Annales Mathematicae et Informaticae
62 (2025) pp. 99–115
DOI: 10.33039/ami.2025.10.019
URL: https://ami.uni-eszterhazy.hu

Parameter identification in a simple chemostat model using neural networks

Anes Moulai-Khatir

Institute of Maintenance and Industrial Safety, University of Oran 2, Algeria, and Laboratory of Nonlinear Analysis and Applied Mathematics, University of Tlemcen, Algeria
anes.mkh@gmail.com & moulaikhatir.anes@univ-oran2.dz

Abstract. This paper investigates the use of a neural network approach for parameter estimation in the chemostat model, relevant to applications like wastewater treatment and bioreactor design. Accurate parameter characterization serves as the foundation for understanding system dynamics and making reliable predictions. Traditional optimization-based methods face challenges such as noise and high-dimensional data. Neural networks offer a promising alternative due to their ability to handle complex datasets. The work applies a simple neural network model, demonstrating its effectiveness for estimating chemostat parameters. While advanced techniques like neural architecture search (NAS) are not included, the approach provides a practical solution for parameter identification in dynamic models.

Keywords: chemostat model, neural network, parameter identification

2020 *Mathematics Subject Classification:* 92D25, 34C60, 68T07, 92B20

1. Introduction

Parameter estimation plays a pivotal role in understanding and predicting the dynamics of biological systems. Among these, the chemostat model stands out as a foundational framework for studying the controlled growth of microorganisms, widely applied in microbial ecology, biotechnology, and environmental sciences [11, 13]. Despite its simplicity, the chemostat model is characterized by nonlinearity and multiple interacting parameters, thus making accurate parameter identification a challenging task [2, 6].

In recent years, neural networks have gained significant attention as powerful tools for solving parameter estimation problems. Unlike traditional approaches,

Submitted: December 26, 2024
Accepted: October 19, 2025
Published online: October 19, 2025

which often rely on linearization or gradient-based optimization, neural networks can capture complex nonlinear relationships in data, even in the presence of noise or sparsity [7, 8]. This ability positions them as a robust alternative for tackling the inherent challenges of parameter identification in systems like the chemostat model [4, 9].

This study aims to develop a neural network-based approach for parameter estimation in a simple chemostat model. By leveraging synthetic data generated from the model, we explore the effectiveness of neural networks in recovering parameters under various scenarios, including the presence of noise [1, 12].

The integration of machine learning, particularly neural networks, into biological modeling has shown significant potential in related fields such as metabolic pathway reconstruction [4], gene regulatory network inference [10], and ecosystem modeling [9].

2. Mathematical model

The chemostat model describes the interaction between microorganisms and the substrate in a well-mixed environment with continuous input and output. This section introduces the mathematical formulation of the chemostat system, outlining its key variables, parameters, and governing differential equations [13]. The focus will be on the biological significance of these components and their roles in determining system dynamics. We also briefly review the assumptions underlying the model, such as constant dilution rates and perfect mixing, which are critical for its applicability. The chemostat model we consider is derived under the following assumptions:

- The culture vessel has a constant volume, denoted by V.
- The inflow and outflow rates of the medium are equal and characterized by a constant flow rate F.
- Microbial growth is dependent on the uptake of a single limiting nutrient, modeled using Monod kinetics [11].

2.1. Governing equations

We consider the following variables and parameters to describe the dynamics of the chemostat system:

- $S(t)$: Concentration of the limiting nutrient at time t (units: g/L).
- $x(t)$: Concentration of the microorganism population at time t (units: g/L).
- S_0: Concentration of the nutrient in the inflowing medium (units: g/L).
- $D = \frac{F}{V}$: Dilution rate, defined as the ratio of the flow rate F to the reactor volume V (units: h^{-1}).

- m: Maximum specific growth rate of the microorganism (units: h^{-1}).

- a: Half-saturation constant, representing the nutrient concentration at which the growth rate is half its maximum (units: g/L).

- γ: Yield coefficient, a dimensionless parameter reflecting the conversion efficiency of the nutrient into biomass.

The dynamics of the system are governed by a set of ordinary differential equations. The evolution of the nutrient concentration $S(t)$ is determined by nutrient inflow, outflow, and microbial uptake, expressed as:

$$S'(t) = (S_0 - S(t))D - g(S(t), x(t)),$$

where $g(S, x)$ denotes the nutrient consumption rate. Using the Monod kinetic model, the consumption rate is given by:

$$g(S, x) = \frac{mS}{a + S}\frac{x}{\gamma}.$$

Substituting $g(S, x)$ into the nutrient balance equation, we obtain:

$$S'(t) = (S_0 - S)D - \frac{mS}{a + S}\frac{x}{\gamma}.$$

The dynamics of the microorganism concentration $x(t)$ are governed by microbial growth and washout:

$$x'(t) = \left(\frac{mS}{a + S}\right)x - Dx.$$

Combining these equations, the full system of governing equations is:

$$\begin{cases} S'(t) = (S_0 - S)D - \frac{mS}{a+S}\frac{x}{\gamma}, \\ x'(t) = \left(\frac{mS}{a+S}\right)x - Dx. \end{cases}$$

This system captures the interaction between nutrient availability and microorganism growth in a chemostat.

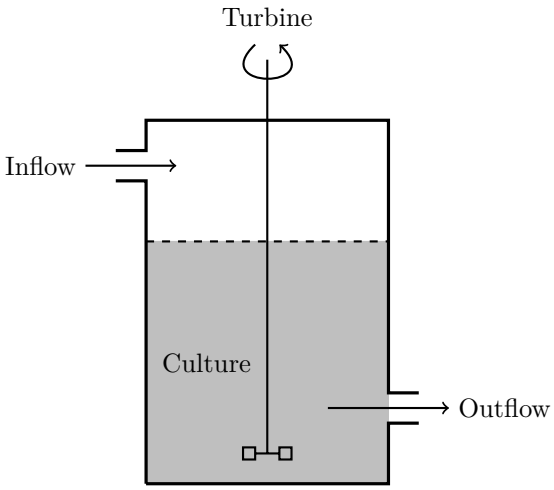

Figure 1. Schematic of the chemostat reactor.

2.2. Rescaling the system

To simplify the analysis and reduce the number of parameters, the system is nondimensionalized using appropriate scalings. This process eliminates redundant units and focuses on the essential dynamics, making the model more tractable for analysis and parameter estimation. For simplicity, we retain the original symbols t, S, and x for the rescaled time, nutrient concentration, and microorganism concentration, respectively.
We introduce the following dimensionless variables and parameters:

$$S = S_0 s, \quad X = \gamma S_0 x, \quad t = \frac{\tau}{D}, \quad A = \frac{a}{S_0}, \quad M = \frac{m}{D}.$$

Substituting these scalings into the original equations and dividing through by appropriate factors, the system transforms into:

$$\begin{cases} \frac{dS}{dt} = 1 - S - \frac{mSx}{a+S}, \\ \frac{dx}{dt} = x\left(\frac{mS}{a+S} - 1\right), \end{cases}$$

with initial conditions: $S(0) \geq 0$ and $x(0) \geq 0$.

This rescaled system eliminates the dependence on S_0 and D, simplifying the equations and emphasizing the key parameters m (maximum growth rate) and a (half-saturation constant). This version will be used for further analysis.

Remark 2.1. This process involved nondimensionalizing the equations, which altered the numerical values and interpretation of the parameters m and a. For consistency and ease of interpretation, we continue to refer to m as the maximal growth rate and a as the half-saturation constant in their original biological context.

2.3. Analysis of the rescaled system

This section reviews existing theoretical results on the stability of the chemostat model under rescaled variables. Key equilibria and their stability properties are discussed, along with conditions for the persistence or extinction of the microorganism population. These results serve as a foundation for understanding the behavior of the system, guiding the parameter estimation process. The rescaled system simplifies the analysis by focusing on the equilibrium behavior. We can compute the equilibrium points for this rescaled system. At equilibrium, we have:

- $S' = 0$, which implies that the nutrient concentration remains constant:

$$1 - S - \frac{mSx}{a+S} = 0.$$

- $x' = 0$, which means that the microorganism population remains constant:

$$x\left(\frac{mS}{a+S} - 1\right) = 0.$$

  From the second equation, we find two possible equilibrium points for x:

  - $x = 0$ (extinction of the microorganism),
  - $\frac{mS}{a+S} = 1$, which gives $S = \frac{a}{m-1}$ for $m > 1$. This is the *break-even concentration* $\lambda = \frac{a}{m-1}$.

Thus, the equilibrium points are:

$$(S_0, x_0) = (1, 0) \quad \text{and} \quad (S_1, x_1) = (1 - \lambda, \lambda).$$

The stability of these equilibrium points depends on the values of m and λ:

- If $m < 1$, the microorganism population is washed out ($x \to 0$) because the maximum growth rate is less than the dilution rate.
- If $m > 1$, and $\lambda \geq 1$, the microorganism population is washed out because insufficient nutrient is available to sustain the growth ($x \to 0$).
- If $0 < \lambda < 1$, the microorganism population reaches a stable equilibrium at $x = 1 - \lambda$, and this is the long-term stable population.

This result is intuitive: if the maximum growth rate of the microorganism is less than the dilution rate, the microorganism population will be washed out. If the maximum growth rate is greater than the dilution rate, and the nutrient concentration is sufficient, the microorganism population will stabilize at a positive equilibrium value determined by the break-even concentration.

3. Numerical simulation and synthetic data generation

Numerical simulations are fundamental to studying the chemostat model and generating synthetic data for machine learning-based parameter estimation. Below, we describe the methods and processes involved.

3.1. Numerical simulation of the dynamics

To approximate the solutions of the chemostat model's ordinary differential equations (ODEs), we employ the *Euler method*, a simple and efficient numerical scheme. The discrete-time update rules are given by:

$$S[t] = S[t-1] + \Delta t \cdot \frac{dS}{dt},$$
$$x[t] = x[t-1] + \Delta t \cdot \frac{dx}{dt},$$

where Δt is the time step, and $\frac{dS}{dt}$, $\frac{dx}{dt}$ are derived from the governing equations of the model.

Simulation steps

1. **Initialization**: Set initial conditions $S(0)$ and $x(0)$, and define model parameters m and a.

2. **Iterative Computation**: Apply the Euler update rules at each time step, evaluating the rates of change and updating the state variables.

Algorithm 1 Euler method for chemostat dynamics simulation.

procedure EULERSIMULATE($S_0, x_0, a, m, n\_steps, dt$)
  Initialize state vectors S, x of length n_steps
  $S[0] \leftarrow S_0, x[0] \leftarrow x_0$
  for $t = 1$ to $n\_steps - 1$ **do**
    Compute substrate rate of change:
    $dS/dt \leftarrow 1 - S[t-1] - \frac{m \cdot S[t-1] \cdot x[t-1]}{a + S[t-1]}$
    Compute microorganism rate of change:
    $dx/dt \leftarrow x[t-1] \cdot \left(\frac{m \cdot S[t-1]}{a + S[t-1]} - 1 \right)$
    Update state variables with non-negative constraint:
    $S[t] \leftarrow \max(0, S[t-1] + dt \cdot dS/dt)$
    $x[t] \leftarrow \max(0, x[t-1] + dt \cdot dx/dt)$
  end for
  return S, x
end procedure

3.2. Generating synthetic data

To emulate real-world experimental scenarios, we generate synthetic data with added noise to simulate measurement inaccuracies. This data serves as input for the machine learning model used for parameter estimation.

Steps for data generation

1. **Simulate True Dynamics**: Solve the chemostat ODEs numerically over a defined time period to obtain S_{true} and x_{true}.

2. **Introduce Noise**: Add Gaussian noise to mimic sensor inaccuracies:

$$S_{\text{noisy}} = S_{\text{true}} + \mathcal{N}(0, \sigma_S), \quad x_{\text{noisy}} = x_{\text{true}} + \mathcal{N}(0, \sigma_x),$$

   where σ_S and σ_x are noise levels representing measurement precision.

3. **Downsample**: Simulate discrete sampling intervals by selecting every k-th data point, reducing temporal resolution.

Algorithm 2 Synthetic Data Generation with Controlled Noise

procedure GENERATESYNTHETICDATA($S_0, x_0, a, m, n\_steps, dt, noise\_level$)
  $(S_{true}, x_{true}) \leftarrow$ EULERSIMULATE($S_0, x_0, a, m, n\_steps, dt$)
  $noise\_S \leftarrow \text{Normal}(0, noise\_level \cdot |S_{true}| + \epsilon)$
  $noise\_x \leftarrow \text{Normal}(0, noise\_level \cdot |x_{true}| + \epsilon)$
  $S_{noisy} \leftarrow S_{true} + noise\_S$
  $x_{noisy} \leftarrow x_{true} + noise\_x$
  return $S_{noisy}, x_{noisy}, S_{true}, x_{true}$
end procedure

Remark 3.1. Our results demonstrate parameter estimation performance on synthetic data generated from the same model class used for estimation, representing an idealized scenario. While this controlled approach isolates algorithmic performance, real experimental systems present additional challenges: measurement artifacts (e.g., sensor drift), ecological complexities (e.g., multiple microbial strains [5]), and non-stationary operating conditions. The robustness and generalization tests partially address these concerns, indicating that neural networks may offer practical advantages in noisy environments but remain sensitive to discrepancies between assumed and actual system dynamics.

4. Neural network for parameter estimation

In the mathematical modeling of biological systems, estimating parameters accurately from available data is essential for reliable predictions. The effectiveness of

a model depends largely on correctly identifying these parameters. While traditional optimization techniques often assume a specific functional form or exploit convexity, such assumptions may not be valid in complex biological settings. Neural networks, with their universal approximation property, offer a flexible approach to modeling nonlinear relationships without strict parametric assumptions. By learning directly from data distributions, neural networks capture the underlying system dynamics implicitly, making them especially suitable for intricate models like the chemostat, where nonlinear interactions play a central role.

4.1. Neural network architecture

The architecture of the neural network used for parameter estimation is designed to balance simplicity and expressiveness. A standard feedforward neural network was employed, with input nodes corresponding to features derived from synthetic data generated by the chemostat model. The network consists of multiple hidden layers, each utilizing a sufficient number of neurons to capture nonlinear dependencies in the data. Activation functions such as ReLU were chosen to introduce nonlinearity, allowing the network to model complex interactions between parameters. The output layer is structured to provide direct estimates of the parameters a and m, representing the key quantities in the chemostat model.

The model is constructed using TensorFlow/Keras [3] and consists of:

- **Input Layer**: Takes the noisy synthetic data S_{noisy} and x_{noisy}, flattened into a 1D array of size $2 \times n$, where n is the number of time steps.
- **Hidden Layers**: Three fully connected layers with ReLU activation functions:
  - First layer: 128 neurons.
  - Second layer: 64 neurons.
  - Third layer: 32 neurons.

  These layers capture complex nonlinear relationships between the input data and the parameters m and a.
- **Output Layer**: Two neurons representing the predicted values of m and a.

Remark 4.1. One should note that, in this work, we employed a feedforward neural network for parameter identification in the chemostat system, which involves time-dependent data. To address this, the entire time series of substrate and microorganism concentrations was concatenated into a single input vector. This design allows the network to implicitly capture temporal patterns through nonlinear transformations, despite lacking explicit temporal processing capabilities. We selected this straightforward approach to enable the network to learn complex nonlinear relationships from the data while maintaining simplicity in the model structure.

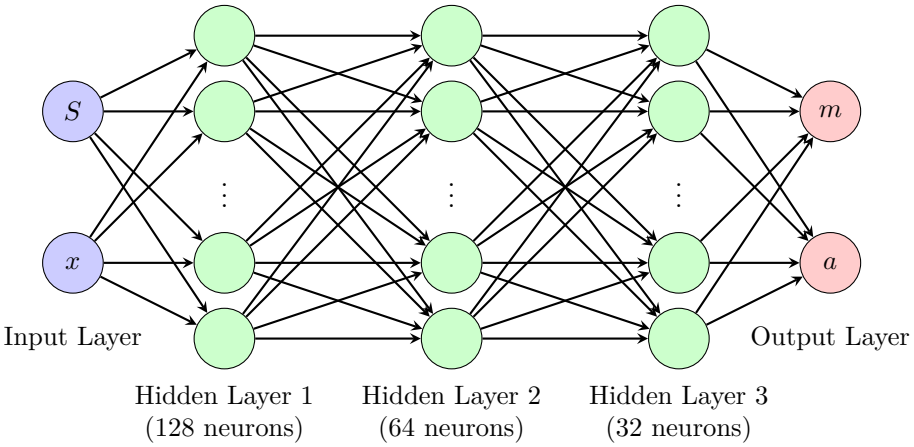

Figure 2. Schematic of the used neural network.

4.2. Training process

The network is trained on noisy synthetic data with the following configuration:

- **Loss Function**: The Mean Squared Error (MSE) is used to evaluate the accuracy of the neural network's parameter predictions by comparing the true and estimated state variable dynamics.
  - $S_{\text{true}}(t)$ and $x_{\text{true}}(t)$: The true values of the substrate concentration and microbial population, respectively, derived by solving the chemostat model using the true parameter values a_{true} and m_{true}.
  - $S_{\text{est}}(t)$ and $x_{\text{est}}(t)$: The estimated values of S and x, obtained by solving the chemostat model using the neural network's predicted parameters a_{est} and m_{est}.

  The MSE for each state variable is defined as:

$$\text{MSE}_S = \frac{1}{N}\sum_{i=1}^{N}(S_{\text{true}}(t_i) - S_{\text{est}}(t_i))^2,$$

$$\text{MSE}_x = \frac{1}{N}\sum_{i=1}^{N}(x_{\text{true}}(t_i) - x_{\text{est}}(t_i))^2.$$

  The total MSE combines the errors across both state variables:

$$\text{MSE} = \text{MSE}_S + \text{MSE}_x,$$

  where N is the total number of time points considered in the simulation.

  This total MSE provides a quantitative measure of the discrepancy between the true and estimated dynamics, reflecting the accuracy of the parameter estimation.

- **Optimizer**: Adam, for adaptive and efficient gradient descent.
- **Data Split**: Divide synthetic data into training and validation sets.
- **Epochs**: Train over 100 epochs using backpropagation to minimize the loss function.

Algorithm 3 Chemostat parameter estimation.

procedure ParameterEstimation($S_0, x_0, true\_a, true\_m, n\_steps, dt, noise\_level$)
    ($S_{noisy}, x_{noisy}, S_{true}, x_{true}$) ← GenerateSyntheticData($S_0, x_0, true\_a, true\_m, n\_steps, dt, noise\_level$)
    $X_{train} \leftarrow$ Concatenate(S_{noisy}, x_{noisy})
    $y_{train} \leftarrow [true\_a, true\_m]$
    $model \leftarrow$ CreateNNModel($input\_size$)
    (a_{est}, m_{est}) ← $model.predict(X_{train})$
    (S_{est}, x_{est}) ← EulerSimulate($S_0, x_0, a_{est}, m_{est}, n\_steps, dt$)
    return a_{est}, m_{est}
end procedure

5. Simulation results

To evaluate the performance of the neural network in estimating the parameters of the chemostat model, two synthetic datasets were generated, each corresponding to a distinct scenario. Both cases were constructed using the same initial conditions and data generation process but differ in the true values of the microbial growth rate.

5.1. Case 1: Washout Scenario

In this scenario, the system parameters are set to induce a washout condition, where the microbial population gradually declines to zero, while the substrate concentration stabilizes at its inflow value. The true parameter values selected for the simulations were $a_{\text{true}} = 0.5$ (half-saturation constant) and $m_{\text{true}} = 1.2$ (maximum growth rate).

The initial conditions were set as $S(0) = 0.1$ and $x(0) = 0.1$, reflecting low initial concentrations of both substrate and microorganisms. The simulations spanned a total duration of $T = 20$ units, with the time domain divided into $n_{\text{steps}} = 200$ steps, resulting in a uniform time step size of $\Delta t = 0.1$.

To replicate the uncertainties and variability inherent in experimental data, Gaussian noise with a standard deviation of 0.01 was added to the simulated measurements. This incorporation of noise evaluates the model's robustness and reliability in predicting system behavior under realistic, noisy conditions.

Under these parameter values, the microbial growth rate is insufficient to sustain the population, leading to a washout condition where the microorganism is eradicated from the reactor over time.

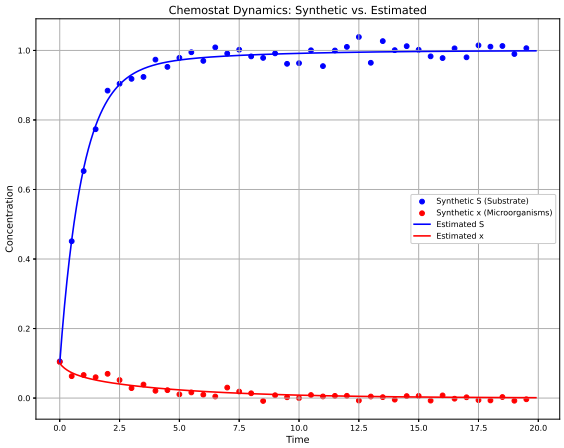

Figure 3. True vs Estimated dynamics of $S(t)$ and $x(t)$ in the washout case.

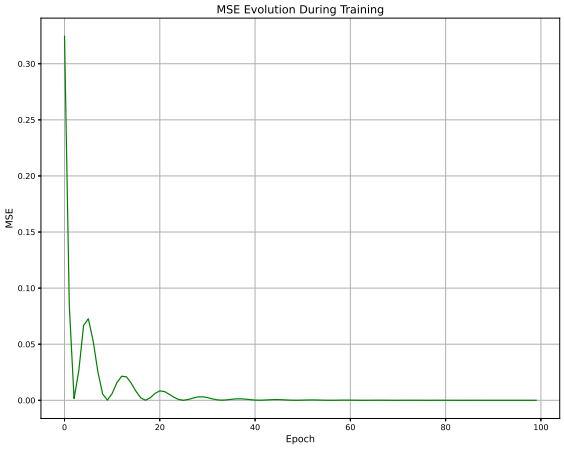

Figure 4. Convergence of Mean Squared Error (MSE) over training epochs in the washout case.

5.2. Case 2: Coexistence Scenario

This scenario represents a more favorable condition for microbial growth, resulting in a stable coexistence of the microbial population and the substrate. The parameter values chosen for this case were $a_{\text{true}} = 0.5$ (half-saturation constant) and $m_{\text{true}} = 2.2$ (maximum growth rate). The initial conditions for the state variables were set as $S(0) = 0.1$ and $x(0) = 0.1$.

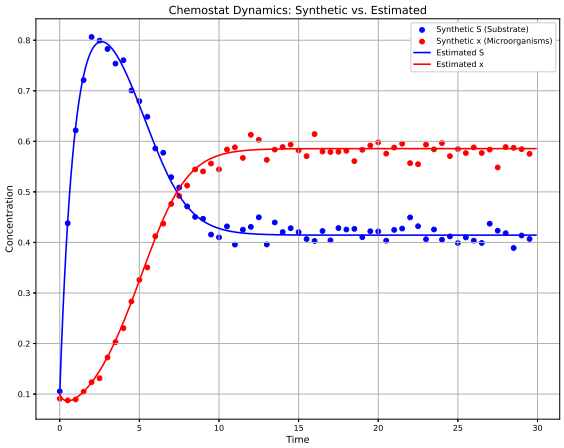

Figure 5. True vs Estimated dynamics of $S(t)$ and $x(t)$ in the microorganism existence case.

The simulation was performed over a total time span of $T = 30$ units, with the time interval discretized into $n_{\text{steps}} = 300$ steps, resulting in a time step size of $\Delta t = 0.1$.

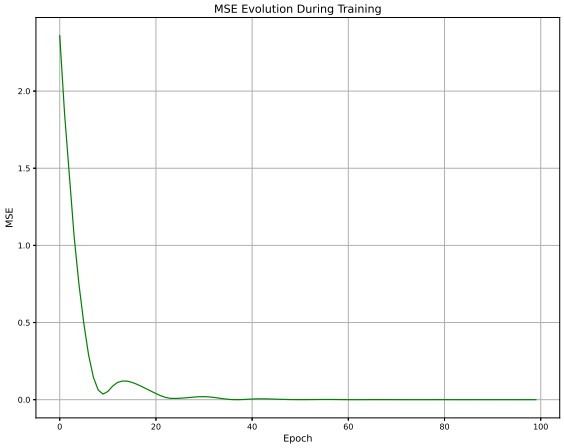

Figure 6. Convergence of MSE over training epochs in the microorganism existence case.

The parameter values enable the microbial population to grow and stabilize at a positive equilibrium, resulting in coexistence between the microorganism and the substrate.

5.3. Parameter estimation results

To evaluate the performance of the neural network in estimating parameters, the true values of a and m were randomly generated within predefined intervals. Specifically, a was sampled uniformly from the interval $(0.1, 1)$, representing plausible half-saturation constants, while m was sampled from the interval $(0.5, 2.5)$, reflecting realistic ranges for maximum specific growth rates in the chemostat model.

The neural network demonstrated consistent accuracy across various test cases, with estimated values closely aligning with the true parameters. These results validate the robustness of the neural network approach, even under conditions of added noise and sparsity in the synthetic data.The small mean squared error values reported indicate the network's capacity to generalize and accurately recover parameters across diverse scenarios. Table 1 summarizes the results, including the Mean Squared Error (MSE) for each run.

Table 1. Summary of parameter estimation results, including Mean Squared Error (MSE).

Set	True a	True m	Estimated a	Estimated m	MSE
1	0.4371	2.4014	0.4338	2.4017	1.556×10^{-5}
2	0.7588	1.6973	0.7578	1.6982	2.246×10^{-7}
3	0.2404	0.8120	0.2407	0.8124	1.150×10^{-11}
4	0.1523	2.2324	0.1504	2.2272	2.234×10^{-6}
5	0.6410	1.9161	0.6376	1.9187	6.247×10^{-5}
6	0.1185	2.4398	0.1203	2.4464	1.565×10^{-6}
7	0.8492	0.9247	0.8479	0.9240	3.133×10^{-12}
8	0.2636	0.8668	0.2632	0.8669	3.031×10^{-10}
9	0.3738	1.5495	0.3745	1.5497	1.538×10^{-6}
10	0.4888	1.0825	0.4869	1.0807	7.052×10^{-11}

5.4. Comparative analysis: Neural network and least squares

In this part we evaluate the performance of the neural network compared to the least squares (LS) method for parameter estimation in the treated chemostat model under ideal conditions. Both approaches were tested on synthetic data generated from the true system dynamics with Gaussian noise, focusing on estimating parameters a and m. The LS method minimizes the sum of squared residuals between simulated and observed trajectories using iterative optimization, while the NN leverages direct mapping from data to parameters after training.

Results (Figure 7, Table 2) demonstrate the NN's superior accuracy, achieving exact matches ($a = 0.4$, $m = 2.3$) compared to LS estimates ($a = 0.42$, $m = 2.34$). The NN's inferred trajectories align closely with ground truth, whereas LS exhibits deviations in biomass dynamics, likely due to sensitivity to noise or local optima. Additionally, the NN provides faster inference post-training, avoiding the computational cost of iterative optimization.

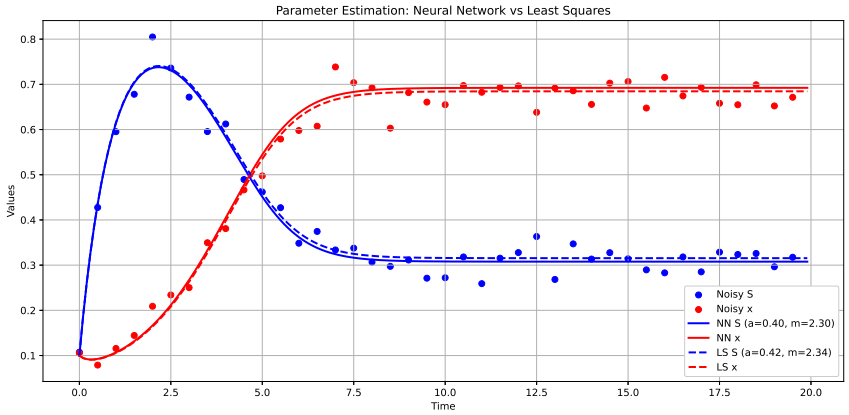

Figure 7. Comparison of reconstructed trajectories using parameter estimates from the Neural Network and Least Squares methods.

Table 2. True Values: $a = 0.4$ and $m = 2.3$.

Method	**Estimated a**	**Estimated m**
Neural Network	0.40	2.30
Least Squares	0.42	2.34

Table 3. Comparison of parameter estimates: Neural Network vs Least Squares method.

Set	**True Params**		**NN Estimates**		**LS Estimates**	
	a	m	a	m	a	m
1	0.4371	2.4014	**0.4367**	**2.3980**	0.4575	2.4554
2	0.7588	1.6973	**0.7559**	**1.6955**	0.6472	1.5900
3	0.2404	0.8120	**0.2399**	**0.8119**	0.2128	0.8184
4	0.1523	2.2324	**0.1534**	**2.2343**	0.1511	2.2256
5	0.6410	1.9161	**0.6375**	**1.9174**	0.6607	1.9420
6	0.1185	2.4398	0.1133	**2.4349**	**0.1167**	2.4207
7	0.8492	0.9247	**0.8484**	**0.9215**	0.0752	0.3958
8	0.2636	0.8668	**0.2637**	**0.8671**	0.0170	0.4909
9	0.3738	1.5495	**0.3722**	**1.5487**	0.3938	1.5756
10	0.4888	1.0825	**0.4891**	**1.0810**	0.1526	0.7629

A broader evaluation across ten parameter sets (Table 3) reveals the NN's consistency: estimates remain within 10^{-2}-10^{-4} relative error of true values. In contrast, LS fails in critical cases (e.g., Set 7, 8, 10), producing unrealistic estimates (e.g., $a = 0.017$ vs. true 0.2636), highlighting its vulnerability to noise and parameter identifiability challenges. These findings underscore the NN's robustness and

reliability in idealized scenarios.

5.5. Discussing the model mismatch case

To assess the robustness of the parameter estimation technique proposed in this work, we consider a more realistic scenario in which the data-generating process deviates slightly from the assumed model. In particular, we introduce an inhibition term in the true biomass dynamics of the form $-\delta x^2$, where $\delta = 0.02$. Despite this structural discrepancy, the NN outperforms LS in capturing trajectory trends (Figure 8, Table 4), with smaller parameter deviations ($a = 0.30$, $m = 1.80$ vs. LS: $a = 0.32$, $m = 1.83$). LS compensates for model inadequacy by distorting parameter values, whereas the NN's flexible architecture adapts to partial mismatches, preserving trajectory fidelity.

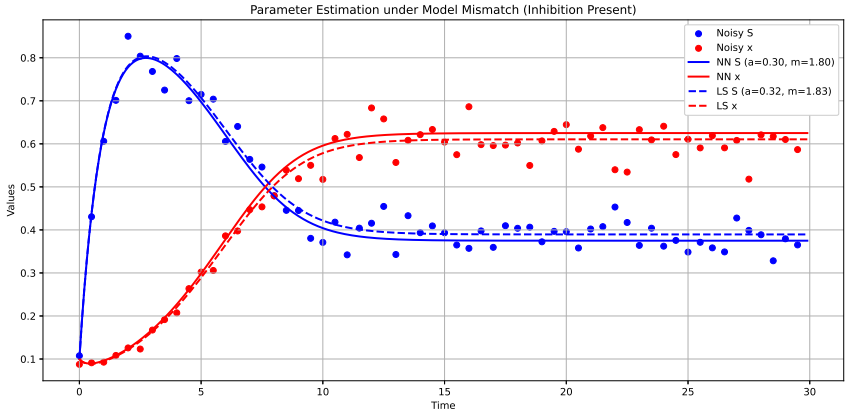

Figure 8. Reconstructed trajectories using parameter estimates from Neural Network and Least Squares methods under model mismatch.

Table 4. True Values: $a = 0.3$ and $m = 1.8$.

Method	**Estimated a**	**Estimated m**
Neural Network	0.30	1.80
Least Squares	0.32	1.83

This resilience is critical in biological applications, where mechanistic models often omit complex interactions (e.g., microbial inhibition, unmodeled species). The NN's ability to generalize across noisy, imperfectly specified systems makes it a pragmatic alternative to LS, particularly when rapid inference or iterative optimization is impractical. Further, its performance aligns with real-world demands:

low MSE (10^{-5}-10^{-12}) confirms accurate dynamic reconstruction, even under perturbations.

However, the neural network's performance depends on the quality and quantity of the training data, which in this study consisted of synthetic data. In a future work, we could focus on validating the approach with real experimental data, scaling to higher-dimensional systems, and integrating hybrid mechanistic-machine learning frameworks. Despite these challenges, neural network-based estimation shows strong potential as a powerful tool for ecological modeling, enabling accurate parameter identification in scenarios where traditional methods may struggle.

6. Conclusion

This study shows that neural networks can offer a practical and powerful way to estimate key parameters in chemostat models-capturing the dynamics of microbial systems even when the data is noisy or incomplete. Compared to traditional methods, neural networks better handle nonlinearities and unexpected outliers, opening new doors for researchers working in complex or unpredictable environments.

While our results on synthetic data are promising, the real value of this approach will come from testing it with real microorganisms in the lab. We encourage the next steps to involve controlled experiments with model organisms like E. coli, directly comparing neural networks with established estimation techniques. Incorporating uncertainty quantification and adaptive learning will be important for dealing with the natural variability that comes with biological systems. By moving from computational models to hands-on experiments, we can help bridge the gap between data science and bioprocess engineering-potentially transforming how bioreactors are optimized, both in research and industry.

Overall, we believe this methodology provides a strong foundation for future work in AI-driven biological modeling. There is significant potential to extend these techniques to more complex microbial communities and to real-time industrial bioprocess monitoring. Continued collaboration across computational and experimental disciplines will be key to accelerating innovation toward more sustainable bioproduction and a deeper understanding of microbial ecology.

Acknowledgements

The author is grateful to the Editor and the anonymous referees for their valuable comments and suggestions, which have greatly improved the quality of this paper.

References

[1] C. M. Bishop: *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

[2] F. Campillo, C. Fritsch: *A modeling approach of the chemostat*, ArXiv preprint (2014), arXiv: 1405.7963.

[3] F. Chollet: *Deep Learning with Python*, Manning Publications, 2017.

[4] Z. Costello, H. G. Martin: *A machine learning approach to predict metabolic pathway dynamics from time-series multiomics data*, NPJ Systems Biology and Applications 4 (2018), p. 19.

[5] K. Faust, J. Raes: *Microbial interactions: from networks to models*, Nature Reviews Microbiology 10 (2012), pp. 538–550.

[6] A. G. Fredrickson, D. Ramkrishna, H. M. Tsuchiya: *Statistics and dynamics of procaryotic cell populations*, Mathematical Biosciences 1.3 (1967), pp. 327–374.

[7] I. Goodfellow, Y. Bengio, A. Courville: *Deep Learning*, MIT Press, 2016.

[8] K. Hornik, M. Stinchcombe, H. White: *Multilayer feedforward networks are universal approximators*, Neural Networks 2.5 (1989), pp. 359–366.

[9] A. Karpatne, G. Atluri, J. H. Faghmous, M. Steinbach, A. Banerjee, A. Ganguly, S. Shekhar, N. Samatova, V. Kumar: *Theory-Guided Data Science: A New Paradigm for Scientific Discovery from Data*, IEEE Transactions on Knowledge and Data Engineering 29.10 (2017), pp. 2318–2331.

[10] X. Li, H. Gao, Y. Zhang: *Applications of deep learning in understanding gene regulation*, Cell Reports Methods 3.1 (2023), p. 100384.

[11] J. Monod: *The growth of bacterial cultures*, Annual Review of Microbiology 3 (1949), pp. 371–394.

[12] D. E. Rumelhart, G. E. Hinton, R. J. Williams: *Learning representations by back-propagating errors*, Nature 323 (1986), pp. 533–536.

[13] H. L. Smith, P. Waltman: *The Theory of the Chemostat: Dynamics of Microbial Competition*, Cambridge University Press, 1995.