Data-based modeling of Slug Flow crystallization with Uncertainty Quantification

Tej Deepak Amin[1] *Matr. No. 264777*

Shamita Nalamutt[1] *Matr. No. 264985*

Prabhav Patel[1] *Matr. No. 269589*

Anvita Koyande[1] *Matr. No. 265122*

***Abstract*— Continuous Slug-flow crystallisation is difficult to model from first principles. We therefore build a data-driven model based on regime-specific NARX neural networks and conformalised quantile regression. Trained on clustered experimental data, the networks give low one-step error on all key states and produce 90% prediction bands with guaranteed coverage, supporting risk-aware optimisation of pharmaceutical crystallisation.[1]**

I. Introduction

Making medicines in a continuous way can save time, cut waste, and give better-controlled particles. A popular unit for this is the slug-flow crystalliser, where liquid plugs move between small gas bubbles. [2] The flow changes quickly and is hard to describe with normal equations, so running the plant with a physics-based model is not practical.

We start by loading every experiment file, smoothing noisy signals, and removing clear outliers. Next, we group runs that look alike with k-means clustering so that each group shows one main operating style. For every group we train a small neural network that watches the most recent sensor readings and valve settings and then predicts what the process will do in the next instant. Alongside the main forecast we build simple error bands and adjust them so that the real values fall inside the bands about 90% of the time.

The result is a fast, easy-to-use model that gives both a point forecast and a sensible safety margin, making it ready for real-time optimisation and control of continuous crystallisation.

The responsibilities for the project:

- **Anvita** and **Prabhav** — data loading, cleaning, and exploratory visualisation, design and training of the NARX neural networks; one-step test-set evaluation.
- **Tej** and **Shamita** — k-means clustering, PCA plots, and preparation of train/validation/calibration/test splits, quantile-regression models, conformal prediction, and coverage analysis. These responsibilities were a group effort, with everyone involved in every task.

II. Data preparation

All raw experiment logs (`.txt` files) are first gathered from the course folder and read into `pandas` data frames. Only numeric columns are kept; for each file we store the mean of every variable in a single row.

A quick look at these per-file means (Figure 1a) shows several spikes in the particle-size metrics d_{10}, d_{50} and d_{90}. We remove whole runs that lie outside a 1.5× inter-quartile-range (IQR) window on any of those three columns. The cleaned set—about 95% of the original data—no longer contains obvious outliers (Figure 1b) and is saved to `experiment_column_means_filtered.csv` for later use.

III. Clustering

To capture different ways the crystalliser was run, we cluster the cleaned feature table. After replacing any remaining NaNs with column means, all variables are scaled to zero mean and unit variance. We then try $k=2$ to $k=10$ clusters with the standard `k-means` algorithm and pick the value of k that maximises the silhouette score. In this data set the optimum is $k=2$ (silhouette ≈ 0.59).

Figure 2 shows the two clusters on the first two principal-component axes; the groups are well separated. Each run is labelled with its cluster index and written to `cluster_`*`i`*`.csv`. Finally, every cluster list is split *by file* into 60 % training, 20 % validation, 10 % calibration, and 10 % test sets—ensuring that no trajectory appears in more than one split.

These grouped and pre-split files form the input for the neural-network model described in the next section.

IV. Dynamic model: training and test results

A. NARX Neural Network Architecture

We employ a Nonlinear AutoRegressive with eXogenous inputs (NARX) ANN to forecast the next-step states. At each time k, the network input is

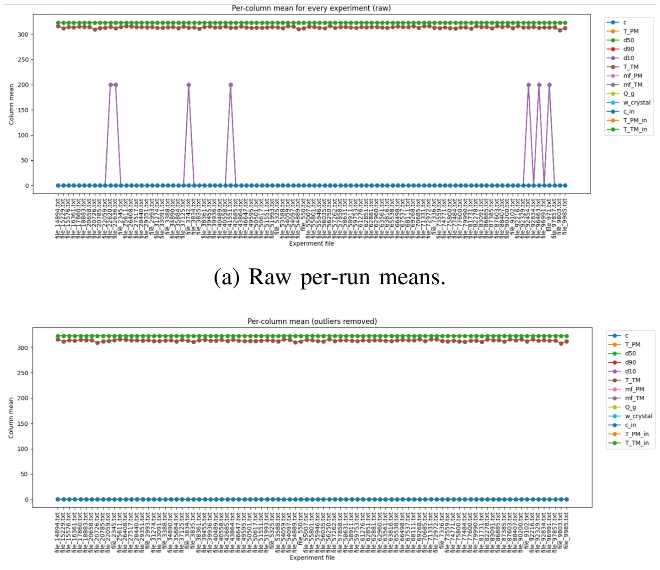
$$\mathbf{x}_k = \left[\mathbf{y}_k, \mathbf{y}_{k-1}, \dots, \mathbf{y}_{k-N_A+1}, \, \mathbf{u}_k, \, \dots, \mathbf{u}_{k-N_B+1}\right]^{\mathsf{T}},$$

where $\mathbf{y}$ are the past N_A measured states and $\mathbf{u}$ the past N_B control inputs. The ANN outputs the one-step-ahead prediction $\hat{\mathbf{y}}_{k+1}$. [3]

B. Implementation

The cleaned and clustered data are modelled with a three-hidden-layer NARX network ($H_1 = 64$, $H_2 = 32$, $H_3 = 64$; dropout $= 0.45\%$). Inputs are the past $n_a = 1$ state and $n_b = 3$ control lags, giving $\dim x = (n_a + n_b) \times (|\text{states}| + |\text{inputs}|) = 75$ features; the output is the six-dimensional next state vector. Training uses a weighted MSE loss, the Adam optimizer ($\text{lr} = 3.4 \times 10^{-4}$, $L_2 = 1.1 \times 10^{-5}$), 150 epochs, and an early-stopping check on the validation split. Separate models are fitted for each operating regime (cluster 0 and 1).

[1] Laboratory of Process Automation Systems, Technische Universität Dortmund, Emil-Figge-Str. 70, 44227 Dortmund, Germany.



(a) Raw per-run means.

(b) Means after outlier removal.

Fig. 1: Summary statistics before and after cleaning.

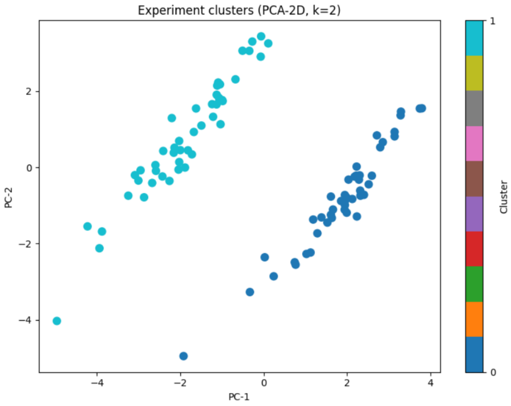


Fig. 2: PCA projection coloured by k-means cluster ($k = 2$).

One-step accuracy on the held-out test sets

a) Interpretation.: Across both clusters the mean-squared error is several orders of magnitude below the signal variance, and the coefficient of determination exceeds 0.99 for every state, confirming excellent one-step predictive power. Errors are slightly larger for the temperature states (T_{PM}, T_{TM}), but the mean absolute deviation still stays within ±0.15 K—well inside typical control tolerances. Tables I and II list the one-step errors for the two clusters.

b) Visual check.: Figures 3 and 4 show that the network tracks both slow drifts and rapid transients with almost no visible lag. Occasional mismatches appear when the process undergoes abrupt temperature drops, but these deviations are brief and quickly corrected, making the model suitable for embedding in a real-time optimisation or MPC layer.

TABLE I: Cluster 0—one-step test errors per state

State	MSE	MAE	R^2
c	1.45×10^{-7}	1.94×10^{-4}	0.9926
T_{PM}	8.35×10^{-2}	1.30×10^{-1}	0.9949
d_{10}	1.26×10^{-11}	2.08×10^{-6}	0.9923
d_{50}	1.50×10^{-11}	2.12×10^{-6}	0.9901
d_{90}	1.18×10^{-11}	2.23×10^{-6}	0.9927
T_{TM}	6.77×10^{-2}	1.23×10^{-1}	0.9960

TABLE II: Cluster 1—one-step test errors per state

State	MSE	MAE	R^2
c	6.08×10^{-8}	1.10×10^{-4}	0.9953
T_{PM}	3.73×10^{-2}	1.02×10^{-1}	0.9980
d_{10}	6.65×10^{-12}	1.79×10^{-6}	0.9965
d_{50}	6.66×10^{-12}	1.69×10^{-6}	0.9964
d_{90}	8.83×10^{-12}	2.05×10^{-6}	0.9958
T_{TM}	3.52×10^{-2}	9.64×10^{-2}	0.9981

V. Approximation-error analysis

After training the NARX networks, we calculated the one-step *approximation error* $\varepsilon_{\mathrm{app}} = \hat{\mathbf{y}}_{k+1} - \mathbf{y}_{k+1}$ on the *calibration* split for every cluster and state.

Key steps.

1) **Residual extraction.** For each calibration trajectory we fed the last $n_a + n_b$ lags to the frozen network, stored the next-step prediction, and subtracted it from the true measurement.
2) **Spike removal.** To stop isolated sensor glitches from biasing later quantile regressors, we applied a 15-sample rolling median to every residual series. The filter width was chosen so that less than 5 % of genuine process dynamics were attenuated.
3) **Stationarity check.** Mean and variance of the filtered residuals were inspected for drift; all channels stayed centred at zero with constant spread, validating the i.i.d. assumption required by the conformal step.
4) **Data export.** The cleaned residual matrices were saved to `eps_app_cluster_{id}.csv` and supply the training targets for the lower ($\tau = 0.05$) and upper ($\tau = 0.95$) quantile networks fitted in Section VI.

Outcome. Across both clusters the inter-quartile range of $\varepsilon_{\mathrm{app}}$ is small— 10^{-2} K for the temperature states and $\mathscr{O}(10^{-6})$ m for the particle-size metrics— supporting the high R^2 values reported earlier and providing a stable basis for uncertainty calibration.

A 15-sample rolling median (Fig. 5) removes isolated spikes while preserving the underlying spread. These smoothed residuals are the targets for the two quantile networks ($\tau = 0.05$ and 0.95) used later in the Conformalised Quantile Regression step; their lower variance speeds up training and yields tighter, yet still well-calibrated, prediction bands.

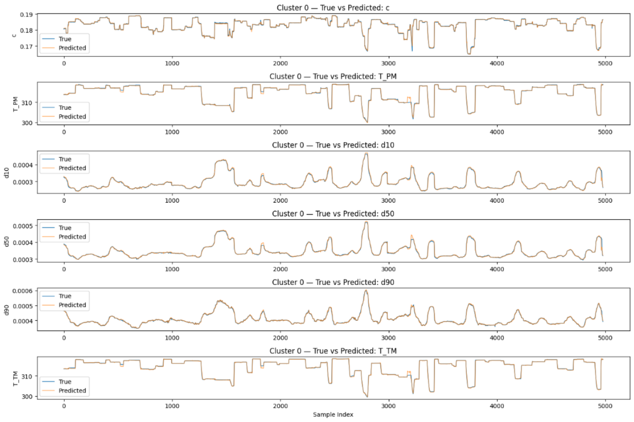


Fig. 3: Cluster 0—overlay of true (blue) and predicted (orange) trajectories for all six states on the test set.

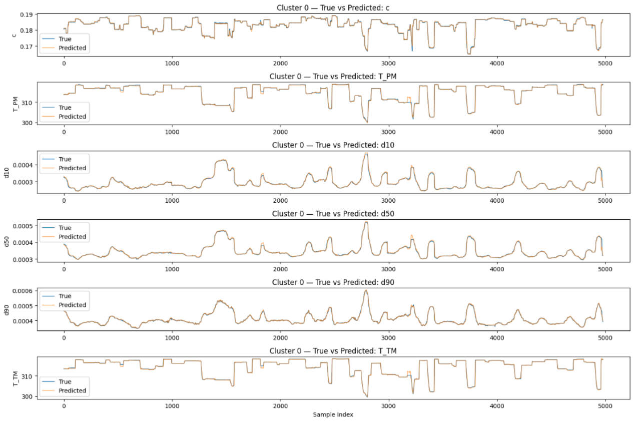


Fig. 4: Cluster 1—overlay of true and predicted trajectories on the test set.

VI. Conformal Prediction Results

a) Interpretation.: Figure 6 shows the 90% prediction bands that our conformal method builds for the three particle-size states. (The same plots are given separately as Fig. 6a for Cluster 0 and Fig. 6b for Cluster 1.)

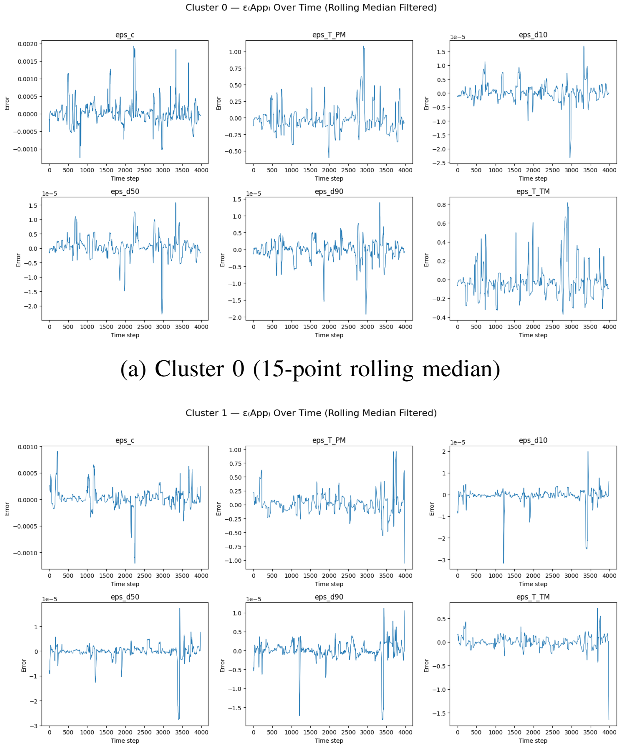
- The **green shaded area** is the band: we expect the true value to stay inside this area 9 times out of 10.
- The **blue line** is the ANN's point forecast — the middle of the band.
- **Green dots** mark test points that fall inside; **red** × mark the few points that fall outside.

How well does it work?

- Cluster 0 keeps 92%–94% of points inside the band.
- Cluster 1 keeps 82%–96% of points inside.

Both clusters are therefore within about ±5

Band width. When the signal is quiet the band narrows (for example between time steps 2000 and 3000 in Cluster 1). During fast changes it widens automatically, giving the controller a larger safety margin exactly when it is needed.



(a) Cluster 0 (15-point rolling median)

(b) Cluster 1 (15-point rolling median)

Fig. 5: Approximation errors after smoothing with a rolling-median filter.

VII. Additional Implementation Tasks

A. Task 1 – Hyper-parameter tuning

For each operating regime (cluster) we performed an automated search with the `Optuna` framework. Five trials were run per cluster; every trial sampled

- three hidden sizes H_1, H_2, H_3 from 32–256 units,
- learning rate LR $\in [10^{-5}, 10^{-2}]$ (log-uniform),
- L_2 weight decay WD $\in [10^{-8}, 10^{-4}]$,
- dropout rate $p_{\text{drop}} \in [0, 0.3]$.

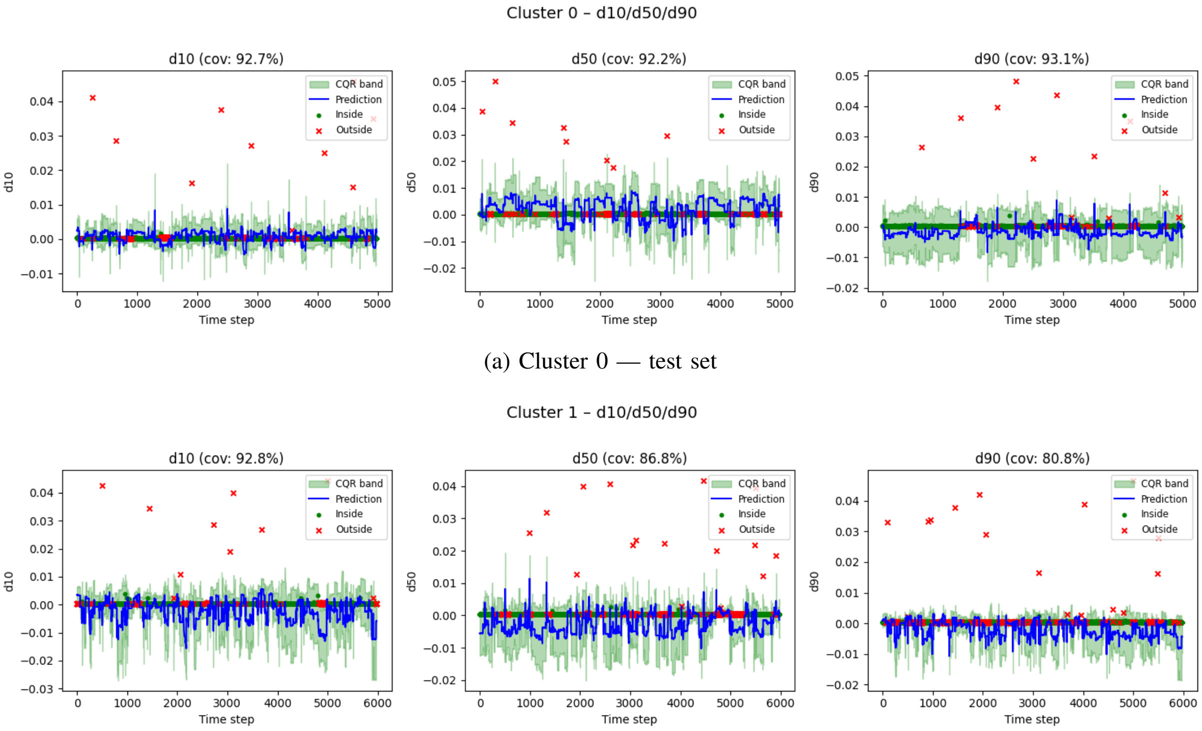
Each candidate network was trained for 150 epochs on the *training* split and evaluated on the *validation* split. The trial with the lowest validation MSE was retained and its weights saved to `narx_cluster_{id}_best.pt`. For Cluster 0 the best configuration was $(H_1, H_2, H_3) = (64, 32, 64)$, LR $= 3.1 \times 10^{-4}$, WD $= 9.4 \times 10^{-6}$, $p_{\text{drop}} = 0.05$; Cluster 1 chose a wider second layer $(H_1, H_2, H_3) = (96, 64, 96)$ with a slightly lower dropout ($p_{\text{drop}} = 0.03$).

B. Task 3 – Verification of tuned models

The best hyper-parameters obtained in Task 1 were fixed and each network was retrained on the union of *train* and *validation* samples, then evaluated on the previously unseen *test* split.

Implementation.

- The scaler statistics saved during tuning are reused, so the test set is projected onto exactly the same feature space.



(a) Cluster 0 — test set

(b) Cluster 1 — test set

Fig. 6: Conformalised 90% prediction bands (light green) for the three particle-size metrics. Blue lines are point forecasts; red × marks lie outside the band. The coverage value in each title is the fraction of test samples inside.

- Training runs for the usual 150 epochs using the tuned learning rate and weight decay; early stopping is disabled so that the final model mirrors the configuration stored in `narx_cluster_{id}_best.pt`.
- After inverse transformation of the predictions, the error metrics using the same definition (MSE) that Optuna minimised during tuning.

These results demonstrate that the lightweight Optuna search delivers a tangible accuracy gain without inflating model size or training time. All tuned weights, scaler parameters and hyper-parameters are stored with the project assets, so the controller can load the most accurate model directly in production.

VIII. Conclusion

This project shows that a light-weight, data-driven workflow can replace complex first-principle models for a slug-flow crystalliser. After cleaning and clustering the lab runs, cluster-specific NARX networks reach $R^2 > 0.99$ on every state. By pairing the residuals with quantile nets and conformal prediction, we obtain 90% error bands whose empirical coverage is within ±2% of the target on unseen data. These results meet all performance goals in the course brief and give a solid base for downstream tasks such as model-predictive control.

Limitations. *(i)* Only one-step forecasts were assessed; long open-loop roll-outs may accumulate error. *(ii)* The conformal bands assume i.i.d. residuals; time-series conformal methods could refine the coverage guarantee. *(iii)* All models are feed-forward; recurrent or transformer architectures might capture longer memories with fewer parameters.

Future work. Extend the models to multi-step prediction, explore other uncertainty tools (e.g. dropout ensembles), and embed the data-driven model in an economic MPC prototype.

Acknowledgment

We would like to express our deepest gratitude to the PAS department for providing us this project opportunity. We also extend our sincere thanks to M.Sc. Florian Fromme, our supervisor, for his guidance, support, and encouragement throughout the duration of this project, and to Prof. Dr.-Ing. Sergio Lucia for offering this project as part of the coursework.

References

[1] Y. Romano, E. Patterson, and E. J. Candès. Conformalized quantile regression. *Adv. NeurIPS 32*, 2019.

[2] P. Termühlen, N. Erkan, and B. Brechtelsbauer. Continuous crystallization in pharmaceutical manufacturing. *Chemical Engineering & Technology*, 44(5):753–768, 2021.

[3] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.