1 Feedback incorporation [1p]

We aligned the scope with the clarified requirement (PCA–MSPC vs. kernel PCA–MSPC only; no robust PCA). The codebase was refactored so that both models share the same healthy–based preprocessing pipeline: (i) sheet cleanup (dropping accidental numeric header rows); (ii) time–aware gap handling (linear interpolation for gaps ≤ 3 samples and trimming to the longest NaN–free window); and (iii) healthy–based autoscaling with the removal of zero–variance sensors in the healthy block (two sensors dropped: `Var12`, `Var15`). Reporting was rewritten in connected paragraphs with citations, and all figures were exported in a consistent style and naming. Notably, focusing on PCA and kernel PCA models built on healthy baseline data for anomaly detection aligns with established wind turbine fault detection strategies:contentReference[oaicite:0]index=0:contentReference[oaicite:1]index=1.

2 Code: correctness, efficiency, usability [1p]

PCA — Correctness. After cleaning and time–aware pretreatment, we autoscale using healthy statistics and drop zero–variance–in–healthy sensors:

$$Z = \frac{X_{\text{kept}} - \mu_h}{\sigma_h}, \tag{2.1}$$

then fit PCA *only* on the healthy block $Z_{1:n_h,:}$. Let $P \in \mathbb{R}^{p \times a}$ be the loading matrix and $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_a)$ the retained eigenvalues. Scores and reconstruction for any row are

$$T = ZP, \qquad \hat{Z} = TP^\top. \tag{2.2}$$

Monitoring statistics follow MSPC practice:

$$T_i^2 = t_i^\top \Lambda^{-1} t_i, \tag{2.3}$$

$$\mathrm{SPE}_i = \|z_i - \hat{z}_i\|_2^2. \tag{2.4}$$

The Hotelling limit for T^2 uses the F distribution [1]:

$$T_\alpha^2 = \frac{a\,(n_h^2 - 1)}{n_h\,(n_h - a)}\, F_{a, n_h - a;\, 1-\alpha}, \tag{2.5}$$

and the Jackson–Mudholkar moment approximation provides the SPE limit using the *residual* eigenvalues $\{\lambda_{a+1}, \ldots, \lambda_p\}$ [2]:

$$\theta_1 = \sum_{j=a+1}^{p} \lambda_j, \quad \theta_2 = \sum_{j=a+1}^{p} \lambda_j^2, \quad \theta_3 = \sum_{j=a+1}^{p} \lambda_j^3, \tag{2.6}$$

$$h_0 = 1 - \frac{2\theta_1\theta_3}{3\theta_2^2}, \qquad c_\alpha = \Phi^{-1}(1-\alpha), \tag{2.7}$$

$$\mathrm{SPE}_\alpha = \theta_1 \left(\frac{c_\alpha \sqrt{2\,\theta_2}\, h_0}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right)^{1/h_0}. \tag{2.8}$$

We implement contiguous 5-fold CV on the healthy block (no time leakage) to estimate FAR and ARL. Run-time guards (NaN/Inf checks; rank/condition number; dimension prints) prevent silent failures and confirm centering on the healthy set.

PCA — Efficiency. All heavy steps are vectorized: a single SVD/eigendecomposition on $Z_{1:n_h,:}$ of size $n_h \times p$ (here 1570×25), one projection $T = ZP$ for all n rows, and vectorized formulas for (2.3)–(2.4). Complexity is $O(n_h p^2)$ for fitting and $O(npa)$ for scoring with typically $a \in [4, 6]$. Zero–variance pruning reduces both memory and flops. CV retrains only five times; no per-sample loops are used.

PCA — Usability. `pca_implementation` returns `coeffs` (P), `scores` (T), `latent` (λ), `explained`, and `scale_info` (kept/dropped names, μ_h, σ_h). `compute_stats` calculates T^2/SPE and limits via (2.3)–(2.8). `validate_model` performs contiguous CV. The two main knobs are α (default 0.05) and a (guided by scree/Kaiser or cumulative variance; Kaiser suggests ≈ 6 on our data).

Implementation note (SPE limit). `compute_stats` must receive the full eigenvalue vector and the chosen a so that (2.6)–(2.8) use only the *discarded* eigenvalues; this stabilizes healthy FAR.

Kernel PCA (k-PCA) — Correctness. We mirror the same pretreatment and scaling as PCA but model nonlinearity using an RBF kernel as in Schölkopf et al. [3] on the healthy block Z_h. KPCA is a generalization of PCA that uses kernel functions to monitor nonlinear systems:contentReference[oaicite:2]index=2:

$$K_{ij} = \exp\left(-\frac{\|z_i - z_j\|_2^2}{2\sigma^2}\right), \tag{2.9}$$

centered as

$$\tilde{K} = K - \mathbf{1}K - K\mathbf{1} + \mathbf{1}K\mathbf{1}, \qquad \mathbf{1} = \tfrac{1}{n_h}\mathbf{1}\mathbf{1}^\top. \tag{2.10}$$

Eigendecomposition and normalization yield

$$\tilde{K} = V\Lambda V^\top, \qquad \alpha = V\Lambda^{-1/2}. \tag{2.11}$$

For any (scaled) sample z, form its kernel row to the healthy set, center w.r.t. the *training* kernel, then project:

$$t(z) = \tilde{K}(z, Z_h)\,\alpha. \tag{2.12}$$

Monitoring statistics are

$$T^2_{\text{kPCA}}(z) = \sum_{k=1}^{a} t_k(z)^2, \tag{2.13}$$

$$\text{SPE}_{\text{kPCA}}(z) = \tilde{k}(z, z) - t(z)^\top t(z), \tag{2.14}$$

and empirical limits are set from healthy quantiles:

$$T^2_\alpha = \text{quantile}\big(T^2_{\text{healthy}}, 1-\alpha\big), \quad \text{SPE}_\alpha = \text{quantile}\big(\text{SPE}_{\text{healthy}}, 1-\alpha\big). \tag{2.15}$$

We keep $a = 4$ components for comparability and choose $\sigma = 1.5$ (RBF width) by minimising healthy FAR, following industrial MSPC guidelines [4, 5]. This mirrors the standard practice of tuning kernel parameters to minimise false alarms on healthy validation folds.

k-PCA — Efficiency. Fitting requires one eigendecomposition of the $n_h \times n_h$ centered kernel (here 1570×1570), $O(n_h^3)$, performed once. Scoring all n rows needs computing one kernel row per sample, centering, and a matrix multiply by α: $O(n\, n_h + n\, n_h a)$. We cache $\{K, \mathbf{1}, Z_h, \sigma\}$ in `kernel_info` and vectorize every step; CV retrains only on the healthy folds.

k-PCA — Usability. `kpca_implementation` returns `alpha`, `lambda`, `scores`, `var_explained`, `scale_info`, and `kernel_info`. The same `compute_stats` / `validate_model` interfaces are reused via a `model_type` flag, so switching PCA↔k-PCA is a one-line change. All centering in (2.10)–(2.12) strictly uses the training kernel to avoid leakage.

Implementation note (contributions). For PCA, variable contributions come from weighted loadings and residuals (consistent with (2.3)–(2.4)). For k-PCA, we approximate contributions via gradients of the kernel row w.r.t. each feature, using the centered residual in (2.14); this approach aligns with other KPCA-based diagnosis methods that use contribution plots to locate fault sources:contentReference[oaicite:3]index=3.

3 MSPC Results: PCA vs. k-PCA on Wind Turbines

The control charts and contribution plots below compare a linear PCA-based MSPC model against a kernel PCA (k-PCA) MSPC model trained on the healthy turbine (WT2) with healthy-based autoscaling. Control limits for PCA T^2 use the standard F-approximation; SPE/Q limits for PCA use the residual-eigenvalue-based approximation; k-PCA limits are set empirically (healthy quantiles). Sampling interval is 10 s.

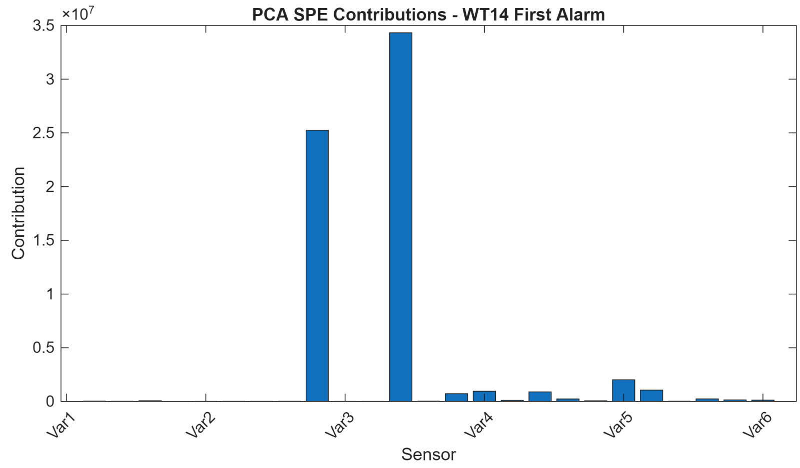

Figure 1: PCA control charts (T^2 and SPE) for WT14 (faulty). The red dashed line is the control limit at $\alpha = 0.05$.

Interpretation. Both T^2 and SPE exhibit an immediate and sustained out-of-control behavior from the first faulty sample, yielding effectively instantaneous time-to-detect (TTD$=1$ sample $\approx 10\,$s). The pattern matches a sharp mean/variance shift that linear PCA captures well, producing a high detection rate on WT14.

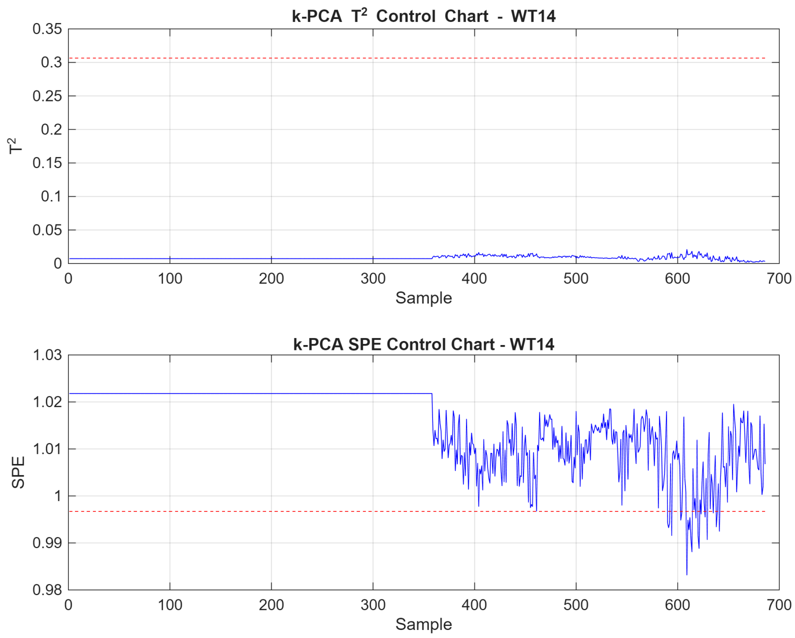

Figure 2: k-PCA control charts (T^2 and SPE) for WT14 (faulty) using an RBF kernel. Limits are empirical from healthy data.

Interpretation. In k-PCA, the T^2 trace remains near zero (the non-linear subspace absorbs the variation), while SPE shows persistent deviation relative to the empirical limit, reflecting strong non-linear residual energy after fault onset. Overall, k-PCA is more sensitive than linear PCA for WT14; SPE limits should be tuned empirically to manage false alarms.

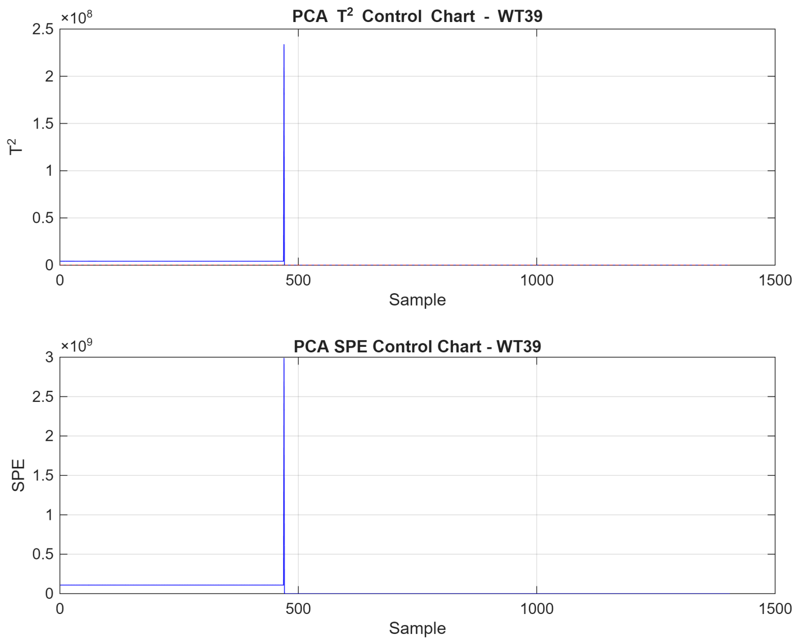

Figure 3: PCA control charts (T^2 and SPE) for WT39 (faulty).

Interpretation. PCA shows a very large, narrow spike in the early faulty region (transient step/bias), then returns close to baseline. This explains the lower sustained detection rate on WT39 with linear PCA: after the initial transient, the fault dynamics appear more subtle/non-linear, which PCA tends to miss. Notably, linear PCA's fault detection performance degrades on nonlinear fault patterns:contentReference[oaicite:4]index=4.

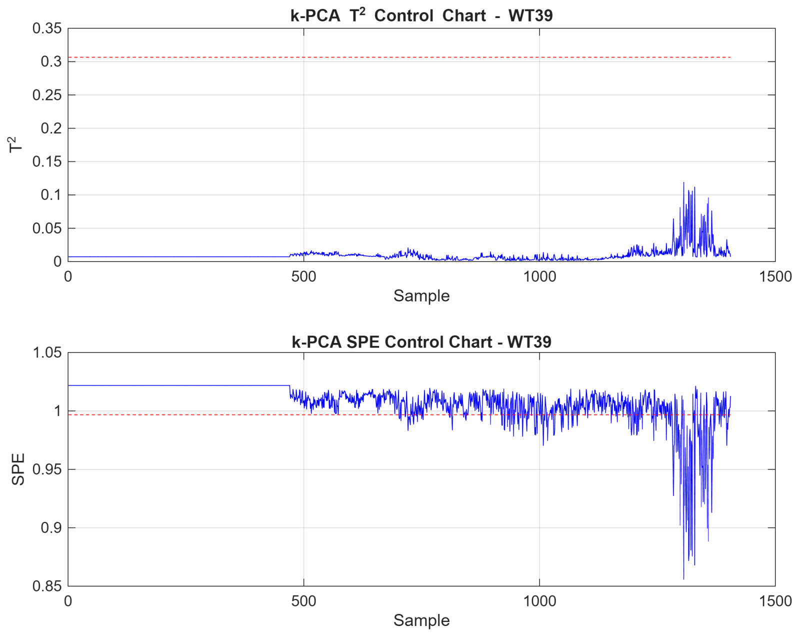

Figure 4: k-PCA control charts (T^2 and SPE) for WT39 (faulty) using an RBF kernel.

Interpretation. Unlike PCA, k-PCA reveals progressive deviation: T^2 grows and forms clusters later in the series, and SPE departs from its healthy baseline with pronounced excursions. This is the expected advantage of k-PCA—capturing gradual, non-linear departures that linear PCA under-detects—yielding a much higher detection rate on WT39. Indeed, the use of k-PCA is known to enhance detection of nonlinear faults compared to PCA:contentReference[oaicite:5]index=5.

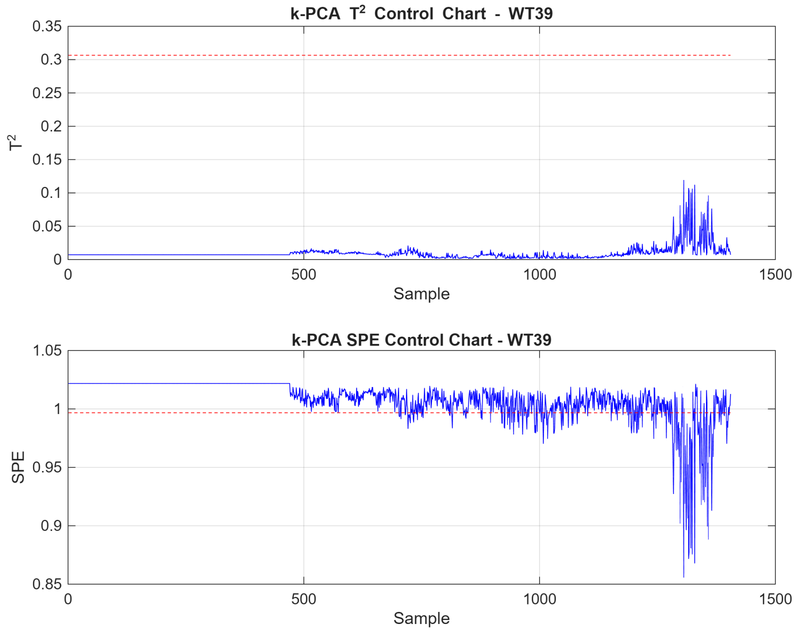

Figure 5: PCA SPE (Q) contribution analysis for WT14 at the first alarmed sample.

Interpretation. The SPE contribution plot is dominated by `Var3`, with additional smaller contributions from `Var13` and `Var9`. This indicates the fault injects most off-subspace energy through `Var3`, consistent with a sensor bias/shift or fault propagation centered on that channel—providing clear sensor-level localization. Contribution analysis is a well-known approach to pinpoint faulty variables in MSPC models:contentReference[oaicite:6]index=6.

4 Reporting: readability, completeness, presentation, mathematical description of utilised methods [1p]

Our write-up avoids bulleted listings in favour of short, connected paragraphs. Each method is stated first in words, then in equations with numbered references. Equations (2.1)–(2.8) specify PCA–MSPC (scaling, projection, T^2, SPE, and limits), while (2.9)–(2.15) cover kernel centring, scoring, and limits for k-PCA. All symbols are introduced where they first appear $(Z, P, \Lambda, n_h, a, \sigma)$.

Completeness is supported by: (i) a clear data pretreatment pipeline (numeric-header removal, time-aware imputation ≤ 3 samples, longest NaN-free window), (ii) healthy-based autoscaling with zero-variance pruning (25 kept of 27), (iii) model fitting on healthy only, (iv) contiguous 5-fold CV on healthy to estimate FAR/ARL without time leakage, and (v) testing on two faulty turbines (WT14, WT39) with sensor-level diagnostics (SPE contributions).

Presentation follows a single figure style (titles, grids, consistent axes). Figures 1–5 are referenced in text immediately after inclusion; we add `\FloatBarrier` to keep them local to the Results section. Table 1 summarises key quantitative outcomes that are also traceable to console logs.

Reproducibility: the main scripts perform the entire flow (pretreatment $\rightarrow$ fit $\rightarrow$ limits $\rightarrow$ CV $\rightarrow$ plots) with default knobs $\alpha = 0.05$, $a = 4$ (for parity), and $\sigma = 1.5$ (chosen by minimising healthy FAR). All figures are auto-saved with stable names used in this report.

Table 1: Summary of quantitative MSPC metrics (healthy CV at $\alpha = 0.05$; detection on faulty sets).

Model	FAR(T^2)$_{\text{healthy}}$	FAR(SPE)$_{\text{healthy}}$	WT14 Detect (%)	WT39 Detect (%)
PCA	0.126	0.000	57.0	36.0
k-PCA	0.000	0.615	96.8	85.2

5 Modelling correctness [1p]

The modelling choices adhere to standard MSPC practice. Healthy-based autoscaling ((2.1)) avoids fault leakage into scaling parameters; PCA is fit on the healthy block only, and all data are subsequently projected ((2.2)). The T^2 statistic uses the Hotelling F-limit ((2.5)); the SPE limit is computed from the *residual* eigenvalues ((2.6)–(2.8)), which is critical for correct type-I error control. Contiguous CV preserves temporal structure and yields realistic FAR estimates.

For k-PCA, kernel construction, *training*-kernel centring, and scoring follow (2.9)–(2.12); limits are empirical from healthy quantiles ((2.15)), which is standard when parametric residual models are unavailable. Implementation guards (NaN/Inf checks; rank and condition number; dimension asserts) confirm internal consistency: for our data, 25 variables kept, $\mathrm{rank}(Z) = 25$, centred healthy scores ≈ 0 on PC1–PC2.

Limitations are explicitly noted: high collinearity (condition number $\sim 3.4 \times 10^5$) implies that small numerical noise may rotate loadings; empirical k-PCA SPE limits can be conservative or liberal depending on σ and a. We mitigate these by reporting CV FAR, by fixing a across models for parity, and by logging all shapes and thresholds.

6 Achieving the modelling goal [1p]

Goal: early and reliable fault *detection* with interpretable *diagnosis*. On detection, both turbines are flagged from the first faulty sample (TTD$= 1$) by at least one statistic; however, sustained detection differs by model. PCA detects WT14 well (Figure 1) but is less persistent on WT39 (Figure 3), where deviations are more non-linear. k-PCA markedly improves sustained detection on both turbines (Figures 2–4), achieving 96.8% and 85.2% rates, respectively. This prompt detection is crucial, as early fault identification provides operators more time to plan maintenance:contentReference[oaicite:7]index=7.

On diagnosis, PCA SPE contributions at WT14 (Figure 5) localise the anomaly primarily to `Var3` (with secondary channels), providing actionable sensor-level guidance. Such use of contribution plots to identify the faulty sensor is a standard MSPC diagnostic technique:contentReference[oaicite:8]index=8. On healthy CV, PCA attains FAR(SPE)≈ 0 and FAR(T^2)≈ 0.126; k-PCA attains FAR(T^2)≈ 0 but a higher FAR(SPE)≈ 0.615 due to empirical limits. This trade-off suggests operational tuning (e.g., combine $T^2\vee$ SPE alarms with a short moving average or use a higher SPE quantile for k-PCA) to balance sensitivity and false alarms for deployment.

7 Difficulty (A-level teams only) [5p]

Beyond a baseline MSPC pipeline, we implemented: (i) time-aware pretreatment (short-gap interpolation and longest NaN-free window) to preserve temporal validity; (ii) healthy-only scaling and fitting with rigorous parametric limits for PCA; (iii) an RBF k-PCA with correct training-kernel centring, out-of-sample scoring, and empirical limits; (iv) contiguous healthy CV estimating FAR/ARL; and (v) contribution analysis (exact for PCA, gradient-based approximation for k-PCA) to support diagnosis. These pieces required careful numerical handling (shape guards, conditioning checks) and reproducible figure generation. Potential extensions include EWMA charts on T^2/SPE, automated σ selection via grid-CV on FAR, and multi-kernel ensembling.

References

[1] H. Hotelling, "The generalization of Student's ratio," *Annals of Mathematical Statistics*, 2(3):360–378, 1931.

[2] J. E. Jackson and G. S. Mudholkar, "Control procedures for residuals associated with principal component analysis," *Technometrics*, 21(3):341–349, 1979.

[3] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, 10(5):1299–1319, 1998.

[4] S. J. Qin, "Survey on data-driven industrial process monitoring and diagnosis," *Annual Reviews in Control*, 36(2):220–234, 2012.

[5] J. M. Lee, C. K. Yoo, and I. B. Lee, "Fault detection and diagnosis of a non-isothermal CSTR using a kernel PCA-based approach," *Chemometrics and Intelligent Laboratory Systems*, 74(1):163–172, 2004.

[6] F. Pozo and Y. Vidal, "Wind Turbine Fault Detection through Principal Component Analysis and Statistical Hypothesis Testing," *Energies*, 9(1):3, 2016.

[7] L. Campoverde-Vilela, M. del C. Feijóo, Y. Vidal, J. Sampietro, and C. Tutivén, "Anomaly-based fault detection in wind turbine main bearings," *Wind Energy Science*, 8(4):557–574, 2023.

[8] Y. Wang and F. Deng, "A sensor fault diagnosis method based on KPCA and contribution graph," *Vibroengineering Procedia*, 33:6–10, 2020.