Predictive Modelling of Desiccant Drying Processes Using Multi-Feature k-Nearest Neighbours Algorithm

Yasser Algoufily, Francesco Borghesan, Mehmet Mercangöz*

**Corresponding Author's E-mail:(e-mail: m.mercangoz@imperial.ac.uk).*
Department of Chemical Engineering, Imperial College London, London SW7 2AZ UK

Abstract: Desiccant dryers play a critical role in industrial sulphonation processes by ensuring that moisture is effectively removed from the air used during SO_2 to SO_3 conversion. This is necessary to prevent the formation of sulphuric acid, which can harm machinery and lower product quality. This paper introduces a novel approach utilizing Multi-Feature k-nearest neighbours (MF-kNN) forecasting to optimise the drying and regeneration cycles of the dehumidification process units. A key advantage of the MF-kNN model is its ability to perform one-shot forecasts relatively early in the cycle, accurately predicting critical transitions without the need for recursive recalculations. The proposed approach was tested using data from a large-scale surfactant production facility. For forecasting the regeneration cycle endpoint, the model incorporates both the regeneration inlet and outlet air temperatures. Hyperparameter tuning results show that assigning 50% of the feature weight to the inlet temperature results in the lowest forecasting error. Two approaches for data window selection were investigated, namely a moving and an expanding window. The moving window approach outperforms the expanding window approach by 35% and 42% reduced errors for endpoint detection and time-series forecasting tasks respectively. Overall, the model is able to predict the endpoint within a 2 min accuracy with a 400 min lead time on the tested cycles.

Keywords: Desiccant dehumidification; k-nearest neighbours (k-NN); Cyclic process optimisation; Time series forecasting; Industrial drying systems; Energy efficiency.

1. INTRODUCTION

1.1 Process Background

Sulfonation is essential in surfactant production for consumer products (Cornwell, 2017). Surfactants facilitate emulsification by lowering surface tension, improving mixing efficiency. Maintaining process efficiency and product quality is critical, particularly in large-scale industrial applications. To prevent sulfuric acid formation, which is highly corrosive, the air used in SO_2 to SO_3 conversion must be extremely dry.

Desiccant dehumidification is widely used to meet strict drying requirements (Rambhad et al., 2016). This technology relies on materials like silica gel to absorb moisture. The process alternates between drying and regeneration phases, where the desiccant is heated to remove absorbed moisture. Since regeneration is energy-intensive, optimizing its timing directly impacts industrial energy consumption.

In practice, phase transitions in desiccant systems are often determined by empirical rules (Oyieke and Inambao, 2021), leading to inefficiencies—either excess energy consumption or insufficient drying. Common data-driven forecasting methods struggle with long-term predictions due to error accumulation in recursive forecasting. The method proposed in this work leverages the cyclic nature of the process, performing one-shot forecasts early in the cycle to improve phase-switching decisions.

1.2 Time Series Forecasting

Time series forecasting is widely applied in industrial optimization. Classical models such as ARIMA and exponential smoothing (Oliveira and Ludermir, 2016) often struggle with nonlinear and cyclic patterns, particularly in processes influenced by external variables like air humidity (Nawi et al., 2021). Recursive forecasting can suffer from error accumulation, degrading long-horizon predictions (Ben Taieb and Bontempi, 2011).

Machine learning techniques provide powerful alternatives, improving accuracy in complex systems. Approaches like support vector machines (SVMs), random forests, and artificial neural networks (ANNs) have shown success in industrial forecasting (Pandey et al., 2023). The k-nearest neighbors (k-NN) method is particularly effective when data is insufficient for deep learning (Elsayed et al., 2021). Unlike parametric models, k-NN does not assume data linearity, making it well-suited for cyclical processes like desiccant dehumidification (Habtemichael and Cetin, 2016).

Although k-NN is widely used in domains such as pattern recognition, banking, and healthcare (Polat et al., 2007), its industrial applications—particularly in energy optimization and process control—are underexplored. Given the periodic nature of drying and regeneration cycles, k-NN can recognize patterns, adapt to disturbances, and make real-time predictions without recursive calculations.

1.3 Contribution to the Literature

This paper builds upon prior work on k-NN time series forecasting in industrial processes by addressing key limitations. Borghesan et al. developed a k-NN algorithm with weighted averaging for process disturbance forecasting, integrating it into Model Predictive Control (MPC) for continuous processes such as buffer tanks and distillation columns. However, their approach was limited to single-feature applications and did not address periodic or hybrid systems (Borghesan et al., 2019).

In contrast, this paper introduces the following key advancements:

- Developed a Multi-Feature k-NN (MF-kNN) model for improved time series forecasting.
- Introduced "time to threshold" as an additional metric for equipment switching prediction.
- Validated MF-kNN on an industrial desiccant dehumidification system.

Through these advancements, MF-kNN enhances scheduling and control. This results in better resource utilization and reduced energy consumption.

The rest of the paper is organized as follows: Section 2 details the methodology. Section 3 analyses model performance using industrial data. Section 4 presents conclusions and future research directions.

2. METHODOLOGY

2.1 Monovariate k-nearest neighbour

The monovariate k-nearest neighbour (k-NN) algorithm is a non-parametric method used for time series forecasting. It operates by finding the most similar past patterns to the current evolving segment of a time series. This process is beneficial when the data exhibits cyclic behavior, which allows the algorithm to leverage historical patterns to predict future states.

In this approach, at a given time N, the algorithm considers the time series as a memory of N samples, denoted as $y_M(N)$ (Borghesan et al., 2019):

$$y_M(N) = [y(1), y(2), \ldots, y(N)] \tag{1}$$

The segment representing the current state of the time series is defined as the most recent m samples, referred to as the "evolution segment" and denoted as y_E by Borghesan et al. (2019).

$$y_E(N) = [y(N-m+1), y(N-m+2), \ldots, y(N)] \tag{2}$$

The embedding dimension m specifies the number of past observations used to define the current state of the system. The choice of m is important, as it determines how much historical context is considered for identifying similar patterns.

To make a forecast, the k-NN algorithm identifies the k most similar segments in the historical data that match the current evolution $y_E(N)$. The similarity is typically measured using the Euclidean distance between the segments:

$$D\left(y_E(N), y_j\right) = \sqrt{\sum_{i=1}^{m} \left(y_E(i) - y_j(i)\right)^2} \tag{3}$$

Where y_j is a candidate historical segment of the same length m. After identifying the k-nearest neighbours, the algorithm retrieves the time series that follow each of these segments, known as "prediction contributions" y_P^j:

$$y_P^j = [y(r_j+m+1), y(r_j+m+2), \ldots, y(r_j+m+h)] \tag{4}$$

Where r_j is the starting point of the j-th nearest neighbour in the memory and h is the prediction horizon. The final forecast is obtained by averaging these prediction contributions. For the unweighted version of k-NN, the forecast at future time $N+i$ is:

$$\hat{y}(N+i) = \frac{1}{k} \sum_{j=1}^{k} y_P^j(i), \quad i = 1, \ldots, h \tag{5}$$

This method provides a simple and interpretable approach for predicting future values of a monovariate time series. While the algorithm can face challenges with data that includes noise or abrupt disturbances, a weighted version was developed primarily to improve robustness in the choice of k. In this approach, the weights w_j are computed based on the Euclidean distance $D\left(y_E(N), y_j\right)$, with closer neighbours receiving higher weights:

$$w_j = \begin{cases} \dfrac{\max\limits_{\ell=1\ldots k} D(\mathbf{y}_E(N), \mathbf{y}_\ell) - D(\mathbf{y}_E(N), \mathbf{y}_j)}{\max\limits_{\ell=1\ldots k} D(\mathbf{y}_E(N), \mathbf{y}_\ell) - \min\limits_{\ell=1\ldots k} D(\mathbf{y}_E(N), \mathbf{y}_\ell)} & \text{if } k \geq 2, \\ 1 & \text{if } k = 1. \end{cases} \tag{6}$$

- for $j = 1, \ldots, k$ if $k \geq 2$,
- for $j = 1$ if $k = 1$.

Using these weights, each nearest neighbour contributes to the forecast in proportion to its similarity to the current evolution:

$$\hat{y}(N+i) = \frac{\sum_{j=1}^{k} w_j y_P^j(i)}{\sum_{j=1}^{k} w_j}, \quad i = 1, \ldots, h \tag{7}$$

This weighted approach allows for more accurate predictions by prioritizing the most similar historical patterns.

2.2 Data pre-processing

Before applying the k-NN algorithm for time series forecasting, we perform essential data pre-processing steps to ensure the accuracy and reliability of the model forecasts.

These steps include the detection and removal of anomalous cycles, padding cycles to a uniform length, scaling of features, and clustering of similar cycles for enhanced early pattern recognition within the unfolding cycle.

Outlier Detection and Removal Outlier cycles can distort forecasting accuracy and increase computational burden in k-NN. To mitigate this, the Local Outlier Factor (LOF) algorithm (Tang and He, 2017) is used to identify and remove anomalous cycles. LOF assigns an anomaly score based on local density, where cycles with significantly lower density than their neighbors are flagged as outliers.

Cycles exceeding a predefined LOF threshold are removed from the training data to enhance model robustness. Since this study focuses on normal operational states and crossing time prediction rather than fault detection, filtering outliers reduces computational complexity.

Padding of cycles to uniform length Given that the time series data may consist of cycles of varying lengths, we apply a padding step to ensure that all cycles have the same length of 800 time steps. This uniformity is crucial for enabling the k-NN algorithm to efficiently compare cycles and identify similar patterns. The shorter cycles are padded with zeros at the end, up to the desired length of 800, allowing for consistent comparison during forecasting.

Cycle Clustering using K-means To enhance early pattern recognition, cycles are grouped using K-means clustering (Nie et al., 2023). This approach restricts k-NN comparisons to relevant historical cycles. This reduces computational demand by narrowing the search space and enhances forecasting accuracy by excluding cycles that do not belong to the current operational mode.

The algorithm partitions cycles by minimizing the within-cluster sum of squares (WCSS):

$$\mathrm{WCSS} = \sum_{i=1}^{k} \sum_{\mathrm{y} \in \mathrm{C}_i} |\mathrm{y} - \mu_i|^2 \tag{8}$$

where μ_i is the centroid of cluster C_i, defined as the mean position of all cycles within that cluster.

2.3 Multi-feature k-NN forecasting

MF-kNN forecasting extends the standard k-NN algorithm by incorporating multiple features for more accurate and robust predictions in time series forecasting. This is particularly beneficial for chemical processes like desiccant dehumidification, where multiple features simultaneously influence the system's performance.

In this section, we discuss the MF-kNN algorithm, how it handles multiple features, and the integration of weighted Euclidean distances. In MF-kNN, it is important to account for the relative importance of different features. For instance, regeneration inlet temperature may be more impactful than the outlet temperature in forecasting the transition time a desiccant bed. To address this, we utilize two types of weights in our MF-kNN approach:

- **Feature weights** (w_f): These weights adjust the Euclidean distance calculation by emphasizing certain features over others, allowing the model to focus more on key variables.
- **Prediction contribution weights** (w_j): These weights determine the influence of each neighbours in the final prediction, with closer neighbours contributing more significantly to the forecast.

By adjusting the feature weights w_f, the algorithm can emphasize or de-emphasize certain features, resulting in more accurate forecasts.

Feature Scaling The distance metric, to be defined in Eq. (10), used for finding the nearest neighbours in the multi-feature space is sensitive to the scale of input features. Unlike the case with monovariate k-NN, feature scaling is crucial for MF-kNN as this prevents any single or a group of features (e.g. inlet temperature, moisture content) from dominating the distance calculation due to their relative larger magnitude. The substantiation is given by:

$$y' = \frac{y - \mu}{\sigma} \tag{9}$$

where μ is the mean and σ is the standard deviation of the feature under consideration.

Weighted Euclidean Distance The MF-kNN algorithm calculates distances between the current evolution segment y_E and historical segment y_r using a weighted Euclidean distance, which prioritizes more critical features:

$$D(y_E, y_r) = \sqrt{\sum_{j=1}^{p} \sum_{i=1}^{m_j} w_{f,j} \left(y_E^{(j)}(i) - y_r^{(j)}(i) \right)^2} \tag{10}$$

where:

- $w_{f,j}$ is the weight assigned to feature j, adjusting its influence in the distance calculation.
- y_E and y_r are the evolution and historical segments, respectively, each containing p features with m_j time steps.

This formulation ensures that feature importance is considered when selecting the k-nearest neighbours. The forecast is computed using weighted contributions from selected neighbours.

2.4 Expanding vs. Moving Window Approaches for the Embedding Dimension

The embedding dimension in k-NN models determines how many past observations are used to forecast future values. Two approaches are commonly employed: the moving window and the expanding window.

Expanding Window Approach The expanding window progressively incorporates all past observations without discarding earlier data, capturing both short-term fluctuations and long-term trends. The observation set at time t is:

$$\mathbf{y}_E(t) = [y(1), \ldots, y(t)] \tag{11}$$

Moving Window Approach The moving window maintains a fixed number of past observations, shifting forward at each time step by adding the latest observation and discarding the oldest. This ensures the model prioritizes recent data while adapting to changing patterns. The observation set at time t follows the same definition as in Eq.(2), where the most recent m samples are retained.

2.5 Hyperparameter Optimisation

To ensure optimal performance, the MF-kNN model undergoes hyperparameter tuning for k (number of nearest neighbours), m (embedding dimension), w_f (feature weights), and whether weighted averaging is applied. Selecting appropriate values for these parameters is crucial for minimizing forecasting errors.

Hyperparameters The primary hyperparameters include:

- k: Number of nearest neighbours, balancing smoothing and sensitivity.
- m: Embedding dimension, tuned for optimal windowing.
- w_f: Feature weights adjusting Euclidean distance importance.
- Weighted averaging: Determines if neighbours are weighted as in Eq.(6) or equally averaged.

Hyperparameter Tuning Procedure Optimal parameters are selected via grid search, evaluating multiple combinations of k, m, and w_f. The model is trained on randomly selected cycles and validated on unseen cycles. Performance is assessed using forecasting accuracy and time-to-threshold prediction.

Objective Function The objective function J balances forecasting accuracy and threshold timing:

$$J = (1-\beta)\cdot \mathrm{MSE}_{\mathrm{series}} + \beta \cdot \mathrm{MSE}_{\mathrm{threshold}} \tag{12}$$

where β controls the weighting between time-series forecasting error and threshold timing error.

$$\mathrm{MSE}_{\mathrm{series}} = \frac{1}{M}\sum_{i=1}^{M}\left(\frac{y_{t_i}-\hat{y}_{t_i}}{|y_{t_i}|}\right)^2 \tag{13}$$

$$\mathrm{MSE}_{\mathrm{threshold}} = \frac{1}{M}\sum_{i=1}^{M}\left(\frac{t_{\mathrm{true},i}-t_{\mathrm{pred},i}}{|t_{\mathrm{true},i}|}\right)^2 \tag{14}$$

Both terms are normalized to ensure fair weighting, preventing one component from dominating due to scale differences.

3. RESULTS AND DISCUSSION

In this section, we analyse the performance of the MF-kNN algorithm on actual industrial data from a desiccant dehumidification process. The focus is the prediction of time-to-threshold for the regeneration outlet temperature which is key for the control of cycle transition timing.

- **Regeneration Outlet Temperature (ROT)**: This is the target feature of our analysis, representing the key variable that dictates the switch between regeneration and drying phases. The behaviour of this feature is depicted in figure 1.
- **Regeneration Inlet Temperature (RIT)**: This auxiliary feature serves as an input to the model and helps capture the operational conditions influencing the ROT. The corresponding data for RIT is shown in figure 2.

3.1 Outlier detection and removal

The LOF algorithm was applied to ensure MF-kNN is trained only on representative data. The cycles identified as outliers were removed to improve the robustness and reliability of the MF-kNN model.

3.2 Bimodal behaviour

After removing the outlier cycles detected using the LOF algorithm, the remaining cycles exhibited a distinct bimodal behaviour in their outlet temperature profiles, as shown in figure 1.

The first mode peaks between 350 and 400 minutes, while the second mode peaks between 400 and 500 minutes. This dual-peak behaviour suggests the existence of two distinct operational profiles or modes.

The dashed red line in the figure represents a critical threshold temperature of 40 °C, at which the system must switch from the regeneration to the drying phase. As illustrated in figure 1, the cycles vary in the time taken to cross this threshold, which underscores the importance of accurate forecasting methods such as the MF-kNN developed and implemented in this work.

To investigate the bimodal behaviour in the desiccant dehumidification cycles, we employed the K-means clustering algorithm to group the cycles based on three key features: cycle length, peak position, and peak height. These features were selected as they capture the essential dynamics of each cycle. The clustering algorithm identified two distinct groups, which are represented in figures 1 and 2, distinguished by shades of grey as indicated in the legend.

- **Cluster 1**: Cycles in this cluster tend to have shorter lengths, with peak positions occurring earlier in the process and lower peak heights.
- **Cluster 2**: This cluster contains longer cycles, with peaks occurring later and at a higher temperature.

These insights help identify the system's operational mode, enabling forecasts based solely on relevant cycles. This targeted approach reduces computational cost and eliminates irrelevant cycles, leading to improved accuracy.

3.3 Forecasting Results

The hyperparameter tuning results highlight the advantage of MF-kNN over monovariate k-NN. Incorporating the regeneration inlet air temperature alongside the target feature (outlet air temperature) significantly improved

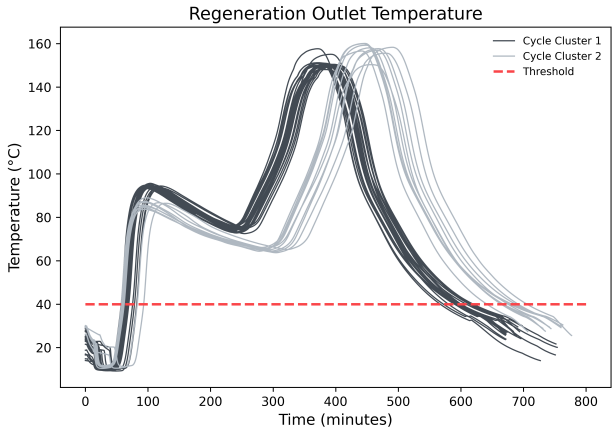

Fig. 1. Regeneration outlet temperature profile. The dashed red line indicates the critical threshold of 40°C.

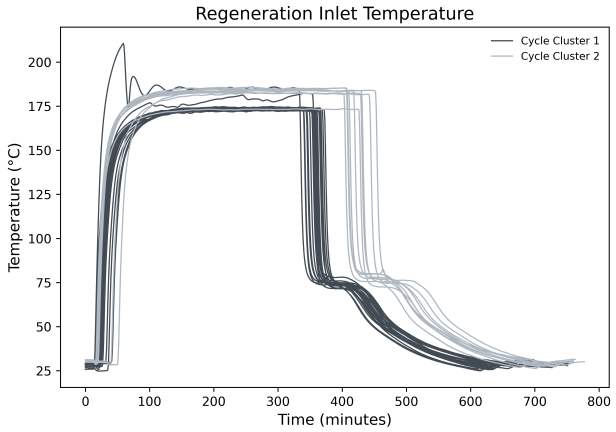

Fig. 2. Regeneration Inlet temperature profile

prediction accuracy. The tuning process evaluated various configurations of the number of neighbours k, embedding dimensions m, feature weights w_f, and weighted averaging, as shown in Table 1.

Separate hyperparameter tuning experiments were conducted for both the moving window and expanding window approaches. In all cases, the moving window configuration minimized the objective function more effectively than the expanding window approach. These findings underscore the suitability of the moving window for capturing recent trends and improving phase-switching predictions in the desiccant dehumidification process.

The results indicate that assigning equal feature weights (50% to each feature) yielded the lowest errors, demonstrating the value of multi-feature forecasting. In contrast, models with a weight of 0% for the second feature effectively became monovariate k-NN, producing the highest errors. Additionally, embedding dimensions of 75–100 achieved the lowest error rates, while smaller dimensions (5–15) performed poorly due to limited historical context.

The hyperparameter configurations in Table 1 were calculated based on an equal weighting between the two objectives in Eq.(12). However, the selected hyperparameter configuration shown in Figures 3 and 4 prioritizes mini-

Table 1. Cluster 1 Moving Window Hyperparameter Tuning Results

Rank	k	m	w_f	Weighted	Error
1	5	100	[0.5, 0.5]	False	0.01
2	7	100	[0.5, 0.5]	True	0.02
3	7	75	[0.5, 0.5]	True	0.02
⋮	⋮	⋮	⋮	⋮	⋮
598	1	15	[1.0, 0.0]	True	0.34
599	1	5	[1.0, 0.0]	True	0.34
600	1	5	[1.0, 0.0]	False	0.35

mizing the time-to-threshold error by assigning a higher weight to $\text{MSE}_{\text{threshold}}$.

- **Figure 3**: Forecasts for Cycle #31 closely align with actual outlet temperature trends, with later forecasts improving as more data becomes available.
- **Figure 4**: Crossing time predictions converge toward actual values over time, demonstrating improved accuracy as the cycle progresses.

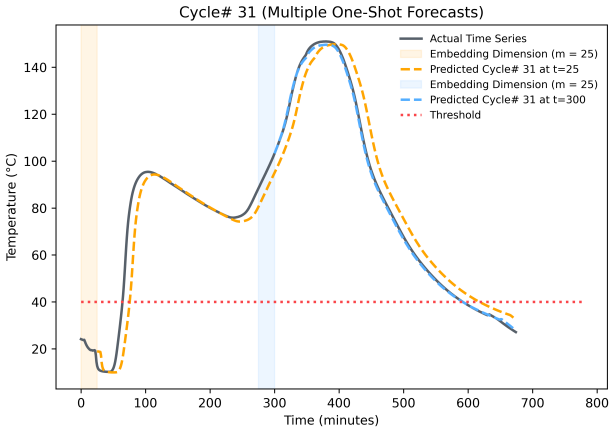

Fig. 3. Actual vs. Predicted Time Series for Cycle #31

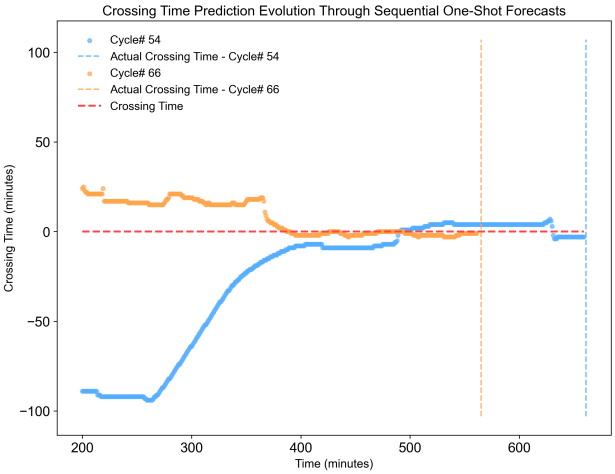

Fig. 4. Evolution of Crossing Time Prediction Through Sequential One-Shot Forecasts

3.4 Discussion of Embedding Dimension Results

The results in Table 2 indicate that the moving window approach delivers lower forecasting errors than the expanding window approach. By dynamically adapting the embedding dimension, the moving window better captures

evolving system dynamics, whereas the expanding window, which retains all past observations, dilutes the relevance of recent patterns, leading to higher errors. However, this improvement comes at the cost of increased distance calculations per prediction, making the moving window computationally more demanding.

Table 2. Comparison of Moving vs Expanding Embedding Dimension

Embedding Dimension Type	**Average Forecasting Error**	
	Time series (°C)	Threshold (min)
Moving	3.60	12.55
Expanding	6.30	20.92

4. CONCLUSION

This paper presented a Multi-Feature k-nearest neighbours (MF-kNN) model for forecasting the transition times in the drying and regeneration cycles in desiccant dehumidification systems. By incorporating both outlet and inlet air temperatures, the MF-kNN model achieved significantly higher accuracy in predicting critical transition points in the cycles compared to traditional, single-feature, k-NN models. The hyperparameter tuning found that assigning equal weights to each feature lead to superior forecasting results.

Additionally, the moving window approach for the embedding dimension proved to be more effective than the expanding window approach, not only in improving overall time series forecasting accuracy but also in accurately forecasting transition times for the desiccant dehumidification system. This adaptability allowed the model to capture variations in system dynamics throughout the drying and regeneration phases, leading to more efficient control of the system and reducing overall energy consumption.

The findings of this study demonstrated that the MF-kNN model, combined with proper hyperparameter tuning and embedding dimension selection, provides a robust and efficient tool for time series forecasting in cyclic industrial processes. Future work may explore extending this approach to other cyclic processes, incorporating additional operational variables, investigating feature selection methods, and integrating the model with real-time control systems to further enhance process optimisation.

REFERENCES

Ben Taieb, S., & Bontempi, G. (2011). Recursive Multi-step Time Series Forecasting by Perturbing Data. In *2011 IEEE 11th International Conference on Data Mining* (pp. 695–704). https://doi.org/10.1109/ICDM.2011.123

Bennett, T., & Myers, J. (2017). Forecasting nonlinear cyclic processes in industrial settings using advanced machine learning methods. *Journal of Industrial Process Control*, 13(2), 85-98.

Borghesan, F., Chioua, M., & Thornhill, N. F. (2019). Forecasting of process disturbances using k-nearest neighbours, with an application in process control. *Computers and Chemical Engineering*, 128, 188–200. https://doi.org/10.1016/j.compchemeng.2019.05.009

Cornwell, P. A. (2017). A review of shampoo surfactant technology: Consumer benefits, raw materials, and recent developments. *International Journal of Cosmetic Science*, 40(1), 16–30.

Elsayed, S., Thyssens, D., Rashed, A., Jomaa, H. S., & Schmidt-Thieme, L. (2021). Do we really need deep learning models for time series forecasting? *arXiv preprint arXiv:2101.02118*.

Habtemichael, F. G., & Cetin, M. (2016). Short-term traffic flow rate forecasting based on identifying similar traffic patterns. *Transportation Research Part C: Emerging Technologies*, 66, 61-78.

Jalali, M. (2015). Energy efficiency improvements and applications of desiccant dehumidification in industrial settings. *Energy Research Journal*, 12(4), 76-89.

Nawi, W. I. A. W. M., Lola, M. S., Zakariya, R., Zainuddin, N. H., Abd Hamid, A. A. K., Aruchunan, E., & Nazzrol, N. S. A. (2021). Improved forecasting of sea surface temperature based on hybrid ARIMA and support vector machines models. *Malaysian Journal of Fundamental and Applied Sciences*, 17(4), 609-620.

Nie, F., Li, Z., Wang, R., & Li, X. (2023). An effective and efficient algorithm for K-means clustering with new formulation. *IEEE Transactions on Knowledge and Data Engineering*, 35(8), 3433-3443.

Oliveira, J. F. L., & Ludermir, T. B. (2016). A hybrid evolutionary decomposition system for time series forecasting. *Neurocomputing*, 180, 27–34.

Oyieke, A. A. Y., & Inambao, F. L. (2021). A review of coupled heat and mass transfer in adiabatic liquid desiccant dehumidification and regeneration systems; advances and opportunities. *International Journal of Low-Carbon Technologies*, 16, 1–20.

Pandey, V. K., Bisoy, S., & Panda, S. (2023). Time series forecasting and prediction of Walmart data using hybrid machine learning techniques. *2023 1st International Conference on Circuits, Power and Intelligent Systems (CCPIS)*, 1-5.

Polat, K., Sahan, S., & Günes, S. (2007). Automatic detection of heart disease using an artificial immune recognition system (AIRS) with fuzzy resource allocation mechanism and k-nearest neighbour method. *Expert Systems with Applications*, 32(2), 625-631.

Rambhad, K. S., Walke, P. V., & Tidke, D. J. (2016). Solid desiccant dehumidification and regeneration methods—A review. *Renewable and Sustainable Energy Reviews*, 59, 73–83.

Tang, J., & He, H. (2017). A local density-based approach for outlier detection. *IEEE Transactions on Knowledge and Data Engineering*, 29(1), 175-187.

Trzciński, K., & Nowak, J. (2019). Predictive control and machine learning in drying systems optimization. *International Journal of Drying Technology*, 35(1), 42-57.

Zhao, W., & Li, X. (2018). Using machine learning techniques to forecast industrial processes with nonlinear cyclic behavior. *IEEE Transactions on Industrial Electronics*, 42(3), 221-235.