Aquaculture America, February 16-19, 2026

Distributed Fish Growth Simulation with Machine Learning-Based Probabilistic Growth Forecasting

Choi Jinseo[1*], Park Jeonghwan[2]

[1)] Electronics and Telecommnuications Research Institute, Daejeon, Republic of Korea

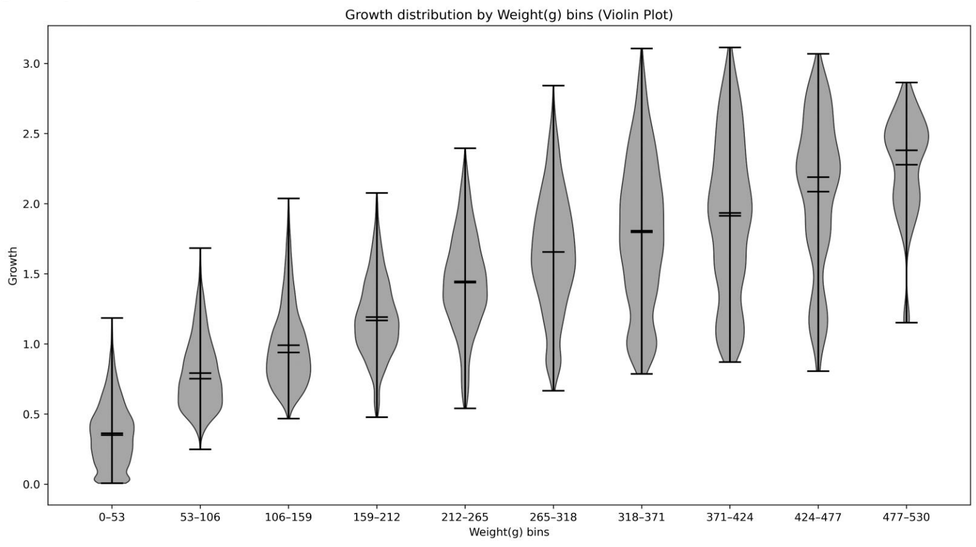
[2)] School of Fisheries, Aquaculture, and Aquatic Sciences, Auburn University, Auburn, AL, USA

Ⅰ. Introduction

Fish populations with the same initial body weight often exhibit substantial individual growth variability, even under identical rearing and feeding conditions. As illustrated by observed growth patterns, individual trajectories diverge over time, forming fast- and slow-growing subpopulations and leading to increasingly wide body weight distributions. This variability is not merely incidental but rather an inherent characteristic of aquaculture systems, which becomes more pronounced as fish grow.

Despite this evident heterogeneity, most conventional growth models rely on average growth rates or deterministic trajectories. Such approaches obscure the underlying distributional structure, failing to capture variance expansion, asymmetry, and extreme growth behaviors that emerge during the production cycle. Consequently, decisions based solely on mean growth estimates offer limited reliability for practical operations, including harvest scheduling, tank transfers, and feed as well as energy demand forecasting.

To support robust decision-making at both the farm and facility levels, growth models must account for uncertainty at the individual level and propagate it over time. Predicting the full probability distribution of growth, rather than a single expected value, enables more realistic planning and risk-aware management. Motivated by these observations, this study proposes a distributed, agent-based fish growth simulation framework that integrates individual-level stochastic growth modeling with machine learning-based probabilistic growth forecasting, thereby enabling distribution-aware growth prediction and operational simulation.



Ⅱ. Data Collection & Preprocessing

Data Source and Scope

Operational data were collected from a commercial Japanese eel (Anguilla japonica) aquaculture farm in Gochang, South Korea, spanning 2018-2021. The dataset comprised 888 rearing cohorts (SETs), each representing an independent production unit from stocking to harvest or transfer. After preprocessing and outlier removal, 795 cohorts yielding 81,876 daily growth records were retained for analysis.

Key variables included body weight (g), pH, water temperature (°C), dissolved oxygen (DO, mg/L), feeding rate (%), and survival rate (%). Environmental conditions were maintained within narrow ranges typical of recirculating aquaculture systems (RAS): temperature 27.5-31.5°C, pH 4.0-6.0, DO 9.5-17.5 mg/L.

Variable	Description	Unit	Data Type
SET	Production cohort	–	Categorical
days	Rearing days	day	Integer
weight	Estimated body weight (SGR-based)	g	Continuous
growth	Daily weight gain	g	Continuous
pH	pH	–	Continuous
Temp	Water temperature	°C	Continuous
DO	Dissolved oxygen	mg/L	Continuous
feedrate	Feeding rate	%	Continuous
survival rate	Survival rate	%	Continuous

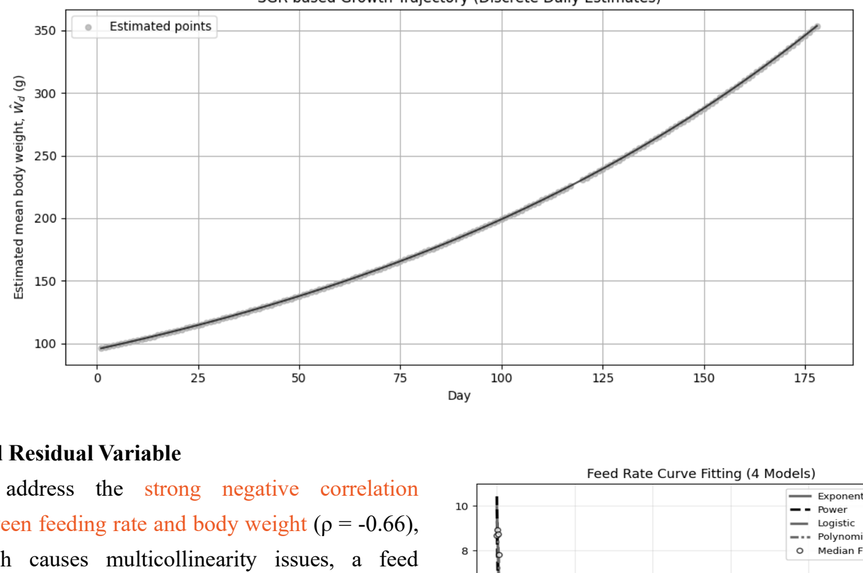
Growth Data Construction

Given the impracticality of daily individual weighing in commercial operations, body weights were estimated daily using Specific Growth Rate (SGR) derived from initial and final measurements recorded at stocking and harvest/transfer. This approach reconstructs continuous growth trajectories from discrete measurements, enabling daily-resolution modeling.

$$SGR(\%/day) = \frac{\ln(W_f) - \ln(W_i)}{T} \times 100$$

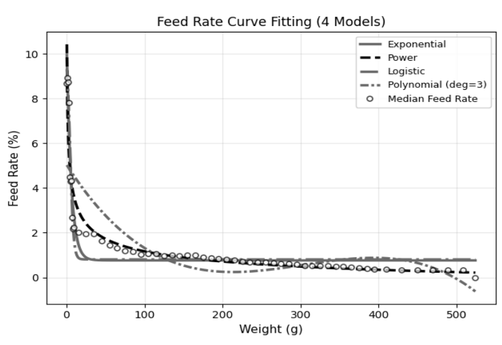
$$W_t = W_i \times \exp\left(\left(\frac{SGR}{100}\right) \times t\right)$$

where W_f and W_i denote mean body weights at harvest and stocking (g), T is the total culture duration (days), and W_t is the estimated weight at day t.



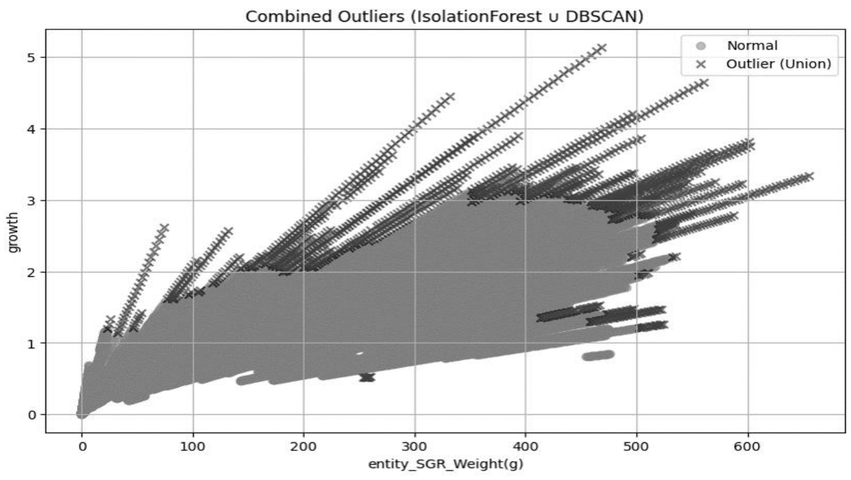
Feed Residual Variable

To address the strong negative correlation between feeding rate and body weight (ρ = -0.66), which causes multicollinearity issues, a feed residual variable (feed_resid) was constructed. A power-law baseline feeding rate was fitted to weight-specific median values, and residuals were calculated as deviations from this baseline. This transformation removes weight dependency while preserving individual-level feeding variability.



Outlier Detection and Removal

Outliers were removed using the union of Isolation Forest and DBSCAN to reduce the influence of sparsely represented observations. Cohorts with daily survival rate drops exceeding 15%, final survival rates below Q1 - 1.5×IQR, or minimum survival Z-scores below -2.5 were removed. Water quality outliers were filtered using IQR-based thresholds (pH: 2×IQR, Temp: 2×IQR, DO: 1.5×IQR). The final dataset represents normal rearing conditions without disease outbreaks or acute environmental stress.



Ⅲ. Model Development

Probabilistic Regression Architectures

Three distributional regression methods were developed to estimate the conditional probability distribution $P(G_t \mid W_t,\ z_t)$ of daily growth G_t given current weight W_t and environmental covariates z_t:

(1) Distributional Regression with Shifted Gamma: An MLP with 2-4 hidden layers (256-512 units) directly estimates shape (α) and scale (β) parameters of a Gamma distribution. Global offset transformation accommodates occasional negative growth. Parameters were optimized via NLL minimization with dropout regularization (0.05-0.20) and early stopping (patience = 20).

$$G_t + c \sim Gamma(\alpha, \beta)$$
$$(\alpha, \beta) = f_\theta(W_t, z_t)$$

(2) Mixture Density Network (MDN): A neural architecture estimating Gaussian mixture parameters—mixing coefficients (π_k), means (μ_k), variances (σ_{k^2}) —with M = 3-6 components to represent multimodal distributions and capture subpopulation heterogeneity. Softmax and softplus activations enforce parameter constraints.

$$p(G_t \mid x) = \sum_{k=1}^{M} \pi_k(x) \mathrm{N}(G_t \mid \mu_k(x), {\sigma_k}^2(x))$$

(3) Natural Gradient Boosting (NGBoost): A gradient boosting framework optimizing Gamma parameters via natural gradients derived from information geometry. Decision trees (max_depth = 3) serve as base learners, with 1,500-3,000 iterations and learning rate 0.03-0.12, efficiently capturing nonlinear relationships while maintaining probabilistic calibration.

$$\theta_{t+1} = \theta_t - \eta \tilde{\nabla}_\theta \mathcal{L}$$

Hyperparameter Optimization

All models underwent rigorous Bayesian hyperparameter optimization using Optuna with Tree-structured Parzen Estimator (TPE) sampling, which constructs probabilistic surrogate models to efficiently guide exploration of the hyperparameter space. The optimization objective minimized validation NLL over 100-200 trials, with median pruning applied to terminate unpromising configurations early and reduce computational overhead.

Hyperparameter	Distributional Regression	MDN	NGBoost
Hidden layer size	256	384	–
Number of hidden layers	3	2	–
Dropout rate	0.198	0.145	–
Learning rate	1.22×10^{-4}	1.81×10^{-4}	0.078
Number of epochs	191	153	–
Batch size	1,024	2,048	–
Number of mixture components	–	6	–
Number of boosting iterations	–	–	3,000

Optimized hyperparameters included neural network architecture specifications (hidden layer dimensions: 256-512 units; depth: 2-4 layers), regularization parameters (dropout rates: 0.05-0.20), training dynamics (learning rates sampled log-uniformly from 10^{-4} to 10^{-2}; batch sizes: 1,024-2,048), and convergence criteria (training epochs: 140-200 with early stopping). For NGBoost, tree-specific parameters including maximum depth (3), boosting iterations (1,500-3,000), and learning rate (0.03-0.12) were similarly optimized. Table 1 summarizes the optimal hyperparameter configurations selected for each model. This systematic procedure ensured fair model comparisons based on architectural differences rather than suboptimal configurations.

Evaluation Metrics

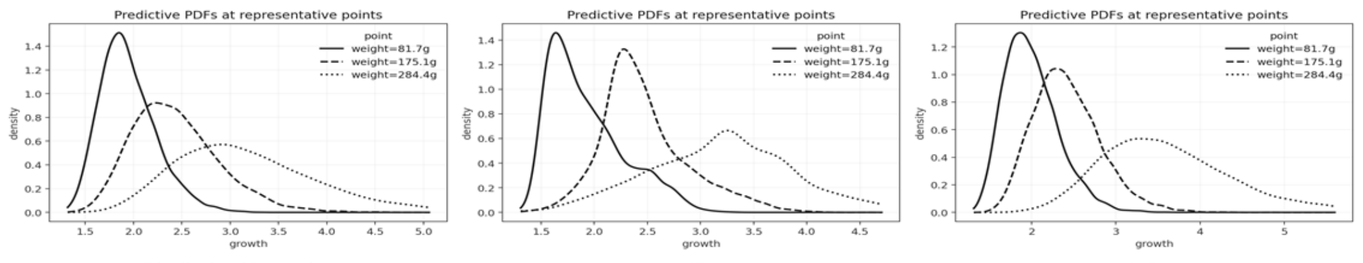
Probabilistic performance was assessed via four complementary metrics. Negative Log-Likelihood (NLL) measures how well the predicted distribution fits observed data, while Continuous Ranked Probability Score (CRPS) quantifies prediction- observation discrepancy by accounting for both location and spread. Coverage indicates the proportion of observations falling within predicted 90% confidence intervals, where ideal coverage equals the nominal level. Probability Integral Transform (PIT) tests distributional calibration by assessing the uniformity of transformed predictions.

Metric	Distributional Regression	MDN	NGBoost
NLL	0.4506	0.4660	0.4374
CRPS	0.2647	0.2604	0.2607
Coverage	0.9106	0.8910	0.9033
Width	1.6326	1.5841	1.6262
PIT (p-value)	< 0.0001	< 0.0001	< 0.0001

Model Performance and Selection

Test set evaluation revealed NGBoost achieved optimal accuracy-calibration balance (NLL = 0.437, CRPS = 0.261, Coverage = 90.3%), outperforming MDN (NLL = 0.466, CRPS = 0.260, Coverage = 89.1%) and Distributional Regression (NLL = 0.451, CRPS = 0.265, Coverage = 91.1%).

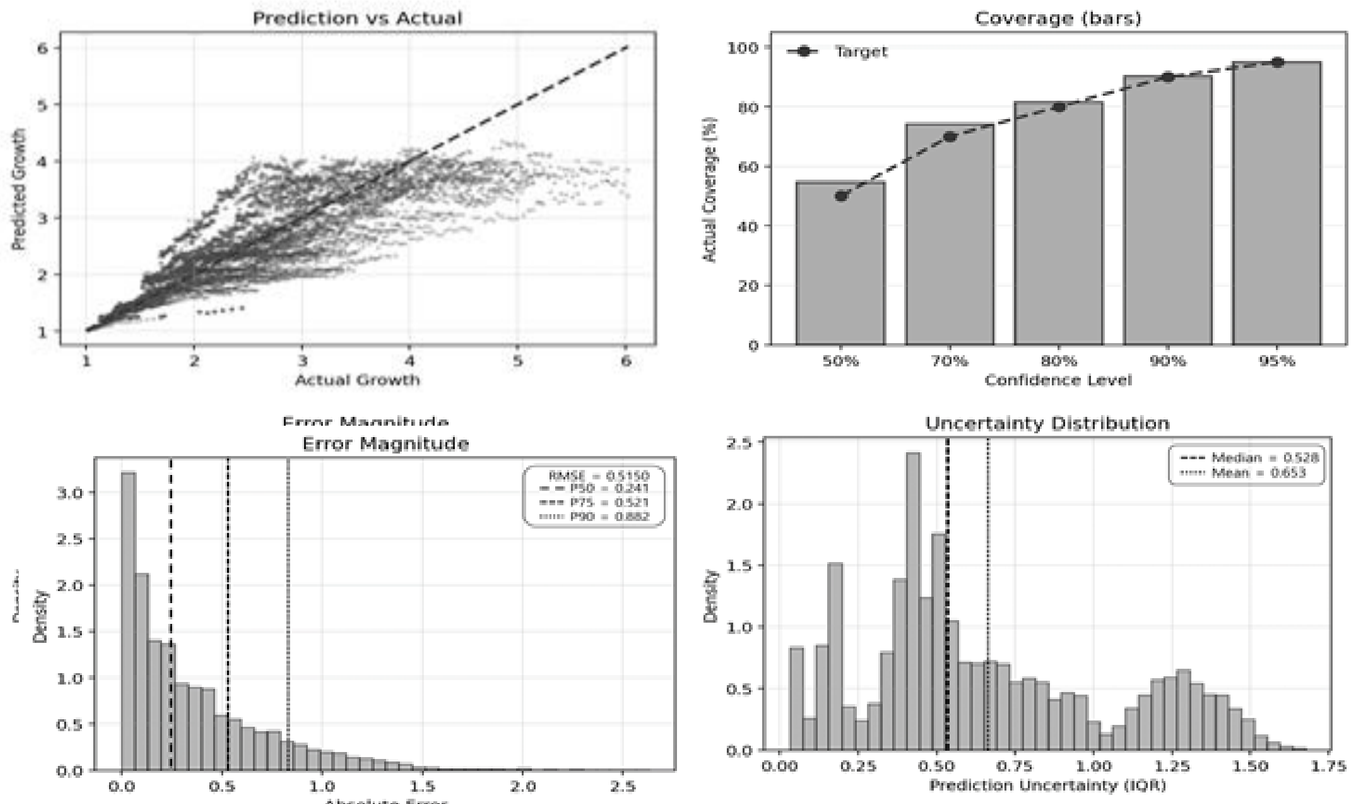
Predicted probability density functions at representative weights (81.7g, 175.1g, 284.4g) confirmed consistent heteroscedasticity capture across models— rightward distribution shifts with increasing variance as weight increased. NGBoost exhibited smooth, consistent distribution tracking and tail behavior. MDN demonstrated high shape flexibility, accurately capturing peaks and asymmetries. Distributional Regression provided a stable, conservative baseline. Based on quantitative and qualitative evaluation, NGBoost was selected for multi-batch simulation.



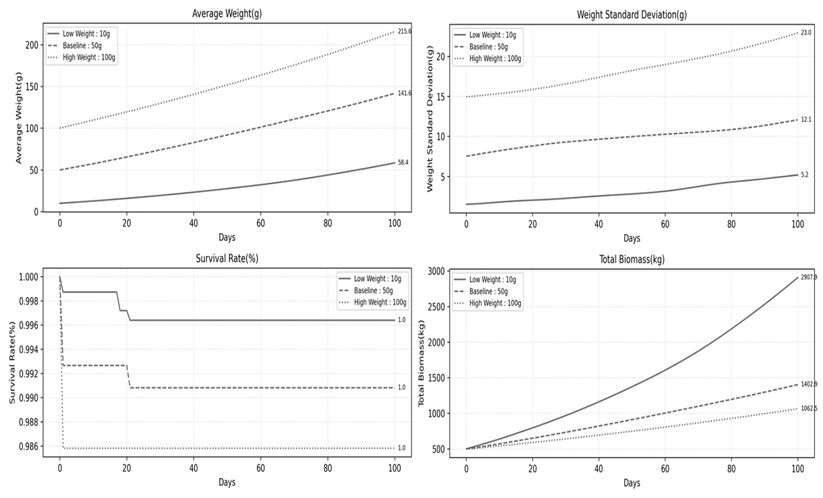
Distributional Regression | MDN | NGBoost

Ⅳ. Results & Conclusion

A detailed reliability assessment was conducted for the selected NGBoost model. As shown in the diagnostic plots, predictions closely align with observations across the full range of daily growth values, without strong systematic bias. Empirical coverage increases monotonically with confidence level, indicating well-calibrated prediction intervals. Most absolute errors remain concentrated within a small range, while the predicted uncertainty distribution reflects realistic heteroscedastic variability rather than overconfident forecasts. Overall, these results support the use of NGBoost as a stable probabilistic growth component for recursive multi-day simulation.



Ⅴ. Application : Recursive Growth Simulation



The selected NGBoost model is applied recursively by updating body weight at each time step and forecasting the conditional next-day growth distribution. This recursive framework enables multi-day simulation of mean growth trajectories, distributional variance, survival dynamics, and aggregate biomass under varying initial stocking conditions. Distribution-aware forecasting thereby supports risk-informed harvest scheduling, tank transfer optimization, and feed as well as energy demand projections. Future extensions will incorporate facility-level operational constraints, including tank capacity and energy availability, enabling end-to-end production optimization under resource limitations.