Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Fabric Drying Prediction Using Physical Kinetics and Neural Parameter Learning

QIYONG YANG

iloveu13141212@163.com

Donghua University https://orcid.org/0009-0009-2446-9333

Research Article

Keywords: Drying Kinetics, Multilayer Perceptron, Parameter Learning, Validation Across Operating Conditions, Applied Computing

Posted Date: November 3rd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11593332/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Fabric Drying Prediction Using Physical Kinetics and Neural Parameter Learning

Qiyong Yang

Glorious Sun School of Economics and Management, Donghua University, Shanghai, China, 200051.

Abstract: Physical equations provide an interpretable temporal structure for drying, while variations in operating conditions are represented through parameter relationships. This study examines whether a small multi-layer perceptron (MLP) can improve Newton–Lewis predictions for unseen operating conditions. The dataset comprises 14665 time points from 79 experiments covering 27 combinations of *temperature, dry-air mass flow rate* and *humidity ratio*. A capacity parameter is calculated for each experiment from its initial and reference masses. A log-linear map and a feedforward neural network share the same exponential drying equation and are calibrated using normalized curve errors. In 27-fold leave-one-condition-out validation, the neural network reduces the median curve root mean square error (RMSE) from 279.63 to 254.48 g and the *median absolute error* in the time required to remove 90% of the reference removable water from 59.51 to 51.84 minutes. Three random initializations yield similar results. The nonlinear representation of operating conditions improves rate prediction, while stage-dependent changes throughout drying appear in the residuals. Embedding parameter learning within analytical kinetics provides a compact computational link between operating conditions, mass trajectories and target times.

Keywords: Drying Kinetics; Multilayer Perceptron; Parameter Learning; Validation Across Operating Conditions; Applied Computing

1 Introduction

Operating conditions influence the rate of water loss during fabric drying and the time required to reach a target mass. *Inlet temperature, dry-air mass flow rate* and *humidity ratio* jointly affect heat and moisture transfer. Predicting drying under new conditions therefore requires a relationship between these inputs and the observed mass trajectories. For datasets with repeated experiments and a limited number of operating conditions, the central challenge is to observe the relationship while retaining a physically interpretable description of the process.

The **Newton-Lewis model** relates the drying rate to the departure of moisture content from equilibrium. Its exponential solution represents the process through a capacity and a rate constant [1,2]. Reviews of thin-layer drying models describe a range of curve formulations for different materials and process conditions [3,4], while computational drying research emphasizes parameter setting and validation [5]. The compact exponential structure facilitates target-time calculations and provides a clear starting point for analyzing the effects of operating conditions.

Neural networks (NN) have been used to predict moisture content and drying trajectories for a range of materials. Early feedforward models learned moisture changes from temperature, air velocity and time [6,7]. Subsequent studies introduced deeper networks and sequence models [8,9,10]. Thin-layer equations have also been combined with deep learning for one-step-ahead moisture-ratio prediction [11], and research on industrial filter media has demonstrated the use of small perceptrons for moisture estimation [12]. These studies illustrate distinct roles for neural networks in mapping operating conditions, estimating states and predicting time series.

Here, the learning task maps operating conditions to a rate constant. The kinetic equation generates the time trajectory, and the neural network learns nonlinear combinations of *temperature, flow rate* and *humidity*.

We examine how much a nonlinear parameter map **improves curve and target-time predictions** for unseen conditions when the exponential structure and mass reference are held consistent. A log-linear baseline is compared with an MLP containing 21 parameters. Performance is evaluated by holding out entire operating conditions, and process residuals are used to examine the relationship between improved parameter prediction and the temporal structure of the curves.

2 Methods

Figure 1 summarizes the study workflow.

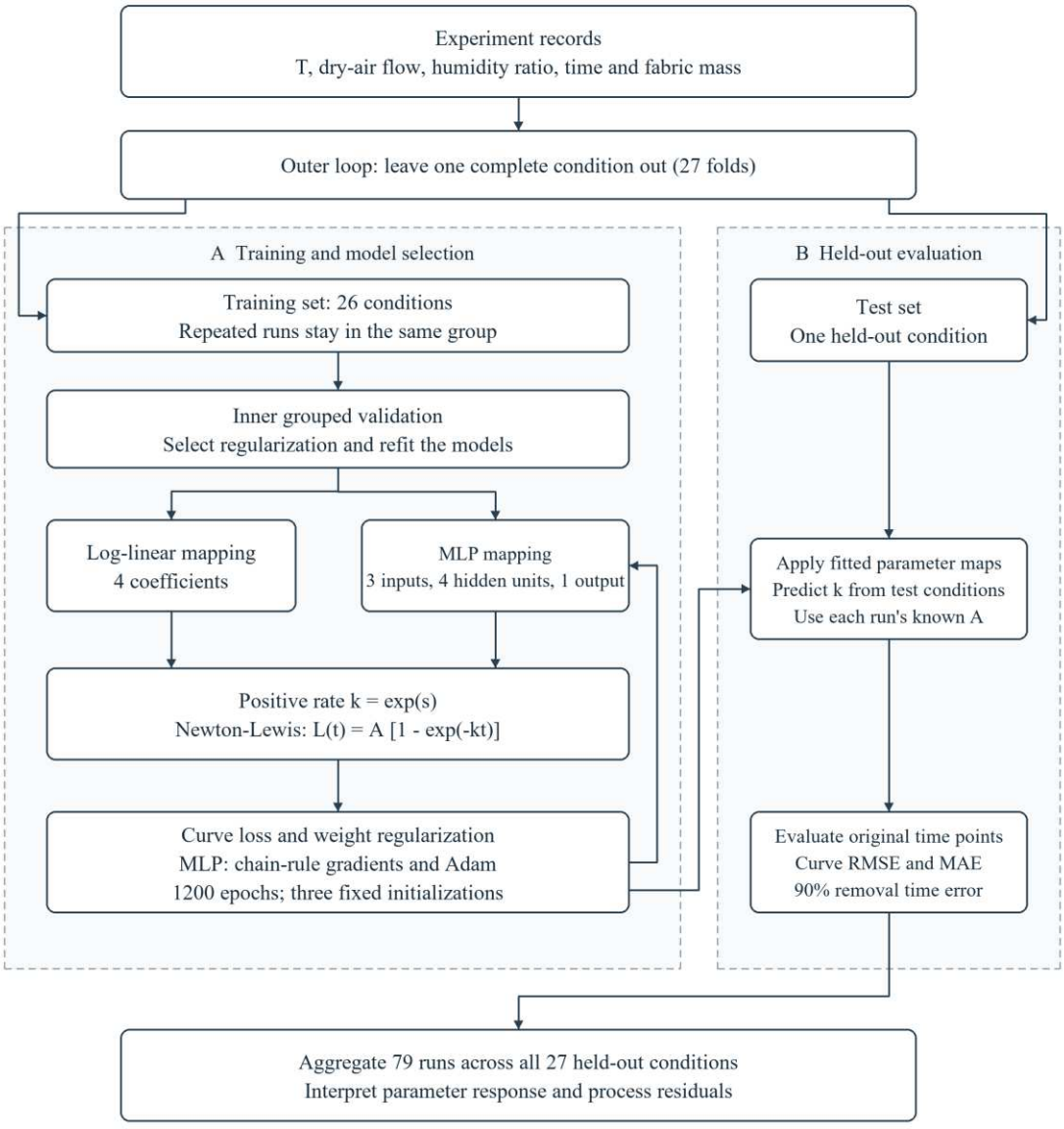


Figure 1. Study workflow.

2.1 Mass reference and kinetic formulation

Kinetic modelling begins by distinguishing *dry matter, water mass* and *environmental equilibrium*. Dry-basis moisture content expresses water mass relative to dry matter, allowing changes in wet-fabric mass to be

translated into changes in moisture state. This mass relationship is consistent with moisture-ratio definitions used in thin-layer drying studies [1,2,3,4]. As initial water loading varies slightly between experiments, capacity is calculated separately for each run. The reference mass after equilibration, initial mass and final mass are retained as distinct quantities to preserve their physical interpretation.

The Newton-Lewis model assumes that moisture in excess of the environmental equilibrium value drives water loss. The proportionality constant k determines the rate of approach to equilibrium [1,2]. Under fixed inlet conditions, a constant k represents the full curve as a lumped first-order kinetic approximation. The solution is determined by the initial state, reference equilibrium state and rate constant. A larger k produces an earlier crossing of a given water-loss fraction. Comparisons of thin-layer models likewise show that model selection depends on the material and process conditions [3,4,5].

This formulation separates capacity from the characteristic time scale. ***A* sets the vertical mass scale**, whereas k sets the horizontal time scale, and their product determines the initial water-loss rate. Calibrating k with a known capacity reduces compensation between jointly estimated capacity and rate, allowing parameter learning to focus on rate differences across conditions. We retain the complete time records and examine mean residuals to assess how the common exponential structure describes different stages of drying.

Let m_d be the dry-matter mass and $m_w(t)$ water mass, with dry-basis moisture content $X(t) = \frac{m_w(t)}{m_d}$, then total mass is $m(t) = m_d[1 + X(t)]$. Within a given operating condition, equilibrium moisture content X_e and rate constant k are treated as constants. The governing equation and initial condition are

$$\frac{dX}{dt} = -k(X - X_e), X(0) = X_0 \tag{1}$$

Separating variables and integrating gives $X(t) = X_e + (X_0 - X_e)e^{-kt}$. Substitution into cumulative water loss, $L(t) = m_d[X_0 - X(t)]$, yields

$$\begin{cases} L(t) = A(1 - e^{-kt}) \\ A = m_d(X_0 - X_e) = m_0 - m_e \end{cases} \tag{2}$$

A is the removable water relative to equilibrium, and k governs the time scale of the approach to equilibrium. Their product determines the initial rate $E(0) = Ak$ (when $t = 0$). Within an experiment, A is fixed and is calculated from that run's initial mass. The reference mass after equilibration in the source table approximates m_e, and the final mass is retained as a separate observation. Appendix A gives the mass relationships and integration steps.

2.2 Mapping operating conditions to the rate parameter

The parameter map determines the rate constant for a new experiment. Fitting k directly to an observed curve characterises that particular run. Predicting unseen conditions additionally requires k to be expressed as a function of *temperature, dry-air flow* and *humidity ratio*. A log-linear relationship provides a compact baseline for main effects, while an MLP allows response slopes to vary across conditions. Previous neural-network studies of drying support the use of operating inputs to learn nonlinear mappings [6,7,8,12].

In the MLP, each hidden neuron first forms a weighted sum of the inputs, then applies a hyperbolic tangent activation. Four neurons provide four learned response components, which the output layer combines into a log-rate s. Composing **purely linear layers** would retain a linear relationship, while the **activation**

function enables curvature and combined effects in the response to operating conditions [13,14,15,16]. **A single hidden layer** keeps the computational relationship between inputs, weights and output transparent.

A positive transformation converts the network output to k, ensuring a positive rate and the prescribed initial water loss. The network receives the three operating variables. Run-specific A and time enter the subsequent kinetic calculation. This division incorporates known initial masses directly into the trajectories while assigning repeated runs at the same nominal condition a common rate prediction. Learning a parameter also exploits the trajectory structure specified by the analytical equation [2,5,15].

The inputs are *inlet temperature* T, *dry-air mass flow rate* G and *inlet humidity ratio* ω. They are normalized by using the design centres and half-ranges:

$$z_T = \frac{T - 60}{20}, z_G = \frac{G - 195.6}{84.65}, z_\omega = \frac{\omega - 27}{14.85} \tag{3}$$

The linear baseline uses an intercept and three main effects in log-rate space. With reference rate $k_{ref} = 1min^{-1}$, the exponent s is dimensionless:

$$s = \beta_0 + \beta_T z_T + \beta_G z_G + \beta_\omega z_\omega, k = k_{ref} e^s \tag{4}$$

The MLP has three inputs, four hyperbolic tangent hidden units and one linear output, which is transformed into a positive k. References [13,14,15,16] describe nonlinear feedforward representations and backpropagation. The network contains **12 input weights**, **4 hidden biases**, **4 output weights** and **one output bias**:

$$\begin{cases} H_h = \tanh(\sum_{j=1}^{3} W_{hj} z_j + b_h) \\ s = \sum_{h=1}^{4} v_h H_h + c \end{cases} \tag{5}$$

The predicted k varies with operating conditions, and the common Newton–Lewis analytical solution then generates the mass trajectory. This concentrates nonlinear learning in the three-dimensional input space while remaining the structural properties including $L(0) = 0$ and $k > 0$.

2.3 Curve-based training and model selection

Training is supervised by measured mass trajectories. The network predicts k from operating conditions, the analytical solution calculates water loss at each time point, and predictions are compared with observations. Gradients propagate from the curve loss through the rate and network output to the weights, so the network is calibrated using the complete process records. The chain rule underlying backpropagation is described in [14,15,16]. Explicit derivatives for the exponential output are used here to make the implementation straightforward to check.

Normalised water loss, $F = L/A$, places experiments on comparable numerical scales during optimisation. Each curve is represented by 41 uniformly spaced time points covering its full recorded duration. Errors are averaged within each run and then across runs, giving long and short records equal overall weight. After training, errors in grams are evaluated on the original minute-resolution records.

Weight regularization controls the magnitude of the nonlinear parameter map. A smaller penalty permits a more flexible response, while a larger penalty favours a smoother map. Three candidate values are compared within each outer training set, and the value with the lowest validation curve error is used to fit the outer model. This separation of tuning and testing follows the principles discussed in [18,19].

Adam maintains exponential moving averages of the gradient and its square, applies bias correction and updates parameters with adaptive step sizes [17]. The learning rate and number of epochs are fixed, and **three initialization seeds are examined independently**. Results are reported for each run to show variation due to initialization alongside differences between the methods.

Each experimental curve is interpolated at 41 equally spaced points over its recorded duration. With $F = \frac{L}{A}$, squared errors are averaged first within each curve and then across experiments, giving long and short records equal to overall weight. The linear map and MLP use identical training points and targets. For the MLP, a weight penalty is added to the curve loss, with biases excluded from regularisation:

$$J = \frac{1}{R}\sum_{r=1}^{R}\frac{1}{41}\sum_{j=1}^{41}(\hat{F}_r(t_j) - F_r(t_j))^2 + \frac{\lambda}{2}(\sum_{h,j} W_{hj}^2 + \sum_{h} v_h^2) \tag{6}$$

The MLP uses explicitly coded chain-rule gradients and Adam updates [17], with **a learning rate of 0.015 and 1,200 training epochs**. Hidden and output weights are initialized with standard deviations of 0.25 and 0.05, respectively. Bias initialization and gradient derivations are given in Appendix B. The linear map is initialized by regressing the logarithm of the individually estimated k values on the inputs, then optimized using the same curve objective.

The outer evaluation uses **27-fold leave-one-condition-out validation**, with all repeats of a condition kept together. Within each outer training set, condition identifiers modulo three define three inner groups, and validation errors are compared for $\lambda \in \{0.0001, 0.01, 0.1\}$. After selecting λ, the model is refitted on the outer training set and predicts the held-out condition. Model selection and performance evaluation are nested [18], with grouping aligned to the dependence structure of the experiments [19]. Seed 17 provides the main results; seeds 41 and 73 provide additional runs with the same architecture and training duration.

2.4 Evaluation of curves and target times

Test performance is evaluated on the original minute-resolution records using per-run curve *RMSE* and mean absolute error (*MAE*) in grams. For parameter evaluation, the rate constant k estimated from each complete experimental curve serves as the reference. The target time is defined consistently **as the time required to remove 90%** of the reference removable water.

$$t_{90} = \frac{\ln 10}{k} \tag{7}$$

Observed t_{90} is obtained by linear interpolation between the two adjacent points bracketing the first crossing of $0.9A$. A common fractional target facilitates comparison between runs with slightly different initial water loads. Within the exponential model, predicted time and rate satisfy:

$$\frac{\hat{t}_{90} - t_{90}}{t_{90}} = \frac{k}{\hat{k}} - 1 \tag{8}$$

Overestimating the rate therefore advances the predicted target time, while underestimating it delays the prediction. Errors relative to the observed crossing additionally reflect differences between the modelled curve shape and the experimental process.

3 Experiments and results

3.1 Experimental design and data preparation

Experiments combine temperatures of 40, 60 and 80℃, dry-air mass flow rates of 111.5, 195.6 and 280.8 kg/h, and humidity ratios of 11.5, 27 and 41.2g/kg, forming 27 conditions and 81 runs (see Figure 2). Run identifiers link the experimental plan to the process records, with initial, reference and final masses retained separately. Run 18 has no valid values in the wet-fabric mass column, and the planned duration of run 61 differs from its recorded endpoint by 10 min. The analysis therefore uses 7**9 runs and 14665 time points**, retaining coverage of all operating conditions.

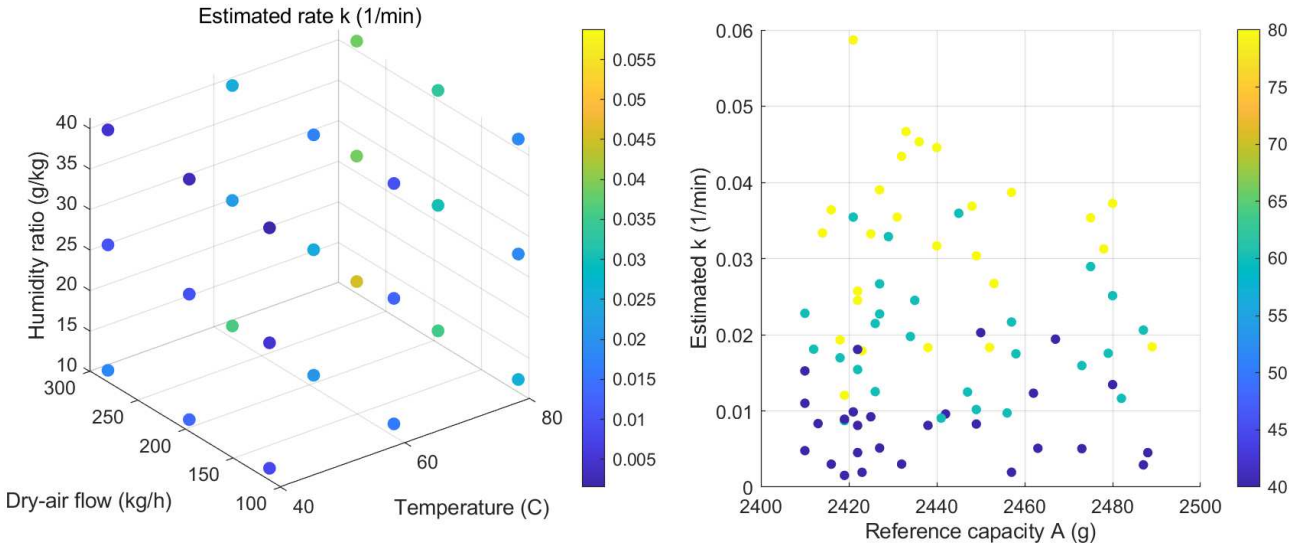


Figure 2. Operating conditions, reference capacities and rate estimates for individual experiments.

Process mass is taken from the smoothed wet-fabric mass column in the source table. All reference masses are 4080g. A ranges from 2410 to 2489g, with a mean of 2440.76g and standard deviation of 23.34g. Initial water loading varies relatively little across experiments, and retaining run-specific A incorporates capacity and rate variation separately. The source table's nominal correction fields are excluded from the analysis.

Wet-fabric mass curves can be reproduced from the smoothed whole-machine weighing signal by linear calibration at the initial and final points. The study uses these existing offline processed curves to evaluate prediction across operating conditions within the dataset. Real-time updating would use an independently calibrated conversion available at the start of the process. All training and plotting are performed in *MATLAB R2023b Update 3*.

3.2 Prediction for held-out operating conditions

In the main MLP run, median curve RMSE decreases by 9.0%, and mean RMSE falls from 312.59 to 302.19 g. The MLP yields a lower curve error in 47 of the 79 experiments. Median absolute relative error in k decreases from 21.34% to 16.59% (as in Table 1), indicating improved estimation of the time scale for unseen operating conditions. Inner validation selects $\lambda = 0.01$ in all 27 outer folds, and the other two initializations produce similar aggregate metrics.

Table 1. Prediction errors in leave-one-condition-out validation.

Parameter map	Median curve RMSE (g)	Median absolute t_{90} error (min)	Median absolute relative k error (%)
Log-linear	279.63	59.51	21.34
MLP seed 17	254.48	51.84	16.59
MLP seed 41	254.48	51.84	16.59
MLP seed 73	254.48	51.84	16.59

In Figure 3, observations are derived from the source wet-fabric mass curves. *Grey lines* show exponential fits to individual runs while *blue and red lines* show held-out log-linear and MLP predictions.

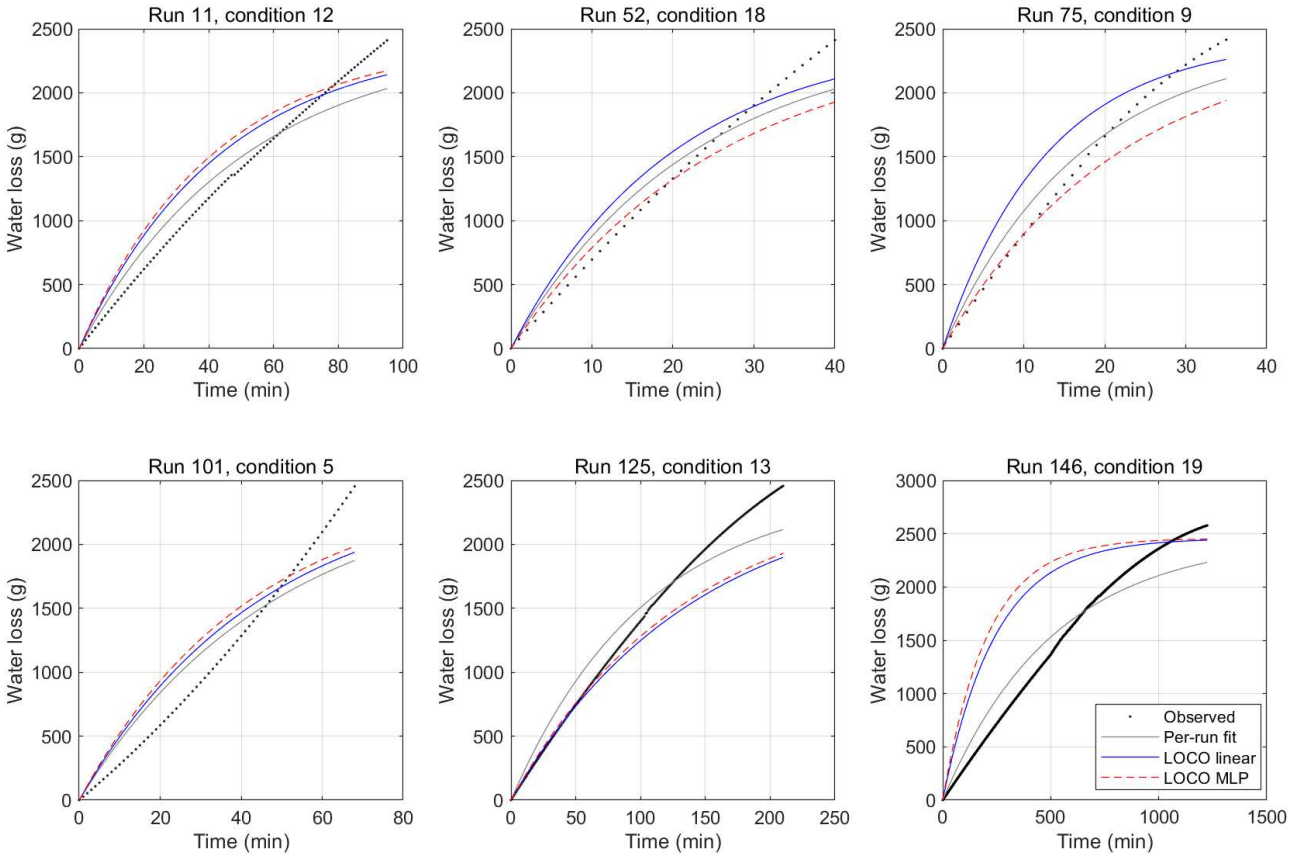


Figure 3. Cumulative water-loss trajectories for six representative experiments.

Errors appear as differences in overall drying speed and as deviations in shape during the middle and final stages. For repeated runs at the same condition, the network predicts a common k, while each trajectory retains its own A. Holding out an entire condition preserves this variation between repeats and evaluates the parameter map under new inlet conditions.

3.3 Target times and process residuals

The three sets of MLP results in Figure 4 correspond to different initializations. All 79 curves cross the $0.9A$ target. The MLP reduces the median absolute t_{90} error from 59.51 to 51.84 mins, a decrease of 7.67 mins

(approximately 12.9% using unrounded values). This agrees with the improvement in k prediction. For long processes, a moderate rate error can produce a substantial error in minutes through the inverse relationship between rate and time. Examining both quantities helps support the consequences of parameter errors.

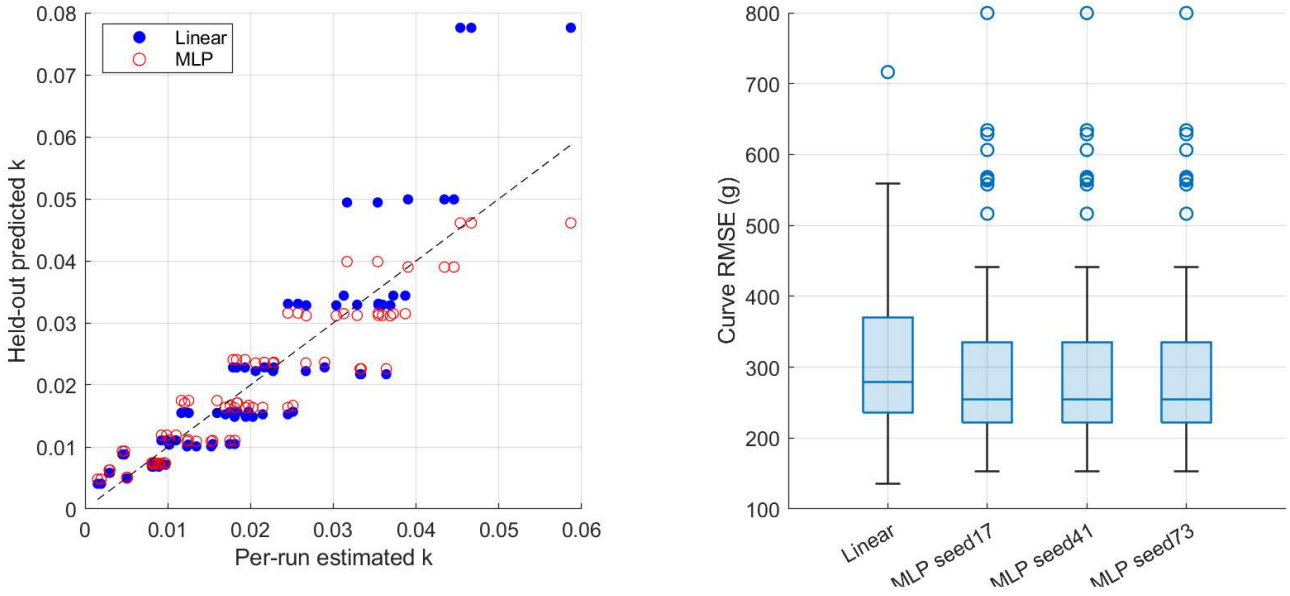


Figure 4. Held-out rate predictions & Curve error distributions.

In Figure 5, Mean residuals are calculated as the observed water-loss fraction minus the exponential fit for each run. In the essay, normalized time is used for plotting and training uses elapsed time in minutes.

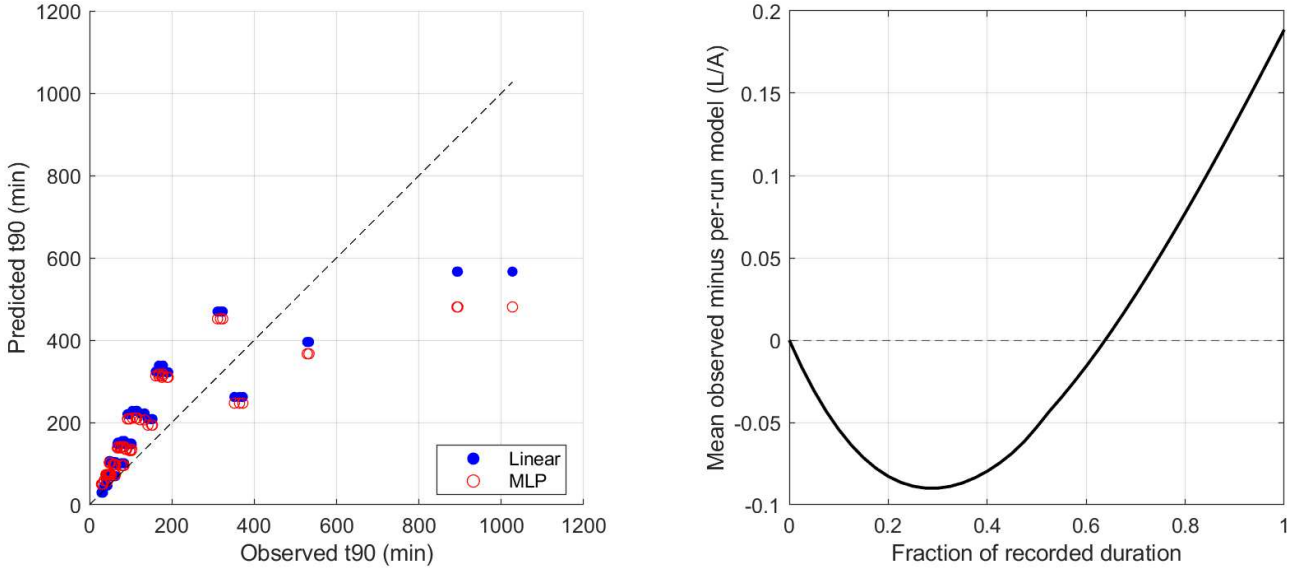


Figure 5. Observed and predicted t_{90} & Mean residuals.

Mean residuals are negative in the early and middle portions of the process and become positive near the end, as shown in Figure 5. Thus, even with a rate estimated separately for each experiment, the common exponential structure leaves systematic temporal deviations. The network improves transfer of the rate parameter across operating conditions, while **warm-up and stage-dependent changes continue to affect**

curve shape. Examining these components within a common framework motivates further study and experiments searching for inlet stabilization times and independent weighing records.

4 Discussion and conclusions

Figure 6 illustrates the fitted parameter maps. Held-out performance is calculated independently using the validation procedure described above. With a common mass reference and analytical kinetic structure, the **21-parameter MLP improves prediction for unseen operating conditions**. The log-linear map captures much of the variation across conditions, and the nonlinear network further reduces median curve and target-time errors. Similar results across three initializations allow interpretation to focus on model structure.

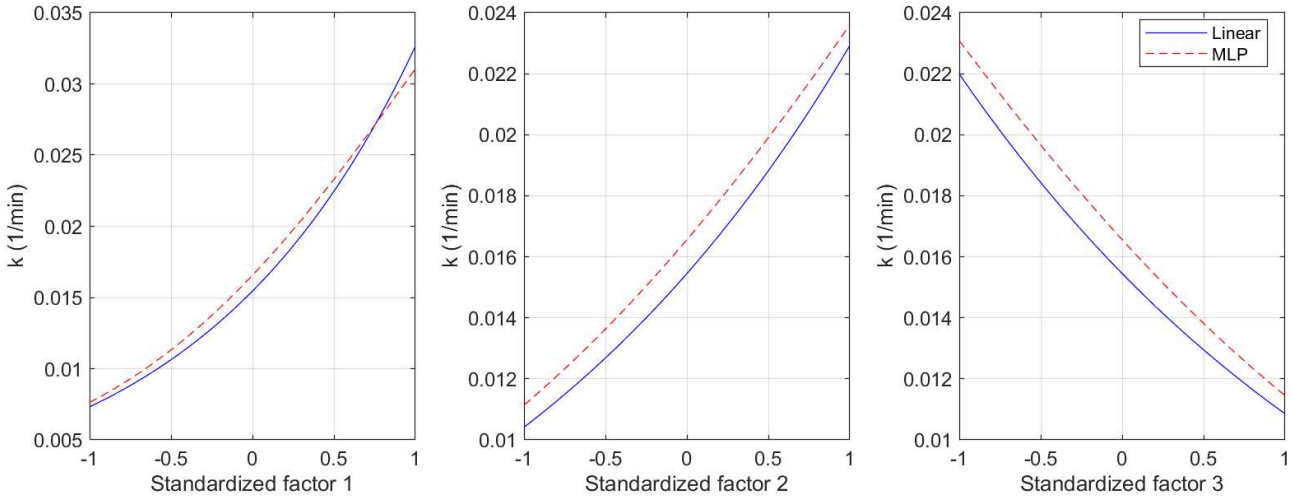


Figure 6. Response slices for the parameter maps fitted to the complete dataset.

The framework assigns a clear role to each component. Mass relationships determine the capacity of each experiment, inlet conditions determine the predicted rate, and the analytical solution converts these quantities into a temporal trajectory. Observations, model parameters and evaluation metrics are consequently linked through an explicit calculation. Training the parameter map directly through the curve objective provides a compact representation for a small set of independent operating conditions, while target times express the engineering consequences of prediction errors.

The dataset comprises 79 experimental processes at 27 operating conditions, with 14665 minute-resolution observations describing the curves. Our study uses Newton-Lewis kinetics as a physical framework and an MLP to learn the rate parameter as a joint function of *temperature, flow rate* and *humidity*. Across 27 validation folds, median curve RMSE decreases by 9.0% and median absolute t_{90} error by 12.9%. The framework connects mathematical modelling, numerical implementation and grouped validation, providing a reproducible baseline for independent experiments and further development of process models.

References

[1] Lewis W K. The Rate of Drying of Solid Materials. Journal of Industrial & Engineering Chemistry, 1921, 13(5): 427–432. https://doi.org/10.1021/ie50137a021

[2] Royen M J, Noori A W, Haydary J. Experimental Study and Mathematical Modeling of Convective Thin-Layer Drying of Apple Slices. Processes, 2020, 8(12): 1562. https://doi.org/10.3390/pr8121562

[3] Kucuk H, Midilli A, Kilic A, Dincer I. A Review on Thin-Layer Drying-Curve Equations. Drying Technology, 2014, 32(7): 757–773. https://doi.org/10.1080/07373937.2013.873047

[4] Ertekin C, Firat M Z. A comprehensive review of thin-layer drying models used in agricultural products. Critical Reviews in Food Science and Nutrition, 2017, 57(4): 701–717. https://doi.org/10.1080/10408398.2014.910493

[5] Defraeye T. Advanced computational modelling for drying processes – A review. Applied Energy, 2014, 131: 323–344. https://doi.org/10.1016/j.apenergy.2014.06.027

[6] Gorjian S, Tavakkoli T, Khoshtaghaza M H. Designing and optimizing a back propagation neural network to model a thin-layer drying process. International Agrophysics, 2011, 25(1): 13–19. https://www.international-agrophysics.org/pdf-106284-37133?filename=37133.pdf

[7] Krishna Murthy T P, Manohar B. Hot air drying characteristics of mango ginger: Prediction of drying kinetics by mathematical modeling and artificial neural network. Journal of Food Science and Technology, 2014, 51(12): 3712–3721. https://doi.org/10.1007/s13197-013-0941-y

[8] Huang X, Li Y, Zhou X, et al. Prediction of Apple Slices Drying Kinetic during Infrared-Assisted-Hot Air Drying by Deep Neural Networks. Foods, 2022, 11(21): 3486. https://doi.org/10.3390/foods11213486

[9] Jia Z, Liu Y, Xiao H. Deep Learning Prediction of Moisture and Color Kinetics of Apple Slices by Long Short-Term Memory as Affected by Blanching and Hot-Air Drying Conditions. Processes, 2024, 12(8): 1724. https://doi.org/10.3390/pr12081724

[10] Simonič M, Ficko M, Klančnik S. Predicting Corn Moisture Content in Continuous Drying Systems Using LSTM Neural Networks. Foods, 2025, 14(6): 1051. https://doi.org/10.3390/foods14061051

[11] Recio-Colmenares R B, Recio-Colmenares C L, Conchas-Cedano R F, et al. Hybrid Thin-Layer and Deep Learning Modeling for One-Step-Ahead Prediction of Solar Drying Kinetics of Whole Charal (Chirostoma spp.) Under Field-Realistic Scenarios. AgriEngineering, 2026, 8(7): 266. https://doi.org/10.3390/agriengineering8070266

[12] Wewer C R, Iosifidis A. Online Non-Destructive Moisture Content Estimation of Filter Media During Drying Using Artificial Neural Networks. arXiv preprint, 2023. https://arxiv.org/abs/2303.15570

[13] Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. Neural Networks, 1989, 2(5): 359–366. https://doi.org/10.1016/0893-6080(89)90020-8

[14] Rumelhart D E, Hinton G E, Williams R J. Learning representations by back-propagating errors. Nature, 1986, 323: 533–536. https://doi.org/10.1038/323533a0

[15] Bishop C M. Neural Networks for Pattern Recognition. Oxford University Press, 1995. https://doi.org/10.1093/oso/9780198538493.001.0001

[16] Goodfellow I, Bengio Y, Courville A. Deep Learning. MIT Press, 2016. https://www.deeplearningbook.org/

[17] Kingma D P, Ba J. Adam: A Method for Stochastic Optimization. ICLR, 2015. https://arxiv.org/abs/1412.6980

[18] Cawley G C, Talbot N L C. On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation. Journal of Machine Learning Research, 2010, 11: 2079–2107. https://www.jmlr.org/papers/v11/cawley10a.html

[19] Roberts D R, Bahn V, Ciuti S, et al. Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. Ecography, 2017, 40(8): 913–929. https://doi.org/10.1111/ecog.02881

Appendix A. Mass relationships and the kinetic solution

Let m_d be the constant dry-matter mass and $m_w(t)$ the water mass. Their sum gives the total mass:

$$m(t) = m_d + m_w(t) \tag{A1}$$

Dry-basis moisture content is the mass of water per unit mass of dry matter:

$$X(t) = \frac{m_w(t)}{m_d} \tag{A2}$$

Multiplying by m_d and substituting into the mass relationship gives

$$m_w(t) = m_d X(t), m(t) = m_d[1 + X(t)] \tag{A3}$$

Let X_e denote equilibrium moisture content under the environmental conditions, with corresponding total mass m_e:

$$m_e = m_d(1 + X_e) \tag{A4}$$

The Newton–Lewis model relates the rate of change in moisture content to its excess above equilibrium, where k is a positive rate constant:

$$\frac{dX}{dt} = -k(X - X_e) \tag{A5}$$

Treating k and X_e as constants within an experiment, separate the variables:

$$\frac{dX}{X - X_e} = -kdt \tag{A6}$$

Apply the initial condition $X(0) = X_0$ and integrate both sides over their respective limits:

$$\int_{X_0}^{X(t)} \frac{du}{u - X_e} = -k \int_0^t ds \tag{A7}$$

Integration gives natural logarithms, which combine as follows:

$$\ln[X(t) - X_e] - \ln(X_0 - X_e) = -kt \tag{A8}$$

Exponentiating both sides gives the normalised moisture ratio MR:

$$MR(t) = \frac{X(t) - X_e}{X_0 - X_e} = e^{-kt} \tag{A9}$$

Multiplying by the denominator and adding X_e gives

$$X(t) = X_e + (X_0 - X_e)e^{-kt} \tag{A10}$$

Cumulative water loss is the difference between the initial and current total masses:

$$L(t) = m_0 - m(t) = m_d[X_0 - X(t)] \tag{A11}$$

Substitute the exponential solution and factor the expression:

$$L(t) = m_d(X_0 - X_e)(1 - e^{-kt}) \tag{A12}$$

Define the capacity parameter A to obtain the core predictive formulation:

$$A = m_d(X_0 - X_e) = m_0 - m_e \tag{A13}$$

Cumulative water loss and current mass are therefore

$$L(t) = A(1 - e^{-kt}), m(t) = m_e + Ae^{-kt} \tag{A14}$$

Differentiate cumulative water loss with respect to time. Differentiating the exponent with respect to t introduces the factor $-k$:

$$E(t) = \frac{dL}{dt} = Ake^{-kt} \tag{A15}$$

The initial rate v is determined jointly by A and k:

$$v = E(0) = Ak \tag{A16}$$

Differentiating again gives the change in the water-loss rate:

$$\frac{dE}{dt} = -Ak^2e^{-kt} \tag{A17}$$

A sets the mass scale, while k sets the time scale. Two trajectories with the same A may have different k values; a larger k produces a faster approach to the same reference equilibrium state. The modelled rate decreases progressively, so fitting the full experimental record yields a parameter that also reflects warm-up, rapid water loss and final-stage changes. The complete records are retained for consistent modelling, and temporal residuals are used to examine these stage-dependent differences.

Appendix B. Neural-network gradients and training settings

To illustrate the chain rule, consider the half-squared error at a single time point. Denote prediction error by e.

Differentiate the single-point loss with respect to predicted water loss:

$$e = \hat{L} - L, \ell = \frac{1}{2}e^2, \frac{\partial \ell}{\partial \hat{L}} = e \tag{B1}$$

At fixed time t, the derivative of the prediction with respect to k is

$$\frac{\partial \hat{L}}{\partial k} = Ate^{-kt} \tag{B2}$$

The derivative of the exponential output with respect to s is k:

$$\frac{\partial k}{\partial s} = k \tag{B3}$$

Multiplying these three derivatives gives the loss gradient with respect to the network output:

$$\delta = \frac{\partial \ell}{\partial s} = eAtke^{-kt} \tag{B4}$$

The gradients for the output weights and bias are

$$\frac{\partial \ell}{\partial v_h} = \delta H_h, \frac{\partial \ell}{\partial c} = \delta \tag{B5}$$

Propagate the gradient to the hidden layer using the derivative of the hyperbolic tangent:

$$\frac{\partial H_h}{\partial a_h} = 1 - H_h^2 \tag{B6}$$

The derivative with respect to an input weight is

$$\frac{\partial \ell}{\partial W_{hj}} = \delta v_h (1 - H_h^2) z_j \tag{B7}$$

For a hidden bias, the corresponding input is one:

$$\frac{\partial \ell}{\partial b_h} = \delta v_h (1 - H_h^2) \tag{B8}$$

Standard gradient descent updates parameters in the direction of decreasing loss:

$$\theta_{new} = \theta_{old} - \eta \frac{\partial J}{\partial \theta} \tag{B9}$$

The implementation uses the same chain of derivatives for F: A cancels, the squared-error derivative introduces a factor of two, and gradients are averaged over time points and experiments. Adam adjusts step sizes using moving estimates of gradient moments. For example, with $A = 2.4kg$, $k = 0.02min^{-1}$ and $t = 30min$, predicted water loss is approximately 1.083 kg. If observed loss is 1.20 kg, e is negative. The remaining factors are positive, so the gradient with respect to s is negative. The update increases s and k, bringing the predicted water loss closer to the observation.

Appendix C. Data & Code Availability

The prepared experimental dataset and MATLAB implementation are available in the project repository. The dataset contains 14,665 observations from 79 runs at 27 operating conditions. It is the processed dataset used in this study, derived from experiments supplied by a collaborating research group. Column definitions, units, preprocessing provenance and excluded runs are documented in the data directory.

- Experimental dataset:
  https://github.com/YQY444/physics-guided-drying-parameter-learning/blob/main/data/drying_curves.csv
- Dataset documentation:
  https://github.com/YQY444/physics-guided-drying-parameter-learning/tree/main/data
- Project repository:
  https://github.com/YQY444/physics-guided-drying-parameter-learning
- Version containing the released dataset:
  https://github.com/YQY444/physics-guided-drying-parameter-learning/tree/9a27c879902c4cb93602d29b2b465e9fc7440d3c

Run *run_study.m* in MATLAB to execute the experimental workflow. The repository includes the Newton–Lewis parameter-learning implementation, a synthetic example, an editable workflow diagram and aggregate results.