aquaculture journal

MDPI

Article

Deep Learning Versus Simple Baselines for Fish Weight Prediction from Aquaponic Fish Pond Sensor Data: A Leakage-Aware Evaluation

Mohamed Elmi Abdi, Dzulkarnain Ahmad, Mokhtazul Haizad Mokhtaram and Intan Faraha Abd Ghani *

Faculty of Engineering and Life Sciences, Universiti Selangor, Bestari Jaya 45600, Selangor, Malaysia; 8222000071@student.unisel.edu.my (M.E.A.); dzulahmad@unisel.edu.my (D.A.); mokhtazul@unisel.edu.my (M.H.M.)
* Correspondence: intanfaraha@unisel.edu.my

Abstract

Fish weight in aquaponics is measured infrequently, and sensor data could help to estimate and predict it. In a public aquaponic dataset the daily weights are interpolated between weighings, so that repeating the previous value is almost exact (mean absolute error 0.0026 g over a test period of one pond). Four deep learning architectures were compared with baselines without water quality data, tabular models with water quality data and a network without sensor input, on 70 real weighings of eight aquaponic fish ponds, using leave-one-pond-out validation and the mean over ponds of the mean absolute log error (S). In weight estimation (35 events, seven ponds), a long short-term memory network (LSTM) had an S of 0.290 against 0.265 for the Cohort mean, with a lower S in three of seven ponds (one-sided $p = 0.656$). No deep model had a lower S value than the primary baseline of its task, and the confirmatory test did not reject, which is not evidence of no effect. For prediction 14 days ahead, the lowest S value was 0.211 for a baseline and 0.295 for a deep model. The LSTM used pond-specific water quality information (permutation $p = 0.005$) without being more accurate.

Keywords: aquaponics; African catfish; fish weight prediction; deep learning; long short-term memory; temporal fusion transformer; baseline models; leave-one-pond-out validation; data leakage; water quality

1. Introduction

Aquaponics couples recirculating aquaculture with soil-less plant cultivation in a closed loop in which fish excreta, converted by nitrifying bacteria, become the nutrient supply for the plants, while the plants in turn purify the water returned to the fish tanks. Because the approach conserves water, requires neither soil nor synthetic fertilizer and yields both animal protein and vegetables from a single footprint, it is increasingly regarded as a component of sustainable food production and food security strategies [1–3]. Figure 1 illustrates a representative aquaponic installation comprising fish tanks, mechanical and biological filtration and several hydroponic grow-bed configurations connected in a recirculating loop.

Check for updates

Academic Editors: Aires Oliva-Teles, Youji Wang, Xiaojun Yan and Bin Xia

Received: 28 August 2026
Revised: 30 October 2026
Accepted: 5 November 2026
Published: 24 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

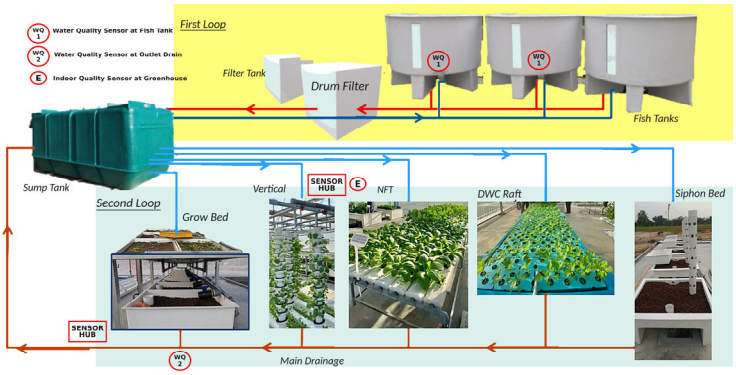

Figure 1. A schematic of an aquaponic system showing the fish tanks, drum filter and filter tank, and the vertical, nutrient film technique (NFT), deep water culture (DWC) raft and siphon grow beds connected in a recirculating loop.

The productivity of such systems depends on maintaining fish and plants within narrow environmental ranges, and management decisions such as feeding rates, stocking density, harvest timing and water quality interventions benefit greatly from the ability to anticipate how fish biomass will develop. Fish weight is therefore a key state variable, but it is measured infrequently and laboriously, whereas water quality variables such as temperature, pH, dissolved oxygen, turbidity, ammonia and nitrate are now recorded continuously by low-cost Internet of Things (IoT) sensors [4,5]. Data-driven models that infer and project fish weight from these continuous measurements would thus provide a practical tool for operational decision-making.

Deep learning has been applied mainly to the prediction of water quality variables: recurrent architectures, in particular long short-term memory (LSTM) networks, have been used to predict the environmental parameters that govern both the aquaculture and the hydroponic components of aquaponic systems [6,7]. The studies that addressed fish length or weight in aquaponics applied regression, tree-based, nearest-neighbor, neural network or neuro-fuzzy models to single sets of measurements [8–10]. None of them tested whether deep sequence models estimate or predict fish weight better than simple baselines, such as a growth curve or the previous weight. The public data make the question hard to answer for three reasons that are not specific to deep learning. First, fish weight is measured by hand about every two weeks [11], and the daily values in the public files are interpolations between the weighings (Section 3.2), so that the label changes once per day and a model that repeats the previous value appears accurate. Second, a random split of windows places windows of the same day, which share the label and have nearly the same inputs, in both the training and the test set; random splits of dependent data underestimate prediction error [12], and leakage of this kind is a documented cause of over-optimistic results in machine learning [13]. Third, the length of a fish, which is measured at the same weighing as its weight, is itself a strong predictor of weight (Section 4.2), so that a model that receives it is not predicting weight from sensor data.

This study therefore compares four deep learning architectures (LSTM, gated recurrent unit (GRU), a hybrid convolutional neural network–LSTM (CNN-LSTM) and the temporal fusion transformer (TFT)) with simple baselines for fish weight in an aquaponic fish pond system, under a validation that addresses these three problems. Two tasks are considered: weight estimation (T1), in which the weight at a weighing is estimated from the time since stocking and the water quality of the preceding 14 days, and weight prediction (T2), in which the weight 14 or 28 days after a weighing is predicted from past weights, time and the water quality of the preceding 28 days. The baselines are growth curves, persistence and other models that use time or past weights only, ridge regression and gradient boosting

with and without water quality features, a network that sees no sensor data, and a reference model that uses fish length. Only real weighings are scored, whole ponds are held out, and the primary models are chosen on inner-validation folds before any test score is computed. The objectives are (i) to test whether the deep models estimate and predict fish weight better than the baselines under this validation and (ii) to assess whether the water quality inputs carry pond-specific information about weight. Section 2 reviews related work; Section 3 describes the dataset, models and evaluation protocol; Section 4 presents the results; Section 5 discusses them; and Section 6 concludes the paper.

2. Related Work

The interaction of biological, chemical and environmental processes in aquaponic systems has motivated the adoption of predictive models that estimate key state variables and support decisions on feeding, stocking and water quality control. Classical machine learning models were among the first to be applied to fish growth estimation. Eneh et al. [8] compared linear, ridge, lasso, k-nearest neighbors and decision tree regressors for predicting fish length and weight from IoT sensor data collected in aquaponic ponds. For fish length, the decision tree regressor gave a root mean squared error (RMSE) of 1.0135 with their initial sensor system and 0.6140 with an improved one (their Table 1). In further tests for fish weight, an extreme gradient boosting (XGBoost) regressor gave an RMSE of 7.0192, a mean absolute error (MAE) of 2.2310 and a coefficient of determination (R^2) of 0.99953 with the initial system, and an RMSE of 0.7143, an MAE of 0.2563 and an R^2 of 0.99825 with the improved system (their Table 2). Although such models are simple and interpretable, they treat each record as an independent sample and therefore disregard the temporal dependencies present in sequential sensor data, which limits their applicability to time-series prediction and, in particular, to forecasting. Similar conclusions apply to the k-nearest neighbors and decision tree regressors evaluated by Kumar et al. [9] on an aquaponic fish pond dataset and to the artificial neural network and neuro-fuzzy models proposed by Debroy and Seban [10] for fish biomass estimation in aquaponics, all of which map a single set of measurements to a weight estimate without exploiting the sequence of observations.

Deep learning has introduced models that learn such dependencies directly from data. LSTM [14] and GRU [15] networks are recurrent architectures designed to retain information over long sequences, and Thai-Nghe et al. [16] demonstrated the effectiveness of LSTM for forecasting water quality in IoT-based systems. Li et al. [17] evaluated recurrent neural network (RNN), LSTM and GRU models for predicting dissolved oxygen in a fishery pond and found the GRU superior in terms of MAE, mean squared error (MSE), mean absolute percentage error (MAPE) and R^2, at a lower computational cost than the LSTM. Putpuek et al. [18] compared LSTM, GRU, bidirectional LSTM (BiLSTM) and bidirectional GRU (BiGRU) for dissolved oxygen prediction in aquaculture and likewise reported the best performance for the GRU, particularly when pH and temperature were included as inputs. Rahul Gandh et al. [19] applied LSTM and GRU models to water quality prediction for smart aquaculture and emphasized their accuracy and computational efficiency, as well as the strong influence of hyperparameter selection. In a hydrological setting, Gao et al. [20] showed that both LSTM and GRU can filter redundant information in short-term runoff prediction, the GRU being more efficient to train owing to its simpler structure. Cardenas-Cartagena et al. [21] compared recurrent variants for pH forecasting in aquaponics and stressed that the choice of architecture should be guided by the characteristics of the data and the prediction objective.

Hybrid architectures combine the feature-extraction capacity of a convolutional neural network (CNN) with the sequence modeling of recurrent layers. Rasheed Abdul Haq and Harigovindan [22] proposed CNN-LSTM models and models that combine a CNN with a

GRU (CNN-GRU) for water quality prediction in smart aquaculture and reported superior accuracy and computation time relative to single-architecture baselines. Jongjaraunsuk et al. [23] likewise found a CNN-LSTM to be the most effective of the CNN, LSTM and CNN-LSTM models compared for predicting dissolved oxygen, total ammonia nitrogen, nitrite and alkalinity in an outdoor recirculating tilapia system; Eze et al. [24] combined empirical mode decomposition with LSTM for multivariate water quality forecasting at a salmon farm; and Sundararajan et al. [5] coupled a convolutional autoencoder with GRU networks to predict water quality in an aquaponic fish pond. Attention-based architectures have also entered the field: the TFT, introduced by Lim et al. [25] for interpretable multi-horizon forecasting, was applied by Metin et al. [26] to nitrate forecasting in aquaponics, where it showed marked improvements in MAE, MSE and explained variance over the benchmark models and was proposed as a means of reducing human intervention through greater automation. Lan et al. [27] applied a transformer-based model, trained on Monte Carlo simulated growth trajectories, to forecast fish weight in open-sea cages with weight errors of about 2–3%.

Table 1 summarizes the studies most relevant to the present work. Most previous research has targeted water quality variables rather than fish growth, and the studies that addressed fish weight in aquaponics [8–10] used models that map a single set of measurements to a weight estimate. The present study adds a comparison of recurrent, hybrid and attention-based deep learning models with baselines that use no water quality data, under a validation that scores only real weighings and holds out whole ponds, and it asks whether the sensor data add information about fish weight beyond the time since stocking and the past weights.

Table 1. Related studies on machine learning in aquaculture and aquaponics.

Study	Data Source	Model(s)	Evaluation Metrics	Remarks
Metin et al. [26]	Aquaponics	TFT	MAE, MSE, explained variance	Long-horizon forecasting challenged by model complexity
Li et al. [17]	Fishery pond	RNN, LSTM, GRU	MAE, MSE, MAPE, R^2	Not aquaponics-specific; dissolved oxygen only
Putpuek et al. [18]	Aquaculture	LSTM, GRU, BiLSTM, BiGRU	MAE, MSE	Not aquaponics-specific; broader aquaculture application
Rahul Gandh et al. [19]	Aquaculture	LSTM, GRU	MAE, MSE	Hyperparameter optimization not detailed
Rasheed Abdul Haq and Harigovindan [22]	Aquaculture	CNN-LSTM, CNN-GRU	MAE, MSE	Relevance to aquaponic systems not examined
Gao et al. [20]	Short-term runoff	LSTM, GRU	MAE, MSE	Hydrological application; not aquaponics
Jongjaraunsuk et al. [23]	Tilapia RAS	CNN, LSTM, CNN-LSTM	R^2	Water quality only; fish growth not addressed
Lan et al. [27]	Open-sea cages	Transformer	Percentage error	Trained on simulated growth trajectories
Kumar et al. [9]	Aquaponics	KNN, decision tree, linear models	MAE, MSE, RMSE, RMSLE	Temporal dependencies not modeled

Table 1. *Cont.*

Study	Data Source	Model(s)	Evaluation Metrics	Remarks
Eneh et al. [8]	Aquaponics	Decision tree regressor (fish length); XGBoost regressor (fish weight)	RMSE, MAE, R^2	Temporal dependencies not modeled
This study	Aquaponics	LSTM, GRU, CNN-LSTM, TFT, with baselines	S, MAE, RMSE, MAPE, R^2	Fish weight estimation and prediction; leave-one-pond-out validation; multi-model comparison

Abbreviations: BiGRU, bidirectional gated recurrent unit; BiLSTM, bidirectional long short-term memory; CNN, convolutional neural network; CNN-GRU, convolutional neural network combined with a gated recurrent unit; CNN-LSTM, convolutional neural network combined with long short-term memory; GRU, gated recurrent unit; KNN, k-nearest neighbors; LSTM, long short-term memory; MAE, mean absolute error; MAPE, mean absolute percentage error; MSE, mean squared error; R^2, coefficient of determination; RAS, recirculating aquaculture system; RMSE, root mean squared error; RMSLE, root mean squared logarithmic error; RNN, recurrent neural network; S, mean over ponds of the mean absolute error of the natural logarithm of weight, defined in the Materials and Methods; TFT, temporal fusion transformer; XGBoost, extreme gradient boosting.

Table 2. Cohort ponds and eligible samples.

Pond	Sensor Rows After QC	Logging Period	Longest Logging Gap (Days)	Weighings	Eligible T1 Events	Eligible T2 Samples (h = 1/h = 2)	Sensor-Days Observed/In Proceedings of the Interpolated (%)
1	83,126	19 June 2021 to 13 October 2021	10.2	6	3	3/3	46.4/7.1
2	172,249	19 June 2021 to 31 October 2021	9.6	9	7	6/5	56.9/8.3
3	169,185	19 June 2021 to 12 October 2021	17.2	6	5	4/3	49.4/9.6
4	89,753	19 June 2021 to 5 September 2021	4.5	14	5	4/3	65.4/14.1
6	91,015	19 June 2021 to 13 October 2021	9.6	15	4	3/3	53.8/7.4
7	279,612	18 June 2021 to 30 December 2021	33.4	9	6	4/4	42.3/7.5
8	70,744	18 June 2021 to 6 December 2021	34.0	3	0	0/0	13.4/0.9
9	151,785	18 June 2021 to 13 October 2021	9.7	8	5	5/3	46.8/9.6

QC, quality control.

3. Materials and Methods

This study asked whether water quality data logged by IoT sensors improve the estimation of African catfish weight beyond the time since stocking, and the prediction of the next measured weights beyond the past measured weights, when models are evaluated only on real weighings and on ponds they have not seen. Four deep learning architectures (LSTM, GRU, CNN-LSTM and TFT) were compared with baselines that do not use water quality and with tabular models that do. The design was fixed in an analysis plan that was frozen before any model was evaluated on test data (Section 3.6). The overall workflow is shown in Figure 2: an audit of the 11 raw pond files [11] identified the real weighings on

the 14-day grid and the sensor stream shared by ponds 1 and 2; quality-control rules and daily aggregates prepared the inputs of the two tasks (Sections 3.1–3.3); the models were compared under leave-one-pond-out validation with nested tuning and under a descriptive calendar hold-out (Sections 3.4 and 3.5); and the comparisons were analyzed with the statistics of Section 3.6.

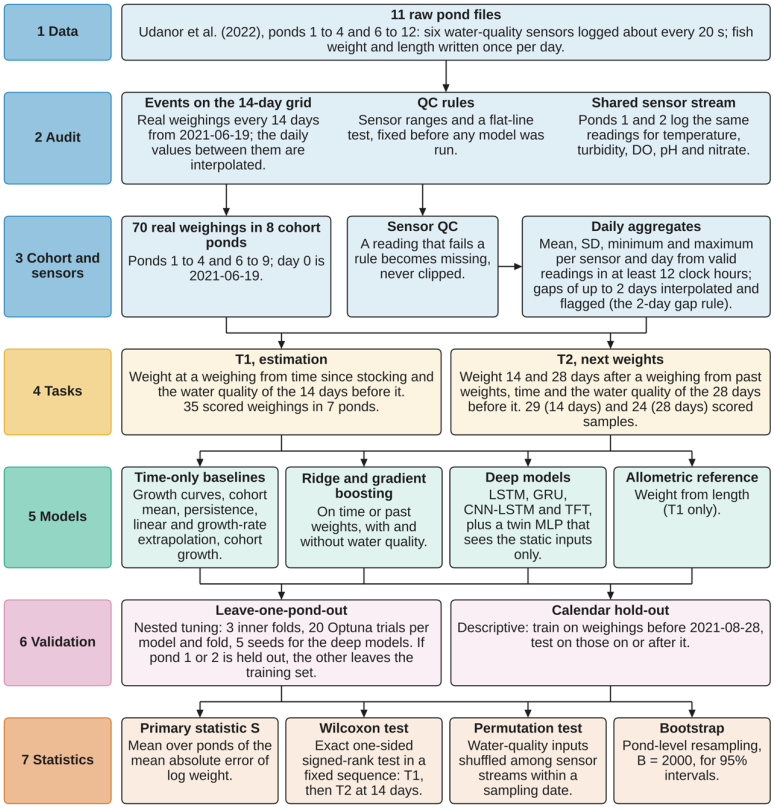

Figure 2. The workflow of the study, from the raw pond files of Udanor et al. [11] to the statistical analysis.

3.1. Dataset

The study used the publicly available IoT-labeled aquaponic dataset published by Udanor et al. [11] and hosted on Kaggle (https://doi.org/10.34740/KAGGLE/DSV/3748790), which is distributed under the Creative Commons Attribution 4.0 (CC BY 4.0) license. The data came from 12 indoor tarpaulin ponds of 2 m × 1 m × 1 m, each stocked with African catfish (*Clarias gariepinus*) fingerlings that were five weeks old and had a mean initial weight of 2 ± 1.1 g [11]. Each pond carried six water quality sensors (temperature, turbidity, dissolved oxygen, pH, ammonia and nitrate) that were read by a microcontroller and uploaded to a cloud platform, and the data paper states a logging interval of 5 s [11].

The public archive contains 11 pond files, for ponds 1 to 4 and 6 to 12; it has no file for pond 5. Each file holds a time stamp (created_at), the six sensor columns, the fish length (cm) and the fish weight (g) as labeled in the files, and the files of ponds 1 to 4, 6 and 10 to 12 also hold a population column, which is 50 in ponds 1 to 3 and 10 to 12 and 75 in ponds 4 and 6. Fish weight was the target of both tasks, and neither fish length nor population was used as a model input; length entered only a reference model and one ablation (Section 3.4).

The sensor log was not uniform. The median interval between consecutive time-stamped rows was 20 s in every pond, not the 5 s stated in the data paper, and logging gaps longer than 10 min summed to between 914 and 3488 h in the cohort ponds. The longest single gap lasted between 4.5 and 34.0 days (Table 2).

Fish length and weight were measured by hand on a fortnightly basis: 10 to 15 randomly sampled fish per pond were weighed on an electronic scale and measured with a measuring tape [11]. The data paper does not state how the sampled fish were combined into the single weight and length given for a pond, and the value was treated as the pond mean. In the files, these values change only once per calendar day and follow straight lines between the sampling dates (Section 3.2), so that thousands of consecutive sensor rows carry the same label, and only the values on the sampling dates are measurements.

Eight ponds (1, 2, 3, 4, 6, 7, 8 and 9) formed the analysis cohort. In each of them the first record with a weight was a fingerling weight of at most 10 g dated 18 or 19 June 2021, consistent with the fingerlings described in the data paper. Ponds 10, 11 and 12 were excluded from model evaluation: their first records are juvenile weights of 27.6, 33.8 and 28.6 g dated 25 June 2021; they have only 310, 627 and 26 time-stamped sensor rows; and the changes in their daily weight values fall on a different 14-day grid (10 October 2021 plus multiples of 14 days), so that no time since stocking comparable with that of the cohort exists for them. Ponds 1 and 2 share one sensor stream: their files have 81,215 common time stamps (98% of the rows of pond 1), at which temperature, turbidity, dissolved oxygen, pH and nitrate are identical in every matched pair, and ammonia is identical in 72% of them. No other pair of ponds has a row that is identical in all six sensors, including when the clock of one pond is shifted by −2 to +2 h. The two ponds were therefore treated as one sensor stream in the validation scheme (Section 3.5).

Table 2 summarizes the cohort ponds. Sensor rows are the time-stamped rows after exact duplicate rows were dropped (44 in pond 4 and none in the other cohort ponds); invalid individual sensor values became missing, and rows were not filtered for completeness. The logging period runs from the date of the first to the date of the last sensor time stamp of each file, in the clock of the file, and fish-only rows after the end of logging do not extend it; the longest logging gap is the longest interval between consecutive time-stamped rows. The weighings include the weighing at day 0 and fish-only weighings, and the T2 samples are given as the counts at 14 and at 28 days ahead. The shares of sensor-days are percentages of the days from the first to the last sensor date of a pond times six sensors: a sensor-day was observed when valid readings existed in at least 12 distinct clock hours and interpolated under the gap rule of Section 3.2, and the remainder were missing. These shares describe the marginal daily coverage, whereas the eligibility of an event also applies the event-specific rule for interpolated days (Section 3.2). Pond 8 had no eligible sample in either task, so that seven ponds were scored (Section 3.3).

The units are as labeled in the source files. Dissolved oxygen, ammonia and nitrate are labeled "g/mL" in the files of ponds 1 to 4, 6, 7 and 9 and "mg/L" in the file of pond 8, and carry no unit label in the files of ponds 10 to 12, whereas the data paper lists mg/l for all three [11]. Temperature is in °C and turbidity in nephelometric turbidity units (NTUs). The values are raw sensor readings, and no unit conversion was applied.

3.2. Weighing Events, Sensor Quality Control and Daily Aggregates

In the files, fish weight and length change once per calendar day, and between sampling dates the daily values follow straight lines. The changes in slope of the daily weight values fall on the dates of a 14-day grid, 19 June 2021 plus 14 k days ($k = 0$ to 16), in the eight cohort ponds. All 70 weights and lengths on the grid dates have at most two decimals, whereas 219 of the 264 daily weights between grid dates in ponds 7 to 9 (83%) have more than two decimals, as expected for the remainders of a division. Only the values on grid dates were therefore treated as real measurements, and the other daily values were never used as targets or inputs.

A weighing event (pond, date) exists when the file has at least one row dated on a grid date with a weight value, and its weight W and length L are the most frequent values among the rows of that date. Grid dates without any row are missing events and were not reconstructed from neighboring interpolation segments. Eight weighings each in ponds 4 and 6 lie after the end of the sensor log of the pond and come from date-only rows that carry fish values but no sensor values; they count as events. Pond 8 has no row on 19 June 2021, and its first record, dated 18 June 2021, is off the grid and is not an event. The cohort has 70 weighing events (6, 9, 6, 14, 15, 9, 3 and 8 in ponds 1, 2, 3, 4, 6, 7, 8 and 9; Table 2), 63 of them after day 0, with weights from 2.91 g to 1011.2 g. Time was expressed as t, the number of days since 19 June 2021, the first sampling date of the cohort, for every pond. Dates were taken from each file's own clock as labeled, which is Central European Time (CET), +01:00 or Coordinated Universal Time (UTC), without time-zone conversion, because the daily weight labels change at midnight on that clock.

Because the daily values are interpolated and the sensor log has gaps, a large difference between the consecutive rows of a file can arise without any measurement. In pond 2, for example, the weight value rose by 29.05 g, from 328.75 g to 357.80 g, between consecutive rows time-stamped 14 October 2021 06:11 and 21 October 2021 18:11. The rows are 180.00 h (7.5 days) apart with no row between them, and the stored daily weight advanced by 4.15 g per day over the 7 calendar days between those dates; the following row, 47.97 h later, added a further 8.30 g. The step is therefore the first row after a logging gap and not a weighing.

These properties of the label determine how an evaluation in which each sensor row is a sample behaves. In a window-level analysis of the file of pond 2 (172,249 rows; Supplementary Section S1), persistence, which predicts each target with the weight of the previous row, had a mean absolute error (MAE) of 0.0026 g on the 34,298 test windows of a time-ordered split, because only 17 of these targets differ from the weight of the previous row. A CNN-LSTM with water quality and the lagged weight as inputs reached an MAE below 1 g only when the windows were assigned to the training and test sets at random, so that windows of the same day were in both sets (0.68 ± 0.022 g over three seeds), and had an MAE of 42.5 ± 4.8 g when the rows were split in time order (Table S19 and Figure S4). Errors of this size measure agreement with the interpolated label and not the prediction of growth, and the present design therefore scores only real weighings and holds out whole ponds.

Sensor values were checked on the time-stamped raw rows, in time order, before any aggregation or scaling. A value that failed a rule became missing, and values were never clipped. The rules followed the measuring ranges of the sensors where the readings were on that scale and physical limits otherwise, and they were fixed from the distributions of the raw values before any model was run.

- Temperature (DS18B20 sensor): 10 to 40 °C, which removes the −127 code of a disconnected sensor;
- pH (DFRobot analog probe): 0 to 14;
- Dissolved oxygen (DFRobot galvanic probe, range 0 to 20 mg/L): above 0 and at most 20;
- Turbidity (TS-300B module, source values on a 0 to 100 scale): 0 to 100;
- Ammonia (MQ137 gas sensor above the water): above 0 and at most 500, the top of the detection range of the sensor, with valid values transformed to base-10 logarithms before daily aggregation;
- Nitrate (MQ135 gas sensor above the water): above 0;
- Every sensor: nonfinite values were invalid, and a run of identical consecutive values spanning at least 24 h of clock time was invalid as a whole, where a missing value or a logging gap of more than 1 h ended a run.

Exact duplicate lines (44 in pond 4) were dropped before these rules were applied. A flat run that crosses an event date is judged on the whole run, so later readings can remove earlier ones; this can only blank values and was accepted. Of the 6,644,814 sensor values of the cohort ponds, 94,692 were missing in the file, 103,590 were nonfinite, 601,639 lay outside the ranges and 326,800 belonged to flat runs, with each value counted under the first rule that it failed, which left 5,518,093 valid values (83.0%). The counts per pond, sensor and rule are given in Table S1, provided as a data file.

For each pond, sensor and calendar date, the daily mean, the standard deviation (population form, ddof 0), the minimum and the maximum were computed when valid readings existed in at least 12 distinct clock hours of that date; otherwise the sensor-day was missing. Missing sensor-days were linearly interpolated only inside runs of at most 2 consecutive missing days with observed days on both sides, and interpolated sensor-days were flagged and counted per pond. Interpolated sensor-days made up between 0.9% and 14.1% of the sensor-days from the first to the last sensor date of a pond, and observed sensor-days between 13.4% and 65.4% (Table 2). An input for a weighing on date d used only days before d: an interpolated sensor-day whose later observed anchor fell on d or after counted as missing for that weighing. A day was covered for a weighing when at least 3 of the 6 sensors had a value on it under this rule. Figure 3 shows the weighings of the eight ponds (panels a to h: ponds 1, 2, 3, 4, 6, 7, 8 and 9) on a logarithmic weight axis. The filled circles are the 35 weighings that enter T1, the open circles are real weighings that do not meet the coverage rule, and the diamonds mark the weighing at stocking ($t = 0$), which pond 8 does not have; the thin gray line is the interpolated daily weight of the source files, broken on days without a row, and the dashed vertical line is the cut-off of the calendar hold-out (Section 3.5). Figure 4 shows, for each day, how many of the six sensors had an observed (blue) or an interpolated (gray) daily value after quality control, with the weighings of the pond marked below each strip in the same way. The order within a stack does not identify a sensor, and the strips show the saved daily status, whereas eligibility also applies the event-specific rule for interpolated days. Ponds 1 and 2 share the stream for temperature, turbidity, dissolved oxygen, pH and nitrate and differ in usable days and in ammonia.

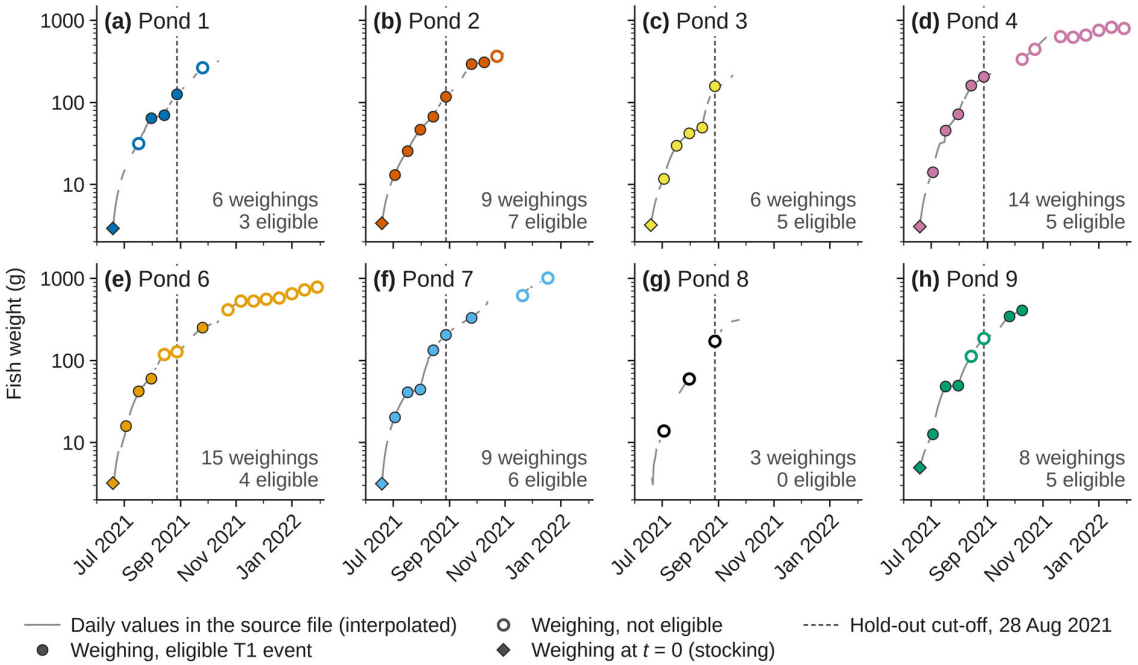

Figure 3. Fish weights at the real weighings of the eight cohort ponds.

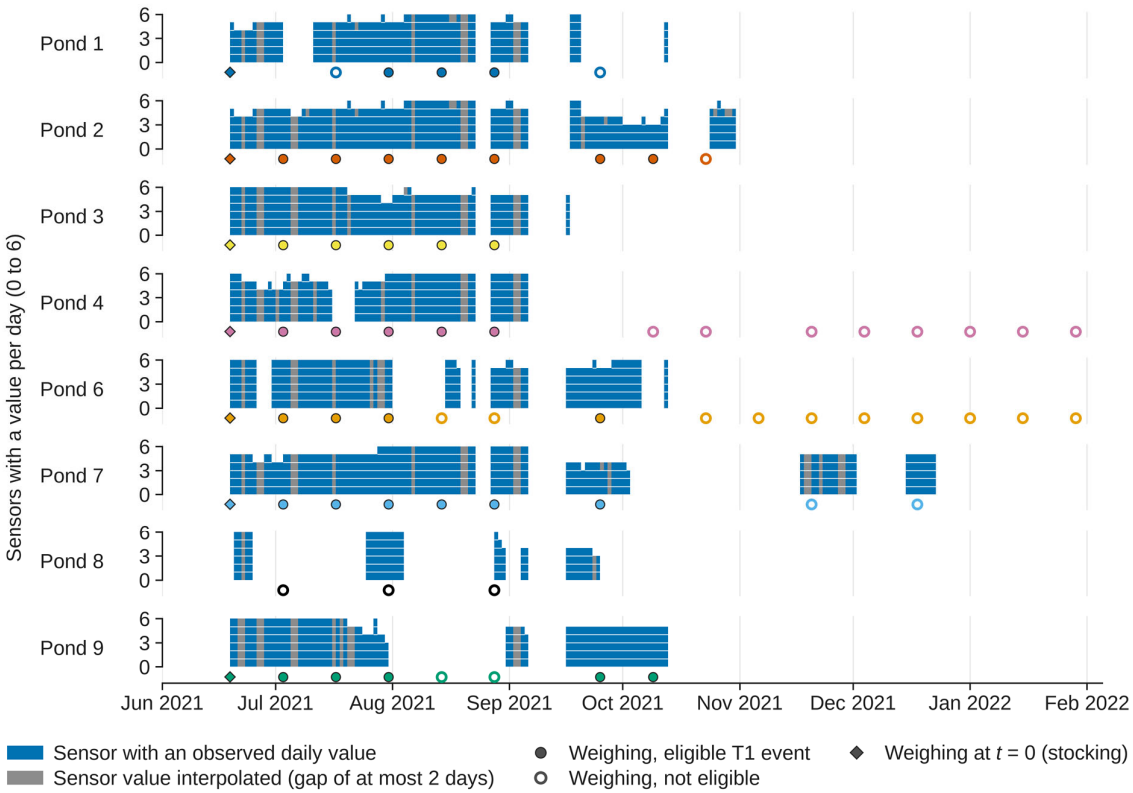

Figure 4. Daily sensor availability after quality control in the eight cohort ponds, 1 June 2021 to 5 February 2022.

3.3. Prediction Tasks

Two tasks were defined on the weighing events. Task T1, weight estimation, estimated the weight W at a weighing from the time since stocking t and the water quality of the 14 days before the weighing date d_e. Its inputs excluded fish length and every weight value, and its target was log W. Task T2, weight prediction from past weights, started from an origin weighing k that had a previous available weighing $k-1$, whatever the interval between them, and predicted the weight at the weighing dated exactly 14 days ($h = 1$) or 28 days ($h = 2$) after the origin date d_k. In the following, the two horizons of task T2 are denoted T2 (14 days ahead) and T2 (28 days ahead). The inputs were log W_k, log W_{k-1}, the time t_k, the interval $t_k - t_{k-1}$, the target gap (14 or 28 days) and the water quality of the 28 days before d_k. The target was the log growth ratio, Equation (1), and the predicted weight was $W_k \exp(\hat{y})$.

$$y = \log\left(\frac{W_{k+h}}{W_k}\right), \quad \hat{W}_{k+h} = W_k \exp(\hat{y}) \tag{1}$$

where W_{k+h} is the weight at the weighing dated 14 h days after the origin date and $\hat{y}$ is the model output. One sample was formed per origin and horizon, and one model served both horizons: the weight 14 or 28 days ahead was predicted directly from the inputs of the origin, with no recursive forecasting, and every prediction was scored against the measured weight of a real weighing. Water quality on the weighing or origin day itself was not used, because the time of day of the weighing is unknown.

The time scale of both tasks is the interval between weighings. Fish weight was measured every 14 days, and the daily values between weighings carry no further information about weight, so growth can be tested only at the scale of the weighings. The water quality windows of 14 days in T1 and 28 days in T2 cover one and two weighing intervals. The sensor logging interval (median 20 s) is not a biological time scale for weight: at that

interval a window of 36 sensor rows spans about 12 min, over which fish weight does not change measurably.

T1 used the weighings with $t > 0$ whose 14 preceding days contained at least 7 covered days. Of the 63 weighings with $t > 0$, 35 were eligible (3, 7, 5, 5, 4, 6 and 5 in ponds 1, 2, 3, 4, 6, 7 and 9), and pond 8 had none; they span $t = 14$ to 112 days. T2 samples needed at least 14 covered days among the 28 days before the origin date. For T2 (14 days ahead), 29 of 43 samples were eligible (3, 6, 4, 4, 3, 4 and 5 in ponds 1, 2, 3, 4, 6, 7 and 9), and for T2 (28 days ahead), 24 of 40 (3, 5, 3, 3, 3, 4 and 3). Their target dates span $t = 28$ to 126 days. Every model of a task was trained on the eligible samples of the training ponds, so that models with and without water quality saw the same training weighings, and all models of a task were scored on exactly the same samples. Ponds without an eligible sample contributed neither training nor test samples, and the baselines without water quality were additionally evaluated on all 63 weighings with $t > 0$ as a secondary descriptive table.

3.4. Models

Five groups of models were compared. Baselines without water quality used only time or past weights, and they included growth curves and a model with a constant specific growth rate (SGR), the daily rate of change in log weight. Tabular models with water quality, namely ridge regression and a gradient boosting machine (GBM), used features computed from the water quality (WQ) sensors, with or without time or past weights. Deep models (LSTM, GRU, CNN-LSTM and TFT) read the daily sequence of WQ aggregates together with static inputs. A static-input network, a multilayer perceptron (MLP) that sees only the static inputs and is called Static MLP, served as the neural counterpart that has no WQ input. A reference model used fish length. Table 3 lists every model, its group and its inputs, and the model names of this paper are those of the table.

Table 3. Models and inputs.

Model	Group	T1 Inputs	T2 Inputs
Growth curve	Baselines without water quality	t	not applicable
Cohort mean	Baselines without water quality	sampling date, t	not applicable
Ridge (time)	Baselines without water quality	t, t^2	not applicable
GBM (time)	Baselines without water quality	t, t^2	not applicable
Persistence	Baselines without water quality	not applicable	W_k
Linear extrapolation	Baselines without water quality	not applicable	W_k, W_{k-1}, t_k, t_{k-1}
Constant SGR	Baselines without water quality	not applicable	W_k, W_{k-1}, t_k, t_{k-1}
Anchored growth curve	Baselines without water quality	not applicable	W_k, t_k, target date
Cohort growth	Baselines without water quality	not applicable	W_k, origin date, target date
Ridge (past weights)	Baselines without water quality	not applicable	past weights, time
GBM (past weights)	Baselines without water quality	not applicable	past weights, time
Ridge (WQ)	Tabular models with water quality	WQ features	not applicable

Table 3. *Cont.*

Model	Group	T1 Inputs	T2 Inputs
GBM (WQ)	Tabular models with water quality	WQ features	not applicable
Ridge (time + WQ)	Tabular models with water quality	time, WQ features	not applicable
GBM (time + WQ)	Tabular models with water quality	time, WQ features	not applicable
Ridge (time + WQ + length)	Tabular models with water quality	time, WQ features, log L	not applicable
GBM (time + WQ + length)	Tabular models with water quality	time, WQ features, log L	not applicable
Ridge (past weights + WQ)	Tabular models with water quality	not applicable	past weights, time, WQ features
GBM (past weights + WQ)	Tabular models with water quality	not applicable	past weights, time, WQ features
LSTM, GRU, CNN-LSTM, TFT	Deep models	daily WQ sequence; static input t	daily WQ sequence; static inputs past weights, time, target gap
Static MLP	Static-input network	static input t	static inputs past weights, time, target gap
Allometric (length)	Reference model using length	log L	not applicable

Abbreviations: GBM, gradient boosting machine; MLP, multilayer perceptron; SGR, specific growth rate; WQ, water quality. t, time since stocking in days; W_k, weight at the origin weighing; L, length at the weighing; time in T2 means t_k, $t_k - t_{k-1}$ and the target gap. The WQ features are defined in the text.

3.4.1. Inputs and Feature Sets

The time features were t and t^2 in T1, and t_k, $t_k - t_{k-1}$ and the target gap in T2. In T2, past weights were entered as log W_k and log W_{k-1}, and length, in one ablation of T1 only, as log L, the logarithm of the length at the weighing. WQ features were computed for each sensor over the days of the window in which that sensor had a value under the gap rule (the 14 days before the weighing date in T1, the 28 days before the origin date in T2): the mean of the daily means, the mean of the daily standard deviations, the minimum of the daily minima and the maximum of the daily maxima, which gave 24 features for the six sensors. T1 added each sensor's mean of the daily means from day 0 to the day before the weighing, which gave 6 further features. A sensor with a value in fewer than 3 days of its window, or of the period from day 0, gave missing features, and coverage (Section 3.2) decided eligibility only.

Ridge models imputed the median of the training fold and added one indicator per sensor, set to 1 when the sensor had no window features, whereas GBM used its native handling of missing values. A feature that was missing in every training sample of a fold was dropped for that fold.

The deep models read daily steps with 30 channels: the 24 daily aggregates, standardized with training-fold statistics and set to 0 where missing, and six validity masks, one per sensor. The T1 sequence ran from day 0 to the day before the weighing and was left-padded with zeros (masks 0) to the longest T1 sequence, and the T2 sequence covered the 28 days before the origin date. The static inputs were t in T1, and log W_k, log W_{k-1}, t_k, $t_k - t_{k-1}$ and the target gap in T2, standardized with training-fold statistics.

The variables were standardized to a common scale using the z-score transformation in Equation (2):

$$n = \frac{X - \mu}{\sigma} \quad (2)$$

where n is the standardized value, X is the observed value, and μ and σ are the mean and standard deviation of the corresponding variable, computed on the training samples only: on the training ponds of the outer fold and again on the inner-training ponds of every inner fold, which also applied to every imputer, median and target standardization. The statistics of a sequence channel were computed over the sensor-days that the training samples could see, each sensor-day counted once, and a channel without a value or without spread received a mean of 0 and a standard deviation of 1. Standardization places all inputs on comparable scales, which improves the stability of gradient-based training; it does not remove outliers, which were handled by the quality-control rules before any scaling.

3.4.2. Baselines Without Water Quality

The growth curve baseline of T1 fitted a growth curve to log W of the training weighings as a function of t by nonlinear least squares in SciPy, using the Gompertz, von Bertalanffy and logistic curves, per Equations (3)–(5):

$$\log W(t) = \log A - b\exp(-ct) \quad (3)$$

$$\log W(t) = \log W_\infty + 3\log(1 - \exp(-k(t - t_0))) \quad (4)$$

$$\log W(t) = \log A - \log(1 + \exp(-k(t - t_m))) \quad (5)$$

where A and W_∞ are asymptotic weights; b, c, k, t_0 and t_m are curve parameters; and $t_0 < 0$. Each curve has three parameters. Three starting values per curve were fixed in the configuration file and tried in order, and the fit with the lowest residual sum of squares was kept. The curve with the lowest Akaike information criterion (AIC) on the training weighings was used [28,29]. A curve that no starting value could fit was skipped and counted, and when all three curves failed, the growth curve used the prediction of the Cohort mean.

The Cohort mean predicted the exponential of the mean log W of the training ponds on the same sampling date when at least 3 training ponds had that date, and the prediction of the growth curve otherwise. Ridge (time) and GBM (time) were ridge regression and GBM on the time features.

In T2, Persistence predicted the weight of the origin, W_k. Linear extrapolation drew the straight line through (t_{k-1}, W_{k-1}) and (t_k, W_k) and read it at the target date; it was scored as it was, including when it predicted a weight of 0 g or less. Constant SGR drew the straight line through log W_{k-1} and log W_k, which corresponds to a constant SGR. Anchored growth curve predicted $W_k\ G(t_{k+h})/G(t_k)$, where G is a growth curve fitted as in T1 to the origin and target weighings of the training samples. Cohort growth predicted $W_k \exp(\bar{y})$, where $\bar{y}$ is the mean y of the training samples with the same origin date and target date, when at least 3 training ponds had them, and the prediction of the Anchored growth curve otherwise. When all three growth curves failed, the Anchored growth curve used Constant SGR. Ridge (past weights) and GBM (past weights) were ridge regression and GBM on past weights and time.

3.4.3. Tabular Models with Water Quality

The tabular models with water quality were ridge regression and GBM, the latter implemented as the HistGradientBoostingRegressor of scikit-learn [30] with early stopping switched off and random_state set to 0. In T1, Ridge (WQ) and GBM (WQ) used WQ

features only, Ridge (time + WQ) and GBM (time + WQ) added the time features, and Ridge (time + WQ + length) and GBM (time + WQ + length) added the logarithm of the length as an ablation. In T2, Ridge (past weights + WQ) and GBM (past weights + WQ) added WQ features to past weights and time. Ridge standardized its features on the training fold and took its regularization strength from the inner folds, and the hyperparameters of GBM were tuned as described below.

3.4.4. Long Short-Term Memory

LSTM networks [14] mitigate the vanishing gradient problem through a gating mechanism that regulates the flow of information into and out of a memory cell. The forget gate, Equation (6), determines which information from the previous cell state is retained:

$$f(t) = \sigma\left(x(t)U_f + h(t-1)W_f\right) \tag{6}$$

where σ is the sigmoid function, $x(t)$ is the input at time t, $h(t-1)$ is the hidden state of the previous time step, and U_f and W_f are weight matrices. The input gate integrates new information through sigmoid and hyperbolic tangent (tanh) activations, as seen in Equations (7)–(9), where $\odot$ denotes element-wise multiplication:

$$i_1 = \sigma(x(t)U_i + h(t-1)W_i) \tag{7}$$

$$i_2 = tanh\left(x(t)U_g + h(t-1)W_g\right) \tag{8}$$

$$i(t) = i_1(t) \odot i_2(t) \tag{9}$$

The cell state is then updated by combining the outputs of the forget and input gates in Equation (10), and the output gate determines the hidden state for the current time step through Equations (11) and (12):

$$C(t) = f(t) \odot C(t-1) + i(t) \tag{10}$$

$$o(t) = \sigma(x(t)U_o + h(t-1)W_o) \tag{11}$$

$$h(t) = o(t) \odot \tanh C(t) \tag{12}$$

Components of the previous cell state multiplied by forget-gate values close to zero are suppressed, the input-gate output augments the cell state, and a sigmoid layer selects which components of the tanh-transformed cell state contribute to the output. This mechanism enables LSTM networks to model long-term dependencies in complex sequential data. Bias terms are omitted from the equations.

In the implementation, the 30-channel daily sequence entered the LSTM of PyTorch, which had 1 or 2 layers and a hidden size of 8, 16, 32 or 64. The hidden state of the last real day was concatenated with the static inputs and passed through a head made of a linear layer with as many units as the hidden size, a rectified linear unit (ReLU) activation, dropout and a linear output layer, so that dropout acted in the head only. The search ranges of all hyperparameters are given in Table S16.

3.4.5. Gated Recurrent Unit

The GRU [15] is a simplified variant of the LSTM that addresses the vanishing gradient problem with fewer parameters. A GRU cell uses an update gate z_t (Equation (13)), which merges the roles of the forget and input gates, and a reset gate r_t (Equation (14)), which determines the contribution of the previous hidden state to the candidate activation:

$$z_t = \sigma(w_z \cdot [h_{t-1}, x_t]) \tag{13}$$

$$r_t = \sigma(w_r \cdot [h_{t-1}, x_t]) \tag{14}$$

where h_{t-1} is the previous hidden state, x_t is the current input and w_z and w_r are weight matrices. The candidate activation and the new hidden state h_t are computed by Equations (15) and (16), the tanh function bounding the candidate activation to the interval $(-1, 1)$:

$$\tilde{h}_t = \tanh\left(W_{\tilde{h}} x_t + r_t \odot \left(U_{\tilde{h}} h_{t-1}\right)\right) \tag{15}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{16}$$

where $W_{\tilde{h}}$ and $U_{\tilde{h}}$ are weight matrices, and the reset gate acts on the recurrent term, as in the implementation used here. GRUs thus simplify the LSTM architecture while maintaining its effectiveness in capturing temporal dependencies [31]. The implementation weights the previous hidden state by z_t and the candidate activation by $1 - z_t$, which is Equation (16) with z_t replaced by $1 - z_t$, and it includes bias terms that the equations omit.

The GRU of PyTorch had the same layers, hidden sizes and head as the LSTM, and the search ranges are those of Table S16.

3.4.6. Hybrid CNN-LSTM

The CNN-LSTM combined convolutional and LSTM layers. The input sequence x_1, x_2, …, x_n of daily steps was first processed by a one-dimensional convolutional layer with 8, 16 or 32 filters of width 3, 5 or 7 as tuned hyperparameters (same padding) and the ReLU activation (Equation (17)), which extracted local temporal patterns from the sequence:

$$f(x) = max(0, x) \tag{17}$$

Max pooling of width 1 (no pooling) or 2 then reduced the length of the feature maps, retaining the most salient information. The pooled features were passed to an LSTM of 1 or 2 layers and a hidden size of 8, 16, 32 or 64, as in the LSTM model, and the LSTM output at the last step was concatenated with the static inputs and passed through the same head as in the LSTM and GRU models. The architecture thereby combined the feature-extraction capability of CNNs with the sequential memory of LSTMs.

3.4.7. Temporal Fusion Transformer

The TFT [25] integrates LSTM-based sequence encoding, gating, variable selection and transformer-style attention (Figure 5). It was implemented as an encoder-only model with a point output, which differs from the original design in three ways: it had no decoder and no known future inputs, because there was one prediction per sequence; it used a single attention query, the last real day, instead of one query per time step; also, it was trained with the mean squared error instead of the quantile loss. Its building block is the gated residual network (GRN), $\mathrm{GRN}(a, c) = \mathrm{LayerNorm}(a + \mathrm{GLU}(W_1\ \mathrm{ELU}(W_2\ a + W_3\ c)))$, where a is the primary input; c is an optional context vector; W_1, W_2 and W_3 are weight matrices (bias terms omitted); ELU is the exponential linear unit; GLU is the gated linear unit; and LayerNorm is layer normalization.

The static inputs (t in T1; $\log W_k$, $\log W_{k-1}$, t_k, $t_k - t_{k-1}$ and the target gap in T2) were each embedded by a linear layer of its own and combined by a variable selection network (VSN), which computed softmax weights over the embeddings with a GRN and formed their weighted sum after a GRN of each variable. Four static covariate encoders, each a GRN, turned the result into the contexts c_s for temporal variable selection, c_e for static enrichment, and c_h and c_c for the initial hidden and cell state of the LSTM encoder. The temporal variables were the six sensors. For each sensor and day, its four standardized daily aggregates and its validity mask were embedded by a linear layer of its own, and a VSN with context c_s weighted the six embeddings at every day. An LSTM encoder of one

layer, started from c_h and c_c, processed the selected inputs, a gated skip connection (a GLU followed by addition and layer normalization) added the selected inputs to its output, and a static enrichment GRN with context c_e followed.

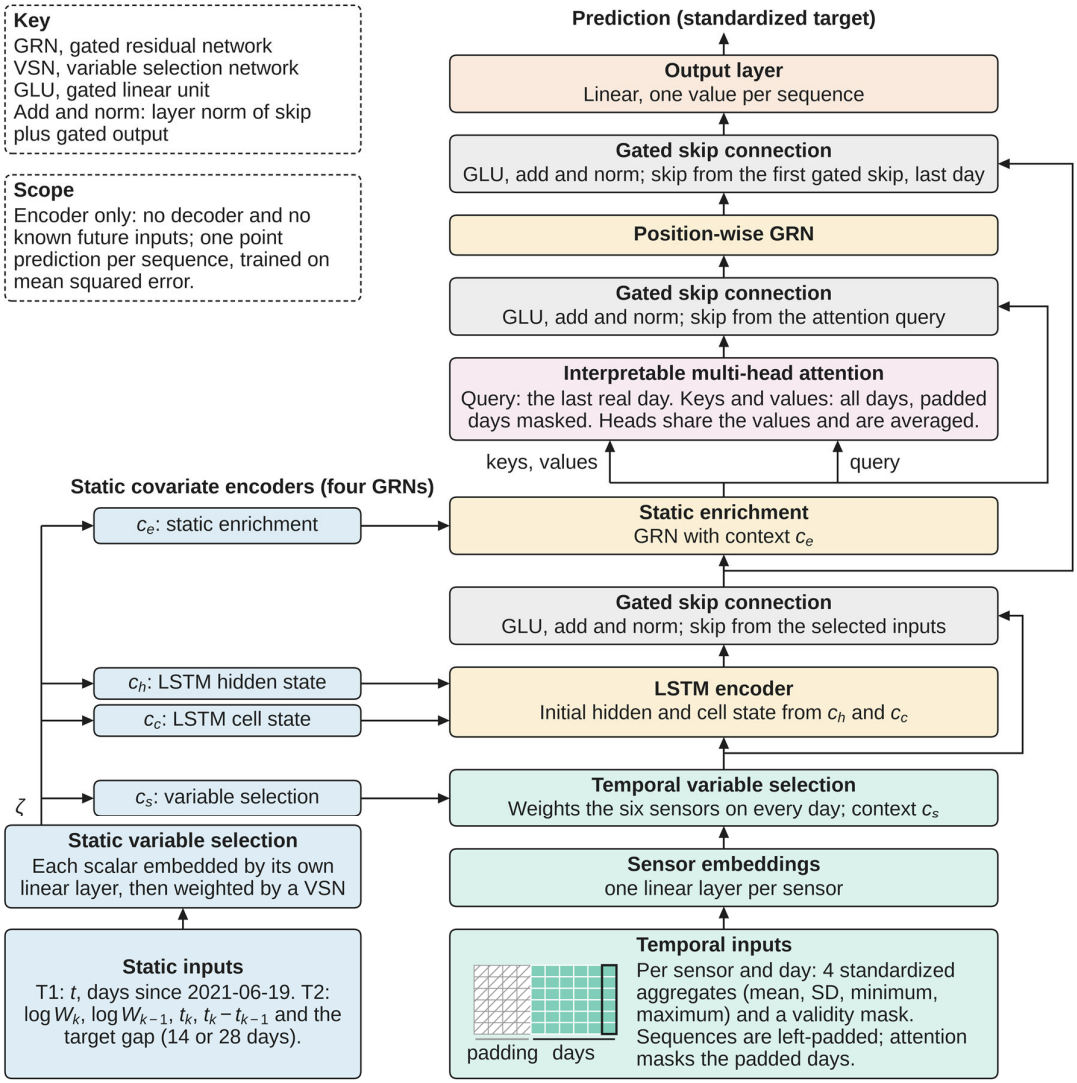

Figure 5. The architecture of the temporal fusion transformer as implemented in this study (own drawing, adapted from Lim et al. [25]).

Interpretable multi-head attention then used the last real day as its only query over all days, with padded days masked, one value projection shared by all heads and the heads averaged. The attended vector passed through a gated skip with the query, a position-wise GRN and a further gated skip to the output of the first gated skip at the last day, and a linear layer returned one value. Dropout acted inside the gated linear units and on the attention weights. Hidden sizes of 8, 16 or 32 and 1, 2 or 4 attention heads were searched (Table S16). In Figure 5, shading groups the blocks into static inputs (blue), temporal inputs and variable selection (green), LSTM and GRN layers (orange), attention (purple), gated skip connections (gray) and the output (vermillion); in the small grid, the rows are sensors, the columns are days, the hatched columns are padding and the outlined column is the last real day.

3.4.8. Static-Input Network

The Static MLP was the twin of the LSTM, GRU and CNN-LSTM: it applied the head that these models placed on their last hidden state (a linear layer, ReLU, dropout and a linear output layer) to the static inputs alone and ignored the daily sequence. In T1 its only input was t, so it was a neural model of time alone. It was tuned, trained and seeded like the deep models, with a hidden size of 8, 16, 32 or 64, so that a difference between a deep model and the Static MLP measures what the WQ sequence adds and not what the training procedure adds.

3.4.9. Reference Model Using Length

The allometric reference fitted the length–weight relationship $W = a\,L^{b}$ to the logarithms of length and weight of the training weighings and predicted the weight from the length at the weighing, in T1 only. Because length is measured on the same occasion as weight, the model was reported as a reference and not as a competitor of the models that do not use length.

3.5. Validation and Hyperparameter Tuning

Models were evaluated by leave-one-pond-out (LOPO) validation: each cohort pond with at least one eligible sample was held out in turn, its eligible samples formed the test set of that fold, and the other ponds were available for training. Whole ponds were held out because observations within a pond are dependent, and random splits of dependent data underestimate prediction error [12]. Seven outer folds were run for each task, because pond 8 had no eligible sample, and eight for the secondary T1 set of all weighings with $t > 0$. When pond 1 or pond 2 was held out, the other was removed from the training ponds of that fold, because the two ponds share one sensor stream.

Hyperparameters were tuned inside the training ponds of each outer fold only. Three inner folds grouped the training ponds by sensor stream: the streams with an eligible training sample were sorted by their smallest pond identifier and assigned to the folds in turn, and ponds 1 and 2 formed one stream. The Optuna framework [32] with the tree-structured Parzen estimator (TPE) sampler [33] and sampler seed 0 ran 20 trials per model and outer fold for each deep model, the Static MLP and GBM, in one study per task, validation scheme, model and outer fold. For Ridge, the regularization strength α was chosen by the same inner folds from 13 values, 10^{-3} to 10^{3} in steps of 0.5 in the exponent, and ties went to the larger α. The objective was the mean over the inner folds of the primary statistic S (Section 3.6) on the inner-validation ponds and, for T2, the mean of its values at the two horizons. Every model, including the baselines, was also scored on the inner folds, which supplied the selection of the primary models (Section 3.6). The search spaces and training settings are listed in Tables S16 and S17.

The deep models and the Static MLP were trained with the AdamW optimizer [34] on the mean squared error of the standardized target for at most 300 epochs. Inside each inner training, early stopping with a patience of 30 epochs on the validation mean squared error ended the training, and the weights of the best epoch were restored. The final model of each outer fold was trained on all training ponds, without a validation set and without early stopping, for the median best epoch of the inner folds of the best trial. Five seeds (0 to 4) were used in each final fit and tuning used seed 0; training was single-threaded, and PyTorch was set to use deterministic algorithms where available. Learning curves, with the inner-validation loss of the chosen trial and the training loss of the final fits per epoch, are shown in Figure S1.

As a descriptive check of prediction at later dates, a calendar hold-out was run with the cut-off date 28 August 2021 ($t = 70$), the latest grid date such that at least 10 eligible T1

events in at least 5 ponds were dated on or after it. In T1, the 24 eligible weighings dated before the cut-off formed the training set, and the 11 eligible weighings dated on or after it, in 7 ponds, formed the test set. In T2, the eligible samples with a target dated before the cut-off formed the training set (19 samples at the shorter and 12 at the longer horizon), and those with a target dated on or after the cut-off and an origin dated on or after the day 14 days before the cut-off formed the test set (10 samples in 7 ponds at the shorter and 5 samples in 4 ponds at the longer horizon). The 7 samples at the longer horizon with a late target and an earlier origin were used for neither, and they were counted. Nothing dated on or after the cut-off entered training, the baselines that use the training ponds on the same date included, which then used their fallback (Section 3.4.2). Tuning used one inner split, whose validation set was the training samples at the latest training date, with the budgets and seeds of the LOPO scheme. In T1 the test dates ($t \geq 70$) lay beyond all training dates ($t \leq 56$), so that tree-based models predicted a constant in t there. No p-values or intervals were computed for the hold-out.

In LOPO validation the training ponds cover the same calendar period as the test pond. T2 under LOPO therefore measures how well growth over the next 14 or 28 days transfers between ponds weighed on the same dates; it does not test forecasting of a later period, which the calendar hold-out addresses.

The ablations were descriptive and used Ridge and GBM: in T1, time only, WQ only, time with WQ, and time with WQ and length; in T2, past weights and time with and without WQ; and the deep models were compared with the Static MLP. The sensitivity analyses were descriptive as well: they varied the WQ window (7 and 28 days in T1, 14 and 42 days in T2) for Ridge and GBM with WQ, omitted the missing-value indicators of Ridge, averaged ponds 1 and 2 into one unit in the confirmatory test, and removed one pond at a time to show the influence of each pond; their results are reported in the Supplementary Materials.

3.6. Metrics and Statistical Analysis

The primary statistic, S, was the mean over ponds of the mean absolute error of the natural logarithm of weight over the scored samples of each pond, Equation (18):

$$S = \frac{1}{P}\sum_{p=1}^{P}\frac{1}{n_p}\sum_{i\in p}\left|\log \hat{W}_i - \log W_i\right| \tag{18}$$

where P is the number of ponds with a scored sample, n_p is the number of scored samples of pond p, and $\hat{W}_i$ and W_i are the predicted and the measured weight of sample i. For T2, the absolute error of the predicted log ratio of Equation (1) equals the absolute error of the predicted log weight at the target date, because both use the same measured weight at the origin. S is the mean absolute log accuracy ratio; it counts an over-prediction and an under-prediction by the same factor equally and avoids the bias of the MAPE towards predictions that are too low [35]. For the deep models and the Static MLP, the per-pond values were averaged over the five seeds. A predicted weight of 0 g or less has an infinite logarithmic error, so that the S of the pond was infinite and the model could not be the primary model; gram-based metrics used the prediction as it was, and the number of such predictions is reported. A nonfinite prediction of a learned model (Ridge, GBM, a deep model or the Static MLP) stopped the run of that model; the stop was logged, and the model was reported as stopped (labeled "stopped" in the tables, following item 9.0 of the analysis plan) and not re-run with other settings. For the Ridge runs that stopped under this rule, one post hoc variant was fixed after the stops and before any test score was computed: in the tuning of α, a candidate whose inner-validation prediction was not finite lost its trial, and everything else followed the frozen procedure. The variant is reported in the Supplementary Materials as post hoc and does not replace a frozen result.

Model performance was additionally quantified, pooled over all test samples of all folds, with three complementary metrics, using Equations (19)–(21): the MAE, the average magnitude of the prediction errors irrespective of sign; the RMSE, the square root of the mean squared error, which assigns greater weight to large errors and is therefore sensitive to outliers; and the MAPE, which expresses the error as a percentage of the observed values and thus provides a scale-independent measure of accuracy:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{19}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{20}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\% \tag{21}$$

where y_i is the measured weight, $\hat{y}_i$ is the predicted weight (not the target of Equation (1)) and n is the number of test samples. The MAE and RMSE are expressed in grams, and a value of 0 for any of the metrics corresponds to perfect prediction. The MAPE is informative when, as here, the target variable is strictly positive and not close to zero. Together, the three metrics characterize different aspects of the error distribution. The coefficient of determination, R^2 (Equation (22)), was also reported, and per pond the MAE, the MAPE and S were reported:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{22}$$

where $\bar{y}$ is the mean of the measured weights. For the deep models and the Static MLP, every metric was computed per seed and is reported as the mean ± standard deviation (SD) over the five seeds, and the spread across ponds is reported as the SD of the per-pond S and MAE. In the tables and the text, S and R^2 are given with three decimals and percentages with one decimal; grams have one decimal below 100 g and none otherwise, and values of 10,000 or more have two significant digits. The differences in S stated in the text were computed from unrounded values and can therefore differ by 0.001 from the differences of the rounded values in the tables.

The primary baseline of a task was the baseline without water quality with the lowest mean inner-validation S over the outer folds of the LOPO scheme, and the primary deep model was chosen in the same way among the LSTM, GRU, CNN-LSTM and TFT models. For T2, the inner S of a model was the mean of its values at the two horizons. A tuned model entered with the mean inner S of its best trial, which favors tuned models, and ties went to the model listed first in Table 3. The choice was made from inner-validation scores only, and its output was committed before any outer prediction was scored. Because the inner folds validate on ponds that are test ponds of other outer folds, the level of the confirmatory test is approximate.

The confirmatory test was an exact one-sided Wilcoxon signed-rank test [36] on the per-pond differences in S between the primary deep model and the primary baseline, with the alternative that the S of the deep model was lower. The units were the ponds with a scored sample; zero differences were dropped and counted, ties received midranks, and the exact distribution over all 2^n sign flips was used, where n is the number of ponds. Differences were rounded to 12 decimals before ties and zeros were identified. The tests followed a fixed sequence: T1 at $\alpha = 0.05$, and T2 (14 days ahead) at $\alpha = 0.05$ only if the test for T1 rejected. Everything else was descriptive: T2 (28 days ahead), the calendar hold-out, the ablations, the sensitivity analyses, all other model pairs and any task and horizon with fewer than 6 scored ponds. Two-sided p-values were reported alongside. With 7 ponds the

test could reject only if the ranks of the ponds in which the deep model was worse summed to at most 3 (5 of 128 sign patterns), and with 6 ponds only if they summed to at most 2.

A permutation test [37] asked whether the WQ inputs carried pond-specific information about weight beyond what the ponds share on a given date. For each permutation r and each date (a T1 weighing date or a T2 origin date) with eligible samples in at least two sensor streams, the WQ inputs of the streams for that date (tabular features and sequences, with their missing-value indicators) were reassigned among the streams by a uniformly random derangement, with the random generator seeded by 20261014 + r; a date with a single stream stayed unchanged and was counted. The reassignment applied to training and test data alike; targets, eligibility and the tuned hyperparameters stayed fixed; and the final fits of the LOPO scheme were re-run without re-tuning. The observed model was re-fitted first and had to reproduce its saved predictions to 10^{-8}. The statistic was S (for T2, S at 14 days ahead), and the p-value was computed as in Equation (23), where R is the number of permutations, S_r the statistic of permutation r, S_{obs} the statistic of the observed data and $I(\cdot)$ the indicator function, which is 1 if its argument is true and 0 otherwise:

$$p = \frac{1 + \sum_{r=1}^{R} I(S_r \leq S_{\text{obs}})}{R + 1} \tag{23}$$

R was 999 for Ridge and GBM with time (or past weights) and WQ, and 199 for the primary deep model of each task, which was fitted with seed 0 in the observed and in the permuted runs.

A pond-level cluster bootstrap [38] with 2000 resamples (seed 20261014) resampled ponds jointly for a model and its comparator and gave 95% percentile intervals of S and of the difference in S to the primary baseline; it was descriptive.

The analysis plan was frozen with the annotated git tag analysis-plan-frozen (commit a8706c3, 14 October 2026) before any model was evaluated on test data. The tag covers the plan, the source code, the configuration files, the processed data tables and the dependency lock file. The run script refused to produce outer test predictions unless the tag existed and these files were identical to it. The plan determined that the first complete outer run would be the reported run; that a model would be re-run only after a crash, with the re-run logged; and that the scorer would refuse to run unless the files of the primary model selection were committed and unmodified. All models, ablations and seeds are reported, including poor results.

The framing of the results followed a rule fixed in the plan. A positive statement for a task required that its confirmatory test rejected, and a statement that water quality contributed to the result additionally required a permutation $p < 0.05$ for the primary deep model and a lower S than the Static MLP, with the per-pond comparison reported. If another baseline without water quality had a lower outer S than the primary deep model, this was to be stated next to the confirmatory result. If neither task rejected, the results were to be presented as a leakage-aware benchmark of the models, and a non-significant result is not evidence of no effect. Causal effects of water quality on growth were not claimed, no claim extended beyond the largest t among the scored samples (112 days in T1 and 126 days in T2), and no claim was made that deep learning is unsuitable for this problem in general, because the data hold 35 scored T1 events.

3.7. Implementation

All analyses were run in Python 3.12.3 on an AMD EPYC 7B12 central processing unit (CPU; AMD, Santa Clara, CA, USA) with 4 cores and no graphics processing unit (GPU), in up to four single-threaded worker processes. The deep models and the Static MLP were implemented in PyTorch 2.14.1 [39], Ridge and GBM with scikit-learn 1.9.1 [30]; the hyperparameter search with Optuna 4.9.0 [32]; and the growth curve fits and the Wilcoxon

test with SciPy 1.18.1 [40]; NumPy 2.5.4 and pandas 3.0.6 were used for data handling. These are the versions in the lock file of the repository. The TFT was implemented directly in PyTorch, as described in Section 3.4.7. The window-level analysis of pond 2 (Supplementary Section S1) used TensorFlow 2.21.0 with Keras 3.15.1. The runtime and the hardware of each training were logged. The code, the configuration files, the processed data tables and the lock file are available at https://github.com/MohamedElminur1/DL_Models_Fishweight (accessed on 4 November 2026).

4. Results

4.1. Primary Models Chosen on the Inner Folds

The primary baseline and the primary deep model of each task were chosen from the mean inner-validation S before any outer prediction was scored (Section 3.6), and Table S3 lists the inner S of every model. In T1, the Cohort mean had the lowest inner S among the four baselines without water quality (0.251), ahead of Ridge (time) (0.259), growth curve (0.266) and GBM (time) (0.272), and the LSTM had the lowest inner S among the four deep models (0.204), ahead of the CNN-LSTM (0.217), the GRU (0.241) and the TFT (0.286). In T2, the inner S of a model was the mean of its values at the two horizons. Ridge (past weights) had the lowest inner S among the seven baselines (0.251), followed by GBM (past weights) (0.252), and the CNN-LSTM had the lowest inner S among the deep models (0.221), ahead of the TFT (0.239), the LSTM (0.243) and the GRU (0.245). The Static MLP, which was not a candidate for either role, had an inner S of 0.219 in T1 and 0.223 in T2, also below that of every baseline.

On the inner folds, three of the four deep models in T1 (the LSTM, the CNN-LSTM and the GRU) and all four deep models in T2 had a lower inner S than every baseline, but on the outer folds no deep model had a lower S than the primary baseline of its task (Section 4.2). A tuned model entered the selection with the inner S of its best trial among 20 trials, which favors tuned models (Section 3.6), and the training of a deep model or of the Static MLP was stopped early on the inner-validation fold on which its inner S was computed (Section 3.5). The outer S of the primary deep model was higher than its inner S: the LSTM had 0.204 on the inner folds and 0.290 on the outer folds in T1, and the CNN-LSTM had 0.221 on the inner folds and 0.317 on the outer folds, averaged over the two horizons, in T2. The selection did not identify the model with the lowest outer S of its group in every case. In T1, the growth curve had a lower outer S (0.261) than the Cohort mean (0.265). In T2, GBM (past weights) had a lower outer S than Ridge (past weights) at both horizons (0.211 against 0.295 at 14 days ahead and 0.217 against 0.243 at 28 days ahead), and the GRU had a lower outer S than the CNN-LSTM at both horizons (0.295 against 0.332 and 0.271 against 0.303). The primary models were not changed after the outer scores were known.

4.2. Leave-One-Pond-Out Comparison of the Models

Table 4 gives the results of the primary baseline, the baseline without water quality with the lowest S, the tabular models with water quality with the lowest S, the four deep models and the Static MLP under leave-one-pond-out validation on the eligible samples, and Table S18 gives the results of every model of Table 3. For each model, Table 4 gives S, the 95% percentile interval of S from the pond-level bootstrap, the MAE and R^2 pooled over all test samples of the task, and the number of ponds, of 7, in which the model had a lower S than the primary baseline, with strict comparisons, so that the count is zero for the baseline itself. The deep models and the Static MLP are given as the mean ± SD over five seeds. The scored ponds were ponds 1, 2, 3, 4, 6, 7 and 9, with 35 samples in T1, 29 in T2 (14 days ahead) and 24 in T2 (28 days ahead). Figure 6 shows the S of every pond for the models of Table 4: the colored circles are the scored ponds, with one color per pond in

all figures; the black diamonds are the *S* of the model; the dashed line marks the *S* of the primary baseline; and the horizontal axis is logarithmic. Figures S5 and S6 show the same for every model. Table S4, provided as a data file, repeats the results with the SD of *S* and of the MAE across ponds; Table S5, provided as a data file, gives the results of every pond; and Table S6, provided as a data file, gives the hyperparameters chosen in every outer fold.

Table 4. The leave-one-pond-out results of the main models.

Task/ Horizon	Model	*S*	95% Interval of *S*	MAE (g)	R^2	Ponds with Lower *S* (of 7)
T1	Cohort mean [a]	0.265	0.215 to 0.310	28.3	0.867	0
	Growth curve	0.261	0.211 to 0.309	27.4	0.881	3
	GBM (time + WQ)	0.436	0.390 to 0.488	47.3	0.581	0
	GBM (time + WQ + length) [c]	0.244	0.178 to 0.331	25.4	0.852	4
	LSTM [b]	0.290 ± 0.009	0.217 to 0.381	28.7 ± 2.3	0.834 ± 0.041	3
	GRU	0.333 ± 0.014	0.265 to 0.437	33.5 ± 1.6	0.803 ± 0.013	4
	CNN-LSTM	0.338 ± 0.050	0.222 to 0.487	33.3 ± 3.7	0.766 ± 0.065	3
	TFT	0.432 ± 0.043	0.384 to 0.485	44.4 ± 5.2	0.603 ± 0.111	0
	Static MLP	0.306 ± 0.009	0.245 to 0.365	33.4 ± 3.5	0.785 ± 0.061	3
T2, 14 days ahead	Ridge (past weights) [a]	0.295	0.254 to 0.339	33.6	0.660	0
	GBM (past weights)	0.211	0.175 to 0.250	19.4	0.929	7
	GBM (past weights + WQ)	0.286	0.234 to 0.329	28.5	0.840	3
	LSTM	0.320 ± 0.028	0.269 to 0.371	33.4 ± 3.8	0.808 ± 0.040	3
	GRU	0.295 ± 0.039	0.248 to 0.345	31.1 ± 6.3	0.806 ± 0.097	4
	CNN-LSTM [b]	0.332 ± 0.024	0.264 to 0.398	33.5 ± 4.1	0.806 ± 0.064	2
	TFT	0.341 ± 0.021	0.304 to 0.372	50.7 ± 8.0	0.053 ± 0.442	1
	Static MLP	0.257 ± 0.010	0.222 to 0.296	28.1 ± 3.5	0.825 ± 0.069	5
T2, 28 days ahead	Ridge (past weights) [a]	0.243	0.176 to 0.308	33.7	0.833	0
	GBM (past weights)	0.217	0.168 to 0.281	32.0	0.836	4
	GBM (past weights + WQ)	0.255	0.198 to 0.314	37.0	0.781	3
	LSTM	0.298 ± 0.049	0.247 to 0.354	45.6 ± 8.0	0.649 ± 0.080	1
	GRU	0.271 ± 0.024	0.206 to 0.342	38.0 ± 7.5	0.744 ± 0.118	3
	CNN-LSTM [b]	0.303 ± 0.022	0.235 to 0.363	43.5 ± 5.0	0.699 ± 0.075	1
	TFT	0.315 ± 0.006	0.248 to 0.387	47.0 ± 6.7	0.585 ± 0.197	1
	Static MLP	0.246 ± 0.011	0.175 to 0.328	34.3 ± 0.7	0.828 ± 0.009	2

[a] Primary baseline. [b] Primary deep model. [c] Uses the fish length measured at the weighing. The deep models and the Static MLP are given as mean ± SD over five seeds. GBM, gradient boosting machine; MAE, mean absolute error; MLP, multilayer perceptron; SD, standard deviation; WQ, water quality.

No deep model had a lower *S* than the primary baseline of its task, in T1 or in either T2 comparison, and in each comparison at least one baseline without water quality had a lower *S* than every deep model (Table 4).

In T1, the four baselines without water quality had *S* values between 0.261 (growth curve) and 0.269 (Ridge (time)), among them the Cohort mean, the primary baseline, with 0.265 (bootstrap 95% interval 0.215 to 0.310). The LSTM, the primary deep model, had an *S* of 0.290 ± 0.009 (mean ± SD over five seeds) and a lower *S* than the Cohort mean in three of seven ponds (ponds 1, 2 and 4). The GRU (0.333 ± 0.014), the CNN-LSTM (0.338 ± 0.050) and the TFT (0.432 ± 0.043) had a higher *S* than the LSTM, and the TFT had a lower *S* than the Cohort mean in 0 of seven ponds. The Static MLP, which received only the time since stocking, had 0.306 ± 0.009.

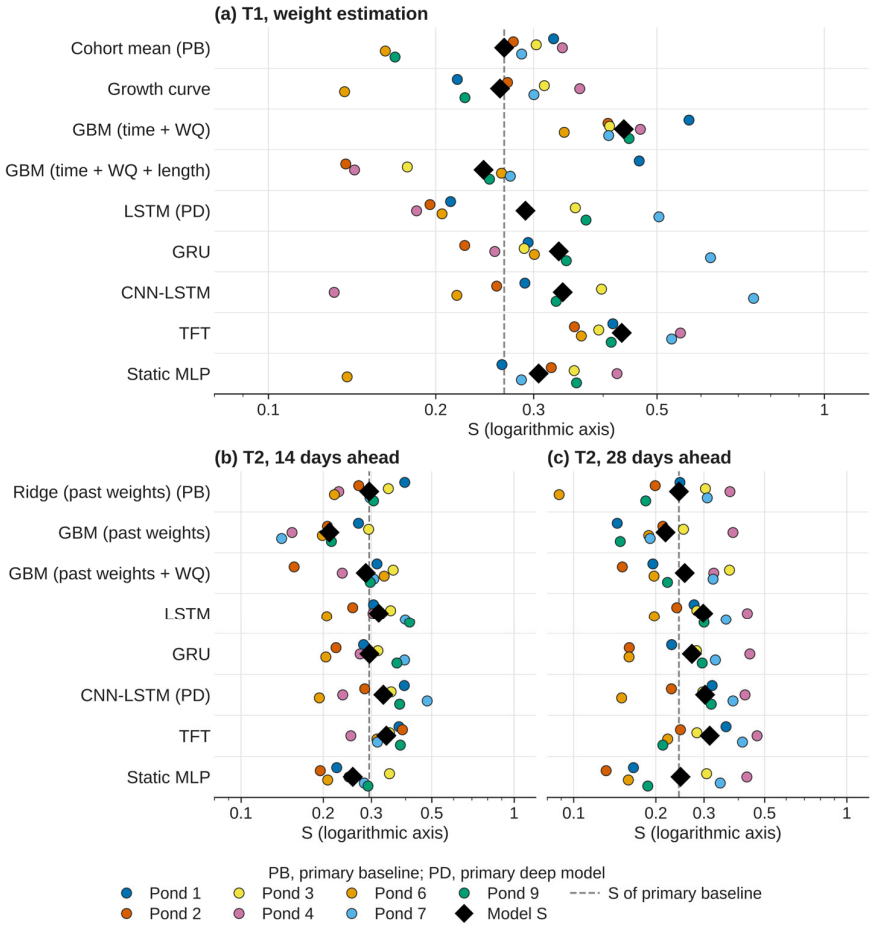

Figure 6. The per-pond *S* of the models of Table 4 under leave-one-pond-out validation for T1 (**a**), T2 (14 days ahead) (**b**) and T2 (28 days ahead) (**c**).

The tabular models with water quality had *S* values between 0.244 and 4.463, the lowest being that of GBM (time + WQ + length) and the highest that of Ridge (WQ) (Table S18). Among them, only GBM (time + WQ + length), which also received the length measured at the weighing, had a lower *S* than the Cohort mean, and it was lower in four of seven ponds. Without length, GBM (WQ) had 0.507 and GBM (time + WQ) had 0.436. The ridge models with water quality had the largest *S* values of T1 (Ridge (WQ) 4.463, Ridge (time + WQ) 1.582 and Ridge (time + WQ + length) 3.429), and pond 7 had the largest *S* values of the ponds for each of them (27.702, 5.769 and 19.915, respectively). Ridge (time + WQ) predicted weights above 1000 g for two samples of pond 2 (6.2×10^4 g and 4.2×10^5 g) and weights below 1 g for five samples of pond 7 (0.13 g to 0.38 g; Figure 7). No model predicted a weight of 0 g or less, so that no *S* value was infinite (Section 3.6).

Figure 7 shows the T1 predictions of the Cohort mean, the LSTM, the Static MLP and the two tabular models with time and water quality for every pond. Filled circles are real weighings, including weighings that were not scored, and open markers with line styles identify the models; predictions exist only at the dates of the eligible events, and the connecting segments are visual guides. The predictions of the LSTM and the Static MLP are means over five seeds, with light bands spanning the minimum and the maximum over the seeds, and a prediction outside the common logarithmic weight axis is drawn as an open triangle at the edge of the panel with its value in grams. The calendar axis ends on 31 October 2021, because later weighings were not scored. The LSTM had a lower *S* than the Cohort mean in ponds 1, 2 and 4 and a higher *S* in ponds 3, 6, 7 and 9, with the largest differences in pond 7 (0.504 against 0.285) and pond 9 (0.373 against 0.169). The allometric

reference, which used the length measured at the weighing, had an *S* of 0.115 (0.092 to 0.142), was lower than the Cohort mean in seven of seven ponds, and had less than half of the *S* value of the best model that did not use length (growth curve, 0.261); this reference is descriptive.

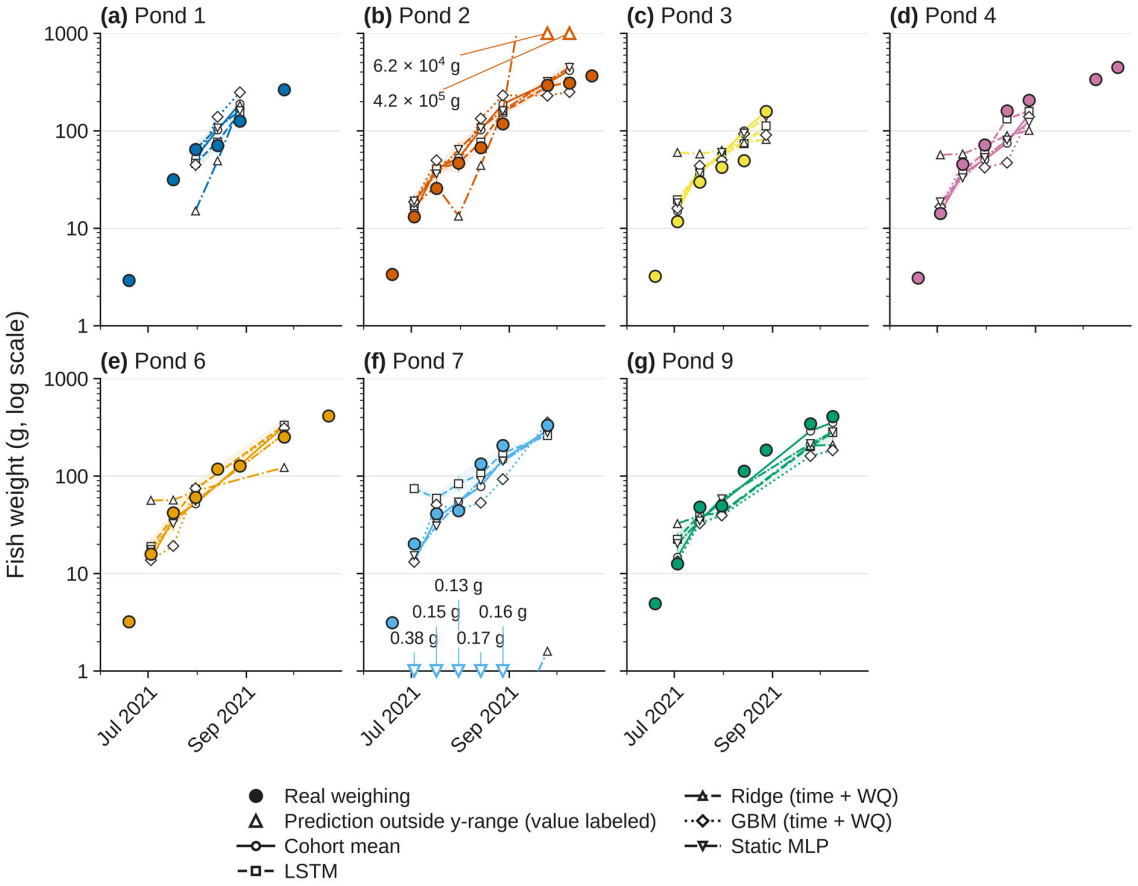

Figure 7. T1 weight estimation under leave-one-pond-out validation for the seven scored ponds (panels (**a**–**g**): ponds 1, 2, 3, 4, 6, 7 and 9).

In T2 (14 days ahead), Persistence had an *S* of 0.529 and an MAE of 38.4 g (Table S18): every target weight exceeded the weight at the origin (29 of 29 samples), so that Persistence under-predicted all samples. The primary baseline, Ridge (past weights), had an *S* of 0.295, and the primary deep model, the CNN-LSTM, had 0.332 ± 0.024 and an MAE of 33.5 ± 4.1 g. Four baselines without water quality had a lower *S* than the CNN-LSTM: GBM (past weights) (0.211), Anchored growth curve (0.274), Cohort growth (0.281) and Ridge (past weights) (0.295). The CNN-LSTM had a lower *S* than Ridge (past weights) in two of seven ponds, than GBM (past weights) in one of seven ponds and than Persistence in seven of seven ponds. The LSTM had 0.320 ± 0.028 and the TFT 0.341 ± 0.021. The GRU had 0.295 ± 0.039, equal to the 0.295 of Ridge (past weights) at three decimals, and a lower *S* than Ridge (past weights) in four of seven ponds. GBM (past weights + WQ) had 0.286, which is higher than the 0.211 of GBM (past weights). The Static MLP, which has no sensor input, had 0.257 ± 0.010, a lower *S* than each of the four deep models. Ridge (past weights + WQ) stopped under the rule of Section 3.6 and has no result (Section 4.4).

T2 (28 days ahead) was descriptive by design and had 24 samples. Persistence had an *S* of 0.983 and an MAE of 85.5 g, Ridge (past weights) 0.243 and the CNN-LSTM 0.303 ± 0.022 (MAE 43.5 ± 5.0 g). The CNN-LSTM had a lower *S* than Ridge (past weights) in one of seven ponds, than GBM (past weights) in one of seven ponds and than Persistence in seven

of seven ponds. Four baselines without water quality had a lower *S* than the CNN-LSTM: GBM (past weights) (0.217), Ridge (past weights) (0.243), Cohort growth (0.254) and Anchored growth curve (0.263). The GRU had 0.271 ± 0.024, the LSTM 0.298 ± 0.049 and the TFT 0.315 ± 0.006, and the Static MLP had 0.246 ± 0.011. GBM (past weights + WQ) had 0.255.

Persistence under-predicted every T2 sample at both horizons (29 of 29 at 14 days ahead and 24 of 24 at 28 days ahead), as its points below the diagonal in Figure 8 show, and its *S* and its MAE grew with the horizon (*S* from 0.529 to 0.983 and MAE from 38.4 g to 85.5 g). The MAE of the CNN-LSTM was lower than that of Persistence at both horizons (33.5 ± 4.1 g against 38.4 g, and 43.5 ± 5.0 g against 85.5 g). In Figure 8, each column shows one model, the upper row the 14-day and the lower row the 28-day horizon; both axes are logarithmic with equal limits, the dashed diagonal marks equality, colors identify the ponds, and the points of the CNN-LSTM are means over five seeds. The comparison at 28 days ahead is descriptive, and neither horizon tests forecasting into a later period (Section 3.5).

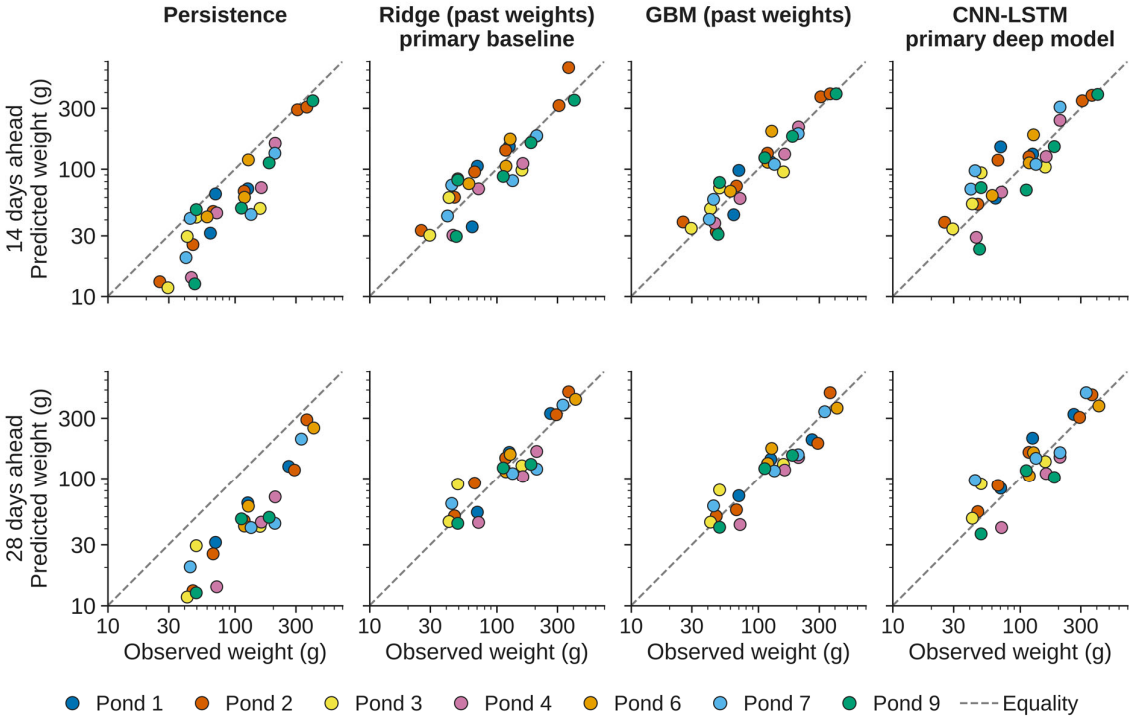

Figure 8. Predicted against observed weight in T2 under leave-one-pond-out validation for four models at 14 days ahead (**upper row**) and 28 days ahead (**lower row**).

Across the three comparisons, the *S* values of the four deep models differed from one another by at most 0.142 in T1, 0.045 in T2 (14 days ahead) and 0.044 in T2 (28 days ahead), computed from the unrounded values (Section 3.6). The TFT had the highest *S* value of the deep models in all three comparisons, and the CNN-LSTM had the third lowest in each. For each of the four deep models and the Static MLP in each comparison, the SD of *S* over the five seeds was smaller than the SD of the per-pond *S* of the same model; the seed SDs ranged from 0.006 to 0.050 and the pond SDs from 0.050 to 0.198.

4.3. Confirmatory Tests and Sensitivity Analyses

Table 5 gives the fixed sequence of confirmatory tests, and Figure S7 shows the per-pond differences in *S* between the primary deep model and the primary baseline. In Table 5, T^+ is the sum of the ranks of the ponds in which the deep model had a higher *S* than the

baseline; the one-sided alternative is a lower S of the deep model, zero differences were dropped, and the exact distribution of the signed-rank statistic with midranks was used. Decisions use the unrounded p-values at $\alpha = 0.05$. The second step is tested only if the first step rejects, so that its p-values are nominal and descriptive here.

Table 5. Fixed-sequence confirmatory Wilcoxon tests.

Step	Task/ Horizon	Model	Comparator	Ponds	Zeros Dropped	T^+	Exact One-Sided p	Exact Two-Sided p	Decision at Alpha 0.05
1	T1	LSTM	Cohort mean	7	0	16	0.656	0.812	not rejected
2	T2, 14 days ahead	CNN-LSTM	Ridge (past weights)	7	0	22	0.922	0.219	not tested (an earlier step did not reject)

CNN-LSTM, hybrid convolutional and long short-term memory network; LSTM, long short-term memory network.

The confirmatory test for T1 did not reject (one-sided $p = 0.656$; Table 5), so that the sequence stopped at its first step. Step 1 compared the LSTM with the Cohort mean in T1. The mean difference in S (LSTM minus Cohort mean) was 0.025 (bootstrap 95% interval −0.073 to 0.131), and the LSTM had a lower S in three of seven ponds (ponds 1, 2 and 4). The sum of the ranks of the ponds in which the LSTM was worse was $T^+ = 16$, and the exact one-sided p-value was 0.656 (two-sided 0.812), so that the test did not reject. The second step, the CNN-LSTM against Ridge (past weights) in T2 (14 days ahead), was therefore not tested. Its values, which are nominal and descriptive, were a difference of 0.037 (−0.002 to 0.091), a lower S of the CNN-LSTM in two of seven ponds, $T^+ = 22$ and a one-sided p-value of 0.922 (two-sided 0.219). For T2 (28 days ahead), which was descriptive by design, the difference was 0.060 (0.030 to 0.090), the CNN-LSTM had a lower S in one of seven ponds, and the one-sided p-value was 0.992 (two-sided 0.031). In all three comparisons the primary deep model had a higher S value than the primary baseline. The interval of the difference at 28 days ahead did not include zero, in the direction of the higher S value of the CNN-LSTM; this comparison is not part of the confirmatory sequence and is not adjusted for multiplicity.

As required by the analysis plan (Section 3.6), the baselines without water quality that had a lower S than the primary deep model are named here. In T1, these were growth curve (0.261), the Cohort mean (0.265), GBM (time) (0.266) and Ridge (time) (0.269), against 0.290 for the LSTM. In T2 (14 days ahead), they were GBM (past weights) (0.211), Anchored growth curve (0.274), Cohort growth (0.281) and Ridge (past weights) (0.295), against 0.332 for the CNN-LSTM, which had a higher S than GBM (past weights) in six of seven ponds. In T2 (28 days ahead), they were GBM (past weights) (0.217), Ridge (past weights) (0.243), Cohort growth (0.254) and Anchored growth curve (0.263), against 0.303 for the CNN-LSTM.

Ponds 1 and 2 share one sensor stream, and the tests were repeated with the two ponds averaged into one unit, which left six units (Table S7). The direction did not change. In T1, the LSTM had an S of 0.304 and the Cohort mean had 0.260 ($T^+ = 14$, one-sided $p = 0.781$, two-sided 0.562). In T2 (14 days ahead), the CNN-LSTM had 0.331 and Ridge (past weights) 0.288 ($T^+ = 17$, one-sided $p = 0.922$, two-sided 0.219), and in T2 (28 days ahead) they had 0.308 and 0.246 ($T^+ = 20$, one-sided $p = 0.984$, two-sided 0.062). With six units, the exact test can reject only if T^+ is at most 2, and none of these descriptive tests did.

Removing one pond at a time from the comparison changed the T1 difference in S between −0.008 and 0.054 (Table S8, provided as a data file). The LSTM had a higher S than the Cohort mean after the removal of any pond except pond 7 (difference −0.008) and pond 9 (difference −0.005), where the sign changed. The sign of the T1 difference therefore

depended on single ponds. In T2, the CNN-LSTM had a higher S than Ridge (past weights) whichever pond was removed, with differences from 0.013 to 0.048 at 14 days ahead and from 0.047 to 0.071 at 28 days ahead.

A non-significant result is not evidence of no effect. With seven ponds, the exact test could reject only if T^+ was at most 3 (five of 128 sign patterns; Section 3.6), and the interval of the T1 difference (−0.073 to 0.131) contains differences of either sign. The T1 data therefore do not show that the LSTM and the Cohort mean are equally accurate. They show that the LSTM was not more accurate than the Cohort mean in these 35 events in seven ponds, and they cannot exclude a difference of either sign within the bootstrap interval.

4.4. Water Quality Inputs

The permutation tests (Section 3.6) asked whether a model used pond-specific water quality information, that is, whether its S with the true inputs was lower than its S with the inputs of the sensor streams reassigned within each date. Table 6 gives the results, and Figure S8 shows the permutation distributions. In Table 6, the statistic is S (for T2, S at 14 days ahead) and p is the permutation p-value of Equation (23); the tuned hyperparameters stayed fixed, and the deep models were fitted with seed 0 in the observed and in the permuted runs, so that their observed S differs from the mean over five seeds in Table 4. The tests are outside the confirmatory sequence and are not adjusted for multiplicity.

Table 6. Within-date permutation tests of the water quality inputs.

Task/Horizon	Model	Seed	Observed S	Permutations	Permuted S ≤ Observed	Permutation *p*
T1	Ridge (time + WQ)	0	1.582	999	947	0.948
T1	GBM (time + WQ)	0	0.436	999	869	0.870
T1	LSTM	0	0.292	199	0	0.005
T2, 14 days ahead	Ridge (past weights + WQ)	–	stopped	–	–	–
T2, 14 days ahead	GBM (past weights + WQ)	0	0.286	999	259	0.260
T2, 14 days ahead	CNN-LSTM	0	0.350	199	190	0.955

A dash means not applicable. GBM, gradient boosting machine; LSTM, long short-term memory network; WQ, water quality.

Only the T1 LSTM had a permutation p-value below 0.05 (Table 6). For the T1 LSTM, the S of the seed-0 fit with the true inputs was 0.292, and none of the 199 permuted fits had an S at or below it. The permuted values ranged from 0.294 to 0.478 with a mean of 0.386, and the permutation p-value was 0.005. The permutation p-values of the other models were 0.948 for Ridge (time + WQ) (947 of 999 permuted values at or below the observed 1.582), 0.870 for GBM (time + WQ) (869 of 999, observed 0.436), 0.260 for GBM (past weights + WQ) in T2 (259 of 999, observed 0.286) and 0.955 for the CNN-LSTM in T2 (190 of 199, observed 0.350). The observed S of the T1 LSTM and of the T2 CNN-LSTM are those of seed 0 and differ from the means over five seeds (0.290 and 0.332). Ridge (past weights + WQ) had no test because its run stopped.

The T1 LSTM therefore used pond-specific water quality information, in the sense that its fit with the true inputs was better than each of the 199 fits with reassigned inputs. This does not make it more accurate than the baselines. Its S (0.290) was higher than that of the Cohort mean (0.265), the confirmatory test did not reject (Section 4.3), and the permutation test compares the true inputs with wrong ones and not with no inputs. The model without sensor input was the Static MLP (0.306 ± 0.009), and the mean S of the LSTM was lower than that of the Static MLP (difference −0.016, interval −0.109 to 0.082), but the LSTM was lower than the Static MLP in only three of seven ponds (one-sided $p = 0.469$, descriptive).

In T2, the Static MLP had a lower S than the CNN-LSTM, with differences (CNN-LSTM minus Static MLP) of 0.075 (0.018 to 0.133) at 14 days ahead and 0.057 (0.012 to 0.105) at 28 days ahead, which is descriptive. A statement that water quality contributed to a result required, in addition to a rejected confirmatory test, a permutation p-value below 0.05 for the primary deep model and a lower S than the Static MLP (Section 3.6). The T1 LSTM met these two conditions on the mean S, but the confirmatory test did not reject, and no such statement is made.

In the ablation (Figure S2, descriptive), adding water quality features raised S for every tabular model that completed and had a counterpart without them: in T1, GBM from 0.266 to 0.436 and Ridge from 0.269 to 1.582, and in T2, GBM from 0.211 to 0.286 at 14 days ahead and from 0.217 to 0.255 at 28 days ahead. Adding length to the T1 models with time and water quality lowered S for GBM (from 0.436 to 0.244) and raised it for Ridge (from 1.582 to 3.429). In T2, the Static MLP had a lower S than each of the four deep models at both horizons, and in T1 it had a lower S than the GRU, the CNN-LSTM and the TFT and a higher S than the LSTM.

Three ridge runs stopped under the rule of Section 3.6 (Tables S9 and S18): Ridge (past weights + WQ) in T2, which serves both horizons, and two sensitivity variants, Ridge (time + WQ) with a window of 28 days in T1 and Ridge (past weights + WQ) without missing-value indicators in T2. Each run stopped because an inner-validation prediction was not finite (one of 10 in T1 and one of 15 in each T2 run). The logs record four attempts of the T1 run and three of each T2 run, all with the frozen settings, and every attempt stopped at the same inner split with the same error. A diagnosis on the inner splits only, which used no test prediction, traced each of the three stops to the same inner-validation row, that of pond 9 on 25 September 2021, in which the minimum of the dissolved oxygen readings over the window was 1.201. In the inner-training ponds of that split (1, 2, 6 and 7) the same window minimum took only the values 0.007 and 0.008, so that the standardized value of the row was 2983 in T1 and 2854 in T2. With a small regularization strength, the ridge model turned this value into a prediction on the log scale above 709.8, the largest argument for which the exponential function is finite in double precision, so that the predicted weight in grams was not finite.

A post hoc ridge variant (Section 3.6), in which a regularization strength with a nonfinite inner-validation prediction lost its trial, was fixed before any test score existed and was run for the three stopped runs. Its results are reported in the Supplementary Materials as post hoc results (Tables S10–S12, the last provided as a data file) and do not replace a frozen result.

4.5. Descriptive Analyses

The analyses of this section are descriptive and were not part of the confirmatory sequence. The calendar hold-out (Section 3.5) trained the models on the weighings dated before 28 August 2021 and tested them on the later ones: 11 events in seven ponds in T1, 10 samples in seven ponds in T2 (14 days ahead) and five samples in four ponds in T2 (28 days ahead; Table S13 and Figure S3). No p-values or intervals were computed. In T1, Ridge (time) (S 0.333) and the Static MLP (0.337 ± 0.123) had the lowest S among the models that did not use length, followed by the CNN-LSTM (0.419 ± 0.180), and the allometric reference had 0.037. The Cohort mean and the growth curve had the same S (0.607), because no training pond had a weighing on the date of a test event, so that the Cohort mean used its fallback, the prediction of the growth curve (Section 3.4.2). Compared with the Cohort mean, the LSTM (0.669 ± 0.107) had a lower S in two of seven ponds, the GRU (0.727 ± 0.262) in four, the CNN-LSTM in five and the TFT (1.348 ± 0.171) in 0. GBM

(time), whose predictions were constant in t because the test dates lay beyond all training dates (Section 3.5), had 0.911.

In T2 (14 days ahead), Anchored growth curve and Cohort growth had the lowest S (0.253, identical because Cohort growth used the prediction of Anchored growth curve when no training sample had the same origin and target dates), followed by the Static MLP (0.302 ± 0.101), the LSTM (0.311 ± 0.085), the GRU (0.321 ± 0.078) and Linear extrapolation (0.338). Ridge (past weights) had 0.407, the CNN-LSTM 0.383 ± 0.051 and the TFT 0.407 ± 0.068. Compared with Ridge (past weights), the LSTM, the GRU, the CNN-LSTM and the Static MLP had a lower S in six of seven ponds each and the TFT in three, but none of these models had a lower S than Anchored growth curve and Cohort growth. GBM (past weights) and GBM (past weights + WQ) had the same S (0.450), because both predicted one constant growth ratio for every test sample. Ridge (past weights + WQ), which stopped under leave-one-pond-out validation, completed in the hold-out and had 1.118. In the leave-one-pond-out comparison, in contrast, the CNN-LSTM had a lower S than Ridge (past weights) in two of seven ponds (Section 4.3). The hold-out has 10 samples and no test was applied, so that this contrast is descriptive.

In T2 (28 days ahead), the five samples in four ponds gave S values between 0.135 (Linear extrapolation) and 0.573 (Persistence). The Static MLP had 0.184 ± 0.022, the LSTM 0.193 ± 0.066, Ridge (past weights) 0.201, the CNN-LSTM 0.214 ± 0.063, which had a lower S than Ridge (past weights) in one of four ponds, the GRU 0.234 ± 0.064 and the TFT 0.280 ± 0.085. These samples are too few for a comparison of models.

The secondary set of all 63 weighings after stocking in eight ponds (Table S14) was scored for the baselines without water quality that need no sensor data. The growth curve had 0.198 (a lower S than the Cohort mean in five of eight ponds), the Cohort mean 0.207, GBM (time) 0.214 and Ridge (time) 0.231.

In the sensitivity analyses (Table S9), the window of the water quality features and the missing-value indicators of Ridge were varied. The variants that completed had S values between 0.366 and 1.679 in T1, all above the 0.265 of the Cohort mean, between 0.257 and 0.873 in T2 (14 days ahead) and between 0.322 and 0.780 in T2 (28 days ahead). For Ridge (time + WQ) in T1, S was 0.396 with a window of 7 days, against 1.582 with the 14 days of the main analysis. In T2 (14 days ahead), only GBM (past weights + WQ) with the 42-day window (0.257) was lower than Ridge (past weights) (0.295), and it remained above GBM (past weights) (0.211); in T2 (28 days ahead), all variants were above Ridge (past weights) (0.243). The sensitivity variants that stopped are described in Section 4.4.

The learning curves of the chosen trials and of the final fits are shown in Figure S1. Early stopping ended 209 of the 210 inner fits of the chosen trials before the limit of 300 epochs, and the exception was a fit of the GRU in T2, which ran for all 300 epochs. The final fits ran for the median best epoch of the inner folds, between four and 174 epochs (Table S6). These curves are training diagnostics and not a model comparison.

Table S15, provided as a data file, gives the logged times of every fit. On the CPU of Section 3.7, tuning a deep model under leave-one-pond-out validation took between 597.3 s and 6428.2 s, and the final fits of all folds and seeds between 18.8 s and 197.6 s. Tuning the Static MLP took 202.2 s in T1 and 273.7 s in T2, tuning a GBM at most 80.1 s, and tuning each of the other models at most 4.4 s.

5. Discussion

This study asked whether deep learning models estimate and predict the weight of African catfish in aquaponic fish ponds better than simple baselines when only real weighings are scored and whole ponds are held out. In these data, no deep model had a lower S than the primary baseline of its task, and the confirmatory test of T1 did not reject

(LSTM S of 0.290 against 0.265 for the Cohort mean, with a lower S in three of seven ponds, one-sided $p = 0.656$). The second step was therefore not tested, and in T2 the primary deep model had a higher S than the primary baseline at both horizons. Baselines without water quality had a lower S than the primary deep model in every comparison, and the four deep models differed from one another by at most 0.142 in S (Section 4.2). A non-significant result is not evidence of no effect: the study has 35 scored T1 events in seven ponds, the interval of the T1 difference (−0.073 to 0.131) contains both signs, and the exact test could reject only if T^+ was at most 3. The results are therefore a leakage-aware benchmark of these models on these data, and they do not show that deep learning is unsuitable for fish weight prediction in general.

Scoring real weighings matters on these data. In the window-level analysis of pond 2 (Section 3.2 and Supplementary Section S1), Persistence had an MAE of 0.0026 g on the 34,298 test windows of a time-ordered split, because the label changes once per day and only 17 of these targets differ from the previous row, and a CNN-LSTM reached an error below 1 g only when windows were assigned to the training and test sets at random (Table S19). On the real weighings, in contrast, Persistence under-predicted every T2 sample and had an MAE of 38.4 g at 14 days ahead and 85.5 g at 28 days ahead, and the lowest MAPE of a model that did not use length was 21.1% (GBM (past weights) at 28 days ahead; Table S18). Errors of a fraction of a gram on the interpolated label therefore measure agreement with the stored daily values and not the prediction of growth, and analyses of these public files should score real weighings and hold out whole ponds.

Among the deep models, the TFT had the highest S value of the deep models in all three comparisons, the LSTM had the lowest in T1 and the GRU in both T2 comparisons, the CNN-LSTM had the third lowest in each, and the differences between the deep models were at most 0.142 in T1 and 0.045 in T2 (Section 4.2). This is consistent with the remark of Cardenas-Cartagena et al. [21] that the choice of architecture should be guided by the characteristics of the data and the prediction objective, although the comparisons between deep models made here are descriptive. Studies that reported advantages of the GRU [17,18], of hybrid convolutional-recurrent models [22,23] or of the TFT [26] concerned water quality variables, and the present results neither confirm nor contradict them, because the target here is the weight of the fish, which was measured only every two weeks. For fish weight in aquaponics, Eneh et al. [8] reported an RMSE of 7.0192 and an MAE of 2.2310 for XGBoost with the data of their initial system (their Table 2). These values cannot be compared with the errors of the present study: their data came from one pond, and their paper does not describe how the test data were separated from the training data.

The LSTM and the GRU differed by at most 0.043 in S, and the better of the two changed between tasks: the LSTM was lower in T1 (0.290 against 0.333), and the GRU was lower in T2 at both horizons (0.295 against 0.320 at 14 days ahead and 0.271 against 0.298 at 28 days ahead). Scored on real weighings, neither of the two recurrent architectures was therefore consistently better than the other.

Several features of the data limit what the comparison can show. The cohort has 70 weighings in eight ponds, of which 35 were eligible for T1 in seven ponds, and the T2 samples number 29 and 24; ponds 1 and 2 share one sensor stream, so that the scored ponds contain six independent streams. The weight of a pond is a single value per weighing, treated as the pond mean of about 10 to 15 sampled fish, and the data paper does not state how the weights of these fish were combined, so that the sampling error of the pond mean is unknown [11]. Many sensor readings were invalid: the share of the readings of the cohort ponds that were missing in the file or failed a quality-control rule was 33.8% for turbidity, 31.2% for ammonia, 24.2% for dissolved oxygen and 12.5% for pH (Table S1). Eneh et al. [8] reported that, after testing their initial system for 10 months against ground-truth values,

the pH sensor, which was not meant for a volume of water as large as a fish pond, and the two gas sensors for nitrate and ammonia, which read "high and unusual values", did not perform well. The data paper lists the same six types of sensors [11], but neither paper states which of the public files were produced by the initial system that Eneh et al. evaluated. The quality of the water quality inputs may therefore have limited what any model could learn from them, and this cannot be separated from the possibility that water quality carries little information about the pond mean weight at the scale of the weighings.

In T1, the ridge models with water quality had the largest S (Section 4.2), and ridge runs stopped under the rule of Section 3.6 (Section 4.4). The diagnosis of the stops points to dissolved oxygen readings that are nearly constant within some ponds: in the inner-training ponds the window minimum was 0.007 or 0.008, so that a row with 1.201 in another pond had a standardized value of 2983. The window maximum showed a similar pattern: it was 19.989 to 19.998 in ponds 3, 4 and 7, whereas pond 2 on 9 October 2021 had 4.181, which is 8199 standard deviations from the mean of those three ponds in the inner split in which they were the training ponds. A linear model extrapolates such values without limit. GBM, which cannot extrapolate beyond the range of its training data, and the neural models did not show this failure: in the leave-one-pond-out runs, all predictions of GBM, of the four deep models and of the Static MLP lay between 10.9 g and 922 g. The diagnosis used inner splits only and covered the stopped runs, and whether the same mechanism explains the large errors of the ridge models with water quality that completed was not examined. This is a data-quality point: readings that are almost constant in some ponds and not in others can make a linear model unreliable when a pond outside the training set reads differently.

The permutation result for the T1 LSTM should be read with care. In the permutation test, the S of the LSTM with the true inputs (0.292) was lower than its S with each of the 199 sets of reassigned inputs (0.294 to 0.478), and the permutation p-value was 0.005. This means that its predictions depended on which pond's water quality it received, and not that water quality made it more accurate: its S was higher than that of the Cohort mean, and the test compares true with reassigned inputs and not with no inputs. The model without sensor input, the Static MLP, had an S of 0.306; the difference of the LSTM from it was −0.016, with an interval from −0.109 to 0.082 that contains zero. The permutation test cannot separate information about fish growth from other pond-specific features of the sensor streams, such as sensor offsets or drift. It was one of five permutation tests that were not adjusted for multiplicity, and the deep model was tested with a single seed and 199 permutations. No causal effect of water quality on growth is claimed.

For practice, the results suggest starting from simple baselines. In T1, the four models that used time only had S values between 0.261 and 0.269, below every deep model and every tabular model with water quality that did not use length (Section 4.2). In T2, GBM (past weights) had the lowest S value of the models of Table 4 at both horizons (0.211 and 0.217). The deep models also had the longest tuning times of all models (Section 4.5). A model that used the length measured at the weighing had a much lower S value than any model that used sensor data: the allometric reference had an S of 0.115, less than half of the 0.261 of the growth curve, which is expected, because length and weight are measured at the same weighing (Section 3.4.9). In these data, the deep models of this study did not improve on simple baselines.

Several limitations should be acknowledged. The evaluation rests on a single public dataset from aquaponic fish ponds stocked with African catfish, on eight ponds and 70 weighings, which gave 35 scored events in T1 and 29 and 24 scored samples in T2 (14 and 28 days ahead, respectively), and generalization to other systems, species and sensor configurations therefore remains to be verified. Leave-one-pond-out validation tests the transfer between ponds weighed on the same dates; it does not test forecasting of a later period, and its results are not forecasting

results. The calendar hold-out looked at later dates but had 11 events in T1 and 10 and five samples in T2 and was descriptive. No claim extends beyond 112 days in T1 and 126 days in T2, the largest times since stocking of the scored samples. With seven ponds the confirmatory test could reject only for five of 128 sign patterns, and its level is approximate (Section 3.6). The selection of the primary models on the inner folds did not choose the baseline with the lowest outer S in either task or the deep model with the lowest outer S in T2 (Section 4.1). Every deep model had a lower inner-validation S than outer S in both tasks (Section 4.1), so the inner S was optimistic for the deep models. Each tuned model received the same budget of 20 trials per outer fold, and whether a larger budget or other search spaces would change the ranking was not tested. Three ridge runs stopped, and the post hoc ridge variant is not a frozen result. No causal effect of water quality on growth is claimed. The results show that deep learning did not beat simple baselines on these data, and not that it is unsuitable for this problem in general.

Future work needs more ponds and more systems, more frequent weighings, a calibration of the sensors against laboratory measurements, in particular of the gas sensors for ammonia and nitrate, and a record of the spread of the weights of the sampled fish. A design for forecasting should reserve a chronological hold-out from the start, with enough later weighings to test it. Any new comparison should score only real weighings, hold out whole ponds, and report persistence and simple growth baselines next to the deep models.

6. Conclusions

This study compared four deep learning architectures with simple baselines for estimating and predicting fish weight from aquaponic fish pond sensor data, using only real weighings and leave-one-pond-out validation. Because the stored daily weight is a linear interpolation between fortnightly weighings, repeating the previous value gave an MAE of 0.0026 g on a window-level test set of pond 2, and only real weighings were therefore scored. On the real weighings, no deep model had a lower S value than the primary baseline of its task, the confirmatory test did not reject (one-sided $p = 0.656$), and baselines without water quality had a lower S value than the primary deep model in every comparison. A non-significant result is not evidence of no effect, and 35 events in seven ponds cannot show that deep learning is unsuitable in general. The T1 LSTM used pond-specific water quality information (permutation $p = 0.005$) without being more accurate than the baselines, and no causal effect of water quality on growth is claimed. Three ridge runs with water quality stopped on a nonfinite prediction, which a diagnosis on the inner splits traced to dissolved oxygen readings that were nearly constant in some ponds. In T1, a reference model that used the length measured at the weighing had an S of 0.115, less than half of the 0.261 of the best model without length. The results suggest starting from simple baselines, scoring only real weighings and holding out whole ponds. They concern the ponds and the times since stocking of this dataset, up to 112 days in T1 and 126 days in T2, and not the forecasting of later periods. More ponds and systems, more frequent weighings and calibrated sensors are needed to test whether water quality data improve the prediction of fish weight.

Supplementary Materials: The following supporting information can be downloaded at https://www.mdpi.com/article/doi/s1, Section S1: Window-level evaluation of pond 2; Figure S1: Learning curves for LSTM, GRU, CNN-LSTM, TFT and Static MLP under leave-one-pond-out validation on eligible samples; Figure S2: Descriptive ablation scores on eligible leave-one-pond-out samples for T1, T2 (14 days ahead) and T2 (28 days ahead); Figure S3: Descriptive T1 calendar hold-out predictions on eligible test events, with one panel per scored pond; Figure S4: Pond 2 test period of the window-level evaluation, with the interpolated label, persistence and CNN-LSTM predictions; Figure S5: Per-pond S of every T1 model under leave-one-pond-out validation; Figure S6: Per-pond S of every T2 model under leave-one-pond-out validation; Figure S7: Per-pond differences in S between the primary deep model and the primary baseline; Figure S8: Permutation distributions of S under

leave-one-pond-out validation; Table S1: Sensor quality control of the cohort ponds, with raw readings by quality-control rule and sensor-days by status for each pond and sensor; Table S2: Descriptive window-level diagnostic scores for Persistence and CNN-LSTM on pond 2 under Design S0 (scaler fitted on all rows), the chronological split and the random split of windows and their specified evaluation subsets; Table S3: Inner-validation scores and primary model selection for T1 and T2 under nested leave-one-pond-out validation on eligible samples; Table S4: Full model scores for T1, T2 (14 days ahead) and T2 (28 days ahead) under leave-one-pond-out validation on eligible samples, including descriptive sensitivity and length-reference analyses; Table S5: Per-pond errors for T1 and each T2 horizon under leave-one-pond-out and descriptive calendar hold-out validation on eligible samples, with the descriptive leave-one-pond-out set of all weighing events reported separately; Table S6: Chosen hyperparameters for T1 and T2 by model and outer fold under leave-one-pond-out validation, with eligible and descriptive fits on all weighing events kept separate; Table S7: Descriptive merged-stream sensitivity comparisons for T1, T2 (14 days ahead) and T2 (28 days ahead) under leave-one-pond-out validation on eligible samples; Table S8: Descriptive pond-influence comparisons for T1 and each T2 horizon under leave-one-pond-out validation on eligible samples; Table S9: Descriptive sensitivity scores for T1, T2 (14 days ahead) and T2 (28 days ahead) under leave-one-pond-out validation on the unchanged eligible sample sets; Table S10: Post hoc ridge variant scores for T1 and each T2 horizon under leave-one-pond-out validation on eligible samples; Table S11: Post hoc chosen ridge alphas by model and outer fold for T1 and T2 under leave-one-pond-out validation on eligible samples; Table S12: Post hoc agreement of the ridge variant with an independent re-implementation for T2 Ridge (past weights + WQ) under leave-one-pond-out validation on eligible samples, with both horizons pooled within each fold; Table S13: Descriptive model scores for T1, T2 (14 days ahead) and T2 (28 days ahead) under calendar hold-out validation on eligible samples; Table S14: Descriptive T1 model scores under leave-one-pond-out validation on all weighing events after stocking, including events ineligible for water-quality models; Table S15: Logged wall times and hardware for T1 and T2 fits under leave-one-pond-out and calendar hold-out validation, with eligible samples and the descriptive set of all weighing events identified separately; Table S16: Window-level errors on pond 2 of persistence and of the CNN-LSTM under three designs. Table S17: Search spaces and training settings of the deep models and the Static MLP; Table S18: Search spaces of the tabular models; Table S19: Leave-one-pond-out results of all models.

Author Contributions: Conceptualization, M.E.A. and D.A.; methodology, M.E.A. and M.H.M.; software, M.E.A.; validation, M.E.A., D.A. and M.H.M.; formal analysis, M.E.A. and M.H.M.; investigation, M.E.A.; resources, D.A. and I.F.A.G.; data curation, M.E.A.; writing (original draft preparation), M.E.A.; writing (review and editing), D.A., M.H.M. and I.F.A.G.; visualization, M.E.A.; supervision, D.A., M.H.M. and I.F.A.G.; project administration, D.A. and I.F.A.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Selangor State Government under the Fundamental Research Grant Scheme, grant number I/INTEGRATEDFARMING/ST-2021/01.

Institutional Review Board Statement: As this manuscript does not involve any research on animals or human subjects, ethical approval from an Institutional Review Board was not required.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset analyzed in this study is publicly available and was published by Udanor et al. [11] (https://doi.org/10.1016/j.dib.2022.108400). The pond files were taken from the Kaggle release of the dataset (https://doi.org/10.34740/KAGGLE/DSV/3748790), which is distributed under the CC BY 4.0 license. The code, the configuration files, the processed data tables, the saved predictions and the result tables of this study are available at https://github.com/MohamedElminur1/DL_Models_Fishweight (accessed on 4 November 2026). No new data were collected in this study.

Acknowledgments: During the preparation of this manuscript, the authors used Claude (Claude 3.5 Sonnet, Anthropic, San Francisco, CA, USA), a generative artificial intelligence tool, for the purposes of writing and checking the analysis code, running the analysis, and drafting and editing the text. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

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

Abbreviations

The following abbreviations are used in this manuscript:

AIC	Akaike information criterion
BiGRU	Bidirectional gated recurrent unit
BiLSTM	Bidirectional long short-term memory
CC BY	Creative Commons Attribution
CET	Central European Time
CNN	Convolutional neural network
CNN-GRU	Convolutional neural network combined with gated recurrent unit
CNN-LSTM	Convolutional neural network combined with long short-term memory
CPU	Central processing unit
DWC	Deep water culture
ELU	Exponential linear unit
GBM	Gradient boosting machine
GLU	Gated linear unit
GPU	Graphics processing unit
GRN	Gated residual network
GRU	Gated recurrent unit
IoT	Internet of Things
KNN	k-nearest neighbors
LOPO	Leave-one-pond-out
LSTM	Long short-term memory
MAE	Mean absolute error
MAPE	Mean absolute percentage error
MLP	Multilayer perceptron
MSE	Mean squared error
NA	Not available
NFT	Nutrient film technique
NTU	Nephelometric turbidity unit
QC	Quality control
R^2	Coefficient of determination
RAS	Recirculating aquaculture system
ReLU	Rectified linear unit
RMSE	Root mean squared error
RMSLE	Root mean squared logarithmic error
RNN	Recurrent neural network
S	Mean over ponds of the mean absolute error of the natural logarithm of weight
SD	Standard deviation
SGR	Specific growth rate
T1	Task of weight estimation at a weighing
T2	Task of weight prediction from past weights
T^+	Sum of the ranks of the ponds in which the model had a higher S than the comparator (Wilcoxon signed-rank statistic)
TFT	Temporal fusion transformer
TPE	Tree-structured Parzen estimator
UTC	Coordinated Universal Time
VSN	Variable selection network
WQ	Water quality
XGBoost	Extreme gradient boosting

References

1. Okomoda, V.T.; Oladimeji, S.A.; Solomon, S.G.; Olufeagba, S.O.; Ogah, S.I.; Ikhwanuddin, M. Aquaponics Production System: A Review of Historical Perspective, Opportunities, and Challenges of Its Adoption. *Food Sci. Nutr.* **2023**, *11*, 1157–1165. [CrossRef] [PubMed]
2. Ibrahim, L.A.; Shaghaleh, H.; El-Kassar, G.M.; Abu-Hashim, M.; Elsadek, E.A.; Alhaj Hamoud, Y. Aquaponics: A Sustainable Path to Food Sovereignty and Enhanced Water Use Efficiency. *Water* **2023**, *15*, 4310. [CrossRef]

3. Nair, C.S.; Manoharan, R.; Nishanth, D.; Subramanian, R.; Neumann, E.; Jaleel, A. Recent Advancements in Aquaponics with Special Emphasis on Its Sustainability. *J. World Aquac. Soc.* **2025**, *56*, e13116. [CrossRef]
4. Chiu, M.-C.; Yan, W.-M.; Bhat, S.A.; Huang, N.-F. Development of Smart Aquaculture Farm Management System Using IoT and AI-Based Surrogate Models. *J. Agric. Food Res.* **2022**, *9*, 100357. [CrossRef]
5. Sundararajan, S.C.M.; Shankar, Y.B.; Sinthia, P.; Manogaran, N.; Seerangan, K.; Deepa, N.; Selvarajan, S. IoT-Based Prediction Model for Aquaponic Fish Pond Water Quality Using Multiscale Feature Fusion with Convolutional Autoencoder and GRU Networks. *Sci. Rep.* **2025**, *15*, 1925. [CrossRef] [PubMed]
6. Qin, H.; Tian, Y.; Wang, H.; Li, Q.; Cong, X.; Sui, J. Research Progress of Aquaculture Environmental Factor Prediction Model Based on Machine Learning. In Proceedings of the 2023 9th International Conference on Control Science and Systems Engineering (ICCSSE), Shenzhen, China, 16–18 June 2023; pp. 263–269. [CrossRef]
7. Tran-Quang, V.; Ha-Ngoc, A. Aquaculture Environment Prediction Based on Improved LSTM Deep Learning Model. In Proceedings of the 2021 8th NAFOSTED Conference on Information and Computer Science (NICS), Hanoi, Vietnam, 21–22 December 2021; pp. 486–491. [CrossRef]
8. Eneh, A.H.; Udanor, C.N.; Ossai, N.I.; Aneke, S.O.; Ugwoke, P.O.; Obayi, A.A.; Ugwuishiwu, C.H.; Okereke, G.E. Towards an Improved Internet of Things Sensors Data Quality for a Smart Aquaponics System Yield Prediction. *MethodsX* **2023**, *11*, 102436. [CrossRef] [PubMed]
9. Kumar, P.; Tiwari, P.K.; Reddy, U.S. Estimating Fish Weight Growth in Aquaponic Farming through Machine Learning Techniques. In Proceedings of the 2023 3rd International Conference on Intelligent Technologies (CONIT), Hubli, India, 23–25 June 2023. [CrossRef]
10. Debroy, P.; Seban, L. A Fish Biomass Prediction Model for Aquaponics System Using Machine Learning Algorithms. In *Machine Learning and Autonomous Systems*; Springer: Singapore, 2022. [CrossRef]
11. Udanor, C.N.; Ossai, N.I.; Nweke, E.O.; Ogbuokiri, B.O.; Eneh, A.H.; Ugwuishiwu, C.H.; Aneke, S.O.; Ezugwu, A.O.; Ugwoke, P.O.; Christiana, A. An Internet of Things Labelled Dataset for Aquaponics Fish Pond Water Quality Monitoring System. *Data Brief* **2022**, *43*, 108400. [CrossRef] [PubMed]
12. Roberts, D.R.; Bahn, V.; Ciuti, S.; Boyce, M.S.; Elith, J.; Guillera-Arroita, G.; Hauenstein, S.; Lahoz-Monfort, J.J.; Schröder, B.; Thuiller, W.; et al. Cross-Validation Strategies for Data with Temporal, Spatial, Hierarchical, or Phylogenetic Structure. *Ecography* **2017**, *40*, 913–929. [CrossRef]
13. Kapoor, S.; Narayanan, A. Leakage and the Reproducibility Crisis in Machine-Learning-Based Science. *Patterns* **2023**, *4*, 100804. [CrossRef] [PubMed]
14. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
15. Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations Using RNN Encoder–Decoder for Statistical Machine Translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1724–1734. [CrossRef]
16. Thai-Nghe, N.; Thanh-Hai, N.; Ngon, N.C. Deep Learning Approach for Forecasting Water Quality in IoT Systems. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*, 686–693. [CrossRef]
17. Li, W.; Wu, H.; Zhu, N.; Jiang, Y.; Tan, J.; Guo, Y. Prediction of Dissolved Oxygen in a Fishery Pond Based on Gated Recurrent Unit (GRU). *Inf. Process. Agric.* **2021**, *8*, 185–193. [CrossRef]
18. Putpuek, N.; Putpuek, A.; Sungthong, A. A Comparative Study of LSTM, GRU, BiLSTM and BiGRU to Predict Dissolved Oxygen. In Proceedings of the 2023 20th International Joint Conference on Computer Science and Software Engineering (JCSSE), Phitsanulok, Thailand, 28 –1 July June 2023; pp. 67–72. [CrossRef]
19. Rahul Gandh, D.; Rasheed Abdul Haq, K.P.; Harigovindan, V.P.; Bhide, A. LSTM and GRU Based Accurate Water Quality Prediction for Smart Aquaculture. *J. Phys. Conf. Ser.* **2023**, *2466*, 012027. [CrossRef]
20. Gao, S.; Huang, Y.; Zhang, S.; Han, J.; Wang, G.; Zhang, M.; Lin, Q. Short-Term Runoff Prediction with GRU and LSTM Networks without Requiring Time Step Optimization during Sample Generation. *J. Hydrol.* **2020**, *589*, 125188. [CrossRef]
21. Cardenas-Cartagena, J.; Elnourani, M.; Beferull-Lozano, B. Forecasting Aquaponic Systems Behaviour with Recurrent Neural Networks Models. In Proceedings of the Northern Lights Deep Learning Workshop (NLDL), Tromsø, Norway, 10–12 January 2022; Volume 3. [CrossRef] [PubMed]
22. Rasheed Abdul Haq, K.P.; Harigovindan, V.P. Water Quality Prediction for Smart Aquaculture Using Hybrid Deep Learning Models. *IEEE Access* **2022**, *10*, 60078–60098. [CrossRef]
23. Jongjaraunsuk, R.; Taparhudee, W.; Suwannasing, P. Comparison of Water Quality Prediction for Red Tilapia Aquaculture in an Outdoor Recirculation System Using Deep Learning and a Hybrid Model. *Water* **2024**, *16*, 907. [CrossRef]
24. Eze, E.; Kirby, S.; Attridge, J.; Ajmal, T. Aquaculture 4.0: Hybrid Neural Network Multivariate Water Quality Parameters Forecasting Model. *Sci. Rep.* **2023**, *13*, 16129. [CrossRef] [PubMed]
25. Lim, B.; Arık, S.Ö.; Loeff, N.; Pfister, T. Temporal Fusion Transformers for Interpretable Multi-Horizon Time Series Forecasting. *Int. J. Forecast.* **2021**, *37*, 1748–1764. [CrossRef]

26. Metin, A.; Kasif, A.; Catal, C. Temporal Fusion Transformer-Based Prediction in Aquaponics. *J. Supercomput.* **2023**, *79*, 19934–19958. [CrossRef]
27. Lan, H.-Y.; Ubina, N.A.; Zhang, K.-X.; Cheng, S.-C.; Li, S.-Y. Fusion of Transformer-Based Deep Learning and Monte-Carlo Fish Growth Simulation for Aquaculture Smart Transformation. *Eng. Comput.* **2025**. [CrossRef]
28. Akaike, H. A New Look at the Statistical Model Identification. *IEEE Trans. Autom. Control* **1974**, *19*, 716–723. [CrossRef]
29. Katsanevakis, S. Modelling Fish Growth: Model Selection, Multi-Model Inference and Model Selection Uncertainty. *Fish. Res.* **2006**, *81*, 229–235. [CrossRef]
30. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830. Available online: https://jmlr.org/papers/v12/pedregosa11a.html (accessed on 4 November 2026).
31. Yigit, G.; Amasyali, M.F. Simple but Effective GRU Variants. In Proceedings of the 2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA), Kocaeli, Turkey, 25–27 August 2021; pp. 1–6. [CrossRef]
32. Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A Next-Generation Hyperparameter Optimization Framework. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD), Anchorage, AK, USA, 4–8 August 2019; pp. 2623–2631. [CrossRef]
33. Bergstra, J.; Bardenet, R.; Bengio, Y.; Kégl, B. Algorithms for Hyper-Parameter Optimization. In *Advances in Neural Information Processing Systems 24*; Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., Weinberger, K., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2011. Available online: https://papers.nips.cc/paper_files/paper/2011/hash/86e8f7ab32cfd12577bc2619bc635690-Abstract.html (accessed on 4 November 2026).
34. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. In Proceedings of the International Conference on Learning Representations (ICLR), New Orleans, LA, USA, 6–9 May 2019. [CrossRef]
35. Tofallis, C. A Better Measure of Relative Prediction Accuracy for Model Selection and Model Estimation. *J. Oper. Res. Soc.* **2015**, *66*, 1352–1362, Erratum in *J. Oper. Res. Soc.* **2017**, *66*, 524. [CrossRef]
36. Wilcoxon, F. Individual Comparisons by Ranking Methods. *Biom. Bull.* **1945**, *1*, 80. [CrossRef]
37. Ojala, M.; Garriga, G.C. Permutation Tests for Studying Classifier Performance. *J. Mach. Learn. Res.* **2010**, *11*, 1833–1863. Available online: https://jmlr.org/papers/v11/ojala10a.html (accessed on 4 November 2026).
38. Field, C.A.; Welsh, A.H. Bootstrapping Clustered Data. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **2007**, *69*, 369–390. [CrossRef]
39. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*; Wallach, H., Larochelle, H., Beygelzimer, A., D'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2019. Available online: https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html (accessed on 4 November 2026).
40. Virtanen, P.; Gommers, R.; Oliphant, T.E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nat. Methods* **2020**, *17*, 261–272. Erratum in *Nat. Methods* **2020**, *17*, 352. [CrossRef] [PubMed]

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