# Supplement

# A Hybrid Deep Learning and Statistical Approach for Fault Detection and Diagnosis in AGRU Systems: Integration with Aspen Plus and Explainable AI

Nadia Khan[1], Syed Ali Ammar Taqvi[2*], Maria Waqas[3]

[1] Department of Polymer and Petrochemical Engineering, Faculty of Chemical and Process Engineering, NED University of Engineering and Technology, University Road, 75270 Karachi, Pakistan
[2] Department of Chemical Engineering, Faculty of Chemical and Process Engineering, NED University of Engineering and Technology, University Road, 75270 Karachi, Pakistan
[3] Department of Computer and Information Systems Engineering, Faculty of Electrical and Computer Engineering, NED University of Engineering and Technology, University Road, 75270 Karachi, Pakistan
[*] Corresponding author, e-mail: aliammar@neduet.edu.pk

**Algorithm S1** Hybrid fault detection and diagnosis algorithm

Step 1: Generate fault and fault-free datasets using Aspen Plus for fault analysis.
Step 2: Split the dataset based on selected features ($x_1$ to $x_{18}$) and fault labels ($F_1$ to $F_6$).
Step 3: Data preprocessing involves normalizing the dataset.
Step 4: Built an LSTM AE/PCA model to detect the fault using the following steps
Divide the normalized data into normal and fault data sets.
Define the threshold of RE, $T^2$, and SPE using a normal data set.
In the PCA space model, calculate the $T^2$ and SPE values using the faulty dataset.
In the LSTM AE model, calculate RE values using the faulty dataset.
Faults are detected when RE, $T^2$, and SPE exceed their statistical thresholds.
Step 5: Tune the hyperparameters using a random search method (e.g., learning rate, batch size, number of units, dropout rate) for bidirectional LSTM /GRU models.
Step 6: Select the best hyperparameters based on performance metrics.
Step 7: Perform $K$-fold cross validation
Set $K$-folds $K = 10$
Split the dataset into $K$ subsets.
Compute the average performance across all $K$ runs.
Step 8: Train the final model using the entire dataset with the best hyperparameters.
Step 9: Test the model.
Step 10: Performance assessment of the model using Accuracy, Precision, Recall, and F1 score.
Step 11: Apply SHAP for a model explanation and compare it with LIME.

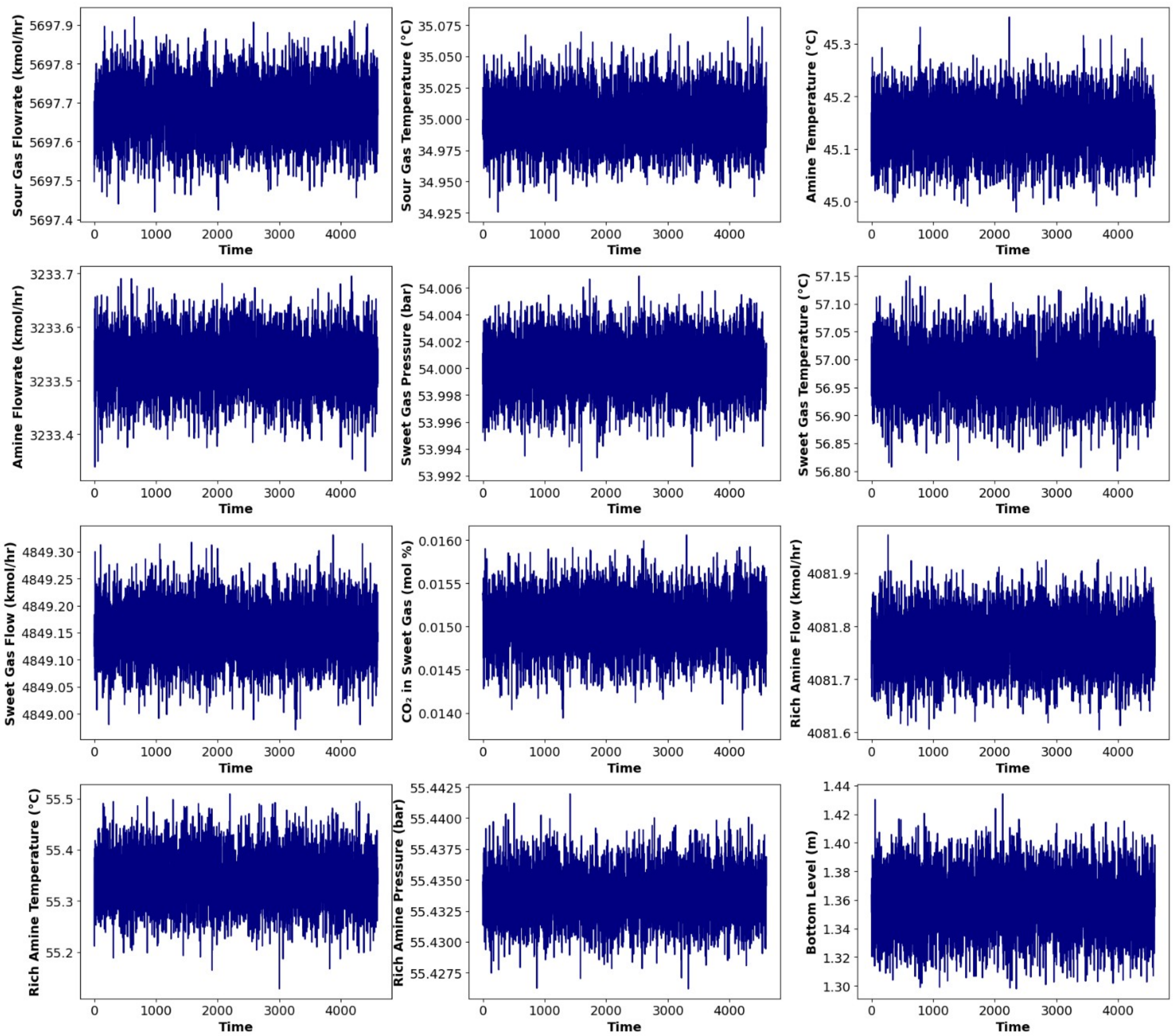


**Fig. S1** Normal operation

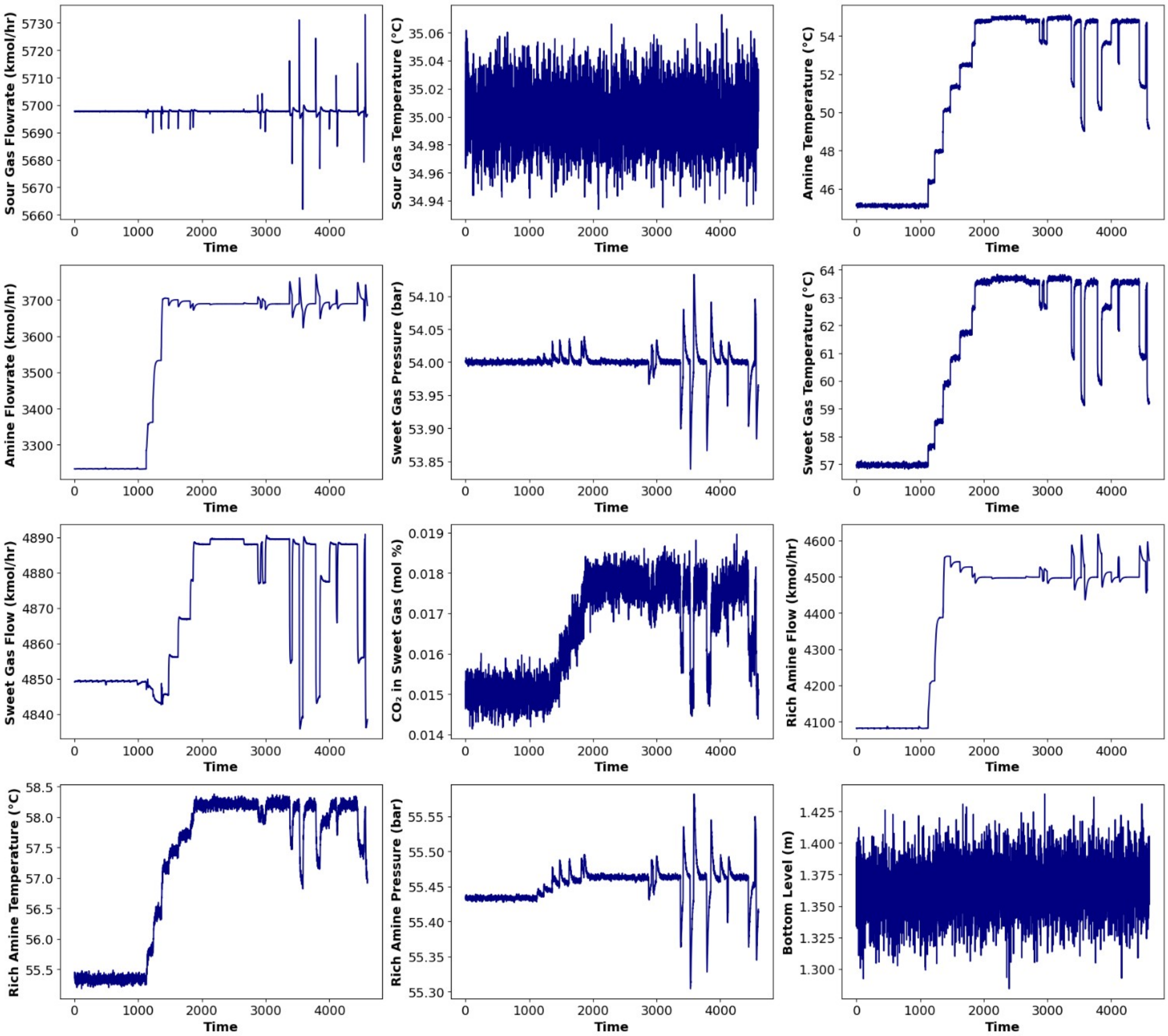


**Fig. S2** Fault 1 cooler fouling

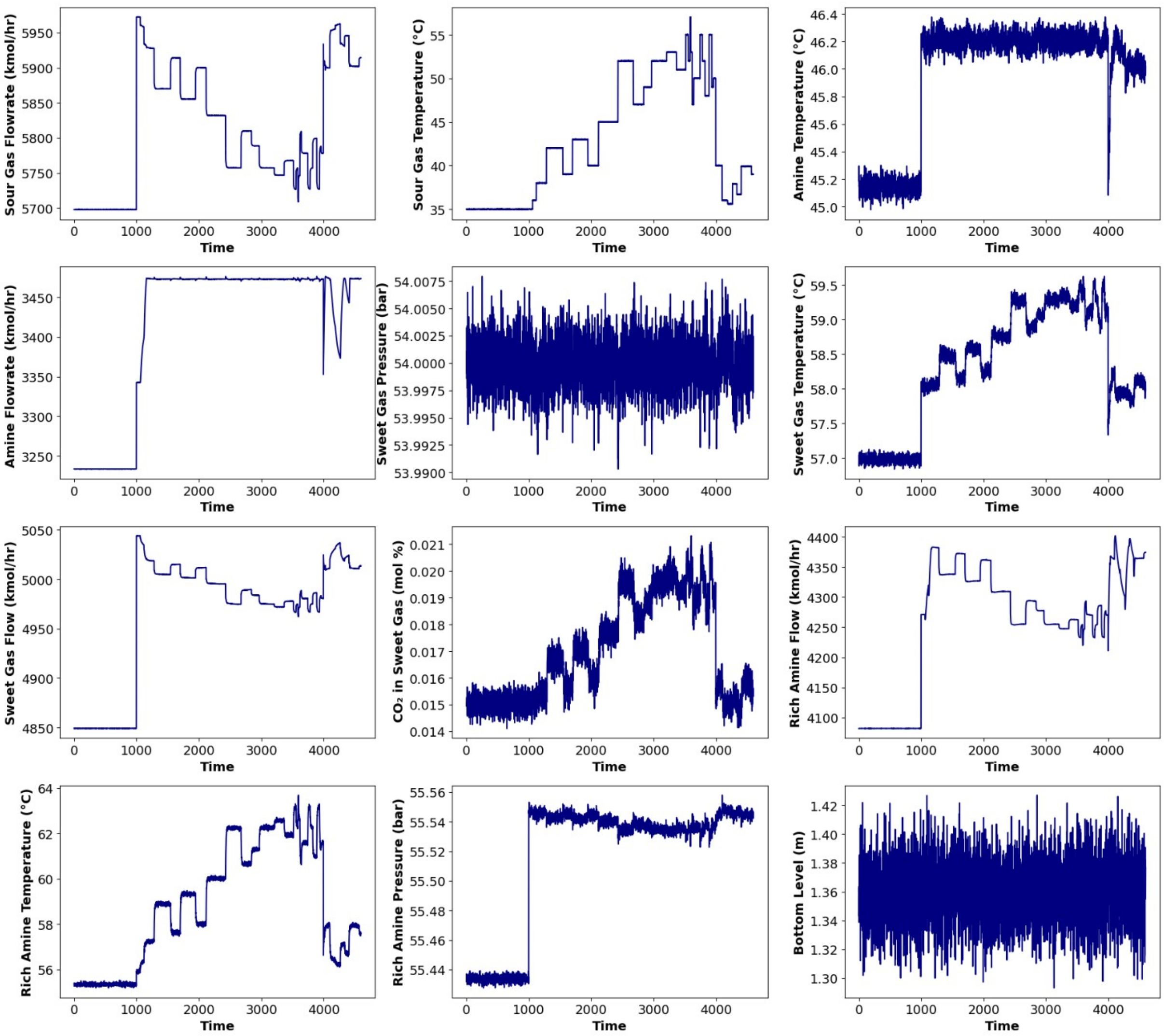


**Fig. S3** Fault 2 high sour gas temperature

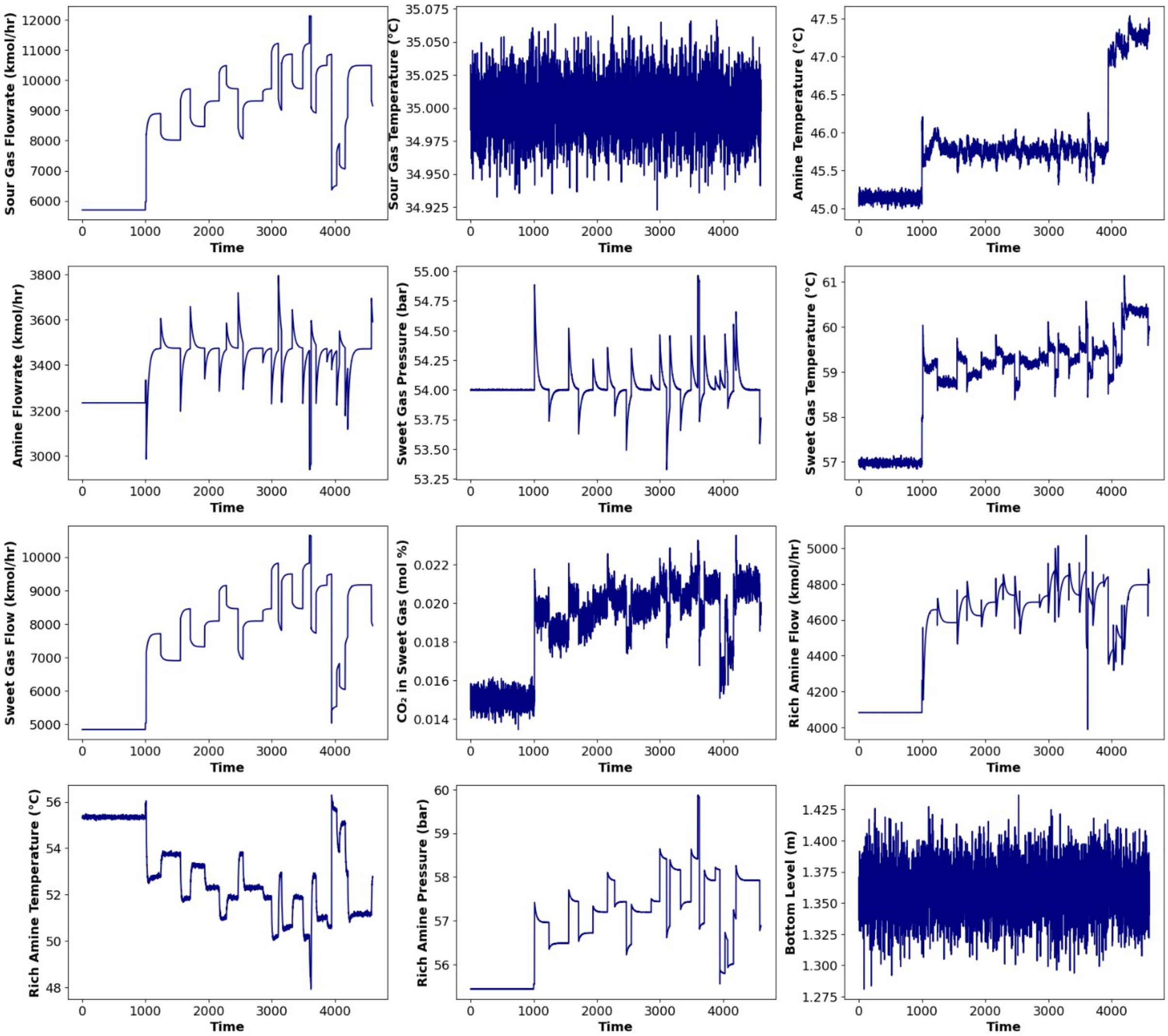


**Fig. S4** Fault 3 high sour gas flow

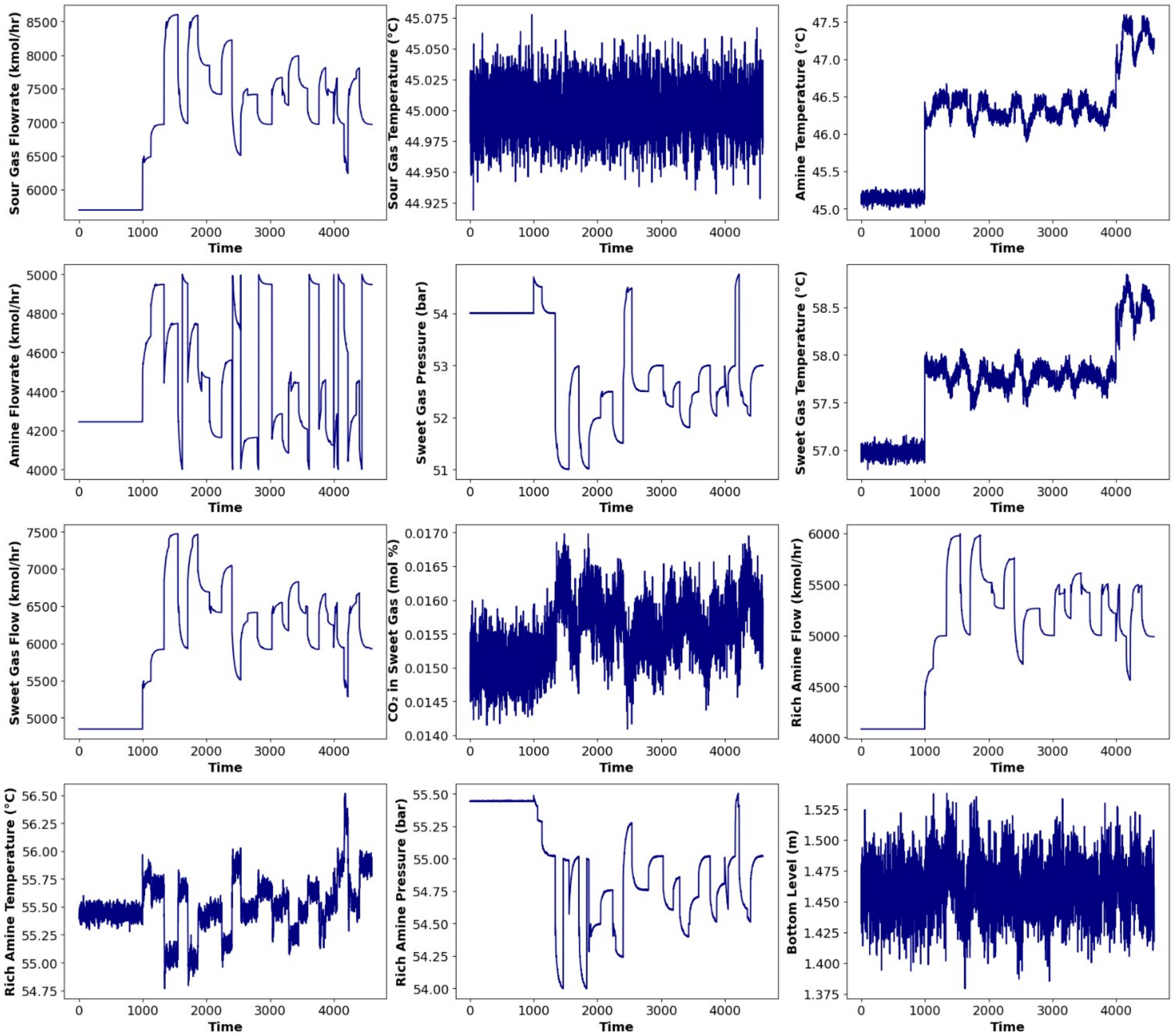


**Fig. S5** Fault 4 increase in differential pressure

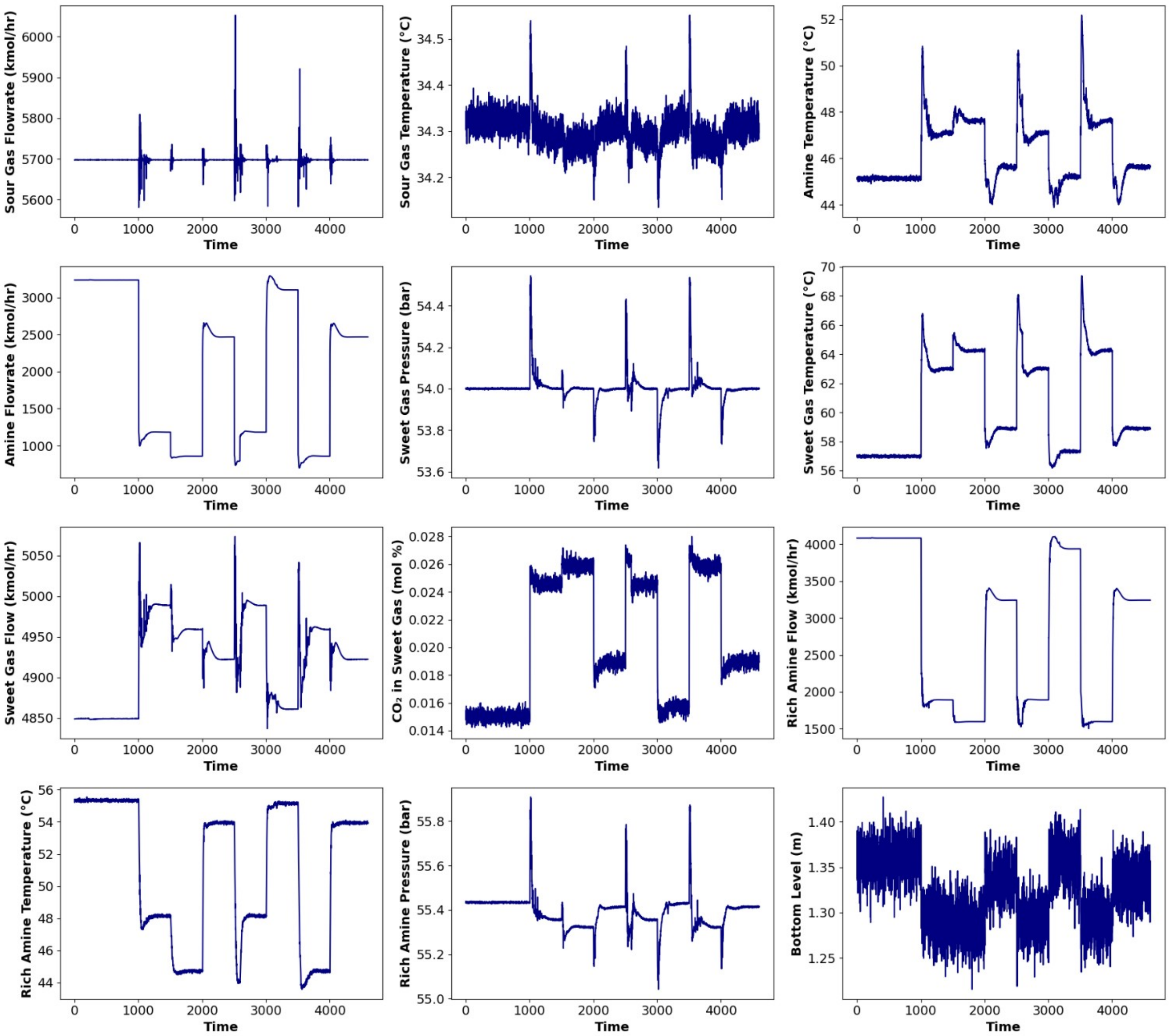


**Fig. S6** Fault 5 loss of amine flow

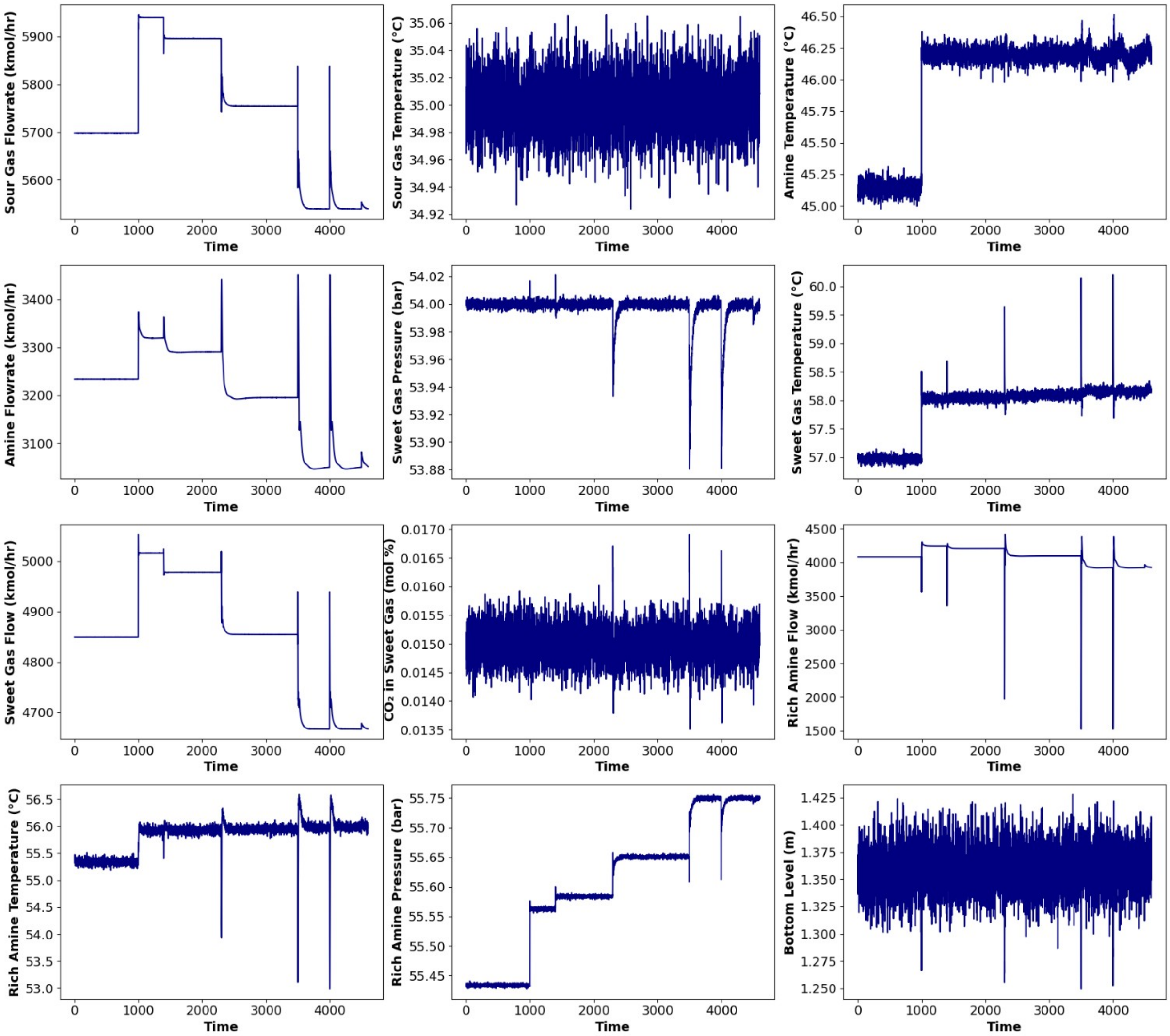


**Fig. S7** Fault 6 damaged tray