

University of Wales Trinity Saint David

Wales Institute of Science and Art



Prifysgol Cymru

Y Drindod Dewi Sant

University of Wales

Trinity Saint David

Predictive Maintenance Strategies

Control Technology

ACMA6001

Jordan Hessey

2315713

28/09/2026

Contents

Table of Tables	2
Table of Figures.....	2
1 Introduction	3
2 Maintenance strategies in industrial operations.....	4
2.1 Role of maintenance.....	4
2.2 Reactive maintenance	4
2.3 Preventive maintenance.....	4
2.4 Predictive maintenance	4
3 Condition monitoring and fault diagnosis.....	6
3.1 Condition-monitoring techniques	6
3.2 Statistical and feature-based classification	6
3.3 Neural-network diagnosis	6
3.4 Comparison and synthetic-data limitations	6
4 Case study model and method	8
4.1 Synthetic dataset generation and preparation	8
4.2 Feature-based SVM workflow	8
4.3 Raw-spectrum neural-network workflow	9
5 Comparative results.....	10
5.1 Evaluation design.....	10
5.2 Results	10
6 Discussion.....	12
7 Conclusions and recommendations.....	13
8 References	14
9 Appendices.....	16
9.1 Neural Network code	16
9.2 SVM Code	19

Table of Tables

Table 1 - Case study data summary	9
Table 2 - Classification accuracy by method and evaluation procedure.	10

Table of Figures

Figure 1 - Simulink transmission-casing model (MathWorks 2026)	8
Figure 2 - Classification Learner configuration for five-fold SVM cross-validation. (MathWorks 2026)	9
Figure 3 - Five-fold cross-validation confusion matrices for the linear SVM models. (MathWorks 2026)	10
Figure 4 - Five-fold cross-validation confusion matrices for the raw-spectrum neural networks. (MathWorks 2026)	11

1 Introduction

Maintenance strategy determines how effectively an industrial organisation controls equipment failure, production interruption and operational risk. This is particularly important in a refinery, where failure of a critical pump can interrupt material flow, affect downstream processes and create safety concerns. Predictive maintenance uses measured condition data to identify deterioration before functional failure, allowing maintenance to be planned according to equipment condition rather than a fixed interval. Its effectiveness nevertheless depends on reliable sensors, representative data and diagnostic models capable of distinguishing genuine faults from normal operating variation.

The purpose of this report is to evaluate maintenance strategies and compare conventional and artificial-intelligence-based approaches to fault diagnosis. The case study examines crude-charge pump P-2201A using a MATLAB and Simulink model of its motor, two-speed gearbox, output shaft, casing accelerometer and shaft tachometers. Synthetic vibration data are generated to represent sensor drift, output-shaft wear and gear-tooth damage.

The objectives are to prepare and analyse the simulated dataset, train feature-based linear SVM and raw-spectrum neural-network classifiers, compare their performance using confusion matrices and classification measures, and assess their suitability for improving equipment availability, safety and process quality. Recommendations are then made for further validation and potential industrial implementation.

2 Maintenance strategies in industrial operations

2.1 Role of maintenance

Maintenance ensures that physical assets continue to perform safely, reliably and economically. Its performance affects equipment availability, production cost, safety and product quality rather than repair expenditure alone (Muchiri et al. 2011). For example, (Alsyof 2007) paper-mill case study demonstrated that improved maintenance could increase production and profitability.

The selected strategy should reflect asset criticality and failure consequences. A non-critical component may reasonably operate until failure, whereas crude-charge pump P-2201A requires greater control because failure could interrupt refinery throughput, disturb downstream processes and increase the risk of mechanical damage or loss of containment.

2.2 Reactive maintenance

Reactive maintenance restores an asset after failure. It avoids routine intervention and can be economical for inexpensive, non-critical items with available replacements (Mobley 2002). Its disadvantages are unplanned downtime, emergency labour and possible secondary damage.

Run-to-failure is unsuitable for P-2201A because gearbox or shaft failure could stop crude flow and affect downstream availability. Degradation before failure could also cause unstable flow or vibration, affecting process consistency, while overheating, seal damage or seizure could introduce safety risks. Reactive maintenance is therefore appropriate only when failure consequences are tolerable.

2.3 Preventive maintenance

Preventive maintenance intervenes at calendar or operating-hour intervals. It supports availability by allowing labour, materials and production access to be planned. Scheduled lubrication, seal inspection and protection-system testing can also reduce safety risks.

However, time is an imperfect indication of condition. Conservative intervals may replace healthy parts and consume planned downtime, while long intervals can allow rapidly developing faults to fail between services. (Jardine et al. 2006) Therefore distinguish scheduled maintenance from condition-based decision-making. For P-2201A, preventive tasks remain necessary, but periodic servicing alone may not identify tooth damage, shaft wear or sensor drift developing between inspections.

2.4 Predictive maintenance

Predictive maintenance uses condition data to identify deterioration and plan intervention before functional failure. It can increase availability by providing time to organise spares, labour and a controlled shutdown. It may also protect safety and process quality where mechanical deterioration causes excessive vibration, leakage or unstable flow.

Machine-learning models can combine several condition indicators and recognise patterns that cannot be represented by a single threshold (Carvalho et al. 2019). (Susto et al. 2015) for example, applied multiple classifiers to predictive maintenance in semiconductor manufacturing, demonstrating an industrial use of data-driven maintenance decisions.

Predictive maintenance nevertheless requires sensors, communication systems, data storage, model validation and competent review. False negatives may allow deterioration to progress, whereas repeated false alarms waste maintenance capacity and weaken operator confidence. Under the COMAH framework, controls should be proportionate to major-accident risk (Executive 2015). Predictive monitoring should therefore support rather than replace preventive tasks, protection systems and engineering judgement.

Overall, reactive maintenance offers low initial cost but the greatest exposure to unplanned failure. Preventive maintenance improves planning but may intervene too early or too late. Predictive maintenance provides the strongest opportunity to improve availability, safety and quality but depends on reliable data and validated models. P-2201A should therefore use a layered strategy combining predictive monitoring with scheduled preventive tasks, while reactive repair is reserved for non-critical or unpredictable failures.

3 Condition monitoring and fault diagnosis

3.1 Condition-monitoring techniques

Condition monitoring can use vibration, temperature, acoustic emission, lubricant or wear-debris analysis, motor current and process-performance measurements. The selected technique should reflect the machine, operating conditions and failure mechanism rather than applying one method universally (ISO 2018). Vibration is suitable for P-2201A because gear and shaft faults change drivetrain behaviour, while sensor drift directly affects accelerometer output. Oil-debris and acoustic-emission measurements could provide complementary evidence where vibration signatures overlap (Loutas et al. 2011).

Conventional vibration analysis includes amplitude monitoring, threshold alarms, time-domain statistics, FFT spectra, envelope analysis and time-synchronous averaging. These methods are transparent because changes can be related to shaft rotation, gear meshing or impulsive events. However, their reliability depends on appropriate sampling, knowledge of characteristic fault frequencies and control of speed and load variation (Randall and Antoni 2011). Fixed thresholds can detect significant changes but may generate false alarms when operating conditions vary or miss weaker patterns distributed across several measurements.

3.2 Statistical and feature-based classification

Statistical indicators such as RMS, variance, kurtosis and crest factor reduce vibration records to manageable measures. Feature-based machine learning combines several time-, nonlinear- and frequency-domain indicators. SVMs can perform effectively with relatively small datasets, but their accuracy depends strongly on the relevance of the engineered inputs (Susto et al. 2015). Recent reviews identify SVMs and neural networks as established approaches to rotating-machinery diagnosis, while emphasising that their performance is affected by data quality, operating conditions and feature representation (Das et al. 2023). Feature-based models are relatively interpretable and computationally efficient, but weak fault information may be lost when the original vibration signal is reduced to selected indicators.

3.3 Neural-network diagnosis

Neural networks can learn nonlinear relationships from higher-dimensional vibration data, reducing dependence on manually selected indicators. This may preserve fault information that is lost through feature reduction and reveal patterns not apparent from thresholds or summary statistics (Das et al. 2023). Their limitations include greater computational demand, reduced interpretability and sensitivity to overfitting, class imbalance and changes in operating conditions (Khan and Yairi 2018; Zhao et al. 2019). Independent test data and class-specific measures are therefore required rather than relying on training accuracy alone.

3.4 Comparison and synthetic-data limitations

Conventional methods provide interpretable evidence but depend on engineering knowledge and carefully selected alarm limits. Contemporary classifiers can combine more information and represent nonlinear behaviour but depend strongly on data quality and may be harder to explain. The approaches should therefore be complementary rather than mutually exclusive.

Synthetic simulation supplies known labels because industrial failures are scarce and deliberately damaging refinery equipment is unacceptable. However, differences between training and operational data distributions can reduce the field performance of rotating-machinery classifiers (Zhang et al. 2023). Real measurements contain sensor noise, mounting variation, changing operating conditions, process disturbances and degradation mechanisms that may be absent from

the model. The classifiers should therefore remain decision-support prototypes until validated using representative plant data.

4 Case study model and method

4.1 Synthetic dataset generation and preparation

Synthetic data were used because labelled industrial fault data are scarce, while deliberately damaging refinery equipment would be unsafe and expensive. Simulation also allowed the three conditions to be introduced at known severities and in controlled combinations. However, simulated results may not fully reproduce the noise, operating variation and interacting degradation found on real equipment.

The Simulink transmission model contained a torque drive, drive shaft, clutch, high and low gears, output shaft, casing accelerometer and tachometers. Sensor drift was represented by a DC offset, shaft wear changed output-shaft damping, and tooth damage introduced a disturbance torque at a fixed angular position.

Gridded and random combinations of the three fault parameters were defined using arrays of Simulation Input objects. Producing 208 healthy and faulty simulations. The first 10 seconds of each run were removed to exclude the start-up transient. Vibration and tachometer data were then retained, and Boolean fault labels were derived from the known model parameters using the specified tolerance thresholds. A composite FaultCode from 0 to 7 represented every possible combination of the three conditions.

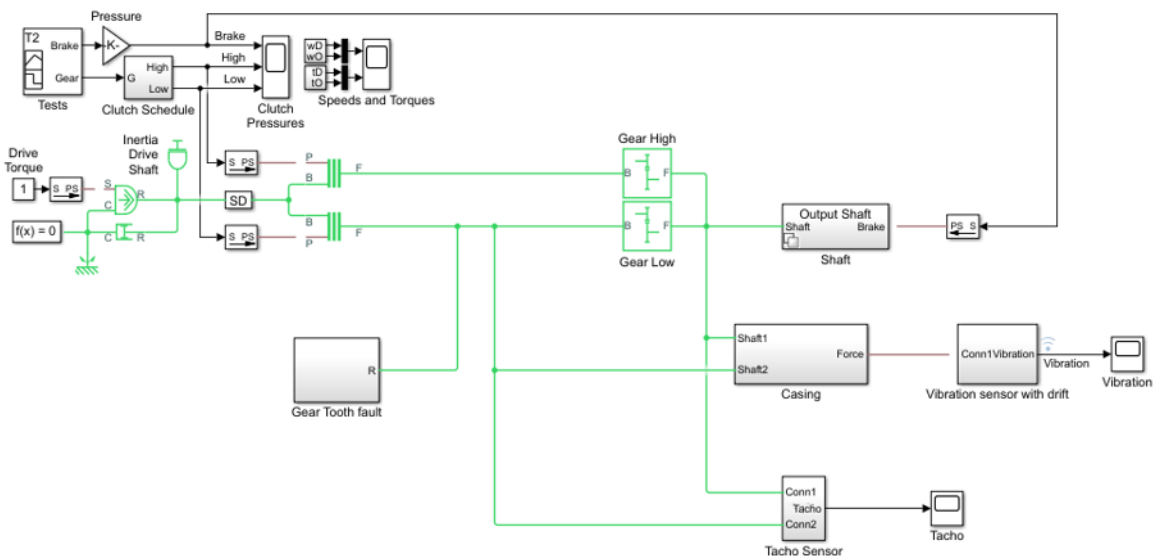


Figure 1 - Simulink transmission-casing model (MathWorks 2026)

4.2 Feature-based SVM workflow

Eighteen time, nonlinear and frequency-domain characteristics were extracted from each simulation. These included mean, RMS, variance, peak-to-peak value, skewness, kurtosis, crest factor, cumulative-sum range, entropy, Lyapunov exponent and TSA-derived spectral indicators. The features were combined with the three Boolean labels and FaultCode, producing a 208×22 dataset.

Neighbourhood component analysis was used to examine feature importance. Separate linear SVM models were then developed in Classification Learner for sensor drift, shaft wear and tooth condition using five-fold cross-validation.

Data Set

Data set variable: [How to prepare data for import](#)
 featureData 208x22 table

Response

☒ From data set variable
☐ From workspace
 ShaftWear logical 2 unique

Predictors

	Name	Type	Range
☒	SigCorrDimension	double	0.919431 .. 1.15153
☒	SigApproxEntropy	double	0.0315649 .. 0.0384987
☒	SigLyapExponent	double	-0.0979127 .. 139.845
☒	PeakFreq	double	0 .. 9.99803
☒	HighFreqPower	double	4.91556e-12 .. 6.75299e-06
☒	EnvPower	double	8.23108e-08 .. 3.23488e-07
☒	PeakSpecKurtosis	double	162.125 .. 230.393
☐	SensorDrift	logical	2 unique
☐	ShaftWear	logical	2 unique
☐	ToothFault	logical	2 unique
☐	FaultCode	double	0 .. 7

Add All Remove All

Validation

Validation scheme: [Read about validation](#)
 Cross-validation
 Cross-validation folds: 5

Protects against overfitting. For data not set aside for testing, the app partitions the data into folds and estimates the accuracy on each fold.

Test

Data selection method: [Read about test data](#)
 No test data

Do not set aside any data for testing. To import a test set instead of partitioning the current data set, use the Import Test Data button on the Test tab after starting an app session.

Refresh Start Session Cancel

Figure 2 - Classification Learner configuration for five-fold SVM cross-validation. (MathWorks 2026)

4.3 Raw-spectrum neural-network workflow

For the neural-network approach, each steady-state vibration signal was resampled to 1,024 points and transformed using the FFT. The first 512 log-scaled magnitude values were retained as inputs. Three binary networks were trained, each containing one hidden layer of 20 ReLU neurons.

A reproducible 70/15/15 division produced 146 training, 31 validation and 31 test observations. Normalisation was calculated using the training data, while class weighting and early stopping reduced the risks associated with imbalance and overfitting. Five-fold cross-validation was also conducted to provide a comparable evaluation against the SVM models.

Table 1 - Case study data summary

Case-study item	Value
Simulation runs	208
Engineered dataset	208 x 22 (18 features & 4 variables)
Neural-network input	208 x 512 spectral bins
Train / validation / test	146 / 31 / 31
Faults	Sensor drift, shaft wear, gear-tooth damage

5 Comparative results

5.1 Evaluation design

The feature-based approach used 18 engineered time-domain, nonlinear-domain and frequency-domain indicators as inputs to separate linear SVM models. The features were standardised during training and evaluated using five-fold cross-validation in MATLAB Classification Learner (MathWorks 2026). Each observation was therefore predicted by a model that had not been trained on that fold.

The neural-network approach instead used 512 log-scaled FFT magnitude values from each vibration signal. Initial evaluation used a reproducible 70/15/15 division, producing 146 training, 31 validation and 31 held-out test observations. Normalisation parameters were calculated from the training data and applied to the validation and test data. Five-fold cross-validation was subsequently used across all 208 observations to provide a more comparable assessment against the SVM models.

Accuracy summarised overall performance, while the confusion matrices identified false positives and false negatives. The 31-observation neural-network test was retained as an additional check but was not compared directly with the cross-validated SVM results because it used a smaller and different partition.

5.2 Results

Table 2 - Classification accuracy by method and evaluation procedure.

Fault	Linear SVM CV	Neural network CV	NN held-out test
Sensor drift	92.8%	95.7%	90.3%
Shaft wear	59.1%	73.1%	51.6%
Tooth fault	95.2%	100.0%	100.0%

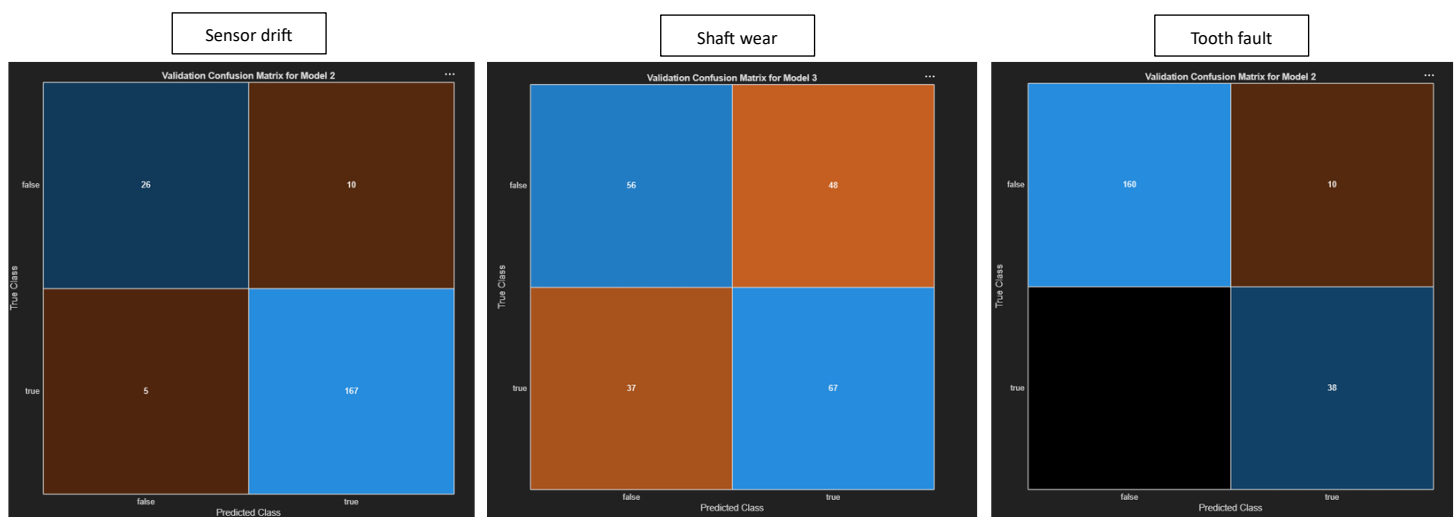


Figure 3 - Five-fold cross-validation confusion matrices for the linear SVM models. (MathWorks 2026)

The neural networks achieved higher cross-validation accuracy for every condition. Compared with the SVMs, accuracy increased by 2.9 percentage points for sensor drift, 14.0 points for shaft wear and 4.8 points for tooth damage. This suggests that the raw spectra retained useful fault information that was not fully represented by the engineered features.

Shaft wear remained the most difficult condition. The SVM produced 37 false negatives and 48 false positives, whereas the neural network reduced these to 31 and 25 respectively. The neural network consequently achieved 73.1% accuracy and a 72.3% F1-score, compared with 59.1% SVM accuracy. Sensor-drift performance was strong for both models, while the neural network correctly classified all tooth-fault observations.

The feature table encoded the tooth-condition Boolean in the opposite direction from the neural-network label. The SVM matrix was therefore interpreted after aligning the class meaning. This did not alter its 95.2% accuracy but ensured that missed faults and false alarms were identified correctly.

The cross-validation results provide the fairest available comparison, although the models used different inputs and were not necessarily evaluated using identical fold assignments. The results therefore demonstrate comparative performance on the simulated dataset rather than definitive superiority in industrial operation.

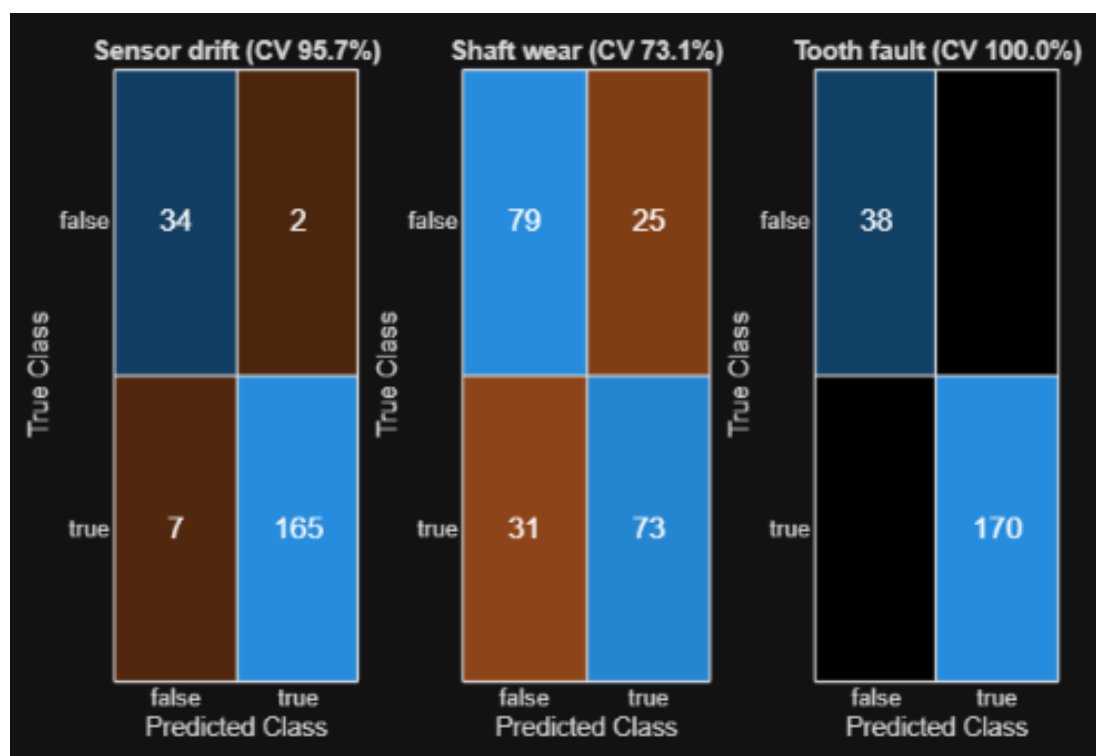


Figure 4 - Five-fold cross-validation confusion matrices for the raw-spectrum neural networks. (MathWorks 2026)

6 Discussion

Shaft wear was the most difficult condition for both approaches. The linear SVM achieved 59.1% cross-validation accuracy, while the raw-spectrum neural network achieved 73.1%. Shaft wear changes output-shaft damping and may produce broad vibration changes that overlap with healthy operation rather than a distinctive fault frequency. This difficulty is consistent with research showing that machine operating conditions can alter vibration features and complicate fault classification (Randall and Antoni 2011). The neural network's stronger result suggests that the 512-bin spectrum retained information that was weakened when the signal was reduced to selected indicators. Conversely, tooth damage produced a repeatable rotational disturbance, creating a clearer spectral pattern and 100% neural-network accuracy.

The neural network missed 31 worn-shaft cases and incorrectly classified 25 healthy cases as worn. A false negative could allow degradation to progress until heat, seal damage or seizure threatens availability and containment. A false positive may prompt unnecessary inspection or controlled shutdown, while repeated alarms could weaken operator confidence. (Jardine et al. 2006) explain that condition-based maintenance decisions must connect diagnosis to the operational consequences of intervention and failure. Any future decision threshold should therefore reflect equipment criticality and the relative cost of missed faults and false alarms rather than accuracy alone.

The results support the layered maintenance strategy discussed in Section 2. Availability could improve through earlier notification, allowing the standby pump, labour and long-lead gearbox parts to be prepared. Safety requires conservative escalation of shaft and gear indications, confirmation that the accelerometer remains reliable, and retention of independent protection systems and inspections. Product quality may be affected if drivetrain deterioration causes unstable crude-feed flow or downstream process variation. Using operational data from UK water-pumping equipment, (Upasane et al. 2023). demonstrated that interpretable fault predictions can support informed maintenance decisions and improve user acceptance. For P-2201A, the classifier should therefore be presented alongside vibration trends and sensor-health information rather than as an unexplained automated alarm.

The models should also complement the conventional techniques discussed in Section 3. Signal trends, FFT spectra and fault-frequency information provide interpretable evidence, while machine learning supports consistent analysis of larger datasets. Reviews of machine-health monitoring recommend this type of validation because high model accuracy under controlled conditions does not demonstrate reliable field performance (Khan and Yairi 2018).

The dataset remains small, simulated and imbalanced for sensor drift and tooth condition. The held-out division was not stratified by FaultCode, while the models used different input representations and potentially different cross-validation folds. Deployment should therefore begin as engineer-reviewed decision support, followed by validation against representative plant vibration, operating-state and maintenance records.

7 Conclusions and recommendations

Predictive maintenance is appropriate for P-2201A because advance warning could support controlled intervention and reduce exposure to an expensive unplanned outage. The raw-spectrum neural networks outperformed the feature-based linear SVMs in five-fold cross-validation, achieving 95.7% accuracy for sensor drift, 73.1% for shaft wear and 100% for tooth damage. However, shaft wear remained difficult for both approaches and should be prioritised for model improvement.

A pilot deployment is recommended in which the neural-network outputs support, rather than replace, engineering decisions. The classifier should be combined with conventional vibration trends, fault-frequency analysis, sensor-health checks and existing protection systems. This layered approach treats diagnosis as an input to maintenance decision-making rather than an isolated automated decision (Jardine et al. 2006). Explainable outputs should also be considered because evidence from operational pumping equipment indicates that interpretable predictions can improve the acceptance and usefulness of predictive-maintenance systems (Upasane et al. 2023).

The principal limitations are the small, simulated dataset, class imbalance, non-stratified held out split and absence of representative plant noise and operating variation. Reviews of machine-health applications warn that strong performance on controlled data does not demonstrate reliable field operation (Khan and Yairi 2018). Further work should therefore use stratified partitions, identical cross-validation folds, additional measurement noise and plant vibration data. Decision thresholds should also be tested against the operational costs of false alarms and missed faults before maintenance actions are automated.

8 References

- Alsyouf, I. 2007. The role of maintenance in improving companies' productivity and profitability. *International Journal of Production Economics* 105(1), pp. 70–78. doi: 10.1016/j.ijpe.2004.06.057.
- Carvalho, T.P., Soares, F.A.A.M.N., Vita, R., Francisco, R. da P., Basto, J.P. and Alcala, S.G.S. 2019. A systematic literature review of machine learning methods applied to predictive maintenance. *Computers & Industrial Engineering* 137, p. 106024. doi: 10.1016/j.cie.2019.106024.
- Das, O., Das, D.B. and Birant, D. 2023. Machine learning for fault analysis in rotating machinery: A comprehensive review. *Heliyon* 9, p. e17584. Available at: <https://doi.org/10.1016/j.heliyon.2023.e17584>.
- Executive, H. and S. 2015. *The Control of Major Accident Hazards Regulations 2015*. London: The Stationery Office. Available at: <https://www.legislation.gov.uk/uksi/2015/483/contents>.
- ISO. 2018. *ISO 17359:2018 Condition monitoring and diagnostics of machines: General guidelines*. Geneva: International Organization for Standardization. Available at: <https://www.iso.org/standard/71194.html>.
- Jardine, A.K.S., Lin, D. and Banjevic, D. 2006. A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing* 20(7), pp. 1483–1510. doi: 10.1016/j.ymssp.2005.09.012.
- Khan, S. and Yairi, T. 2018. A review on the application of deep learning in system health management. *Mechanical Systems and Signal Processing* 107, pp. 241–265. doi: 10.1016/j.ymssp.2017.11.024.
- Loutas, T.H., Roulias, D., Pauly, E. and Kostopoulos, V. 2011. The combined use of vibration, acoustic emission and oil debris on-line monitoring towards a more effective condition monitoring of rotating machinery. *Mechanical Systems and Signal Processing* 25(4), pp. 1339–1352. doi: 10.1016/j.ymssp.2010.11.022.
- MathWorks. 2026. generateSimulationEnsemble: Generate ensemble data by running a Simulink model. Available at: <https://www.mathworks.com/help/predmaint/ref/generatesimulationensemble.html>.
- Mobley, R.K. 2002. *An Introduction to Predictive Maintenance*. 2nd ed. Oxford: Butterworth-Heinemann.
- Muchiri, P., Pintelon, L., Gelders, L. and Martin, H. 2011. Development of maintenance function performance measurement framework and indicators. *International Journal of Production Economics* 131(1), pp. 295–302. doi: 10.1016/j.ijpe.2010.04.039.
- Randall, R.B. and Antoni, J. 2011. Rolling element bearing diagnostics - A tutorial. *Mechanical Systems and Signal Processing* 25(2), pp. 485–520. doi: 10.1016/j.ymssp.2010.07.017.
- Susto, G.A., Schirru, A., Pampuri, S., McLoone, S. and Beghi, A. 2015. Machine Learning for Predictive Maintenance: A Multiple Classifier Approach. *IEEE Transactions on Industrial Informatics* 11(3), pp. 812–820. doi: 10.1109/TII.2014.2349359.
- Upasane, S.J., Hagra, H., Anisi, M.H., Savill, S., Taylor, I. and Manousakis, K. 2023. A Type-2 Fuzzy Based Explainable AI System for Predictive Maintenance within the Water Pumping Industry. *IEEE*

Transactions on Artificial Intelligence 5(2), pp. 490–504. Available at: <https://doi.org/10.1109/TAI.2023.3279808>.

Zhang, S., Su, L., Gu, J., Li, K., Zhou, L. and Pecht, M. 2023. Rotating machinery fault detection and diagnosis based on deep domain adaptation: A survey. *Chinese Journal of Aeronautics* 36(1), pp. 45–74. Available at: <https://doi.org/10.1016/j.cja.2021.10.006>.

Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P. and Gao, R.X. 2019. Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing* 115, pp. 213–237. doi: 10.1016/j.ymssp.2018.05.050.

9 Appendices

9.1 Neural Network code

Sensor drift

```
rng(0)
sensorDriftNet = fitcnet( ...
    normalisedSpectra(trainingRows,:), driftTrainingLabels, ...
    'Weights',          driftWeights, ...
    'LayerSizes',       20, ...
    'Activations',      'relu', ...
    'Lambda',           1e-3, ...
    'Standardize',       false, ...    % we already normalised in Step 3
    'ValidationData',    {normalisedSpectra(validationRows,:),
sensorDriftLabels(validationRows)}, ...
    'ValidationPatience', 10, ...
    'IterationLimit',     2000, ...
    'ClassNames',        [false; true]);

% Predict on the unseen test rows and report accuracy
driftPredictions = predict(sensorDriftNet, normalisedSpectra(testRows,:));

driftTestAccuracy = mean(driftPredictions == sensorDriftLabels(testRows));
driftTrainAccuracy =
mean(predict(sensorDriftNet, normalisedSpectra(trainingRows,:)) ==
driftTrainingLabels);

fprintf('SensorDrift test accuracy: %5.1f%% (training: %5.1f%%)\n', ...
    100*driftTestAccuracy, 100*driftTrainAccuracy);
figure('Name', 'Per-fault confusion matrices (raw spectra)')
confusionchart(sensorDriftLabels(testRows), driftPredictions, ...
    'Title', 'Sensor drift')
```

Shaft wear

```
shaftTrainingLabels = shaftWearLabels(trainingRows);

shaftWeights = zeros(size(shaftTrainingLabels));
shaftWeights(shaftTrainingLabels) = 0.5 / sum(shaftTrainingLabels);
shaftWeights(~shaftTrainingLabels) = 0.5 / sum(~shaftTrainingLabels);

rng(0)
shaftCrossValModel = fitcnet( ...
    normalisedSpectra(trainingRows,:), shaftTrainingLabels, ...
    'Weights',          shaftWeights, ...
```

```

    'LayerSizes',      20, ...
    'Activations',     'relu', ...
    'Lambda',          1e-3, ...
    'Standardize',     false, ...
    'ValidationData',   {normalisedSpectra(validationRows,:),
shaftWearLabels(validationRows)}, ...
    'ValidationPatience', 10, ...
    'IterationLimit',    2000, ...
    'ClassNames',       [false; true]);

% Predict on the unseen test rows and report accuracy
shaftCrossValPredictions = predict(shaftCrossValModel,
normalisedSpectra(testRows,:));

shaftTestAccuracy = mean(shaftCrossValPredictions ==
shaftWearLabels(testRows));
shaftTrainAccuracy =
mean(predict(shaftCrossValModel,normalisedSpectra(trainingRows,:)) ==
shaftTrainingLabels);

fprintf('Shaft wear test accuracy: %5.1f%% (training: %5.1f%%)\n', ...
    100*shaftTestAccuracy, 100*shaftTrainAccuracy);
Shaft wear test accuracy: 51.6% (training: 61.6%)

figure('Name','Per-fault confusion matrices (raw spectra)')
confusionchart(shaftWearLabels(testRows), shaftCrossValPredictions, ...
    'Title', 'Shaft Wear')

```

Gear Tooth fault

```

toothTrainingLabels = toothFaultLabels(trainingRows);

toothWeights = zeros(size(toothTrainingLabels));
toothWeights(toothTrainingLabels) = 0.5 / sum(toothTrainingLabels);
toothWeights(~toothTrainingLabels) = 0.5 / sum(~toothTrainingLabels);

rng(0)
toothCrossValModel = fitcnet( ...
    normalisedSpectra(trainingRows,:), toothTrainingLabels, ...
    'Weights',      toothWeights, ...
    'LayerSizes',   20, ...
    'Activations',   'relu', ...
    'Lambda',        1e-3, ...
    'Standardize',   false, ...

```

```
'ValidationData',    {normalisedSpectra(validationRows,:),  
toothFaultLabels(validationRows)}, ...  
'ValidationPatience', 10, ...  
'IterationLimit',    2000, ...  
'ClassNames',        [false; true]);  
  
toothCrossValPredictions = predict(toothCrossValModel,  
normalisedSpectra(testRows,:));  
  
toothTestAccuracy = mean(toothCrossValPredictions ==  
toothFaultLabels(testRows));  
toothTrainAccuracy = mean(predict(toothCrossValModel,  
normalisedSpectra(trainingRows,:)) == toothTrainingLabels);  
  
fprintf('Tooth fault test accuracy: %5.1f%% (training: %5.1f%%)\n', ...  
        100*toothTestAccuracy, 100*toothTrainAccuracy);  
Tooth fault test accuracy: 100.0% (training: 100.0%)  
  
figure('Name','Per-fault confusion matrices (raw spectra)')  
confusionchart(toothFaultLabels(testRows), toothCrossValPredictions, ...  
               'Title', 'Tooth fault')
```

9.2 SVM Code

```
rng('default') % For reproducibility
cvp = cvpartition(size(classifyTF,1), 'KFold',5); % Randomly partition the data
for training and validation
classifierTF = fitcsvm(...
    classifyTF(cvp.training(1),1:end-1), ...
    classifyTF(cvp.training(1),end), ...
    'KernelFunction','polynomial', ...
    'PolynomialOrder',2, ...
    'KernelScale','auto', ...
    'BoxConstraint',1, ...
    'Standardize',true, ...
    'ClassNames',[false; true]);
% Use the classifier on the test validation data to evaluate performance
actualValue = classifyTF{cvp.test(1),end};
predictedValue = predict(classifierTF, classifyTF(cvp.test(1),1:end-1));

% Check performance by computing and plotting the confusion matrix
confdata = confusionmat(actualValue,predictedValue);
h = heatmap(confdata, ...
    'YLabel', 'Actual gear tooth fault', ...
    'YDisplayLabels', {'False','True'}, ...
    'XLabel', 'Predicted gear tooth fault', ...
    'XDisplayLabels', {'False','True'}, ...
    'ColorbarVisible','off');
```