataset, and study metrics. In Section 3, we illustrate the performance of these models as well as their forecasting capability. In Section 4, we discuss the results, scientific and practical values of the study, and potential future work. The study is concluded in Section 5.

2. Methodology

2.1. Study Area and Dataset

This study exemplifies the development and use of novel ML models in an estuarine system: the Delta of California (Figure 1). The Delta is a transition zone between freshwater and saltwater, where freshwater inflows from the Sacramento and San Joaquin rivers are conveyed westward toward the San Francisco Bay through a series of channels and tributaries. Managing Delta salinity is important to maintain the region's ecological health, freshwater water supply reliability, and regulatory compliance. Salinity is monitored at sparse locations across the Delta and is typically represented as electrical conductivity (EC) in micro-Siemens/cm (µS/cm) which indicates the amount of dissolved salt in water. Specifically, this study focuses on 23 salinity-monitoring locations with a reasonably long record of observed data (Figure 1). These locations include freshwater pumping locations, key flow junctions, and locations of ecological significance in the Delta. Hourly salinity measurements during the study period from 1 January 2000 to 31 December 2019 at these 23 stations are used for ML model training and testing.

Salinity in the Delta can vary between fully marine to near-zero, depending on the location and interaction between tides and freshwater inflows [52]. Figure 2 visualizes the range of EC values at each station over the 20-year study period. The orange line on each box represents the median values of the corresponding metrics for each of the 23 study locations. Each box represents the interquartile range from the 25th to the 75th percentiles. The location of the upper bar indicates the maximum metric value within 1.5 times the interquartile range above the 75th percentile. The location of the lower bar indicates the minimum metric values within 1.5 times the interquartile range below the 25th percentile. The open circles represent outliers. Figure 2 illustrates that the EC between the least saline and most saline locations can span two orders of magnitude. Generally, water is least saline toward the northern Delta, where lower salinity water from the Sacramento River and eastern tributaries (e.g., Cosumnes River, Mokelumne River, and Calaveras River) enter the Delta. The northernmost locations considered in this study (RSMKL008 #1, RSAN032 #2 and RSAN037 #3) exhibit median EC between 100 and 300 µS/cm. Median salinity near the major pumping locations (CHVT000 #7, CHSWP003 #8, CHDMC006 #9) ranges from 350 to 450 µS/cm. San Joaquin River inflows into the southern Delta (RSAN072 #12) have a higher median EC between 600 and 800 µS/cm, due to having higher salt content from agricultural drainage [53]. In the brackish Suisun Marsh (SLMZU011 #20, SLSUS012 #21, SLCBN002 #22), where the saltwater from the San Francisco Bay meets the freshwater from the Sacramento and San Joaquin Rivers, the median salinity ranges from 8000 to 10,000 µS/cm. The westernmost locations considered in this study (RSAC075 #16 and RSAC064 #23) have high median salinity and variability due to their proximity to the ocean and influence from tidal cycles [53].

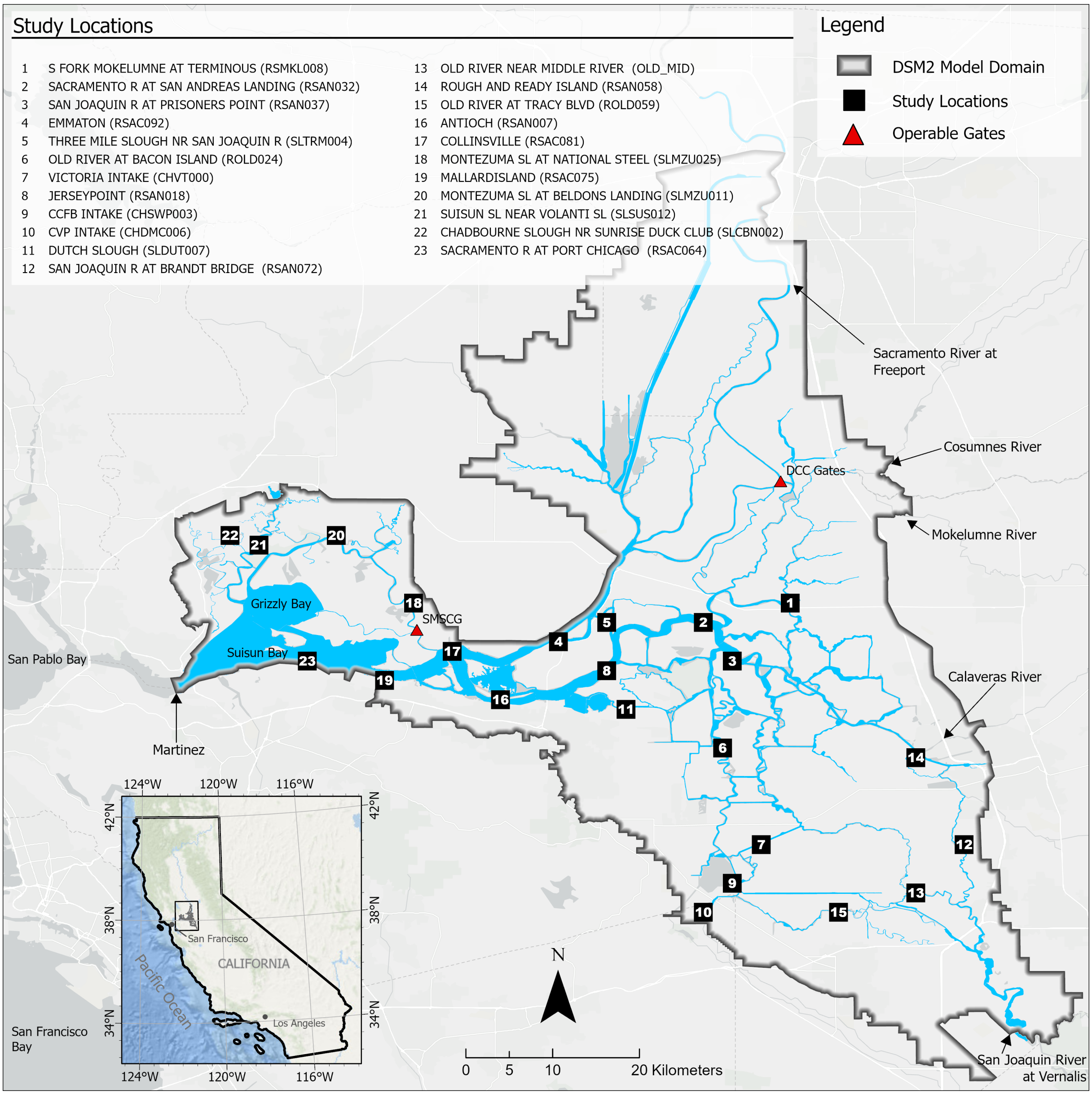

Study Locations
1 S FORK MOKELUMNE AT TERMINOUS (RSMKL008)
2 SACRAMENTO R AT SAN ANDREAS LANDING (RSAN032)
3 SAN JOAQUIN R AT PRISONERS POINT (RSAN037)
4 EMMATON (RSAC092)
5 THREE MILE SLOUGH NR SAN JOAQUIN R (SLTRM004)
6 OLD RIVER AT BACON ISLAND (ROLD024)
7 VICTORIA INTAKE (CHVT000)
8 JERSEYPOINT (RSAN018)
9 CCFB INTAKE (CHSWP003)
10 CVP INTAKE (CHDMC006)
11 DUTCH SLOUGH (SLDUT007)
12 SAN JOAQUIN R AT BRANDT BRIDGE (RSAN072)
13 OLD RIVER NEAR MIDDLE RIVER (OLD_MID)
14 ROUGH AND READY ISLAND (RSAN058)
15 OLD RIVER AT TRACY BLVD (ROLD059)
16 ANTIOCH (RSAN007)
17 COLLINSVILLE (RSAC081)
18 MONTEZUMA SL AT NATIONAL STEEL (SLMZU025)
19 MALLARDISLAND (RSAC075)
20 MONTEZUMA SL AT BELDONS LANDING (SLMZU011)
21 SUISUN SL NEAR VOLANTI SL (SLSUS012)
22 CHADBOURNE SLOUGH NR SUNRISE DUCK CLUB (SLCBN002)
23 SACRAMENTO R AT PORT CHICAGO (RSAC064)
Legend
DSM2 Model Domain
Study Locations
Operable Gates
Sacramento River at Freeport
Cosumnes River
DCC Gates
Mokelumne River
Calaveras River
San Joaquin River at Vernalis
Grizzly Bay
Suisun Bay
SMSCG
San Pablo Bay
Martinez
San Francisco Bay
124°W 120°W 116°W
42°N 38°N 34°N
San Francisco
CALIFORNIA
Pacific Ocean
Los Angeles
N
0 5 10 20 Kilometers

Figure 1. Schematic showing the Sacramento–San Joaquin Delta (Delta), the 23 study locations, and the DSM2 model domain.

As with any real-time observations, there can be missing data. The ratios of available data in Figure 2, expressed in percent, indicate the data available during the 20-year observation period. An ideal ratio of 100% represents no missing data in the 20-year period. Most (14 out of 23) stations have over 90% data availability. Stations generally have low data availability because there are no observations for consecutive years in the early part of the 20-year period likely because sensors were not installed. For example, data for Dutch Slough (SLDUT007) are available starting from 2010. From 2010 onward, the data are continuously available for all stations with minor, intermittent, dropouts.

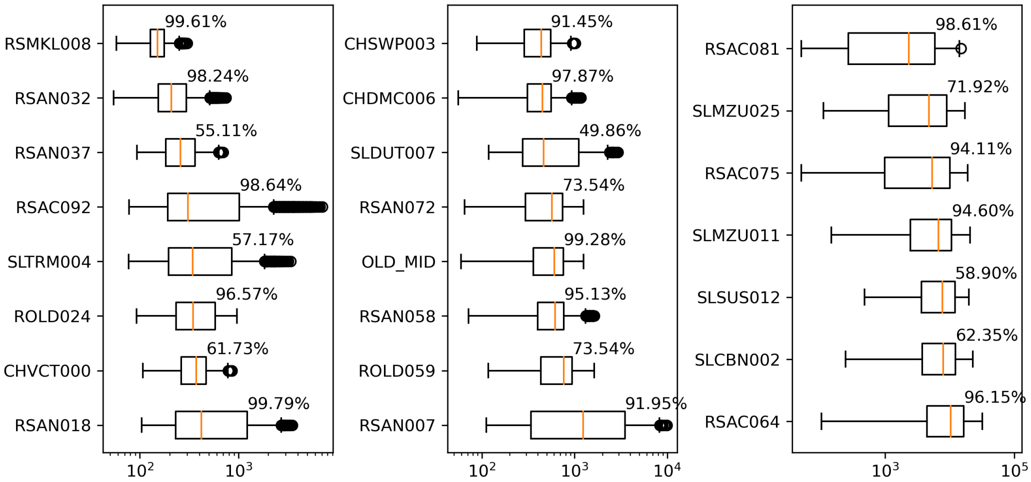


Figure 2. Boxplot of salinity observations (represented by electrical conductivity) at study locations, sorted by their medians. Numbers next to each station's box represent the ratios of available observations in the dataset during the 20-year study period. Each box represents the interquartile range from the 25th to the 75th percentiles. The line inside each box represents the median value. The open circles represent outliers.

To maintain consistency with our previous study [29], we use the same set of eight variables as input features to the ML models proposed in the current study (Table 1). Both salinity measurements and input variables during the study period have been rigorously quality controlled and applied to calibrate Delta Simulation Model II (DSM2), the operational hydrodynamics and water quality model providing simulations on flow, water stage, and water quality variables (including salinity) to guide real-time and long-term planning of water operations in the Delta [54]. The source of data utilized in the current study is provided in Appendix A, and the details of the datasets utilized in previous studies are summarized in Appendix B. Following [29], we randomly select 70% of the historical salinity measurements for training and use the remaining 30% for testing the ML models proposed.

Table 1. Input features to proposed ML models.

Index	Input Feature Name	Definition
1	Northern Flow	Sum of Sacramento, Yolo Bypass, Mokelumne River, Cosumnes River, and Calaveras River flows.
2	San Joaquin River Flow	San Joaquin River at Vernalis Flow.
3	Pumping	Sum of pumping from Banks Pumping Plant, Jones Pumping Plant, and Contra Costa Water District at Rock Slough, Old River, and Victoria Canal.
4	Delta Cross-Channel Gate Operation	Delta Cross-Channel Gate Openings.
5	Consumptive Use	Net Delta Consumptive use estimated by Delta Channel Depletion (DCD) and Suisun Marsh Channel Depletion (SMCD) models.
6	Martinez Tidal Energy	Tidal energy at Martinez, calculated as the daily maximum–the daily minimum astronomical tide at Martinez.
7	San Joaquin River EC	Electrical conductivity measured at San Joaquin River at Vernalis.
8	Sacramento River EC	Electrical conductivity measured at Sacramento River at Greens Landing.

2.2. Machine Learning Models

This study explores the use of six different ML models in a multi-task learning framework (i.e., one ML model for all study locations). Four of them were investigated in our previous study [29] where they were trained and tested using noise-free salinity simulations rather than real-world field observations of salinity. The current study adopts the same architectures applied in [29]: a multi-layer perceptron (MLP) network, a residual network (ResNet), a long short-term memory (LSTM) network, and a gated recurrent unit (GRU) network. The workflow of each of these four models is provided in Appendix C. These models are briefly described as follows. For detailed explanation on them, the readers are referred to [29].

The MLP model consists of one input layer, two fully connected (FC) hidden layers, and an output layer. The number of neurons in the two hidden layers is the number of study locations multiplied by 8 and 2, respectively. The input time series are pre-processed before being sent to the MLP model, which will be discussed in detail in Section 2.3. Results in [29] show that an MLP model achieves satisfactory salinity estimates, but the pre-defined input pre-processing procedure can still lead to unavoidable information loss in the input data. To improve the performance of the MLP model, a ResNet [55] model is developed by adding a shortcut side path including two convolutional layers and a FC layer to the vanilla MLP model to skip the pre-processing step so that the temporal information in the inputs is preserved. In addition to MLP and ResNet, Ref. [29] also explored two recurrent neural networks (RNNs): GRU and LSTM. RNNs maintain a memory internally in order to preserve essential temporal information from the inputs and have also achieved great success on time series analysis and processing tasks. GRU is one type of the popular RNN architecture designed for processing sequential data, which consists of a reset gate and an update gate connected to its hidden state. Similar to GRU, LSTM models keep a hidden state, which is the short-term memory, while storing an additional cell state, also known as long-term memory. Following [29], in the current study, we set the numbers of neurons in FC layers in ResNet, or numbers of RNN units in LSTM and GRU models equivalent to the numbers of neurons in the baseline MLP model.

In addition to these four models, we devise two novel architectures: Res-LSTM and Res-GRU. LSTM and GRU models applied in [29] are beneficial for time-series-related tasks, as they are capable of keeping track of memory when processing sequential inputs in an iterative manner. However, they have two major drawbacks. Firstly, they usually appear as complex models with more parameters than non-RNN models such that sufficient memory can be retained. Secondly, as the subsequent cell states in GRU and LSTM depend on the previous ones, computations cannot be processed in parallel, making them run slower than MLP or ResNet by design.

In order to address these two issues, inspired by the shortcut design in ResNet, we propose the Res-LSTM and Res-GRU models, which are less complex than the baseline LSTM or GRU models, respectively. According to our previous work [29], a vanilla MLP model already yields satisfactory results. Therefore, we directly use the MLP model as the main branch in Res-LSTM or Res-GRU models, as illustrated in Figures 3 and 4 below. The numbers of neurons, 184 and 46, of the two hidden FC layers in the main branch are identical to those in the original MLP baseline model. In addition, we add the shortcut connection, consisting of a single LSTM or GRU layer, such that the error, or the residual, between the ground truth and the outputs of the MLP model can be captured. Taking advantage of the powerful MLP baseline model, we can reduce the complexity of the LSTM or GRU layer in the shortcut path in comparison with the baseline LSTM or GRU models. Here, we arbitrarily pick the number of units in the LSTM or GRU layers to be 46, which is equivalent to the number of neurons in the second hidden layer of the MLP baseline model. The number of parameters of the six aforementioned models can be found in Table 2.

The LSTM or GRU layers in the shortcut branch takes the time series of the eight input variables as inputs. At each of the 118 daily time steps in inputs, the RNN layers process one set of daily values of the eight input variables to update their hidden state and/or cell state accordingly. At the final 118-th daily time step, the LSTM or GRU layer outputs its final hidden state, which is supposed to fit the error or residual defined above. When applying the Res-RNN models for salinity forecasting, we simply shift the target salinity forward by a given lead time such that the outputs generated by these Res-RNN models represent the salinity values in the future.

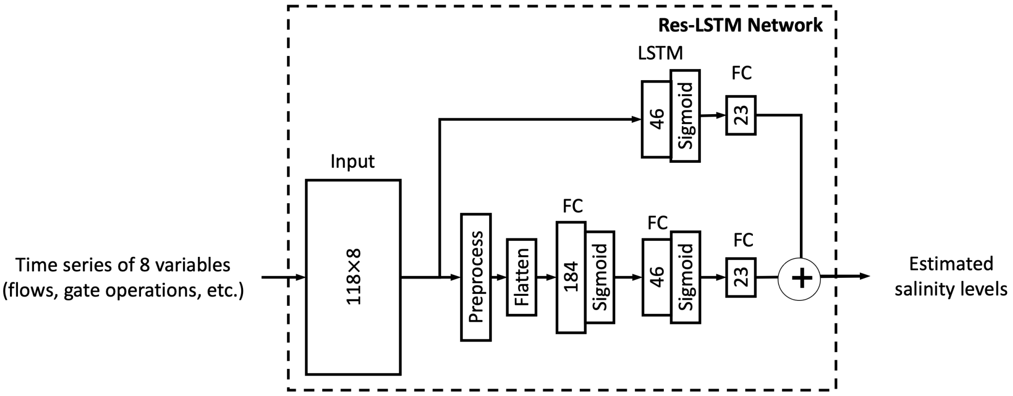


Figure 3. Diagram of the proposed Res-LSTM network. The number in the input layer denotes input shape and those in the subsequent layers represent the numbers of units / neurons of those layers.

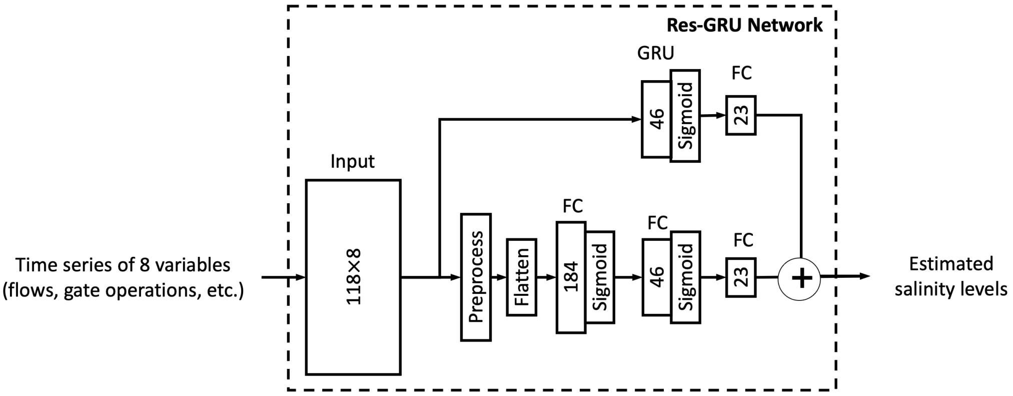


Figure 4. Diagram of the proposed Res-GRU network. The number in the input layer denotes input shape and those in the subsequent layers represent the numbers of units / neurons of those layers.

As can be seen in Table 2, the Res-LSTM and Res-GRU use fewer parameters than their corresponding vanilla LSTM and GRU models. Meanwhile, in theory, they ought to outperform the vanilla MLP model since the shortcut side branch can learn to compensate for residual errors of the MLP main branch. Additionally, they need shorter training times.

Table 2. Number of parameters of the six ML models.

Architecture	MLP	ResNet	LSTM	GRU	Res-LSTM	Res-GRU
Number of parameters	36, 271	357,799	227,263	201,687	111,300	100,812

2.3. Input Preprocessing

The inputs for the proposed ML models are generated using the same pre-processing strategy as introduced in [29]. Specifically, given the long memory of the Delta, where flow and operations in the past several months have lagged impacts the current salinity conditions [26–28], we aggregate 118 antecedent daily values of each of the 8 input variables into 18 values per input variable. These 18 values consist of one measurement from the current day and 7 daily measurements from the most recent 7 preceding days, together with 10 non-overlapping 11-day averages of the prior 110 days.

We use bold symbols for vectors and regular symbols for scalars. We utilize superscript to represent vector indices and subscript to represent input and output variable indices. For instance, x_i^t stands for the i-th input variable on day t. We write the mean of the i-th input variable computed based on a time frame from day t_1 to t_2, with $t_1 < t_2$, as $\overline{x_i^{t_1 \to t_2}}$, where

$$\overline{x^{t_1 \to t_2}} = \frac{1}{t_2 - t_1 + 1} \sum_{t=t_1}^{t_2} x_i^t \tag{1}$$

We apply linear min-max normalization on the input time series and the salinity at each study station to the range of [0, 1]. That means that for each input feature or each salinity sequence at each station over the 20-year time span, the minimum value is transformed to 0, while the maximum value is transformed to 1. To be more specific, taking x_i^t, the i-th input variable on day t, as an example, it is normalized according to Equation (2), where T is the total number of samples.

$$\hat{x}_i^t = \frac{x_i^t - (\min\limits_{k=1,\dots,T} x_i^k)}{(\max\limits_{k=1,\dots,T} x_i^k) - (\min\limits_{k=1,\dots,T} x_i^k)} \tag{2}$$

The inputs for the proposed MLP network used to estimate reference salinity levels on day t consist of 8 daily values $\hat{x}_i^t, \dots, \hat{x}_i^{(t-7)}$ $(1 \le i \le 8)$, and 10 average values $\overline{\hat{x}_i^{(t-8) \to (t-18)}}, \dots, \overline{\hat{x}_i^{(t-107) \to (t-117)}}$ computed on 10 successive 11-day sliding windows, without overlapping, on the prior 110 days. The 18 processed values of each input variable make up the $8 \times 18 = 144$ inputs parameters for the MLP network. This pre-defined pre-processing method reduces the dimension of the input vector and avoids unnecessarily increasing the complexity of proposed MLP model, as well as preserving historical memory in the input time series.

For the proposed ResNet, LSTM, GRU, Res-LSTM and Res-GRU models, as they are designed to be more complex than the MLP model and require more detailed information from the input data, we directly input the 118 daily values $\hat{x}_i^t, \dots, \hat{x}_i^{t-117}$ of each input variable $\hat{x}_i$ $(1 \le i \le 8)$.

Model inputs are prepared in the same way as discussed above. For salinity estimation on day t, namely, the case where lead time $t_l = 0$, the salinity levels observed on day t at each of the 23 monitoring stations, depicted as y_k^t, where $k = 1, 2, \dots, 23$ is the index of the monitoring stations, are set as target outputs during training or testing.

2.4. Forecasting Setup

In our previous work [10,29], we focused only on the investigation of same-day salinity estimation (i.e., the lead time is zero). In practice, forecasting near-term salinity is critical to informing real-time water management decision making. In this work, we extend the scope of proposed Res-LSTM and Res-GRU models to salinity forecasting up to 14 days into the future (lead time equals 14 days). Specifically, one ML model is trained for each lead day. A total of 14 Res-LSTM models and 14 Res-GRU models are developed.

For salinity forecasting on day t with a lead time of t_l, we perform the following pre-processing steps:

Step 1: We prepare model inputs the same way as discussed in Section 2.3, which consists of $\hat{x}_i^t, \dots, \hat{x}_i^{(t-7)}$ $(1 \le i \le 8)$ and $\overline{\hat{x}_i^{(t-8) \to (t-18)}}, \dots, \overline{\hat{x}_i^{(t-107) \to (t-117)}}$.

Step 2: We formulate the target output values by shifting the salinity values forward by t_l days, represented by $y_k^{t+t_l}, k = 1, 2, \dots, 23$.

In this way, after training, the models shall be capable of predicting daily salinity levels at the 23 monitoring stations ahead of time by t_l days.

In the remaining of the paper, ML models trained with a lead time of zero ($t_l = 0$) are referred to as "salinity estimation" models, while models trained with a lead time of greater than or equal to 1 ($t_l \geq 1$) are referred to as "salinity forecasting" models. It is worth noting that forecasting models here differ from models applied in real-time forecasting operations which use forecast model inputs to drive the model and generate forecasts. The forecasting models developed for each lead time (i.e., day 1 through day 14 into the future) in the current study use purely historical data up to the lead time of zero.

2.5. Evaluation Metrics

The proposed models are trained with the Adam optimization algorithm [56] based on the widely used mean squared error (MSE) loss function. Four statistical evaluation metrics, consisting of the square of the correlation coefficient (r^2), bias, root mean standard deviation ratio (RSR), and the Nash–Sutcliffe efficiency coefficient (NSE), are employed to assess the ML model performance. Each of the four metrics evaluates the modeled salinity performance from a different perspective: r^2 quantifies the strength of the linear relationship between modeled salinity and the target salinity; percent bias indicates whether the models over- or underestimate the salinity; RSR is a standardized representation of the root mean squared error (RMSE) between model outputs and targets; and NSE compares the predictive capacity of the models with the global mean of target sequences. For r^2 and NSE, a value close to 1 indicates desirable model performance. For percent bias and RSR, a value close to 0 designates good model performance. Table 3 provides detailed descriptions and definitions of these five metrics. Here, S represents the salinity sequence, $\bar{S}$ indicates the overall average value of the salinity levels S, and the subscripts ANN and $Observed$ indicate ANN-estimated and observed salinity, respectively.

Table 3. Study metrics.

Name	Definition	Formula
MSE	Mean Squared Error	$\text{MSE} = \sum_{t=t_l+1}^{T} (S_{Observed}^{t} - S_{ANN}^{t})^2$
r^2	Squared Correlation Coefficient	$r^2 = (\frac{\sum_{t=t_l+1}^{T} \lvert (S_{Observed}^{t} - \overline{S_{Observed}}) \times (S_{ANN}^{t} - \overline{S_{ANN}}) \rvert}{T \times \sigma_{Observed} \times \sigma_{ANN}})^2$
Bias	Percent Bias	$\text{Bias} = \frac{\sum_{t=t_l+1}^{T} (S_{ANN}^{t} - S_{Observed}^{t})}{\sum_{t=t_l+1}^{T} S_{Observed}^{t}} \times 100\%$
RSR	RMSE-observations standard deviation ratio	$\text{RSR} = \frac{\sqrt{\sum_{t=t_l+1}^{T} (S_{Observed}^{t} - S_{ANN}^{t})^2}}{\sqrt{\sum_{t=t_l+1}^{T} (S_{Observed}^{t} - \overline{S_{Observed}})^2}}$
NSE	Nash-Sutcliffe Efficiency coefficient	$\text{NSE} = 1 - \frac{\sum_{t=t_l+1}^{T} (S_{Observed}^{t} - S_{ANN}^{t})^2}{\sum_{t=t_l+1}^{T} (S_{Observed}^{t} - \overline{S_{Observed}})^2}$

2.6. Implementation Details

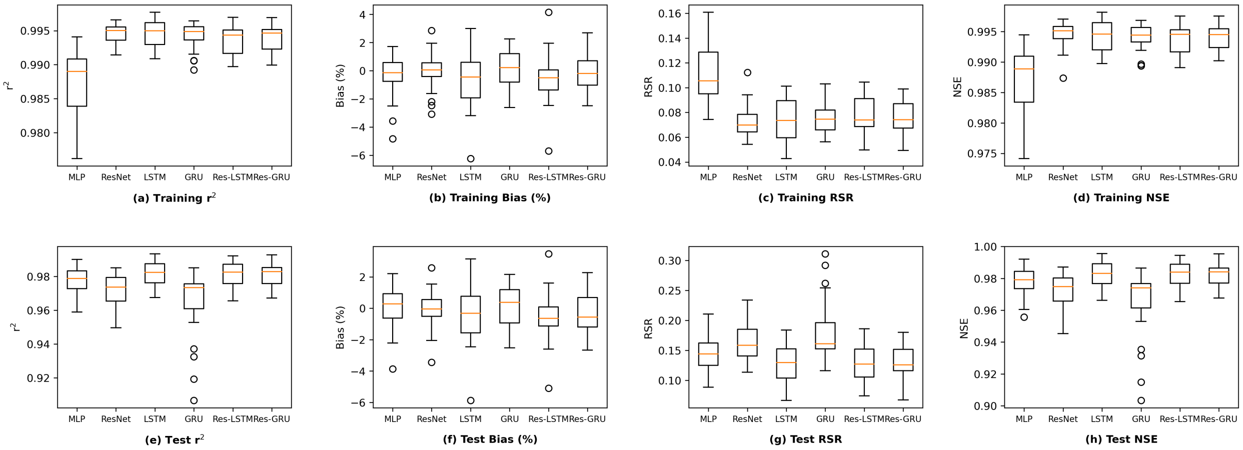
Our experiments are carried out on a public platform: the Google Colaboratory. Hyperparameters such as batch size, learning rate and numbers of epochs may affect model performance. The authors of a previous study [57] optimized some hyper-parameters in the backpropagation neural network (BPNN) architecture, including the number of nodes in the model, the learning rate, and number of epochs. In a different manner, we used a constant small learning rate of 0.001 with the Adam optimizer [56] to train our models and stopped training if the mean squared error (MSE) on the test set did not decrease for 50 epochs. In addition, to prevent overtraining, we limited the maximum number of epochs to 5000. In this way, we did not have to specifically optimize the learning rate or the number of epochs.

3. Results

This section presents the results of the proposed ML models, particularly those two novel models. Firstly, the training and testing performance of all six models is scrutinized in terms of the skill metrics described in Section 2.5 as well as visual inspection of modeled salinity against the corresponding observations. Next, the forecasting capability of the two proposed novel models is examined. Finally, model performance is evaluated on the hourly time step.

3.1. Model Performance on the Daily Scale

Figure 5 illustrates the performance of the two exploratory ANNs, Res-LSTM and Res-GRU, in comparison with the original four basic networks, MLP, ResNet, LSTM, and GRU, in terms of the four study metrics (r^2, Bias, RSR, and NSE). The figure reveals that the two new models (Res-LSTM and Res-GRU) have satisfactory performance. Specifically, the training results in Figure 5a–d indicate that both Res-LSTM and Res-GRU outperform MLP, while being at a similar level with ResNet, LSTM, and GRU. The former is most likely due to the learning compensation of the shortcut side branch from Res-LSTM and Res-GRU compared to MLP. The latter suggest that the new and simpler structures of Res-LSTM and Res-GRU could successfully achieve similar performance to their more complex counterparts, LSTM and GRU. Meanwhile, the test results in Figure 5e–h suggest that Res-LSTM and Res-GRU yield better or similar results to the four original models. All in all, Res-LSTM has a slight edge over other models, as it has slightly more desirable metrics. Simulations from Res-LSTM are further examined and compared with the observed salinity in different ways.



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

(e) Test r^2 **(f) Test Bias (%)** **(g) Test RSR** **(h) Test NSE**

Figure 5. Comparison of six models on observed data at daily time step.

Figure 6 shows the corresponding exceedance probability curves and daily time series plots of Res-LSTM simulations compared with the observed data at selected locations. Both types of plots demonstrate that the simulations mimic the target observed salinity very well, with the latter showing capture of the temporal pattern and magnitude in general. Another notable pattern is that, despite the marginal discrepancies in the full salinity spectrum between the two models, time series plots reveal that Res-LSTM slightly underestimates high salinity, especially for RSAC092 and RSAN018.

Figure 7 shows the statistical metrics for each study location, calculated at three ranges, illustrating the performance of the Res-LSTM model compared to observed data on a daily time step. For metrics r^2, NSE, and RSR, "yellow" indicates satisfactory performance. For the percent bias metric, shaded blue and orange represent underestimation and overestimation, respectively. Overall, model performance is most satisfactory when the salinity is in the low–middle range (0–75%) and decreases with high (75–95%) and extremely high (95–100%) salinity ranges. Several notable observations are further discussed below.

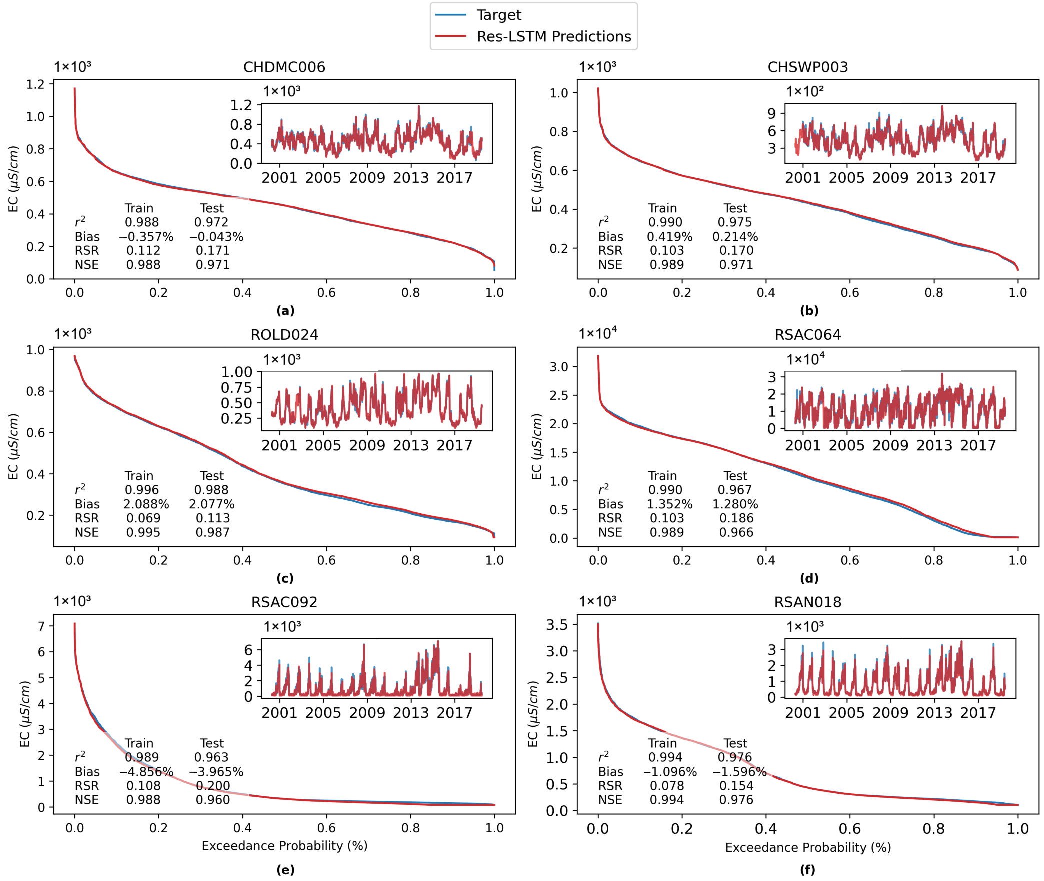


Figure 6. Exceedance probability plot and time series plot of Res-LSTM simulated versus observed salinity at daily time step.

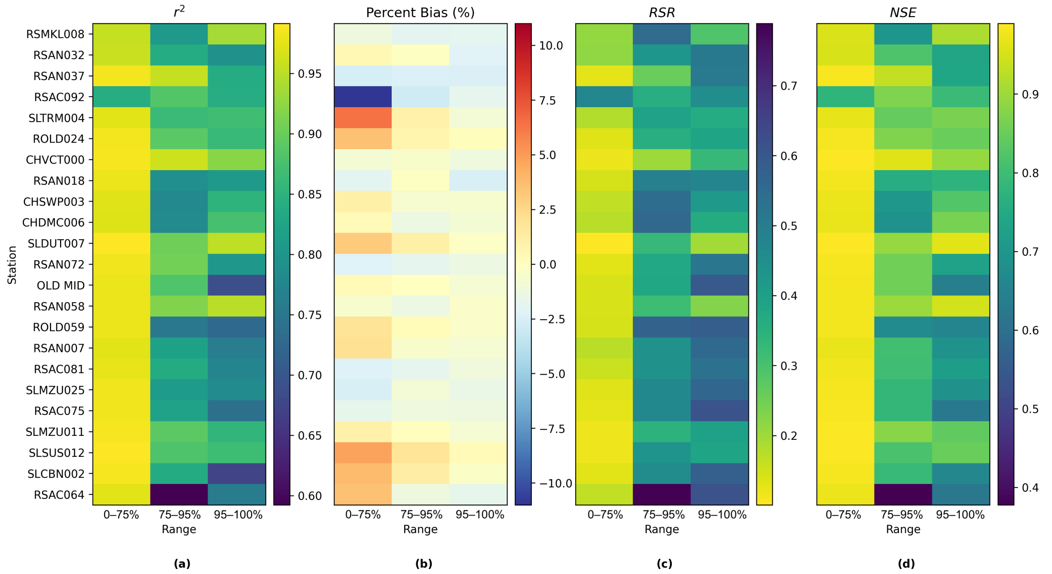


Figure 7. Heatmap showing Res-LSTM performance at different salinity ranges on the daily time step: low–middle range (lowest 75%), high range (75 to 95 percentile), and extreme high range (highest 5%) at the study locations.

Performance at location RSAC092 (Sacramento River at Emmaton) is lower in the low salinity range but is consistent with the other locations in the high and extremely high ranges. Despite the departure from the other stations, overall, r^2, NSE, and RSR for RSAC092 are acceptable. The Res-LSTM model underestimates salinity in the low–middle range, where the percent bias is −11%. This is because the Res-LSTM often estimates zero EC at Emmaton, but this generally does not occur in the observed data. At location RSAC064 (Port Chicago), r^2, NSE, and RSR are acceptable in the low–middle and extremely high ranges, but less satisfactory under the high range. The Res-LSTM is less able to capture the salinity variability at Port Chicago under the high range, but the percent bias is acceptable and consistent with other locations. At locations in the Suisun Marsh (SLMZU011, SLSUS012, SLCBN002), the Res-LSTM tends to overestimate salinity, especially in the low–middle range.

In short, the novel Res-LSTM and Res-GRU models can satisfactorily estimate salinity at the locations studied, while achieving similar or better performance compared with their more complex LSTM and GRU counterparts. Generally, performance is best at stations with lower median salinity and variability and degrades at stations with higher salinity and variability. The combination of a simpler architecture paired with comparably good performance to vanilla LSTM and GRU models indicate that the new models show promise in estimating Delta salinity on a daily time step.

3.2. Forecasting Performance

Figure 8 compares the forecasting performance of the Res-LSTM model during the training (Figure 8a–d) and test (Figure 8e–h) runs, using box and whisker plots for four types of metrics consisting of r^2, percent bias, RSR, and NSE. Each plot includes one box and whisker for each lead time evaluated. The statistical meaning of the boxes, whiskers and circles is the same as in Figure 2.

Generally speaking, model performance declines smoothly as the lead time increases, and all the metrics are within a reasonable range. Even with a lead time of 14 days, the Res-LSTM model predictions are satisfactory. For all lead times evaluated under training and testing, NSE is above 0.94, r^2 is above 0.95, and percent bias centers around zero percent, indicating excellent predictive performance without a tendency to systematically underestimate or overestimate.

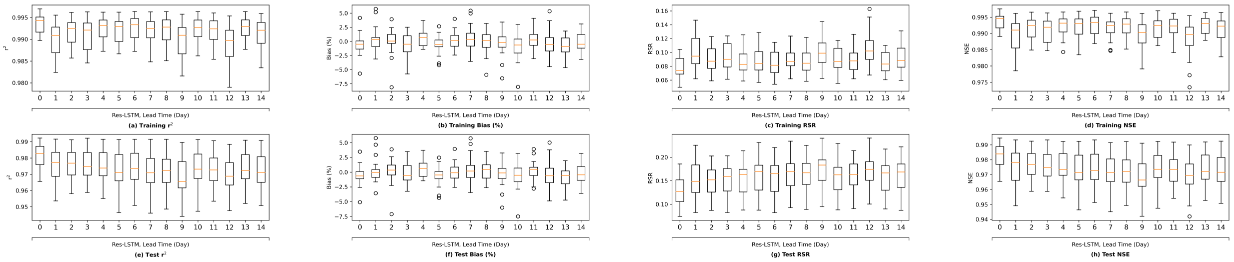


Figure 8. Salinity forecasting performance of Res-LSTM.

Figure 9 shows the corresponding performance of Res-GRU models based on four criteria (r^2, bias, RSR, and NSE) in two rows. The first row (Figure 9a–d) and the second row (Figure 9e–h) display the performance of Res-GRU for training and test datasets, respectively. As a performance indicator for ML algorithms, results of the test dataset (Figure 9e–h) indicate that the nowcasting (forecasting with 0 lead time) model has the best performance, and the forecasting model's accuracy decreases when the lead time increases, which is reasonable for every forecasting model. However, the forecasting model with 6 and 12 days lead time does not follow this pattern, and the forecasting model with a 6-day lead time provides the worst performance but still is satisfactory (r^2 and NSE are high, while RSR and bias are low). This suggests that the historical data up to lead time 0 alone may not be ideal to forecast these two days for the Res-GRU model. Overall, all the metrics are within a reasonable range. For all lead times evaluated under training and testing, NSE is above 0.94 and r^2 is above 0.94, indicating satisfactory performance overall. The percent bias metric indicates higher variability than the Res-LSTM predictions (Figure 5) but does not show a clear systematic bias toward underestimation or overestimation.

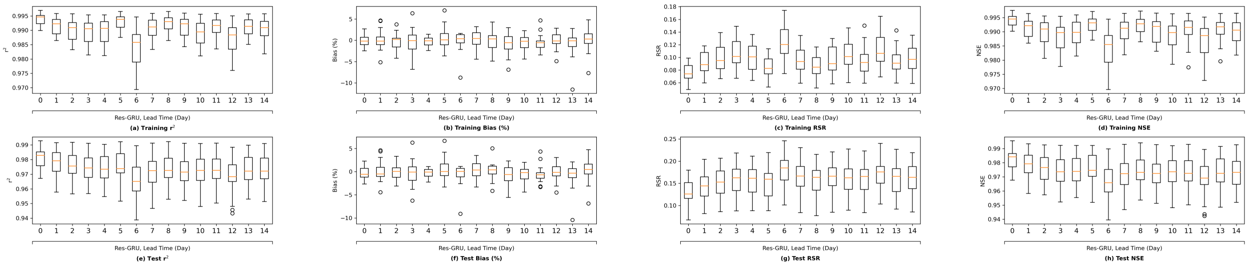


Figure 9. Salinity forecasting performance of Res-GRU.

All in all, for all lead times considered, Res-LSTM and Res-GRU can forecast salinity levels at all study locations with satisfactory performance. Model performance generally decreases as the lead time increases.

3.3. Model Performance on the Hourly Scale

The results presented so far are all trained and tested using daily salinity data aggregated from the hourly observations of salinity. In this sub-section, the six ML models proposed are trained using the hourly observations directly though the input data supplied to the models and are still on the daily scale. Figure 10 compares the performance of these models during the training (Figure 10a–d) and test (Figure 10e–h) runs, using box and whisker plots for four types of metrics consisting of r^2, percent bias, RSR, and NSE. Each plot includes one box and whisker for each model evaluated. Based on these metrics, the Res-LSTM model generally outperforms all the other ML models tested during training and test runs. On average, Res-LSTM has the highest r^2 and NSE. It also has the lowest bias and RSR for both training and testing. The performance of Res-GRU is close to but not as ideal as that of Res-LSTM. In contrast, ResNet has slightly inferior performance compared to other ML models, followed by MLP. Compared to their counterparts on the daily scale (Figure 5), the skill metrics r^2, RSR, and NSE are notably inferior, indicative of stronger performance on the daily (versus hourly) scale for all six models.

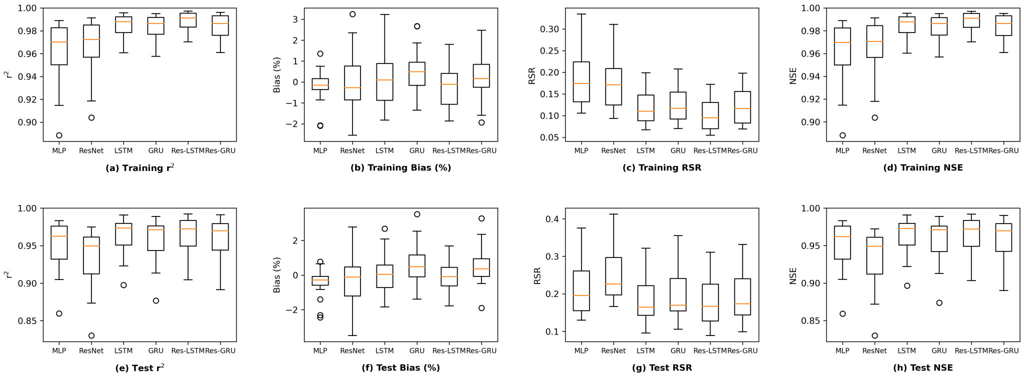


Figure 10. Comparison of six models on observed data at hourly time step with daily inputs.

Figure 11 shows the corresponding exceedance probability curve and hourly time series plots to evaluate the performance of Res-LSTM at six selected locations in the Delta. In general, the differences between model simulations and the corresponding observations are marginal. However, the time series subplots indicate that the Res-LSTM models slightly underestimate the peak values at some of these specified locations. Nevertheless, the plots show remarkable similarity between models and observations, and the Res-LSTM model can skillfully capture the temporal pattern of observed salinity. Comparing Res-LSTM performance on the daily scale (Figure 6) versus the hourly scale (Figure 11), the metrics associated with the daily scale are generally superior. In particular, the r^2 and NSE are slightly higher while the RSR is generally lower on the daily scale. This is also observed for other models as illustrated in Figures 5 and 10.

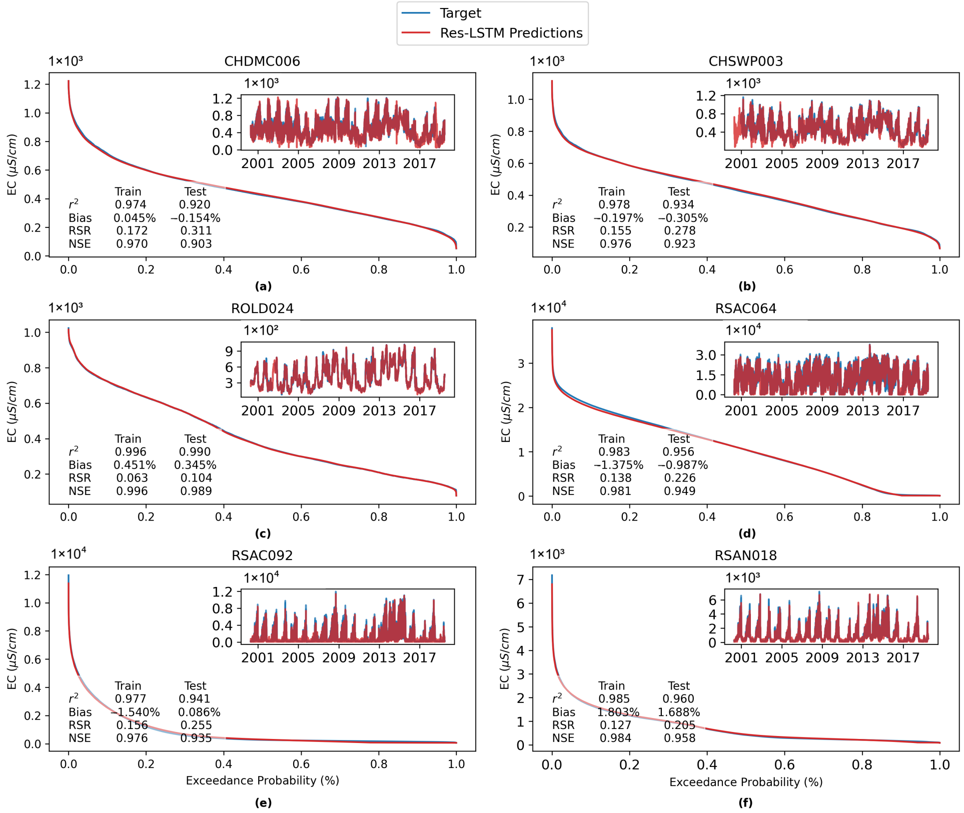


Figure 11. Exceedance probability plot and time series plot of Res-LSTM simulated versus observed salinity at daily time step with daily inputs.

As in Figure 7, Figure 12 shows heatmaps which summarize the performance of the Res-LSTM model with hourly time steps using the statistical metrics r^2, percent bias, RSR, and NSE for each study location. In general, model performance is the most satisfactory for salinities in the low–middle range across most stations, but lower for the high and extreme high ranges. Compared to the daily time step simulation results in Figure 7, the metrics associated with the hourly time step are inferior for most locations. The detailed values of all four study metrics on both daily and hourly scales are provided in the Appendix D.

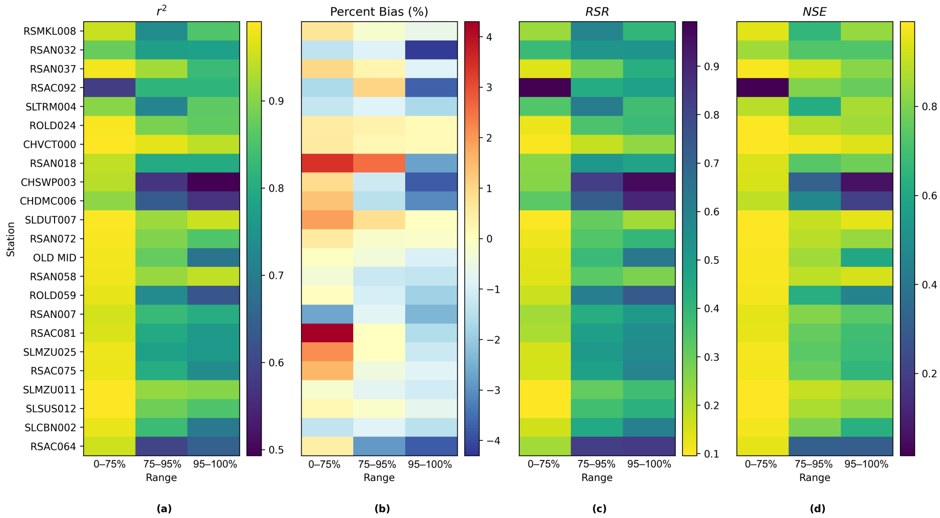


Figure 12. Heatmap showing Res-LSTM performance at different salinity ranges on the hourly time step: low–middle range (lowest 75%), high range (75 to 95 percentile), and extreme high range (highest 5%) at the study locations.

In a nutshell, all six proposed models can achieve satisfactory performance at a finer hourly scale, and Res-LSTM slightly outperforms the other five. The differences between model simulations and the corresponding observations are small on average. The performance of Res-LSTM is highest in the low–middle range, but relatively lower for the high and extreme high ranges. Compared to their counterparts on the daily scale, the ML models on the hourly scale generally have slightly degraded performance.

4. Discussions

This section first discusses the overfitting potential of the ML models proposed in the study. Second, the performance of a selected ML model is compared with that of the process-based model DSM2, which is widely used to inform water operations in the Delta. Next, the section discusses the scientific and practical implications of the study, followed by discussions on study limitations and planned future work.

4.1. Overfitting Potential versus Model Complexity

Overfitting happens when a ML model picks up the details, including noise, and fits exactly on the training data but does not generalize well on unseen data [58,59]. Overfitting is a central problem in the field of data-driven ML, as it negatively impacts the model's generalization performance on new data. Overfitting is more likely to occur when a model's structure is too complex for the task. To avoid this problem, the number of neurons or units in the layers need to be determined carefully.

In this study, we proposed six different ML models. In this sub-section, we explore the relationship between model complexity and salinity estimation performance by reducing the number of neurons in the hidden layers of each model. For the MLP, ResNet, Res-LSTM and Res-GRU models, we adjust the number of neurons in two fully connected hidden layers in the main branch, depicted by n_1 and n_2, respectively. In addition to the original settings, where $n_1 = 184$ and $n_2 = 46$, we pick four combinations, including $\{n_1 = 138, n_2 = 46\}$, $\{n_1 = 92, n_2 = 46\}$, $\{n_1 = 46, n_2 = 46\}$ and $\{n_1 = 46, n_2 = 23\}$ to build four simplified versions as well as five additional combinations of $\{n_1 = 184, n_2 = 92\}$, $\{n_1 = 184, n_2 = 138\}$, $\{n_1 = 184, n_2 = 184\}$, $\{n_1 = 368, n_2 = 92\}$, and $\{n_1 = 368, n_2 = 184\}$ that lead to complicated versions for each of the four models. For the vanilla LSTM and GRU models, we change the number of units in the recurrent layer from 184 units to 322, 276, 230, 138, 92, 46 or 23 units. A detailed list of the number of parameters in these models can be found in Tables A10 and A11 of Appendix E.

We plot the model performance (in terms of the average value of each evaluation metric across the 23 study locations) versus model complexity (in terms of number of parameters) in Figure 13. In general, model performance improves as complexity grows. In both the training and test plots of r^2, RSR and NSE, MLP models show the best complexity–performance trade-off. Namely, the MLP model can achieve a comparable performance with other models with a relatively smaller number of parameters. However, during the grid search process when designing these models, we observed that the MLP model hits a performance plateau earlier than other models, namely, the test performance stops improving, even if we keep increasing the number of neurons in its hidden layers. In contrast, the ResNet model gives the worst complexity–performance trade-off because the extra FC hidden layer in its shortcut branch adds a large number of parameters to the model. Comparing the vanilla RNNs (i.e., LSTM and GRU) and their corresponding Res-RNN models, we see that the simple RNN module in the residual path ensures a satisfying model performance while bringing down the complexity.

In brief, model performance improves as the complexity grows but both training and test performance will hit a plateau at some point. No obvious overfitting was observed during our exploration, suggesting that the proposed models are not over-parameterized and well-trained.

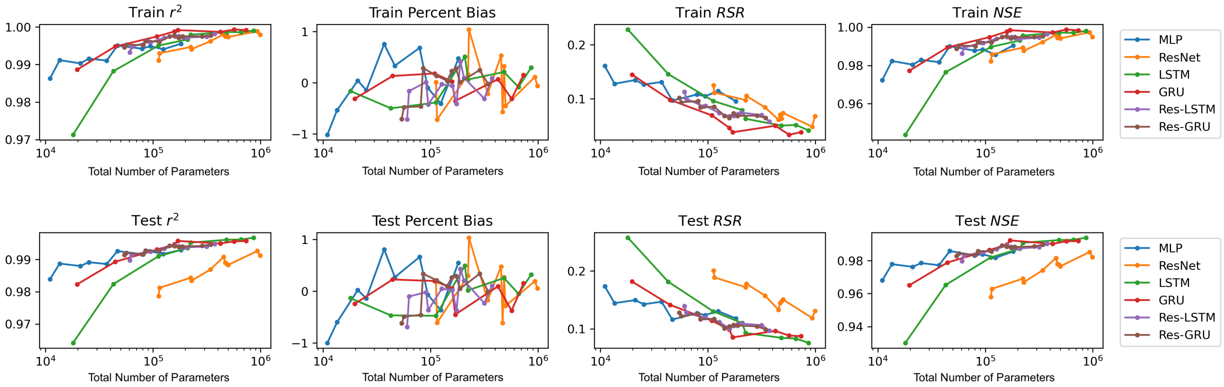


Figure 13. Model performance versus total numbers of parameters of the proposed models and their variants with varying structural complexities.

It is worth noting that there are other methods that can be applied to assess the overfitting potential of machine learning models. The current study briefly explored two of those methods, namely data distortion and cross validation for demonstration purposes. The results (Figures A5–A8 in Appendix F) also indicated that no evident overfitting was observed.

4.2. Comparing with a Process-Based Model

It is shown in Section 3 that the proposed ML models in this study, particularly the novel Res-LSTM and Res-GRU models, can simulate and predict real-world salinity in the Delta well. Traditionally, process-based models, including the operational hydrodynamics and water quality model DSM2, have been applied to simulate and predict the spatial and temporal variations of the salinity across the Delta. It is imperative to assess the performance of these ML models against that of the operational process-based models. For illustration purposes, this sub-section compares simulations from the Res-LSTM model and its counterparts from DSM2.

The comparison is conducted by means of both visual inspection and evaluating statistical metrics. Figure 14 shows the comparison among time series plots of the measured DSM2-simulated and Res-LSTM-simulated EC data at six selected key stations. Four study metrics (r^2, Bias, RSR, NSE) of both sets of models are also displayed side-by-side to conduct quantitative comparison. The corresponding comparison for all 23 study locations is provided in Appendix G. Both simulations mimic the target filed measurements of salinity very well via visually inspecting the time series plot. The overall biases of both models are generally small. For Res-LSTM, the absolute bias ranges from 0.3% to 4.6%. The r^2, RSR and NSE values of Res-LSTM are notably better than their counterparts of DSM2 for all six locations. Collectively, these observations indicate that Res-LSTM yields comparable or more desirable salinity simulations compared to the operational DSM2 model.

Nevertheless, it should be pointed out that DSM2 can generate salinity simulations not only at the 23 study locations, but at all channels across the Delta. The ML models proposed in the study can only be applied to study locations where they have been trained, and thus are not meant to be substitutes of process-based models, including DSM2.

4.3. Implications

This study has important scientific implications. Firstly, the study exemplifies the feasibility of applying ML, particularly deep learning models, as a new scientific exploratory tool to tackle a complex problem. Secondly, this study proposes two novel deep learning models (i.e., Res-LSTM and Res-GRU) that have never been explored before. These novel models can be applied to simulate other variables, including water temperature, suspended sediment, dissolved oxygen, and other water quality variables, that are important to guide water management practices in the Delta. Thirdly, this study illustrates the forecasting capability of newly developed deep-learning models. Effective and efficient forecasting models are valuable tools that can guide real-time operations in light of forecast near-term salinity conditions.

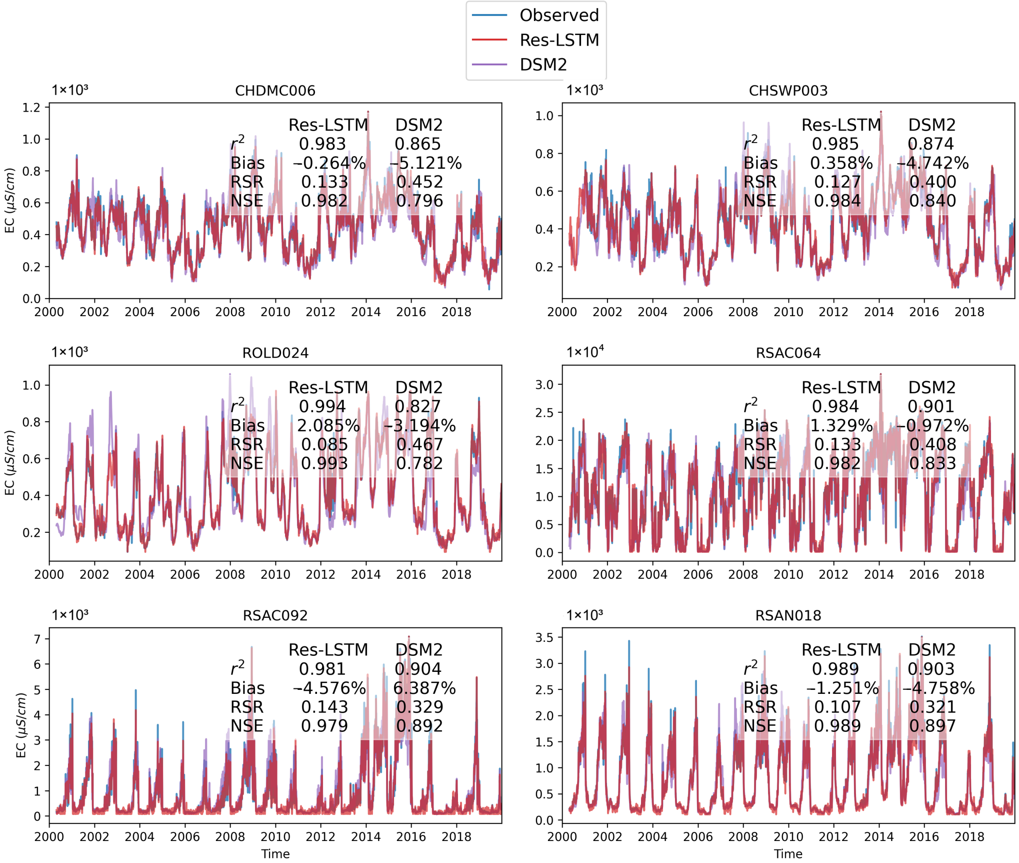


Figure 14. Time series plots of observed salinity levels versus Res-LSTM simulations and DSM2 simulations of six key stations. Detailed values of four evaluation metrics of Res-LSTM and DSM2 are marked for each station.

There are also important practical implications of this study. The study demonstrates that the proposed ML models are capable of generating desirable salinity simulations and predictions even on the hourly scale. The overall absolute biases are generally less than 5%. The correlation coefficients, RSR, and NSE values are generally satisfactory. They are either comparable or superior compared to the corresponding metrics of the DSM2 model. In addition to accuracy, the proposed ML models are also of high efficiency. DSM2 runs can take hours depending upon the simulation period length [28], while the runtime for trained ML models for the same inputs is measured in seconds. This is particularly appealing, for instance, for real-time operations which require quick turn-around time and also studies that require running multiple scenarios during the historical period.

4.4. Limitations and Future Work

Despite those scientific and practical implications, this study has several limitations. First of all, the current study randomly splits the dataset into a training subset and a test subset. Though random split of training/testing in not uncommon in ML studies in the Delta [2,10,26–28,54], there are other data-split methods available. Ref. [29] examined chronological split and manual split and found that the performance of the random split method yielded improved results over the other methods. Since the observed data are about one-third shorter than the simulated data used in [29], this study did not employ the chronological/manual methods. In our future work focusing on larger datasets, we will explore chronological split and manual split methods. In addition, explainable artificial intelligence (XAI) is an active research area [60]. There have been a number of XAI approaches developed. In our future work, we plan to explore various XAI approaches, including the gradient-based method [61] and the backpropagation-based method [62], and the input variable sensitivity analysis [57], and conduct in-depth investigation of the importance of different inputs features in different regions and locations in the Delta.

This study used eight empirical variables as input features to the proposed models. They were shown to yield desirable salinity estimation at the study locations. However, other variables, including precipitation and wind speed, also influence the circulation and mixing of freshwater and sea water and thus affect the salinity level in the study area. In our future work, we will also explore the impacts of additional input features on the ability of the proposed models to estimate Delta salinity.

In this study, the ML models are trained and tested on historical time series. The selected range of data may not capture potential hydrologic extremes or increased water use. Climate change is expected to cause larger storm-driven streamflow and altered runoff timing [63]. In addition, in the coming decades, municipal, industrial and agricultural water demand in California is projected to increase from increasing urbanization and changing agricultural practices [64]. The models trained on historical data, therefore, will not be exposed to the range of inflows and operational constraints resulting from potential future conditions. Additionally, the ML models are trained using data from 23 study locations and can thus only be applied to those study locations. In the future, we will explore generic ML models capable of generating salinity estimates at locations they are not trained upon.

Exclusively using the historical record for training may also introduce shortcomings when conducting long-term planning studies, as the training dataset is not modified beyond the scope of historical operational considerations. In real-time operations and planning, measures such as emergency barriers and temporary operational regimes may be implemented to manage flow and salinity. Operational measures include emergency temporary barriers [65] to manage salinity intrusion and maintain acceptable water quality at pumping locations. The historical time series reflects limited use of emergency operational measures and does not consider the operation of the Suisun Marsh Salinity Control Gates (SMSCG). The limitations associated with using a historical dataset during training will be addressed in future work, where the model will be trained using synthetically augmented datasets.

Data augmentation is a technique to generate synthetic data for model training, which incorporates an enlarged and diversified dataset to better represent extreme conditions and possible future conditions. In our ongoing preliminary tests, we use DSM2 historical simulation as the baseline, then apply several modifications, including (1) scaling the magnitude of major boundary flows; (2) temporally shifting major boundary flows; and (3) changing operations of key Delta structures, such as operable gates. All of the above aim to reduce overfitting and thus improve the generalization ability of the trained neural networks. Another benefit of data augmentation is that it can provide sufficient time series for chronological split training to bypass the limit of random split method.

In another follow-up work, we plan to explore the physics informed neural network (PINN), a cutting-edge neural network algorithm which can embed the knowledge of physical laws into data training. Most of the physical laws governing the dynamics of a system can be described by partial differential equations (PDEs). PINN adds the underlying PDE of the dynamics directly into the loss function of the neural network [66,67]. For our study, we plan on implementing PINN with the one-dimensional advection–dispersion equation for salinity transport [14,68].

PINN can be viewed as a regularization limiting the space of admissible solutions, through adding the prior knowledge of general physical laws in the training of neural networks. Its benefits include increasing the correctness of the function approximation, facilitating the learning algorithm to capture the right solution, generalizing well, even with a low amount of training examples, and providing a meshfree alternative to traditional approaches. PINN could also be viewed as a paradigm that bridges the gap between the process-based models, which are developed from the known PDEs of the dynamics, and the machine-learning models, which are driven purely by existing data. By integrating the best of both methods, namely, the physics information of the process-based models and the training data employed in the machine-learning models, PINN could learn the underlying solution of the dynamics more accurately and more efficiently.

Moreover, this study explored the forecasting capability of the two proposed novel ML models using historical data up to the forecast time only. The main reason is that we do not have forecasted input variables available. It is expected that the performance of the ML models would be even more satisfactory should forecasted model inputs be used to drive the models. In a follow-up study, we plan to collect archived forecasted inputs and salinity data from real-time operations, and develop ML models based on them, and test the trained model in a hindcasting mode.

Lastly, we are developing an interactive dashboard to integrate and visualize our modeling results. It is designed as a front-end to the trained neural network model engine, allowing users to customize data inputs, run the ANNs, and query results. The proposed dashboard could generate a visualization of the time-series output and their proposed metrics for all the specified locations across the Delta. This tool could facilitate management decisions in historical, real-time, and forecast applications.

5. Conclusions

Built upon the success of relevant previous studies that explored machine learning applications in salinity modeling in the Delta, this study develops two novel deep learning models and applies them in both salinity simulation and prediction as well as on a finer hourly time scale that had never been investigated before. The study shows that the novel models proposed can effectively simulate and predict salinity at all study locations across the Delta. In addition, compared to traditional process-based models, the proposed models run significantly faster once trained. Their effectiveness and efficiency make them viable supplements to operational process-based models in terms of providing salinity estimates to inform both real-time and long-term water management and planning practices.

Author Contributions: Conceptualization, P.S., Z.B., Z.D. and M.H.; Data curation, B.T., R.H., P.N. and Y.Z.; Funding acquisition, P.S.; Investigation, S.Q. and M.H.; Methodology, S.Q., M.H., Z.B., Z.D., P.S. and J.A.; Project administration, P.S.; Software, S.Q.; Validation, M.H., R.H., P.N. and Y.Z.; Visualization, S.Q., M.H.; Writing—original draft, S.Q., M.H., Y.Z., B.T., R.H., P.N. and J.A.; Writing—review and editing, Z.B., P.S., J.A., F.C. and D.M.R. 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: Data availability is described in Appendix A.

Acknowledgments: The authors would like to thank the editors and two 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. Data Sources

Input data for the ML models along with observed salinity data are available at the following link: https://data.cnra.ca.gov/dataset/dsm2-v8-2-1 (accessed on 1 July 2022).

Appendix B. Summary of Datasets from Previous Studies

Table A1. Datasets applied in relevant previous studies.

Study	Dataset
Rath et al. (2016) [2]	The input features of this study are daily freshwater flow to the estuary, daily mean coastal water level, and the daily tidal range for water years 1922–2012. Labels are salinity data from nine locations collected from sensors in the Delta.
Chen et al. (2018) [28]	In this study, the machine learning emulator is based on data generated using DSM2 (a process-based model) including its outputs at 17 locations for 10 scenarios (two decades each). The use of 10 scenarios is intended to augment the dataset to bound the range of possible water management operations. The study period 1990–2010 was strategically selected as it contains widely varying hydrology and is a period where the DSM2 model is well-calibrated.
Mosavi et al. (2018) [31]	In this study, the authors examined studies that used field data from rain gauges and other sensing devices, including data from remote sensing technologies.
He et al. (2020) [10]	The dataset includes a 24-year period (WY 1991–2014) of daily observed water stage at Martinez, Martinez salinity and the net Delta outflow.
Jayasundara et al. (2020) [26]	Input features used are Northern flow, San Joaquin River flow, exports, Delta cross channel gate operation, net Delta consumptive use, tidal energy, and San Joaquin River inflow salinity at Vernalis. Labels include multiple sets of DSM2 simulated salinity data representing a range of operational conditions.
Qi et al. (2021) [27]	The input features in this study are the same as Jayasundara et al. (2020) and the labels are DSM2-simulated salinity data at 12 locations.
Qi et al. (2022) [29]	The input features of this study are eight inputs representing boundary flows or operating rules for Delta flow and salinity management. DSM2 simulated daily salinity at the 28 study locations during 1990–2019 is used as the training label dataset.

Appendix C. Diagrams of MLP, ResNet, LSTM and GRU Networks

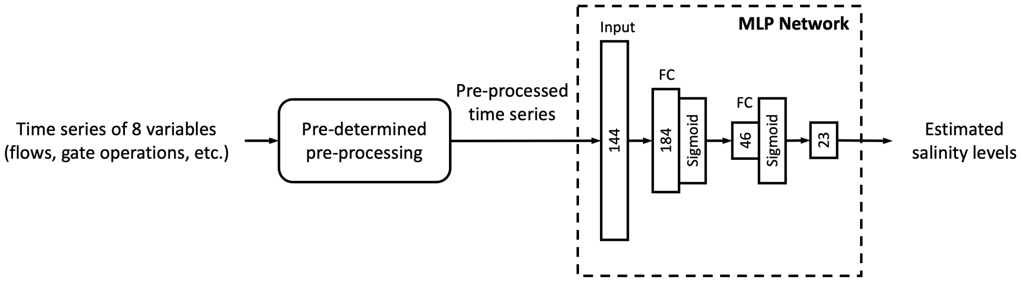

Time series of 8 variables
(flows, gate operations, etc.)
Pre-determined pre-processing
Pre-processed time series
Input
MLP Network
144
FC
184
Sigmoid
FC
46
Sigmoid
23
Estimated salinity levels

Figure A1. Diagram of the MLP network from [29]. The number in the input layer denotes input shape and those in the subsequent FC layers represent the numbers of neurons of the layers.

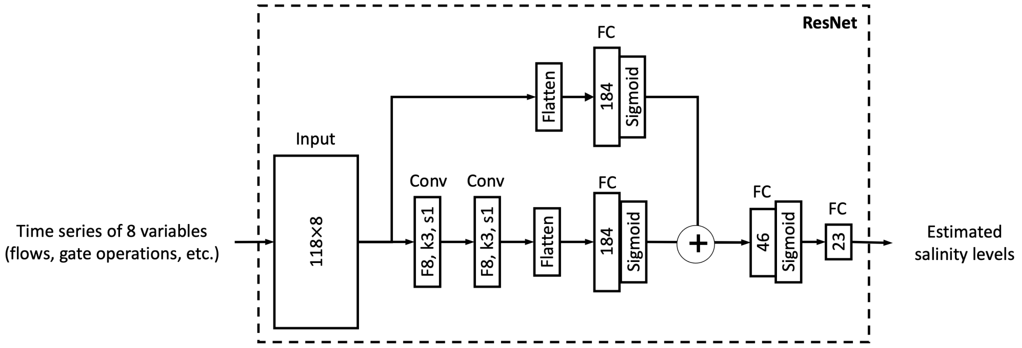


Figure A2. Diagram of the ResNet network from [29]. The number in the input layer denotes input shape and those in the FC layers represent the numbers of units/neurons of the layers. In the convolutional layers following the input layer, "f" denotes the number of convolutional filters, "k" denotes size of convolutional kernels and "s" denotes stride.

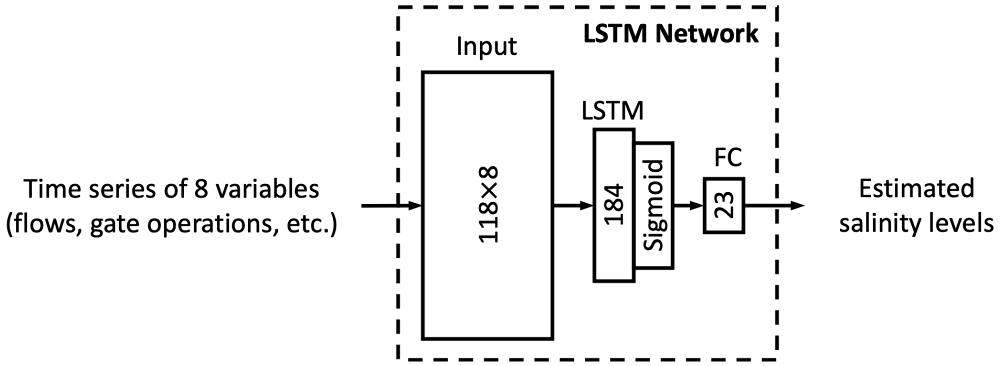


Figure A3. Diagram of the LSTM network from [29]. The number in the input layer denotes input shape and those in the subsequent layers represent the numbers of units/neurons of the layers.

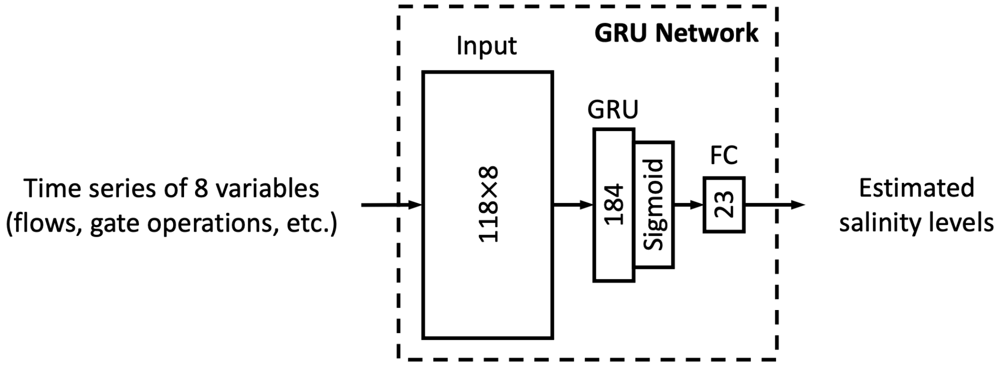


Figure A4. Diagram of the GRU network from [29]. The number in the input layer denotes input shape and those in the subsequent layers represent the numbers of units/neurons of the layers.

Appendix D. Detailed Values for Figures 7 and 12

Table A2. Station-wise r^2 results of Res-LSTM at hourly time step with daily inputs.

r^2	0∼75%	75∼95%	95∼100%
RSAC064	0.9529	0.5940	0.6464
SLCBN002	0.9748	0.8334	0.6973
SLSUS012	0.9909	0.8803	0.8518
SLMZU011	0.9904	0.9111	0.9003
RSAC075	0.9760	0.8008	0.7332
SLMZU025	0.9777	0.7759	0.7619
RSAC081	0.9623	0.7956	0.7625
RSAN007	0.9568	0.8286	0.8028
ROLD059	0.9728	0.7287	0.6267
RSAN058	0.9809	0.9135	0.9437
OLD MID	0.9817	0.8729	0.6846
RSAN072	0.9860	0.8975	0.8538
SLDUT007	0.9905	0.9156	0.9509
CHDMC006	0.9064	0.6368	0.5679
CHSWP003	0.9379	0.5695	0.4932
RSAN018	0.9434	0.7981	0.7969
CHVCT000	0.9894	0.9708	0.9405
ROLD024	0.9860	0.8916	0.8700
SLTRM004	0.9027	0.7200	0.8676
RSAC092	0.5819	0.8165	0.8177
RSAN037	0.9815	0.9221	0.8310
RSAN032	0.8751	0.7713	0.7742
RSMKL008	0.9484	0.7347	0.8541

Table A3. Station-wise percent bias results of Res-LSTM at hourly time step with daily inputs.

Bias	**0∼75%**	**75∼95%**	**95∼100%**
RSAC064	0.4684	−2.8600	−3.7659
SLCBN002	−1.3600	−0.8149	−1.4523
SLSUS012	0.2163	−0.1661	−0.6855
SLMZU011	−0.2336	−0.7538	−1.0799
RSAC075	1.5492	−0.4411	−0.8134
SLMZU025	2.1184	−0.0241	−1.2001
RSAC081	4.8211	−0.0190	−1.6114
RSAN007	−2.6612	−0.8044	−2.3899
ROLD059	0.0202	−1.0317	−1.8503
RSAN058	−0.3692	−1.2292	−1.3330
OLD MID	−0.0381	−0.3858	−0.9698
RSAN072	0.5610	−0.1732	−0.1627
SLDUT007	1.9058	0.8568	−0.0001
CHDMC006	1.3642	−1.4991	−3.1108
CHSWP003	0.9000	−1.1438	−3.7689
RSAN018	3.4133	2.5636	−2.7383
CHVCT000	0.5407	0.2339	0.1670
ROLD024	0.5134	0.3675	0.1518
SLTRM004	−1.2958	−0.8454	−1.6861
RSAC092	−1.6766	1.0572	−3.6682
RSAN037	0.9873	0.2697	−0.8718
RSAN032	−1.3944	−0.8808	−4.2074
RSMKL008	0.6971	−0.2226	−0.6141

Table A4. Station-wise RSR results of Res-LSTM at hourly time step with daily inputs.

RSR	**0∼75%**	**75∼95%**	**95∼100%**
RSAC064	0.2200	0.8297	0.8356
SLCBN002	0.1609	0.4508	0.6052
SLSUS012	0.0960	0.3660	0.4214
SLMZU011	0.0981	0.3109	0.3718
RSAC075	0.1582	0.4841	0.5874
SLMZU025	0.1552	0.5096	0.5669
RSAC081	0.2087	0.4888	0.5559
RSAN007	0.2201	0.4363	0.5086
ROLD059	0.1661	0.6113	0.7402
RSAN058	0.1385	0.3238	0.2714
OLD MID	0.1359	0.3856	0.6345
RSAN072	0.1191	0.3383	0.4005
SLDUT007	0.1023	0.3063	0.2214
CHDMC006	0.3243	0.7288	0.8950
CHSWP003	0.2598	0.8280	0.9698
RSAN018	0.2541	0.5150	0.4731
CHVCT000	0.1037	0.1754	0.2459
ROLD024	0.1207	0.3509	0.3848
SLTRM004	0.3391	0.6186	0.3708
RSAC092	0.9923	0.4407	0.4853
RSAN037	0.1392	0.2900	0.4270
RSAN032	0.3860	0.5242	0.5271
RSMKL008	0.2320	0.5844	0.4060

Table A5. Station-wise NSE results of Res-LSTM at hourly time step with daily inputs.

NSE	0～75%	75～95%	95～100%
RSAC064	0.9516	0.3117	0.3017
SLCBN002	0.9741	0.7968	0.6337
SLSUS012	0.9908	0.8660	0.8224
SLMZU011	0.9904	0.9033	0.8618
RSAC075	0.9750	0.7657	0.6549
SLMZU025	0.9759	0.7403	0.6786
RSAC081	0.9564	0.7611	0.6910
RSAN007	0.9516	0.8096	0.7413
ROLD059	0.9724	0.6263	0.4521
RSAN058	0.9808	0.8952	0.9264
OLD MID	0.9815	0.8513	0.5975
RSAN072	0.9858	0.8856	0.8396
SLDUT007	0.9895	0.9062	0.9510
CHDMC006	0.8948	0.4689	0.1990
CHSWP003	0.9325	0.3145	0.0595
RSAN018	0.9354	0.7348	0.7762
CHVCT000	0.9892	0.9692	0.9395
ROLD024	0.9854	0.8769	0.8519
SLTRM004	0.8850	0.6174	0.8625
RSAC092	0.0154	0.8058	0.7645
RSAN037	0.9806	0.9159	0.8177
RSAN032	0.8510	0.7252	0.7222
RSMKL008	0.9462	0.6585	0.8351

Table A6. Station-wise r^2 results of Res-LSTM at daily time step.

r^2	0～75%	75～95%	95～100%
RSAC064	0.9744	0.5923	0.7580
SLCBN002	0.9854	0.8367	0.6724
SLSUS012	0.9915	0.8796	0.8667
SLMZU011	0.9865	0.8907	0.8532
RSAC075	0.9836	0.8220	0.7386
SLMZU025	0.9834	0.8091	0.7832
RSAC081	0.9764	0.8359	0.7744
RSAN007	0.9730	0.8189	0.7607
ROLD059	0.9805	0.7518	0.7275
RSAN058	0.9797	0.9156	0.9489
OLD MID	0.9792	0.8817	0.6887
RSAN072	0.9834	0.9060	0.8040
SLDUT007	0.9910	0.9033	0.9512
CHDMC006	0.9697	0.7823	0.8735
CHSWP003	0.9720	0.7799	0.8500
RSAN018	0.9801	0.7914	0.8068
CHVCT000	0.9855	0.9608	0.9206
ROLD024	0.9852	0.8897	0.8608
SLTRM004	0.9730	0.8626	0.8684
RSAC092	0.8405	0.8825	0.8410
RSAN037	0.9850	0.9560	0.8381
RSAN032	0.9575	0.8381	0.7932
RSMKL008	0.9572	0.8070	0.9379

Table A7. Station-wise percent bias results of Res-LSTM at daily time step.

Bias	**0～75%**	**75～95%**	**95～100%**
RSAC064	3.6008	−1.2224	−1.6747
SLCBN002	3.7842	1.2626	−0.3216
SLSUS012	4.7572	1.8593	0.4100
SLMZU011	0.9901	0.0644	−0.6901
RSAC075	−1.6408	−1.1783	−1.2080
SLMZU025	−2.5367	−0.9088	−1.4725
RSAC081	−2.2831	−1.7392	−1.0961
RSAN007	2.0662	−0.4297	−0.7600
ROLD059	1.9174	0.1989	−0.3832
RSAN058	−0.5765	−1.3418	−0.4266
OLD MID	0.4040	−0.0756	−0.7731
RSAN072	−2.1535	−1.7981	−1.3735
SLDUT007	3.1500	1.0906	−0.1996
CHDMC006	0.3502	−1.3688	−0.7877
CHSWP003	1.0587	−0.6732	−0.5765
RSAN018	−1.9600	−0.1313	−2.5437
CHVCT000	−0.7863	−0.4636	−1.1248
ROLD024	3.5323	0.7066	0.0888
SLTRM004	6.3652	0.9577	−0.8914
RSAC092	−11.2734	−2.8386	−1.7970
RSAN037	−2.6610	−2.3256	−2.4599
RSAN032	0.4398	−0.0980	−2.0932
RSMKL008	−1.2886	−1.8978	−1.8280

Table A8. Station-wise RSR results of Res-LSTM at daily time step.

RSR	**0～75%**	**75～95%**	**95～100%**
RSAC064	0.1681	0.7889	0.6164
SLCBN002	0.1332	0.4571	0.5819
SLSUS012	0.1194	0.4326	0.3862
SLMZU011	0.1183	0.3456	0.3991
RSAC075	0.1287	0.4669	0.6117
SLMZU025	0.1359	0.4665	0.5618
RSAC081	0.1552	0.4445	0.5323
RSAN007	0.1716	0.4434	0.5567
ROLD059	0.1446	0.5735	0.5859
RSAN058	0.1420	0.3179	0.2296
OLD MID	0.1433	0.3778	0.6007
RSAN072	0.1329	0.3776	0.5197
SLDUT007	0.1010	0.3287	0.1968
CHDMC006	0.1745	0.5584	0.3696
CHSWP003	0.1681	0.5464	0.4189
RSAN018	0.1459	0.4913	0.4763
CHVCT000	0.1201	0.2027	0.3275
ROLD024	0.1358	0.3567	0.3865
SLTRM004	0.1801	0.3922	0.3667
RSAC092	0.4723	0.3580	0.4551
RSAN037	0.1303	0.2589	0.5091
RSAN032	0.2149	0.4265	0.5118
RSMKL008	0.2116	0.5490	0.2906

Table A9. Station-wise NSE results of Res-LSTM at daily time step.

NSE	0∼75%	75∼95%	95∼100%
RSAC064	0.9718	0.3777	0.6201
SLCBN002	0.9823	0.7910	0.6613
SLSUS012	0.9857	0.8129	0.8508
SLMZU011	0.9860	0.8806	0.8407
RSAC075	0.9834	0.7820	0.6258
SLMZU025	0.9815	0.7824	0.6843
RSAC081	0.9759	0.8024	0.7166
RSAN007	0.9705	0.8034	0.6901
ROLD059	0.9791	0.6711	0.6567
RSAN058	0.9798	0.8989	0.9473
OLD MID	0.9795	0.8572	0.6392
RSAN072	0.9823	0.8574	0.7299
SLDUT007	0.9898	0.8920	0.9613
CHDMC006	0.9696	0.6881	0.8634
CHSWP003	0.9718	0.7015	0.8246
RSAN018	0.9787	0.7586	0.7731
CHVCT000	0.9856	0.9589	0.8927
ROLD024	0.9816	0.8727	0.8506
SLTRM004	0.9676	0.8461	0.8655
RSAC092	0.7769	0.8718	0.7929
RSAN037	0.9830	0.9330	0.7408
RSAN032	0.9538	0.8181	0.7381
RSMKL008	0.9552	0.6986	0.9155

Appendix E. Numbers of Parameters in Simplified or Complicated Architectures

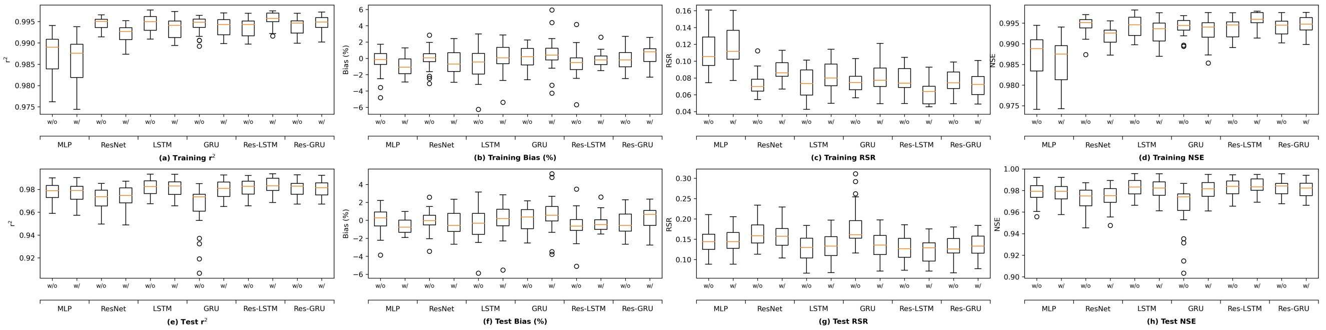
Table A10. Numbers of parameters of simplified or complicated recurrent neural network models.

Number of Units in the Recurrent Layer	LSTM	GRU
322	627,279	486,243
276	486,887	378,695
230	363,423	283,843
184 (Baseline)	256,887	201,687
138	167,279	132,227
92	94,599	75,463
46	38,847	31,395
23	17,319	14,122

Table A11. Numbers of parameters of simplified or complicated MLP, ResNet, Res-LSTM and Res-GRU models.

Numbers of Neurons in Hidden Layers	MLP	ResNet	Res-LSTM	Res-GRU
368,184	125,511	768,119	256,292	237,340
368,92	89,447	732,055	220,228	201,276
184,184	64,975	386,503	140,004	129,516
184,138	55,407	376,935	130,436	119,948
184,92 (Baseline)	367,367	367,367	120,868	110,380
184,46	36,271	357,799	111,300	100,812
138,46	27,485	268,743	88,576	80,204
92,46	18,699	179,687	65,852	59,596
46,46	9913	90,631	43,128	38,988
46,23	8303	89,021	41,518	37,378

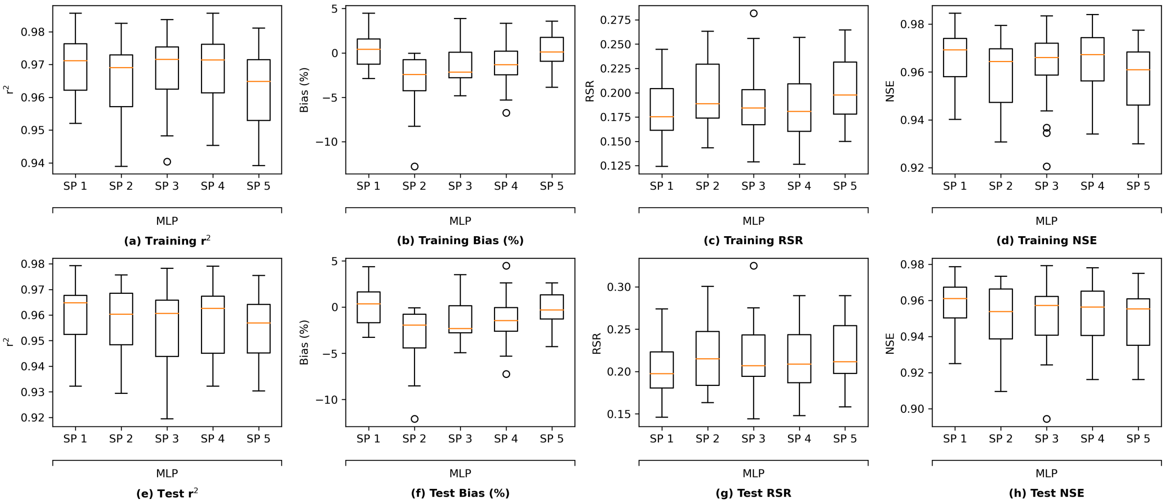
Appendix F. Preliminary Data Distortion and Cross-Validation Results



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

(e) Test r^2 (f) Test Bias (%) (g) Test RSR (h) Test NSE

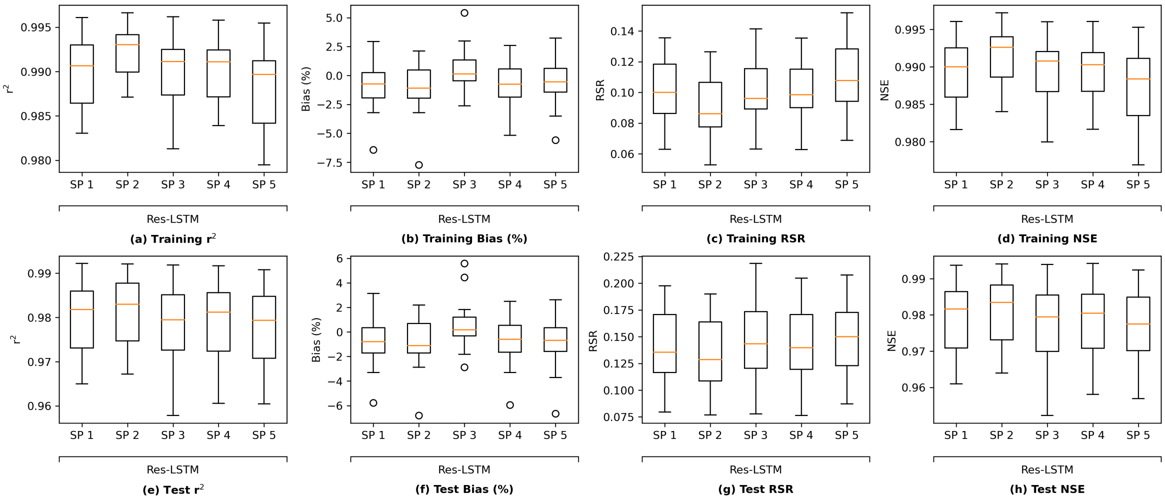
Figure A5. Comparison of six models on observed data without ("w/o") or with ("w/") data distortion.



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

(e) Test r^2 (f) Test Bias (%) (g) Test RSR (h) Test NSE

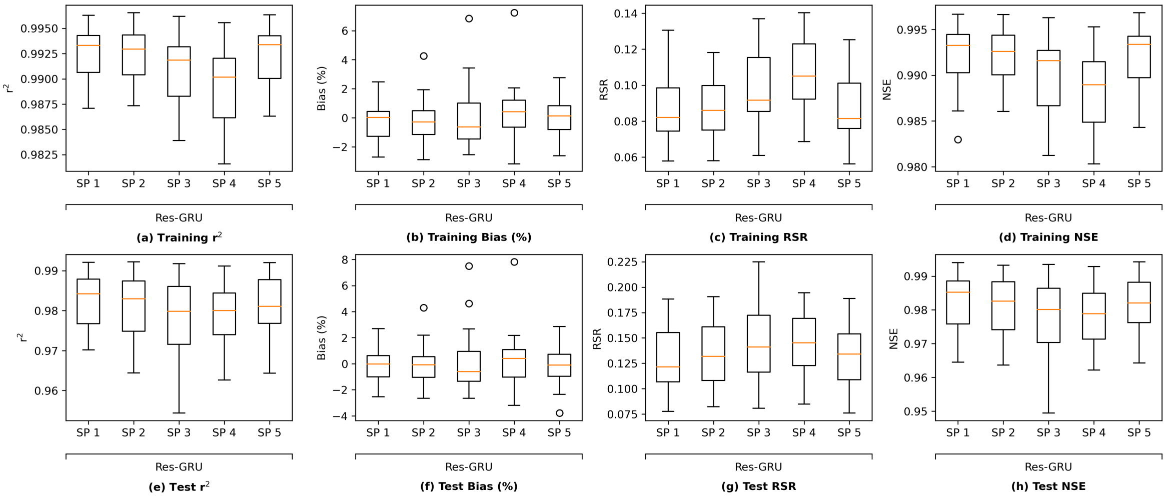
Figure A6. Comparison of the 5-fold cross-validation on the MLP architecture using observed data. "SP" stands for "split".



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

(e) Test r^2 (f) Test Bias (%) (g) Test RSR (h) Test NSE

Figure A7. Comparison of the 5-fold cross-validation on the Res-LSTM architecture using observed data. "SP" stands for "split".



(a) Training r^2

(b) Training Bias (%)

(c) Training RSR

(d) Training NSE

(e) Test r^2

(f) Test Bias (%)

(g) Test RSR

(h) Test NSE

Figure A8. Comparison of the 5-fold cross-validation on the Res-GRU architecture using observed data. "SP" stands for "split".

Appendix G. Time Series Plots of Observed Salinity Levels Versus Model Simulations

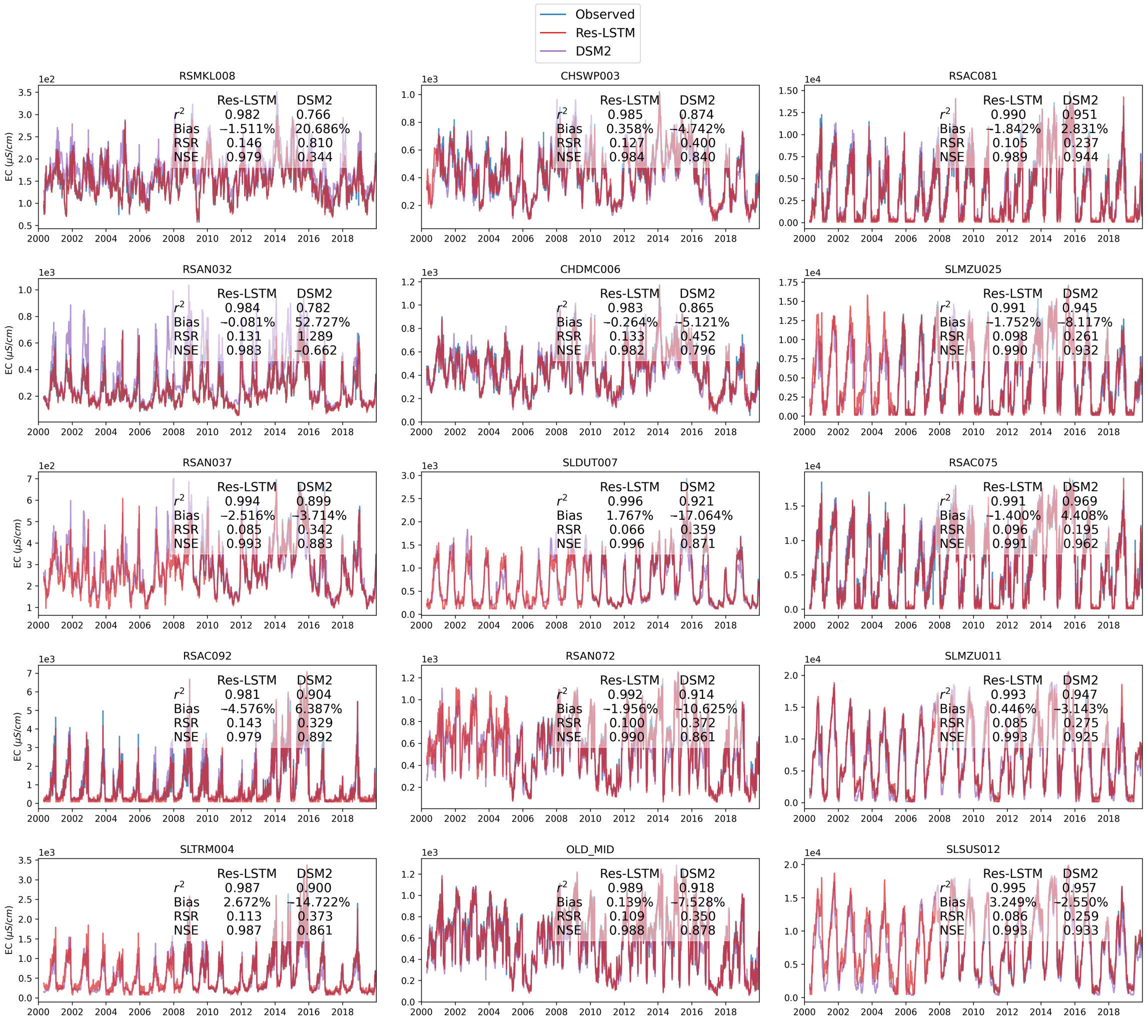


Figure A9. *Cont.*

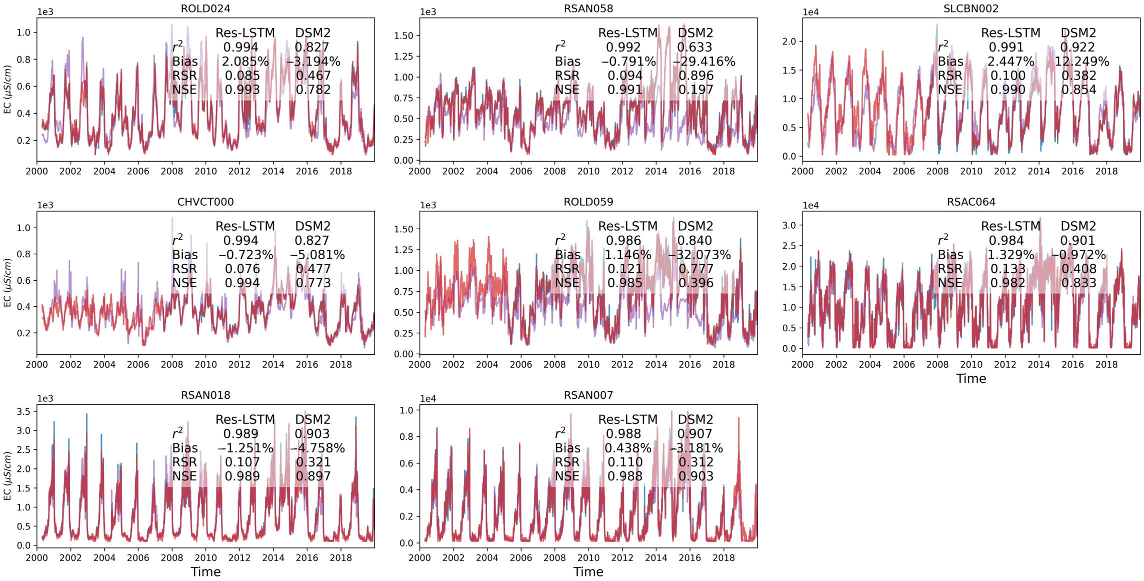


Figure A9. Time series plots of observed salinity levels versus Res-LSTM simulations and DSM2 simulations of the 23 stations. Detailed values of four evaluation metrics of Res-LSTM and DSM2 are marked for each station.

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. MDBMC. The Salinity Audit of the Murray-Darling Basin, A 100-Year Perspective; Murray-Darling Basin Commission: Canberra, Australia, 1999. Available online: https://www.mdba.gov.au/sites/default/files/archived/mdbc-salinity-reports/2072_Salinity_audit_of_MDB_100_year_perspective.pdf (accessed on 1 July 2022).
7. MDBMC. Basin Salinity Management 2030 (BSM2030), *MDBA Publication No 21/15*; Murray–Darling Basin Ministerial Council: Canberra, Australia, 2015.
8. 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]
9. 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]
10. 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]
11. Healey, M.; Dettinger, M.; Norgaard, R. Perspectives on Bay–Delta Science and Policy. *San Fr. Estuary Watershed Sci.* **2016**, *14*. [CrossRef]
12. 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*; CDWR: Sacramento, CA, USA, 1990.
13. 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*; CDWR: Sacramento, CA, USA, 1991.
14. Denton, R.A. Accounting for Antecedent Conditions in Seawater Intrusion Modeling—Applications for the San Francisco Bay-Delta. In *Hydraulic Engineering*; ASCE: Reston, FL, USA, 1993; pp. 448–453.
15. 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]
16. DeGeorge, J.F. *A Multi-Dimensional Finite Element Transport Model Utilizing a Characteristic-Galerkin Algorithm*; University of California: Davis, CA, USA, 1996.
17. Hutton, P.H.; Rath, J.S.; Chen, L.; Ungs, M.J.; Roy, S.B. Nine decades of salinity observations in the San Francisco Bay and Delta: Modeling and trend evaluations. *J. Water Resour. Plan. Manag.* **2016**, *142*, 04015069. [CrossRef]

18. 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]
19. 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]
20. 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]
21. Sandhu, N.; Finch, R. Application of artificial neural networks to the Sacramento-San Joaquin Delta. In *Estuarine and Coastal Modeling*; ASCE: Reston, FL, USA, 1995; pp. 490–504.
22. CDWR. *Modeling Flow-Salinity Relationships in the Sacramento-San Joaquin Delta Using Artificial Neural Networks*; Technical Information Record OSP-99-1; CDWR: Sacramento, CA, USA, 1999.
23. 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*; CDWR: Sacramento, CA, USA, 2001.
24. 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*; CDWR: Sacramento, CA, USA, 2002.
25. 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*; CDWR: Sacramento, CA, USA, 2007.
26. 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]
27. 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]
28. Chen, L.; Roy, S.B.; Hutton, P.H. Emulation of a process-based estuarine hydrodynamic model. *Hydrol. Sci. J.* **2018**, *63*, 783–802. [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. Zounemat-Kermani, M.; Matta, E.; Cominola, A.; Xia, X.; Zhang, Q.; Liang, Q.; Hinkelmann, R. Neurocomputing in surface water hydrology and hydraulics: A review of two decades retrospective, current status and future prospects. *J. Hydrol.* **2020**, *588*, 125085. [CrossRef]
31. Mosavi, A.; Ozturk, P.; Chau, K.W. Flood prediction using machine learning models: Literature review. *Water* **2018**, *10*, 1536. [CrossRef]
32. Yaseen, Z.M.; Sulaiman, S.O.; Deo, R.C.; Chau, K.W. An enhanced extreme learning machine model for river flow forecasting: State-of-the-art, practical applications in water resource engineering area and future research direction. *J. Hydrol.* **2019**, *569*, 387–408. [CrossRef]
33. Tongal, H.; Booij, M.J. Simulation and forecasting of streamflows using machine learning models coupled with base flow separation. *J. Hydrol.* **2018**, *564*, 266–282. [CrossRef]
34. Islam, A.R.M.T.; Talukdar, S.; Mahato, S.; Kundu, S.; Eibek, K.U.; Pham, Q.B.; Kuriqi, A.; Linh, N.T.T. Flood susceptibility modelling using advanced ensemble machine learning models. *Geosci. Front.* **2021**, *12*, 101075. [CrossRef]
35. Shahabi, H.; Shirzadi, A.; Ghaderi, K.; Omidvar, E.; Al-Ansari, N.; Clague, J.J.; Geertsema, M.; Khosravi, K.; Amini, A.; Bahrami, S.; et al. Flood detection and susceptibility mapping using sentinel-1 remote sensing data and a machine learning approach: Hybrid intelligence of bagging ensemble based on k-nearest neighbor classifier. *Remote Sens.* **2020**, *12*, 266. [CrossRef]
36. Costache, R.; Hong, H.; Pham, Q.B. Comparative assessment of the flash-flood potential within small mountain catchments using bivariate statistics and their novel hybrid integration with machine learning models. *Sci. Total. Environ.* **2020**, *711*, 134514. [CrossRef]
37. Tang, Y.; Zang, C.; Wei, Y.; Jiang, M. Data-driven modeling of groundwater level with least-square support vector machine and spatial–temporal analysis. *Geotech. Geol. Eng.* **2019**, *37*, 1661–1670. [CrossRef]
38. El Bilali, A.; Taleb, A.; Brouziyne, Y. Groundwater quality forecasting using machine learning algorithms for irrigation purposes. *Agric. Water Manag.* **2021**, *245*, 106625. [CrossRef]
39. Yin, J.; Medellín-Azuara, J.; Escriva-Bou, A.; Liu, Z. Bayesian machine learning ensemble approach to quantify model uncertainty in predicting groundwater storage change. *Sci. Total. Environ.* **2021**, *769*, 144715. [CrossRef]
40. Kumar, D.; Pandey, A.; Sharma, N.; Flügel, W.A. Daily suspended sediment simulation using machine learning approach. *Catena* **2016**, *138*, 77–90. [CrossRef]
41. Choubin, B.; Darabi, H.; Rahmati, O.; Sajedi-Hosseini, F.; Kløve, B. River suspended sediment modelling using the CART model: A comparative study of machine learning techniques. *Sci. Total. Environ.* **2018**, *615*, 272–281. [CrossRef]
42. Melesse, A.M.; Khosravi, K.; Tiefenbacher, J.P.; Heddam, S.; Kim, S.; Mosavi, A.; Pham, B.T. River water salinity prediction using hybrid machine learning models. *Water* **2020**, *12*, 2951. [CrossRef]

43. Nauman, T.W.; Ely, C.P.; Miller, M.P.; Duniway, M.C. Salinity yield modeling of the Upper Colorado River Basin using 30-m resolution soil maps and random forests. *Water Resour. Res.* **2019**, *55*, 4954–4973. [CrossRef]
44. Derot, J.; Yajima, H.; Jacquet, S. Advances in forecasting harmful algal blooms using machine learning models: A case study with Planktothrix rubescens in Lake Geneva. *Harmful Algae* **2020**, *99*, 101906. [CrossRef]
45. Alizadeh, M.J.; Kavianpour, M.R.; Danesh, M.; Adolf, J.; Shamshirband, S.; Chau, K.W. Effect of river flow on the quality of estuarine and coastal waters using machine learning models. *Eng. Appl. Comput. Fluid Mech.* **2018**, *12*, 810–823. [CrossRef]
46. Shamshirband, S.; Jafari Nodoushan, E.; Adolf, J.E.; Abdul Manaf, A.; Mosavi, A.; Chau, K.w. Ensemble models with uncertainty analysis for multi-day ahead forecasting of chlorophyll a concentration in coastal waters. *Eng. Appl. Comput. Fluid Mech.* **2019**, *13*, 91–101. [CrossRef]
47. Jiang, Y.; Zhang, T.; Gou, Y.; He, L.; Bai, H.; Hu, C. High-resolution temperature and salinity model analysis using support vector regression. *J. Ambient. Intell. Humaniz. Comput.* **2018**, 1–9. [CrossRef]
48. Thai-Nghe, N.; Thanh-Hai, N.; Chi Ngon, N. Deep learning approach for forecasting water quality in IoT systems. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*, 686–693. [CrossRef]
49. Granata, F.; Papirio, S.; Esposito, G.; Gargano, R.; De Marinis, G. Machine learning algorithms for the forecasting of wastewater quality indicators. *Water* **2017**, *9*, 105. [CrossRef]
50. Barzegar, R.; Asghari Moghaddam, A.; Adamowski, J.; Ozga-Zielinski, B. Multi-step water quality forecasting using a boosting ensemble multi-wavelet extreme learning machine model. *Stoch. Environ. Res. Risk Assess.* **2018**, *32*, 799–813. [CrossRef]
51. Ahmed, A.N.; Othman, F.B.; Afan, H.A.; Ibrahim, R.K.; Fai, C.M.; Hossain, M.S.; Ehteram, M.; Elshafie, A. Machine learning methods for better water quality prediction. *J. Hydrol.* **2019**, *578*, 124084. [CrossRef]
52. Ghalambor, C.K.; Gross, E.S.; Grosholtz, E.D.; Jeffries, K.M.; Largier, J.K.; McCormick, S.D.; Sommer, T.; Velotta, J.; Whitehead, A. Ecological Effects of Climate-Driven Salinity Variation in the San Francisco Estuary: Can We Anticipate and Manage the Coming Changes? *San Fr. Estuary Watershed Sci.* **2021**, *19*. [CrossRef]
53. Lund, J.R. California's agricultural and urban water supply reliability and the Sacramento–San Joaquin Delta. *San Fr. Estuary Watershed Sci.* **2016**, *14*. [CrossRef]
54. Namadi, P.; He, M.; Sandhu, P. Salinity-constituent conversion in South Sacramento-San Joaquin Delta of California via machine learning. *Earth Sci. Informatics* **2022**, *15*, 1–16. [CrossRef]
55. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
56. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
57. Panda, S.S.; Amatya, D.M.; Muwamba, A.; Chescheir, G. Estimation of evapotranspiration and its parameters for pine, switchgrass, and intercropping with remotely-sensed images based geospatial modeling. *Environ. Model. Softw.* **2019**, *121*, 104487. [CrossRef]
58. Dietterich, T. Overfitting and undercomputing in machine learning. *ACM Comput. Surv. (CSUR)* **1995**, *27*, 326–327. [CrossRef]
59. Ying, X. An overview of overfitting and its solutions. In *Journal of Physics*; Conference Series; IOP Publishing: Bristol, UK, 2019; Volume 1168, p. 022022.
60. Adadi, A.; Berrada, M. Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access* **2018**, *6*, 52138–52160. [CrossRef]
61. Ancona, M.; Ceolini, E.; Öztireli, C.; Gross, M. Towards better understanding of gradient-based attribution methods for deep neural networks. *arXiv* **2017**, arXiv:1711.06104.
62. Shrikumar, A.; Greenside, P.; Kundaje, A. Learning important features through propagating activation differences. In Proceedings of the International Conference on Machine Learning, PMLR, Sydney, Australia, 6–11 August 2017; pp. 3145–3153.
63. Dettinger, M.; Anderson, J.; Anderson, M.; Brown, L.R.; Cayan, D.; Maurer, E. Climate change and the Delta. *San Fr. Estuary Watershed Sci.* **2016**, *14*. [CrossRef]
64. Wilson, T.S.; Sleeter, B.M.; Cameron, D.R. Future land-use related water demand in California. *Environ. Res. Lett.* **2016**, *11*, 054018. [CrossRef]
65. Kimmerer, W.; Wilkerson, F.; Downing, B.; Dugdale, R.; Gross, E.S.; Kayfetz, K.; Khanna, S.; Parker, A.E.; Thompson, J. Effects of drought and the emergency drought barrier on the ecosystem of the California Delta. *San Fr. Estuary Watershed Sci.* **2019**, *17*. [CrossRef]
66. 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]
67. Lu, L.; Meng, X.; Mao, Z.; Karniadakis, G.E. DeepXDE: A deep learning library for solving differential equations. *Siam Rev.* **2021**, *63*, 208–228. [CrossRef]
68. Gay, P.S.; O'Donnell, J. A simple advection-dispersion model for the salt distribution in linearly tapered estuaries. *J. Geophys. Res. Ocean.* **2007**, *112*. [CrossRef]