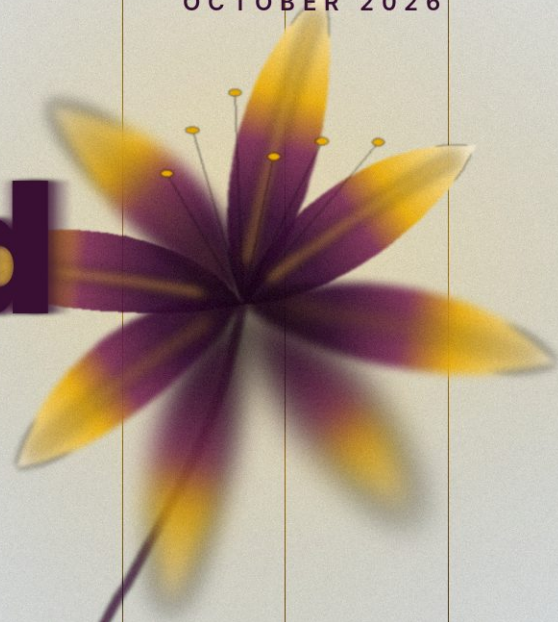


DairyGuard Batch.



Blueprint of a batch-aware spoilage model: the data we will collect, the architecture we will build, how it will be trained, and the tests it must pass

Suryansh Pratap Singh

Department of Mathematics and Computing, Shri Mata Vaishno Devi University, India

STATUS: DESIGN; PROTOTYPE ON SIMULATED DATA ONLY. Nothing in this paper has been trained on, or measured in, real milk. A first prototype of stages 0–2 runs on simulated batches (Section 7). It is written *before* the data exist, so that the data plan, the model and, above all, the tests it must pass are fixed in advance and cannot be tuned to the results. Changes after publication will be listed with their dates in the amendment log (Section 11).

WHAT THIS DOCUMENT IS

White Paper WP-01 [1] describes DairyGuard Kinetic, the monitor we have built and tested. Its own results show where it stops: the biology of each batch of milk, which a temperature log cannot see. This paper is the blueprint for the model that will address that gap, working name **DairyGuard Batch**. It sets out why it is needed, what it must do and not do, the data sources and how we will get access to them, the architecture layer by layer, the training plan, the evaluation protocol with pass and fail criteria, deployment, risks and timeline.

1 Why a second model

DairyGuard Kinetic treats every batch as a draw from one fixed population. Four of its results show that this is now the limiting assumption [1, 2].

1. In 240 000 simulated deliveries, the unseen biology costs 0.27 to 0.40 AUC; a lab-grade thermometer recovers at most 0.02.
2. On real ComBase data, the square-root slope varies between 11 *Listeria* studies with a coefficient of variation of 0.25.
3. Five growth laws, including a monotone neural ensemble, predict 20 held-out studies with accuracy factors between 1.37 and 1.45. A more flexible law fitted to growth *rates* alone does not help.
4. The monitor's estimate of hours left at 4 °C has a median error of 16 h against a median of 22 h. In simulation, one measurement of the batch closes 69–94 % of the accuracy gap.

The conclusion is about data, not formulas: the next model must learn *each batch's* biology, from complete growth curves under changing temperatures and from simple measurements taken when the batch is loaded.

2 Goals and non-goals

The model will	The model will not
predict the bacterial count of a specific batch along any temperature history, with a calibrated uncertainty band	give a food-safety verdict; the output is a count against a stated limit
use a few cheap tests taken at loading to estimate the batch's hidden state	model illness, toxins or non-microbial spoilage
keep DairyGuard Kinetic's generalized-time core and its output interface	replace physics with a large black-box network
run where Kinetic runs: Python, command line, ESP32, browser	need a network connection to make a prediction

Research questions

- Q1** Do real milk curves under temperature shifts collapse onto one curve in generalized time, as the Baranyi–Roberts model implies? If not, where and by how much do they depart?
- Q2** Which cheap test taken at loading (initial plate count, dye-reduction time, pH, calorimetric detection time) best predicts a batch's growth?
- Q3** On batches the model has never seen, how much of Kinetic's error does a batch-aware model remove?

3 Data plan

3.1 Sources and access

Source	What it gives	How we get access	Status
ComBase growth curves [3]	counts against time for milk and chilled foods, steady and some dynamic temperatures	public browser, free registration	IN PROGRESS
Our ComBase collection	373 milk records; 131 carry no stated rate and need a curve fit	already in the repository	AVAILABLE
Published shift studies, e.g. [4]	counts of milk under fluctuating temperature	papers; raw counts requested from authors	TO REQUEST
E1, E2, E3: our lab counts	curves of real Indian pasteurised milk, steady and shifting temperatures, day-one tests	university microbiology and biotechnology labs	TO REQUEST
E4: calorimetry	heat-flow detection times of the same batches	partner lab with an isothermal microcalorimeter	TO FIND
Temperature logs	the exact profile each sample saw	our ESP32 + DS18B20 controller, logged every minute	HARDWARE READY

3.2 Experiments

	Design	Answers
E1	3 batches of pasteurised milk at 4, 7, 10 and 15 °C; counts every 12–24 h in duplicate until the stationary phase	steady-temperature curves with lag; within- and between-batch variation
E2	the same milk under a step up (4 to 12 °C), a step down, and 4–12 °C cycles	Q1
E3	10 or more batches (brands, dates) under one realistic profile; day-one initial count, resazurin dye-reduction time, pH	Q2, Q3
E4	E3 batches in an isothermal microcalorimeter at 20 °C	the heat test on real milk

Counts follow ISO 4833-1 [12], with ISO 17410 [13] for psychrotrophs. E1–E3 need about 300 to 400 plate counts.

3.3 Data schema

Three tables, joined by keys, stored as CSV in the repository with a data card for each source.

- `curves`: source, study, batch, sample_id, time_h, log10_cfu, dilution, colonies, censored
- `batches`: batch, study, product, organism, day0_log10_cfu, dye_time_h, pH, t_det_h, notes
- `temperatures`: study, batch, time_h, temp_c (the profile each sample experienced)

3.4 Splits, fixed now

Data are split *before* any model is fitted. ComBase curves are split by study (60 % train, 20 % validation, 20 % test). Lab batches are split by batch: at least three E3 batches are locked as the final test set and are not looked at until the evaluation of Section 6. The split files are committed to the repository with a hash, so later changes are visible.

4 Architecture

DairyGuard Batch has five layers (Fig. 1). Layers 1, 2 and 4 extend DairyGuard Kinetic; layer 3 is new; layer 5 is switched on only if the data demand it.

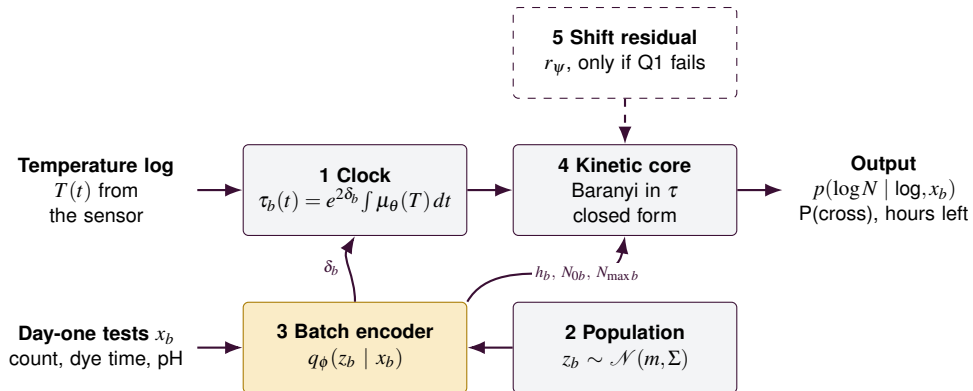


Figure 1 DairyGuard Batch. Purple layers extend DairyGuard Kinetic; the marigold layer is new; the dashed layer is added only if experiment E2 shows that real curves depart from the generalized-time clock.

4.1 Layer 1: the clock

As in Kinetic, temperature enters only through generalized time. The population growth law $\mu_\theta(T)$ is the square-root law by default; the monotone neural law of WP-01 is kept as a candidate and adopted only if it wins on validation data. Each batch b scales the clock by its own factor:

$$\tau_b(t) = e^{2\delta_b} \int_0^t \mu_\theta(T(s)) ds,$$

which is the same as giving batch b its own square-root slope $b_b = b e^{\delta_b}$.

4.2 Layer 2: the population of batches

Each batch has a hidden state $z_b = (\delta_b, h_b, \log N_{0b}, \log N_{\max b})$: rate scale, lag parameter, starting count and maximum count. Batches are drawn from a population, $z_b \sim \mathcal{N}(m, \Sigma)$, whose mean and covariance are learned across studies and batches. This is a hierarchical (mixed-effects) model [7]: a batch with few counts is pulled towards the population, a well-measured batch is not. The fixed spread that Kinetic assumes becomes a learned Σ , and its correlations, for example between lag and rate, are estimated rather than assumed.

4.3 Layer 3: the batch encoder

Before any counts exist for a batch, the encoder estimates its state from day-one tests x_b :

$$q_\phi(z_b | x_b) = \mathcal{N}(m + Wx_b, \Sigma_{|x}).$$

It starts linear, because there will be tens of batches, not thousands. A small network replaces W only if it improves validation likelihood. The encoder answers Q2 directly: the share of Σ that each test explains.

4.4 Layer 4: the kinetic core

In generalized time the Baranyi–Roberts equations no longer contain temperature, and they have a closed-form solution [5, 6]:

$$y_b(\tau) = y_{0b} + A_b(\tau) - \ln\left(1 + \frac{e^{A_b(\tau)} - 1}{e^{y_{\max b} - y_{0b}}}\right), \quad A_b(\tau) = \tau + \ln(e^{-\tau} + e^{-h_b} - e^{-\tau - h_b}),$$

with $y = \ln N$. So for *any* temperature history the core costs one integral and one formula: no ODE solver, cheap enough for a microcontroller. This is the practical payoff of the biological-clock result.

4.5 Layer 5: the shift residual (conditional)

If E2 shows that real curves do not collapse in generalized time after a shift (Q1), a small network $r_\psi(T, \dot{T}, q)$ is added to the physiological-state equation, $dq/dt = \mu(T)q + r_\psi$, with a penalty pulling it to zero. The core then needs an ODE solver for the residual part, as in universal differential equations [9, 10]. The size of r_ψ after training is itself the answer to Q1.

4.6 Observation model

Plate counts are noisy and censored. Each count is modelled as $\log_{10} N^{\text{obs}} \sim \mathcal{N}(y_b / \ln 10, \sigma^2)$, with σ learned (expected 0.1–0.3 $\log_{10}$, from duplicate plates), and counts below the countable range treated as censored, not as zero.

4.7 Output

For a trip and a batch, the model returns the predictive distribution of $\log N$ over time, from which it reports exactly what DairyGuard Kinetic reports: shelf life used, probability the limit was crossed, hours

left at 4 °C (typical and cautious) and a verdict. Without day-one tests, the encoder falls back to the population and the model reduces to a calibrated version of Kinetic.

5 Training plan

Stage	What is fitted	Data	Pass condition
0	everything, on curves simulated by Kinetic with known parameters	synthetic	true parameters inside the 90 % intervals in about 90 % of cases
1	$\theta, m, \Sigma, \sigma$	ComBase curves (train)	better validation likelihood than one shared batch
2	encoder ϕ	E1, E3 day-one tests and curves	better validation likelihood than the population alone
3	residual ψ , only if Q1 fails	shift curves, E2	residual improves held-out shift curves

Method. Stages 0–2 are fitted by maximising the marginal likelihood with batch states integrated out. The prototype does this in two steps: each batch is fitted by least squares, giving an estimate $\hat{z}_b$ with covariance $C_b = s^2(J^\top J)^{-1}$; then m, W and $\Sigma = LL^\top$ are chosen to maximise the likelihood of the estimates under $\hat{z}_b \sim \mathcal{N}(m + W(x_b - \bar{x}), \Sigma + C_b)$. A joint fit of all levels (Laplace approximation, with a variational fit as a cross-check [8]) will replace the two-step version once it works on real curves. Stage 3 is trained by gradient descent through an ODE solver. **Tools.** NumPy and SciPy for stages 0–2; PyTorch with torchdiffeq on a Colab GPU for stage 3. **Habits.** Fixed seeds, every run logged with its settings and result, checkpoints saved, all code in the repository.

6 Evaluation protocol, fixed in advance

6.1 Baselines

(i) DairyGuard Kinetic as released; (ii) a classical Baranyi fit with one shared batch; (iii) a black-box recurrent network (GRU) of similar size, given the same inputs.

6.2 Metrics

On held-out studies and the locked test batches: RMSE of $\log_{10}$ counts; Ross bias and accuracy factors [11]; AUC and calibration error for crossing the limit; coverage of the 90 % band; median error of hours left at 4 °C.

6.3 Pass and fail criteria

DairyGuard Batch replaces Kinetic in the software only if, on the locked test batches, it

1. reduces the median error of hours left at 4 °C by at least 30 % relative to Kinetic;
2. keeps the coverage of its 90 % band between 85 and 95 %;
3. is not worse than Kinetic on AUC for crossing the limit;
4. beats the black-box baseline on RMSE of $\log_{10}$ counts.

If it fails, the failure is reported in full, with which criterion failed and by how much. A negative answer to Q1, Q2 or Q3 is a result.

7 Prototype on simulated data

`experiments/batch_prototype.py` implements layers 1–4 and stages 0–2 with NumPy and SciPy. It simulates batches from a known population, four isothermal curves per batch (4, 7, 10, 15 °C; nine samples each, plate noise $0.2 \log_{10}$) and three day-one tests whose link to the biology is *invented* for testing the code. It then fits 30 training batches and judges 200 new ones on one trip (40 h at 4 °C, 3 h at 15 °C, 10 h at 4 °C).

Seed	median error of hours left at 4 °C		cautious value safe	
	population only	with day-one tests	population only	with tests
1	31.9 h	14.1 h	0.98	0.82
2	16.4 h	15.4 h	1.00	0.98
3	22.1 h	14.7 h	1.00	0.95
4	21.4 h	13.8 h	1.00	0.90

Stage 0 recovered the true parameters inside their 90 % intervals in 87–100 % of cases (seed 1). What this shows: the pipeline works end to end, and informative day-one tests can sharpen “hours left”. What it does not show: anything about real milk, because both the batches and the tests are simulated. It also shows a risk the evaluation must catch: with 30 batches the encoder can be over-confident (coverage 0.82 for seed 1, below criterion 2 of Section 6).

8 Deployment

DairyGuard Batch keeps the interface of WP-01: `update(T, dt)` and `status()`. The trained parameters (θ, m, Σ, W) are exported as a small JSON file. Because the core is closed-form, the firmware needs only the integral of μ_θ , the encoder’s matrix and the formula of Section 4.4; day-one test results are entered once when a batch is loaded. The browser app gains a “batch tests” panel. Tests will check that Python, JavaScript and C agree, as they do for Kinetic.

9 Risks

Risk	Mitigation
Lab access delayed	stages 0–1 need only public ComBase curves; the plan advances without the lab
Too few batches for the encoder	linear encoder, strong population prior, results reported with intervals
ComBase curves are heterogeneous	study-level random effects; data cards; sensitivity run without each source
Overfitting a flexible part	every network enters only if it beats the simpler layer on validation data
No calorimetry partner	E4 is optional; Q2 is answered with the other tests
Plate-count errors	duplicates, censoring model, a bound lab notebook, raw data never edited

10 Timeline

When	Milestone	Status
October 2026	this blueprint published; prototype of stages 0–2 on simulated data	DONE
Nov–Dec 2026	curve pipeline; splits committed; stages 0 and 1; lab protocol agreed	IN PROGRESS
Jan–Feb 2027	E1 and E2; stage 2 on first batches; Q1 answered	PLANNED
Spring 2027	E3 (and E4 if a partner is found); locked-test evaluation; Q2, Q3	PLANNED
After	report of results; peer-reviewed paper if the results stand	PLANNED

11 Amendment log

Date	Change
October 2026	first version
22 October 2026	added the prototype on simulated data (Section 7); the method paragraph of Section 5 now describes the two-step fit it uses

Note on tools

This design was drafted with the help of an AI assistant; the decisions, criteria and responsibility for them are the author's.

References

- [1] Singh SP. DairyGuard Kinetic: architecture of an open cold-chain spoilage monitor. DairyGuard White Paper WP-01; 2026. <https://github.com/CopyNinja07-07/dairyguard>
- [2] Singh SP. Beyond the 4 °C alarm: what a temperature log can and cannot tell about milk spoilage. Kutra: the author; 2026. <https://github.com/CopyNinja07-07/dairyguard>
- [3] ComBase. Database of microbial responses to food environments. <https://www.combase.cc>
- [4] Lin H, Shavezipur M, Yousef A, Maleky F. Prediction of growth of *Pseudomonas fluorescens* in milk during storage under fluctuating temperature. J Dairy Sci. 2016;99:1822–30. <https://doi.org/10.3168/jds.2015-10179>
- [5] Baranyi J, Roberts TA, McClure P. A non-autonomous differential equation to model bacterial growth. Food Microbiol. 1993;10:43–59. <https://doi.org/10.1006/fmic.1993.1005>
- [6] Baranyi J, Roberts TA. A dynamic approach to predicting bacterial growth in food. Int J Food Microbiol. 1994;23:277–94. [https://doi.org/10.1016/0168-1605\(94\)90157-0](https://doi.org/10.1016/0168-1605(94)90157-0)
- [7] Gelman A, Hill J. Data analysis using regression and multilevel/hierarchical models. Cambridge: Cambridge University Press; 2007.
- [8] Kingma DP, Welling M. Auto-encoding variational Bayes. arXiv:1312.6114; 2014.
- [9] Chen RTQ, Rubanova Y, Bettencourt J, Duvenaud D. Neural ordinary differential equations. arXiv:1806.07366; 2018.
- [10] Rackauckas C, Ma Y, Martensen J, et al. Universal differential equations for scientific machine learning. arXiv:2001.04385; 2020.

- [11] Ross T. Indices for performance evaluation of predictive models in food microbiology. *J Appl Bacteriol.* 1996;81:501–8. <https://doi.org/10.1111/j.1365-2672.1996.tb03539.x>
- [12] ISO 4833-1:2013. Microbiology of the food chain: horizontal method for the enumeration of microorganisms, part 1: colony count at 30 °C by the pour plate technique.
- [13] ISO 17410:2019. Microbiology of the food chain: horizontal method for the enumeration of psychrotrophic microorganisms.