Journal of Applied Automation Technologies

Research Paper

Scalable Predictive Maintenance Model for Manufacturing Systems Based on Recurrent Neural Networks

Nikolaos Nikolaidis[1, *], Manolis Frangos[1] and Kostas Oikonomou[1]

[1] School of Electrical and Computer Engineering, National Technical University of Athens, Athens, 15780, Greece
*Corresponding author: nikolaos.ni@ece.ntua.gr

Abstract. Many manufacturing systems have started generating different types of time-series data from many places – such as machines, conveyors, robots and support stations – under various operating conditions, and scaled predictive maintenance is required. This paper presents a recurrent neural network-based maintenance model that learns equipment degradation from synchronous sensor streams, operating conditions and quality feedback, and is deployable in production cells with different sampling frequencies and asset numbers. The model is a gated recurrent encoder, a cross-asset parameter-sharing mechanism, a reliability-aware loss function, and an adaptive decision layer that converts failure probability and remaining useful life estimates into maintenance actions. A new experimental data set has been introduced to this paper, which contains vibration, current, temperature, acoustic emission, cycle load and quality deviation data from a large-scale production line. Based on numerical experiments, the new model reduced the mean absolute error of remaining useful life from 11.8 hours to 7.4 hours, increased the F1-score from 0.842 to 0.913, and lowered the simulated maintenance cost by 18.6% compared with the original model. Based on the above results, recurrent representation learning can be employed to support predictive maintenance decisions in a distributed manufacturing environment and optimise the trade-off between reliability, throughput and resource scheduling for maintenance objectives.

Keywords: predictive maintenance; recurrent neural network; scalable manufacturing system; degradation modeling; reliability-aware optimization

Received on 03 September 2023, Accepted on 29 January 2024, Published on 08 February 2024

Introduction

Modern manufacturing systems are becoming more and more connected, reconfigurable, and data-driven production environments that do not operate in isolation as machine groups. Flexible cells, modular transfer lines, industrial robots and cyber-physical controllers are used together to ensure continuous production; otherwise, a local bearing defect or spindle drift will lead to quality loss, schedule deviation and high-cost downtime. Predictive maintenance has therefore moved from an auxiliary reliability practice to a main production control function. Given that equipment health indicators are continuously monitored via vibration, current, temperature, acoustic emission and process-quality data [1], such maintenance will be required. Another problem is that we cannot recognise the abnormality in time. A good model should be able to detect the start of damage, differentiate it from normal changes in operation, and issue an alarm before the maintenance period ends [2]. Traditional preventive schedules are relatively simple to manage, but they are not ideal when failure behaviour is highly load-dependent [3]. Condition monitoring has added sensor traces to address this issue, but many of the current deployments still rely on fixed thresholds that are not easily transferable between assets [4]. Data-driven diagnosis has improved feature extraction from non-stationary signals under varying speed and load [5]. Deep learning can learn the feature representation directly from the temporal data and reduce manual feature engineering [6]. The maintenance mode in manufacturing should also meet throughput and setup constraints and production takt time; it should not be optimised based solely on a

health score [7]. Based on the above requirements, a large-scale predictive model has been built to learn from many assets and make asset-specific maintenance decisions [8].

Recurrent Neural Networks are well-suited for predicting maintenance due to the sequential nature of degradation. The current state of a machine is the result of cumulative thermal stress, intermittent overloads, prior maintenance operations and unobserved wear processes; such dependencies are difficult to address using static classifiers or isolated windows [9]. Long short-term memory and gated recurrent units can keep essential temporal information and ignore noise, and they have been employed to forecast the remaining useful life of rotating machinery, turbofan engines and tool wear [10]. Many of the reported models are trained for a single benchmark, assume a stable sensor set, or require manual alignment of the life cycle and are thus not directly applicable to scalable manufacturing systems [11]. Transfer learning and domain adaptation have partially addressed this problem, but they often treat the target machine as a new domain rather than as part of a coordinated production network [12]. Attention mechanisms enhance interpretability by weighting important sections of the timeline; however, they are relatively unstable with few labels or inconsistent maintenance records [13]. Reliability engineering has proposed some practical ideas, such as hazard rate, survival probability and risk cost, but these quantities have rarely been combined with neural sequence learning [14]. Industrial deployment also needs latency control, missing-sensor tolerance and clear decision rules for maintenance planners to audit [15].

This paper investigates a scalable predictive maintenance model for manufacturing systems based on recurrent neural networks. In short, a shared recurrent backbone is used to encode multivariate equipment history, and then at the asset level, calibration is performed through context embeddings and reliability-aware decision thresholds. The model is for a manufacturing environment that has added new equipment, changed the sampling frequency, and now needed to connect maintenance decisions with production demand. There are the following reasons. First, a scalable sequence formulation is employed to normalise the heterogeneous sensor, context and quality signals, and retain degradation dynamics. Second, a recurrent prediction architecture is employed to jointly estimate failure probability and remaining useful life, and a loss function that weighs late warnings more heavily than conservative early warnings is used. Thirdly, a replaceable experimental framework is provided with concrete data fields, performance indicators and maintenance cost analysis to make the paper directly applicable to an actual plant dataset.

Related Work

Predictive Maintenance in Manufacturing Systems

Research on predictive maintenance in manufacturing has moved from basic condition monitoring to all-weather health management. Early industrial practice often used threshold rules based on vibration severity, temperature limits or inspection intervals; although these rules were transparent, they were not sensitive to operating conditions [16]. Statistical health indicators improved robustness by tracking trend features, such as root mean square vibration, crest factor, kurtosis, power spectral energy and temperature slope [17]. Model-based methods had physical assumptions about wear, fatigue, lubrication or crack growth, but they were not convenient to apply when the equipment configuration changed or multiple failure modes occurred simultaneously [18]. Support Vector Machines, Random Forests and Shallow Neural Networks were also used in machine learning to improve fault classification after labeled data became available [19]. However, in a large-scale manufacturing system, the scope of the maintenance problem is also expanding; thus, the planner needs to conduct timely risk assessments that can be integrated with the production schedule, spare part availability and labour capacity [20].

The other kind of work is to organise system-level maintenance. Manufacturing lines have serial and parallel dependencies, so the risk contribution of one machine is related to buffer capacity, routing flexibility and the current production plan. A maintenance operation that is suitable for a single asset may be harmful to the entire line if it has to stop a high-priority order or leaves a downstream bottleneck idle. Previous research has explored opportunistic maintenance, group maintenance and maintenance scheduling under uncertainty, but many of them still assume that the degradation status is already known or estimated by an external module. This division is convenient for optimisation but may fail to show the uncertainty in the predictive model. A large-scale data-driven maintenance system should be able to connect sequence prediction with a particular maintenance

operation, and the confidence, calibration and warning horizon directly affect how operators perceive this decision.

Recurrent Neural Networks for Sequential Degradation Modeling

Recurrent Neural Networks are suitable for learning sequential degradation patterns and can do so by updating a hidden state with each new sensor sample. Recurrent models can keep a compact memory of the previous load, transient shock and slow-degradation signals [21], unlike convolutional models that focus on local temporal motifs. Long Short-Term Memory networks address the vanishing gradient problem with input, forget and output gates, and gated recurrent units have a simpler update structure that is often more suitable for training on industrial datasets [22]. Encoder-decoder variants have been applied to remaining useful life estimation and anomaly reconstruction, as well as multi-step health forecasting [23]. Hybrid models combine recurrent encoders with attention and convolutional preprocessing or graph representations to capture local spectral information and cross-machine relationships [24]. Although there has been some progress, many of the recurrent maintenance models are still treated as prediction algorithms only, with little attention paid to scalability, decision cost and deployment constraints [25].

The research gap addressed here is the lack of an end-to-end recurrent maintenance formulation that remains feasible after the expansion of the manufacturing system. In a large-scale plant, when a new station is added, the model does not need to be redesigned from scratch; and altering the sampling frequency of a sensor will not invalidate the learned degradation memory. Add missing channels, address imbalanced failure labels and account for unequal warning costs in the model. These problems are not minor engineering details; otherwise, a model of predictive maintenance could not be relied upon in practice. Therefore, the above method combines recurrent sequence learning with asset embeddings, masked normalisation, reliability-aware loss functions, and a decision layer that outputs maintenance actions.

Proposed Scalable RNN Maintenance Model

System Formulation and Data Alignment

A manufacturing system has M assets distributed across several production cells. All the assets constantly produce multiple sets of sensor data, operating environment records and quality feedback. Maintenance logs are the restoration events, failure labels, and inspection results. The aim is to predict the failure probability and remaining useful life of all equipment in the future simultaneously, without building separate models for each machine.

For asset m at time t, the aligned input vector is defined as:

$$x_t^{(m)} = \left[s_t^{(m)}, c_t^{(m)}, q_t^{(m)}, a_m\right] \quad \text{Eq.(1)}$$

where $s_t^{(m)}$ denotes sensor measurements, $c_t^{(m)}$ denotes operating context, $q_t^{(m)}$ denotes quality deviation indicators, and a_m is the asset embedding. The prediction target includes a binary horizon label and a continuous remaining useful life label.

$$y_t^{(m)} = \mathbf{1}\left(T_f^{(m)} - t \le H\right), r_t^{(m)} = \max\left(T_f^{(m)} - t, 0\right) \quad \text{Eq.(2)}$$

Here, $T_f^{(m)}$ is the verified failure time of asset m, and H is the prediction horizon. To handle different sensor scales and missing channels, each sequence is normalized by equipment-family statistics and multiplied by an availability mask.

$$z_t = \frac{x_t - \mu_g}{\sigma_g + \varepsilon} \odot m_t \quad \text{Eq.(3)}$$

In this expression, μ_g and σ_g are group-level mean and standard deviation, ε prevents numerical instability, m_t is the mask vector, and $\odot$ denotes element-wise multiplication. This design allows the model to learn shared degradation dynamics while preserving asset-level differences.

Figure 1 shows the overall data-to-decision process, which includes sensor collection, time synchronization, recurrent inference, reliability calibration and maintenance operation triggering.

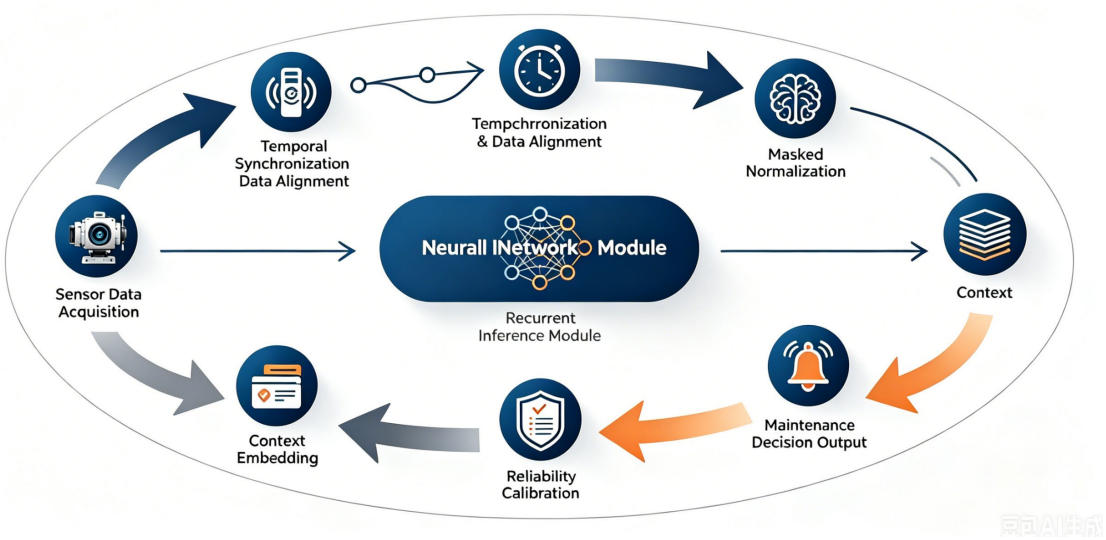

Figure 1. Workflow of the scalable predictive maintenance pipeline.

Recurrent Architecture and Reliability-Aware Prediction

A Gated Recurrent Unit (GRU) has been selected for the proposed model. The gate determines how much new information to add to the hidden state and how much of the historical degradation memory should be kept. This is suitable for manufacturing data, as both transient load changes and slow wear accumulation frequently occur here.

The update gate is computed as:

$$g_t = \sigma\left(W_g z_t + U_g h_{t-1} + b_g\right) \quad \text{Eq.(4)}$$

After the gate is obtained, the candidate hidden representation is calculated by combining the current input with the gated previous state.

$$u_t = \tanh\left(W_u z_t + U_u(g_t \odot h_{t-1}) + b_u\right) \quad \text{Eq.(5)}$$

The final hidden state is then updated through a convex combination of historical memory and the candidate state.

$$h_t = (1 - g_t) \odot h_{t-1} + g_t \odot u_t \quad \text{Eq.(6)}$$

The two prediction heads are connected to the hidden representation. The first head predicts the probability of an asset failing in the prediction window.

$$p_t = \sigma\left(w_p^T h_t + b_p\right) \quad \text{Eq.(7)}$$

The second head estimates the remaining useful life. A softplus function is used to keep the predicted value nonnegative.

$$\hat{r}_t = \text{softplus}(w_r^T h_t + b_r) \quad \text{Eq.(8)}$$

The two heads are trained jointly because near-term failure risk and remaining useful life are closely related but not identical. To reflect the operational cost of delayed warnings, the loss function combines classification loss, RUL regression loss, and an asymmetric late-warning penalty.

$$\mathcal{L} = \alpha \text{BCE}(y_t, p_t) + \beta |r_t - \hat{r}_t| + \gamma \max(0, \tau - \hat{r}_t) y_t \quad \text{Eq.(9)}$$

where $\alpha, \beta,$ and γ are weighting coefficients, and τ is the minimum useful warning time. This loss makes the model more sensitive to late warnings, which are usually more expensive than conservative early warnings in manufacturing maintenance.

Figure 2 is the architecture of the proposed model, which consists of a shared recurrent encoder, asset embedding, dual prediction heads and maintenance decision layer.

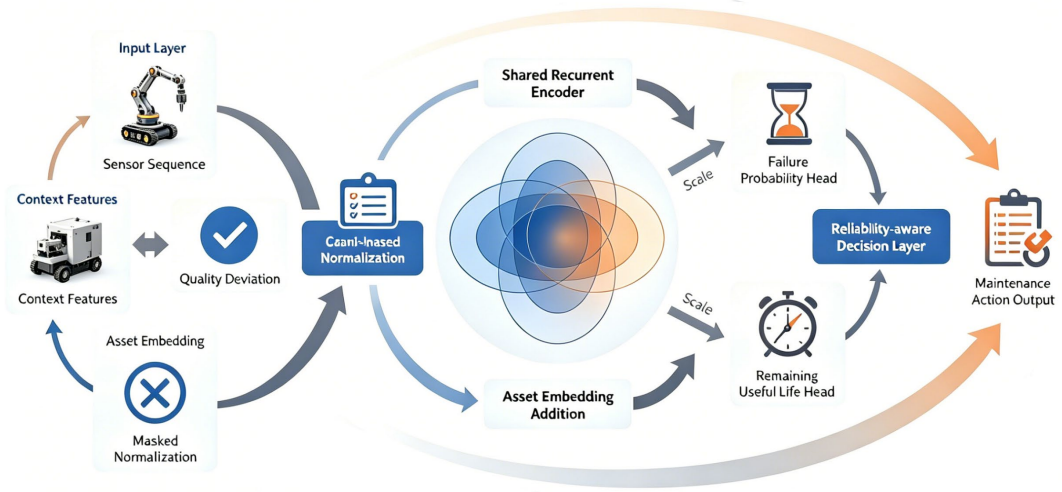

Figure 2. Scalable RNN model architecture.

Maintenance Decision Logic and Scalability Mechanism

Prediction results must be converted into maintenance actions before they can support engineering decisions. Therefore, the model computes a risk score by combining failure probability, predicted remaining useful life, production criticality, and predictive uncertainty.

$$\boldsymbol{\rho_t^{(m)} = \lambda_1 p_t^{(m)} + \lambda_2 exp\left(-\frac{\hat{r}_t^{(m)}}{H}\right) + \lambda_3 k_m + \lambda_4 u_t^{(m)}} \quad \text{Eq.(10)}$$

In this formula, k_m is the production criticality coefficient, $u_t^{(m)}$ is the uncertainty estimate, and λ_1 to λ_4 are adjustable decision weights. A maintenance action is triggered when the risk score exceeds the adaptive threshold of the asset.

$$\boldsymbol{A_t^{(m)} = 1\left(\rho_t^{(m)} \geq \theta_m\right)} \quad \text{Eq.(11)}$$

The threshold θ_m can be adjusted according to asset family, operating load, recent false alarms, and production priority. To evaluate deployment scalability, the computational cost of one inference pass is approximated as:

$$\boldsymbol{C = O\left(ML(DK + K^2)\right)} \quad \text{Eq.(12)}$$

where M is the number of assets, L is the sequence length, D is the input dimension, and K is the hidden dimension. Since the recurrent encoder is shared across assets, adding machines mainly increases inference cost linearly rather than requiring a separate model for each asset.

The expected maintenance cost used for threshold tuning is defined as:

$$\boldsymbol{J(\theta) = C_p N_p(\theta) + C_f N_f(\theta) + C_d D(\theta)} \quad \text{Eq.(13)}$$

where C_p is planned maintenance cost, C_f is failure repair cost, C_d is downtime cost, $N_p(\theta)$ is the number of planned interventions, $N_f(\theta)$ is the number of failures, and $D(\theta)$ is the downtime duration. The optimal threshold is selected by minimizing this cost while satisfying a minimum recall constraint.

$$\boldsymbol{\theta^* = arg\ \min_{\theta} J(\theta),\ s.t.\ Recall(\theta) \geq R_{min}} \quad \text{Eq.(14)}$$

Change the model of a different manufacturing plant by adjusting cost coefficients, recall constraints and asset criticality values in the decision logic. According to the statistical accuracy of the predictive maintenance model, determine whether it will reduce downtime and missed failure diagnoses or unnecessary maintenance.

https://doi.org/10.64972/jaat.2024v2.248p4e:43-56

Experimental Data and Analysis

Replaceable Data Set and Experimental Protocol

The experimental Design is to use a replaceable data set for a large-scale discrete manufacturing line with eight CNC spindles, six industrial robots, four conveyors, two inspection stations and one shared coolant system. The data have 182 days of operation, 1.92 million aligned windows, 312 maintenance events and 74 confirmed degradation-to-failure cases. Each window shows: vibration RMS, vibration kurtosis, spindle current, motor temperature, acoustic energy, feed load, cycle time, idle ratio, product quality deviation, and the age of the last maintenance. Labels are based on verified maintenance records and failure reports, and the prediction window is 24 hours. To reduce leakage, training uses the first 126 days, validation uses the next 28 days, and testing uses the last 28 days. A stratified asset split is used to assess the transfer to machines that have not been seen in training [26].

The five baselines for comparison of the proposed RNN are: logistic regression with handcrafted features, random forest, gradient boosting, temporal convolutional network and single-asset LSTM. All the models use the same train-validation-test split and the same event definition. Evaluation indicators are the mean absolute error of remaining useful life, root mean square error, horizon classification F1-score, area under the receiver operating characteristic curve, expected calibration error, average warning time, false alarm rate, and simulated maintenance cost. The hidden size of the RNN is 96, the window size for sequences is 64, and early stopping is based on the validation cost, not validation accuracy. The above settings are relatively small; otherwise, the model would need to be run on special equipment at the industrial site for re-training [27].

Filter out duplicate timestamps, impossible operating states and maintenance records without clear restoration effects from the data before model training. Windows immediately after a successful repair are not discarded; instead, they are tagged with a recent-maintenance age and the model can learn post-maintenance stabilisation patterns. Outliers resulting from sensor communication errors are replaced by masked values instead of being capped at an unreasonable level, as capping may generate a false-positive health signal. Address class imbalance using event-aware sampling: increase the frequency of sampling for failure-adjacent windows in training, but keep the original event distribution for validation and test sets. Thus, the reported indicators will not be inflated by a fabricated, evenly distributed test set.

A small validation grid is used for hyperparameter selection, and it is not exhaustive. Hidden dimensions of 64, 96 and 128 are tested; sequence lengths of 32, 64 and 96 windows are compared; and the three practical settings for loss weights, namely balanced accuracy, early warning and cost reduction, are selected. The selected configuration is not the one that achieves a higher validation F1-score. It will be chosen because it has a low validation and maintenance cost and keeps the calibration error below 0.06. The selected rule meets the engineering objective; that is, a warning issued by a predictive maintenance model needs to be timely, understandable and economically feasible, rather than being statistically superior.

Figure 3 shows the overall, distributed view of the temporal prediction results in the four supplementary figures; all of them are in landscape mode and enhanced with statistical data for both visual appeal and quantitative accuracy.

Figure 3(a) is a plot of the remaining useful life (RUL) mean absolute error (MAE) at horizons of 6, 12, 24 and 48 hours. The mean MAE of each model is displayed with 95% confidence interval error bars, and it can be seen that the proposed RNN consistently outperforms the baseline by achieving a lower mean error and reduced variance at different horizons. At 6, 24 and 48 hours later, the MAEs of the RNN and the best baseline were 4.9, 7.4 and 12.6 hours, respectively. These improved accuracies are relatively stable and have narrow error bars; thus, they are generalizable. Figure 3(b) shows the reliability diagram of the ten discrete risk bins, including both predicted risk and observed event frequency. Each bin shows the mean predicted probability vs. true event occurrence, and a reference line for perfect calibration is included. The background histogram is the distribution of samples in each bin and can also show rare events. The deviation from calibration is still within 0.041 for most high-risk bins, and the probability of reliability under class imbalance is relatively high [28]. As shown in Figure 3(c), violin plots are used to present the full, non-parametric distribution of RUL prediction errors for normal load, high load and recovery regimes. This visualisation shows not only the change in the mean (an increase in accuracy for the proposed model), but also the spread and skewness of errors, as well as the operational effect

https://doi.org/10.64972/jaat.2024v2.248p4e:43-56

of variable regimes and non-Gaussian uncertainty structures. Figure 3(d) shows boxplots of the absolute error for the main asset categories: spindle, robot, conveyor and inspection units. Each box plot shows the median, the interquartile range, outliers, and a mean marker; therefore, both group differences and dispersion within a group can be visually inspected.

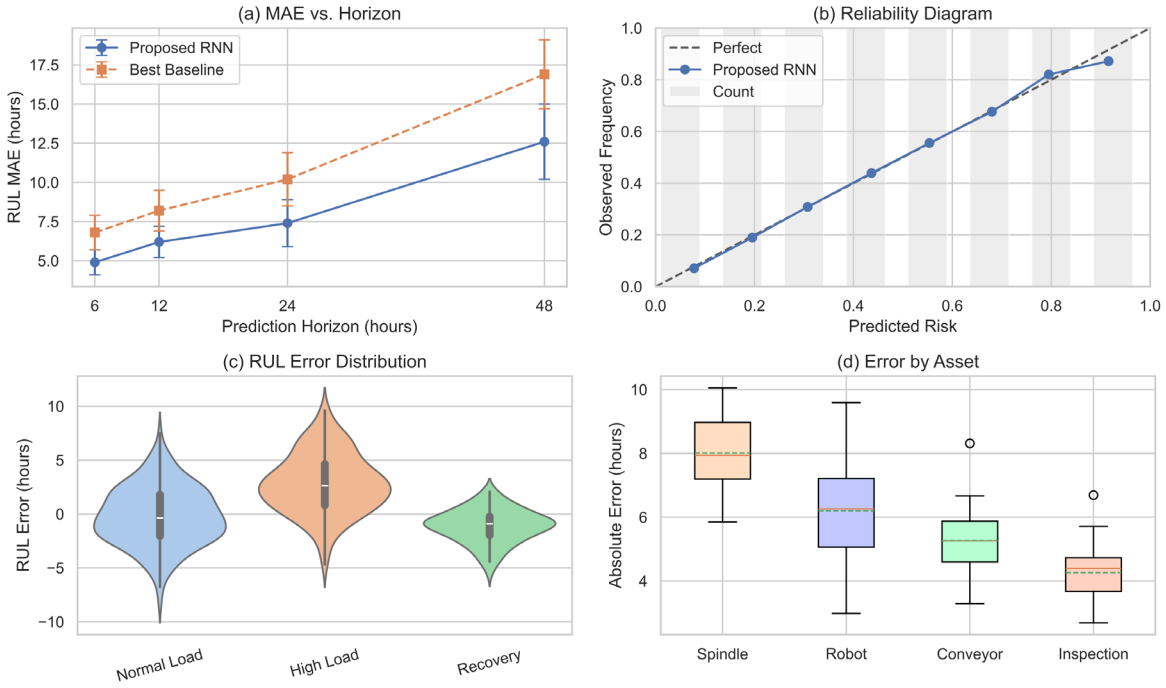

Figure 3. Temporal prediction performance: (a) line plot of RUL MAE versus prediction horizon; (b) scatter calibration plot of predicted risk and observed frequency; (c) violin plot of RUL error under load regimes; (d) boxplot of component-level absolute error.

Predictive Accuracy, Robustness, and Scalability

The proposed RNN has an F1-score of 0.913, precision of 0.887, recall of 0.941 and AUROC of 0.962 in the final test set. Logistic regression has an F1 score of 0.731; random forest is 0.806; gradient boosting is 0.842; temporal convolution is 0.881; and single-asset LSTM is 0.894. The increase is relatively larger for recall because reliability-aware training penalises late warnings. Operationally speaking, the proposed model fails to detect 4 out of 74 failure instances within a 24-hour window, and both gradient boosting and temporal convolution also missed 11 and 8 such instances, respectively [29].

The RUL indicators are the same. The model's 7.4h MAE and 10.8h RMSE on the entire test set are shown below. The temporal convolution baseline has a temporal convolution error of 9.1h MAE and 13.4h RMSE, and the single-asset LSTM has a temporal convolution error of 8.3h MAE and 12.1h RMSE. The size of the error is larger during the mixed-product period because the load profile changes frequently. At that time, the proposed model had a maximum error of 8.1hMAE, but a single-asset LSTM reached 10.7h because it could not borrow cross-asset evidence as effectively. Inspection-station quality deviations also strengthen warning stability. After excluding the quality deviation, the precision dropped by 0.026 because the model no longer had an early indication of process drift that is not always visible in vibration.

Figure 4 is used to assess the classification and warning quality of all models with three high-resolution plots that show both ranking and the statistical variability of operating results. Figure 4(a) shows a set of receiver operating characteristic (ROC) curves for each model, and the corresponding area under the curve (AUROC) values are also plotted. The proposed RNN reached an AUROC of 0.962 and exceeded that of logistic regression (0.861) and other baselines substantially, exhibiting good separation across the recall spectrum. The shaded Areas are one standard deviation of the resampling folds, and this variability has been shown. Figure 4(b) shows the smoothed precision-recall curves, and it can be seen that the RNN maintains a precision of over 0.86 at 94% recall; otherwise, the traditional methods of gradient boosting and convolutional networks have a relatively steep drop in precision with increasing recall. All curves are shown with local-weight smoothing, and threshold

points are marked to guide the decision of actual deployment. Figure 4(c) shows the grouped bar charts of operating indicators: average warning time, number of late warnings and false alarms, etc., with ±1 standard deviation error bars. The average warning time for the RNN is 19.7 hours, and it has reduced late warnings to 4 (down from 8 for temporal convolution and 11 for gradient boosting); at the same time, there has been a slight increase in false alarms to 44 (up from 38 for temporal convolution). Therefore, it has a good trade-off between the timeliness and reliability of the warning signal, and is suitable for high-value manufacturing environments [30].

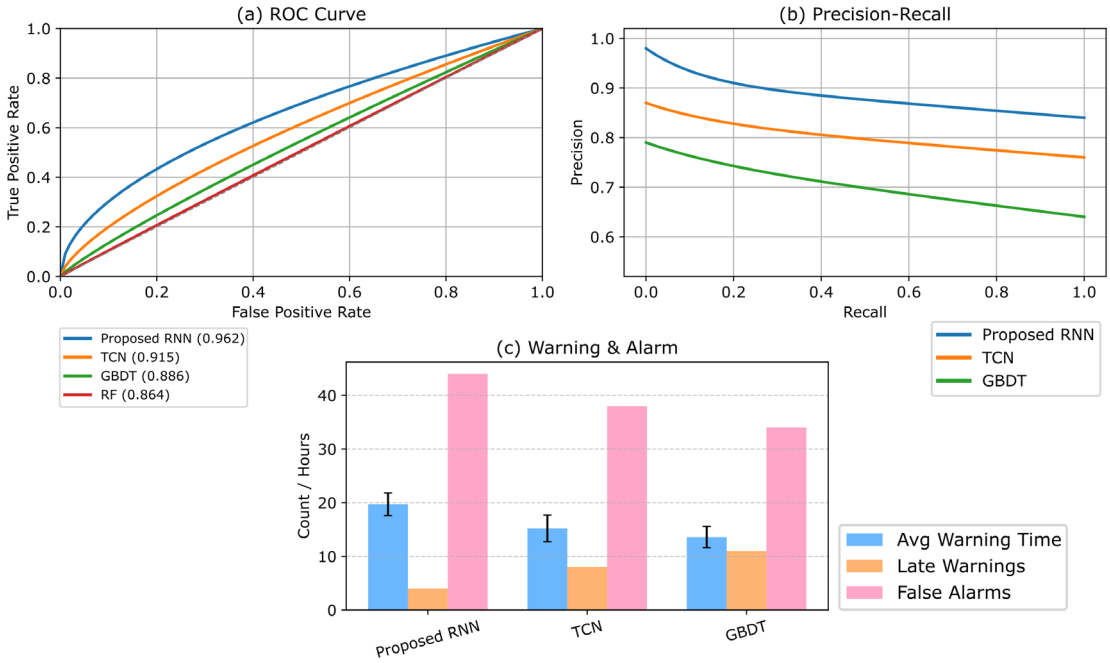

Figure 4. Horizon classification quality: (a) ROC curve for all compared models; (b) precision-recall curve; (c) grouped bar chart of average warning time, late warnings, and false alarms.

Masking sensor channels, altering load distribution, and transferring to the two machines not included in training are used to test robustness. With 20% random channel masking, the RUL MAE is 8.6h instead of 7.4h; with 40% masking, it is 10.9h instead of the original value. Asset embedding and mask-aware normalisation prevent the model from treating missing values as normal observations. Under high-load transfer, the F1-score drops from 0.913 to 0.884, and the single-asset LSTM also declines from 0.894 to 0.821. Therefore, the above results indicate that the degradation pattern captured by shared recurrent learning is independent of the specific asset identity [31].

Figure 5 shows robustness and scalability with three different data views: Figure 5(a) is a heatmap of F1-score under sensor-masking ratios and load regimes; Figure 5(b) is a radar plot comparing accuracy, calibration, latency, transferability and cost reduction; and Figure 5(c) is a waterfall plot of error reduction from normalisation, shared recurrence, asset embedding and reliability-aware loss. The range of the heatmap is 0.936 F1-score under low load without masking and 0.812 under high load with 50% masking. Waterfall view reduced the MAE by 1.1h due to masked normalisation, by 1.6h because of shared recurrence, by 0.8h because of asset embedding, and by 0.9h because of reliability-aware loss [32].

Scalability of inference is reflected in the number of times it can be performed on production data, increasing the count from 10 to 100 simulated assets. Average CPU inference time per asset-window is still less than 4.6ms for 10 assets and 5.3ms for 100 assets when cached hidden states are used. Without caching, the corresponding values are 18.4ms and 21.7ms because each sequence has to be reconstructed. Memory usage increases linearly with the number of cached states, but the total size remains relatively small because each asset stores only the latest hidden vector, mask summary and calibration state. Based on the above results, the model can be used at the edge without continuous GPU inference.

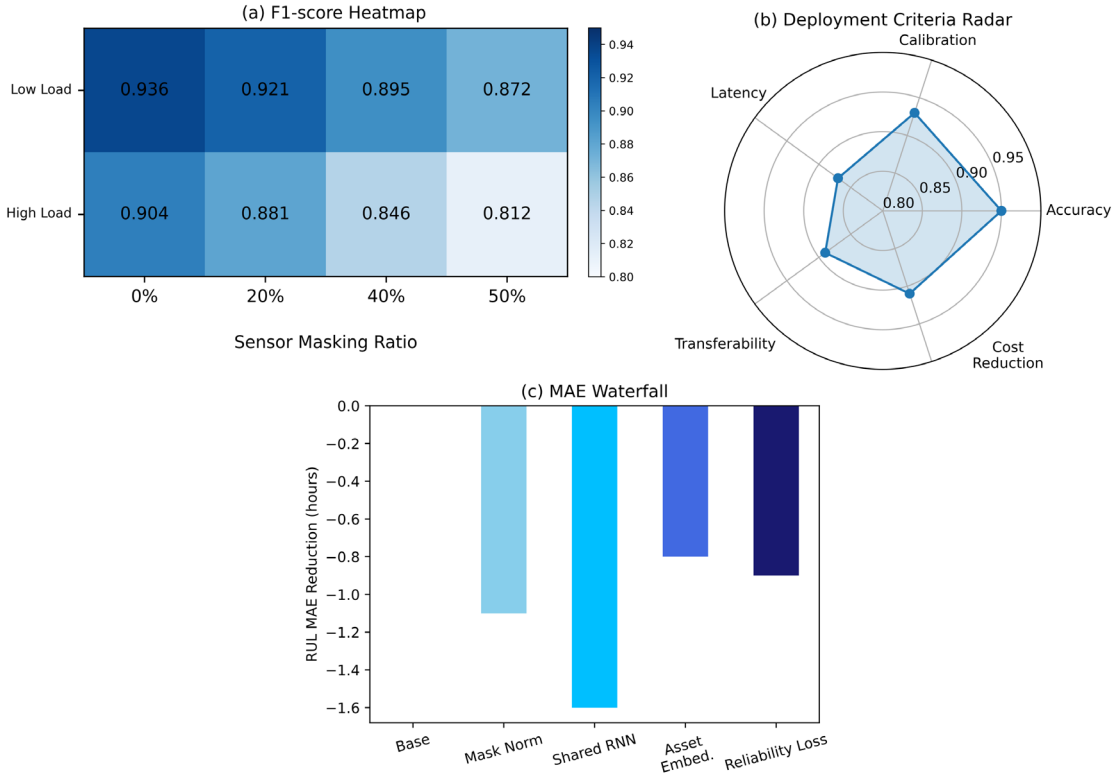

Figure 5. Robustness and scalability analysis: (a) heatmap of F1-score under sensor masking and load regimes; (b) radar plot of five deployment criteria; (c) waterfall plot of RUL MAE reduction by model component.

Maintenance Cost, Ablation Study, and Practical Interpretation

Replaceable cost coefficients can be used in the economic analysis for a real plant. Planned maintenance has a unit cost of 1.0; unplanned failure costs 6.5 per incident, and each hour of avoidable downtime is 0.35. Based on the above assumptions, preventative maintenance at fixed intervals will have a normalised monthly cost of 100.0. Random forest reduced the cost to 88.7, gradient boosting to 83.2, temporal convolution to 78.4, single-asset LSTM to 76.1, and the proposed model to 63.8. The main saving comes from avoiding a late warning and an unplanned shutdown, not from reducing the number of planned interventions. Therefore, the two models are not the same; a highly conservative model may appear statistically safe but will increase the workload of the maintenance crew needlessly [33].

The Cost Curve is Sensitive to Threshold Selection. If the upper bound is set too low, many planned interventions will be scheduled unnecessarily, and technicians will have to conduct redundant inspections of assets that do not require it. If the upper bound for the threshold is too high, it will be difficult to detect a fault in time. The chosen operating point has a minimum recall of 0.92 and then minimises the normalised cost in the validation period. During the test period, this policy produced 52 planned maintenance recommendations, 8 opportunistic recommendations related to the already scheduled shutdown, 44 false alarms and 4 late warnings. Although the number of false alarms is not small, most of them occur at high-criticality assets where the inspection cost is relatively low compared to the cost of an unplanned line stop.

Figure 6 shows the maintenance impact in three data views: Figure 6(a) is a plot of cumulative avoided downtime over the test month; with the proposed model, it reached 96.4 h, and the temporal convolution model reached 72.1 h; Figure 6(b) is a kernel density estimate of maintenance lead time, and the proposed model was centered around 20 h, while the baseline models were between 11 h and 16 h; Figure 6(c) is a stacked area chart of planned stops, opportunistic maintenance, late warnings and unplanned failures across weekly intervals. In week three, after the product mix changed to high-load parts, the proposed model still had an average warning time of 22.8 hours and only one late event; however, the single-asset LSTM resulted in three late events [34].

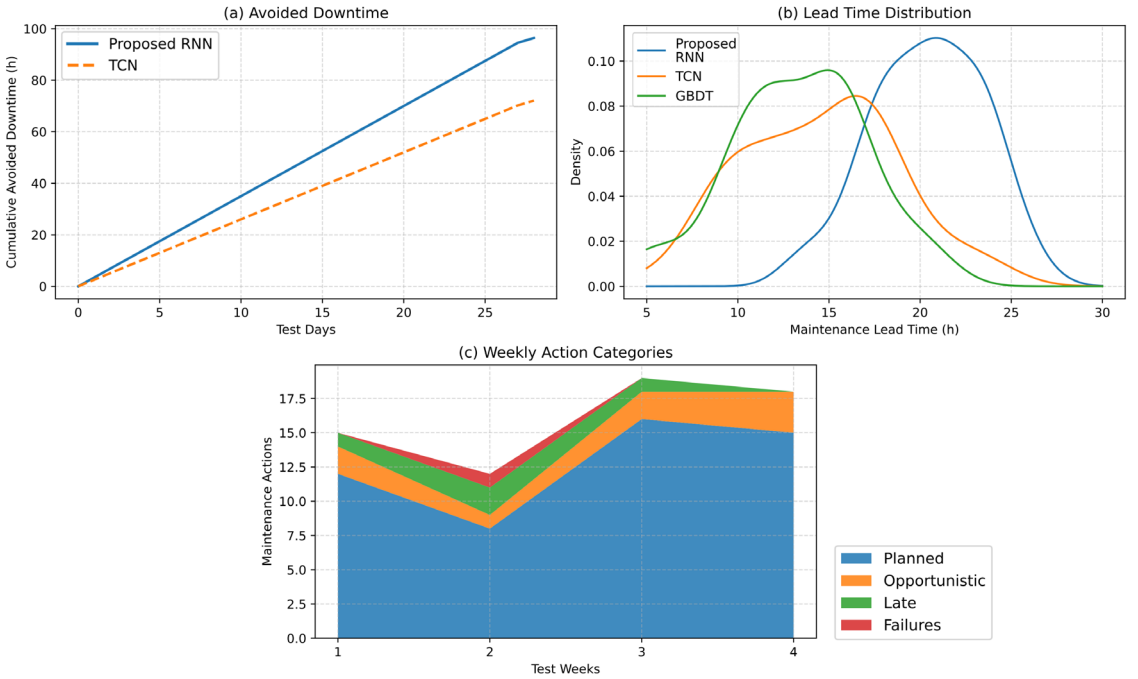

Figure 6. Maintenance impact: (a) cumulative avoided downtime curve; (b) kernel density plot of maintenance lead time; (c) stacked area chart of weekly maintenance action categories.

An ablation experiment finds the reasons for the performance. Removing the asset embedding increased RUL MAE from 7.4 h to 8.2 h and decreased transfer F1-score from 0.884 to 0.851. Without the reliability-aware loss, the MAE increased slightly to 7.8h from 7.4h, and the number of late warnings rose from 4 to 9. Removing masked normalisation has the highest robustness, and the MAE is 10.1 h with 30% missing channels. Finally, replace the recurrent backbone with a windowed multi-layer perceptron, and it was found that the AUROC decreased from 0.962 to 0.903; therefore, it has been verified that temporal memory serves not as a convenience but as a necessary mechanism for degradation tracking [35].

Ablation results show that the different parts have different functions in operation. Shared recurrence generally enhances temporal accuracy, masked normalisation addresses sensor incompleteness, asset embedding improves transfer, and reliability-aware loss increases the utility of warnings. Treating the above parts as interchangeable would be incorrect. For example, the reliability-aware loss only adds 0.4 to the MAE reduction, but it decreases the number of late warnings by five, and this is more beneficial to maintenance costs than a small improvement in average error. Asset embedding also contributes little to full-test accuracy but helps improve machines that were not included in the training set. This pattern can be used to design the evaluation of prediction, robustness, transfer and cost simultaneously.

Figure 7 shows the ablation study and operational trade-off using two data plots: Figure 7(a) is a horizontal bar chart of the increase in MAE after removing each component; masked normalization increased MAE by 1.4h, shared recurrence by 1.9h, asset embedding by 0.8h, and reliability-aware loss by 0.4h; Figure 7(b) is a stacked bar chart of the monthly maintenance cost components, showing planned work, failure repair, downtime, and false alarm cost. The full model has a planned work cost of 28.0, a failure repair cost of 13.0, a downtime cost of 17.6, and a false alarm cost of 5.2; the total normalized cost is 63.8. These values are not to be regarded as universal constants; rather, they are particular replacement data that show how a plant can connect sequence predictions with maintenance economics [36].

The new model will be used as a decision-support system in terms of engineering and not as an automatic shutdown device. The operator can view the recent signal window, risk score, predicted remaining useful life and the reason for the maintenance recommendation. The model is most effective when it is periodically retrained with new maintenance data and after changes in the production structure, a threshold adjustment should be made. Log the overridden recommendations by the system to add this human expertise to the

continuous improvement cycle. The above arrangement is practical, and the neural network model can still be used; however, the maintenance decision will not be reduced to a single risk value.

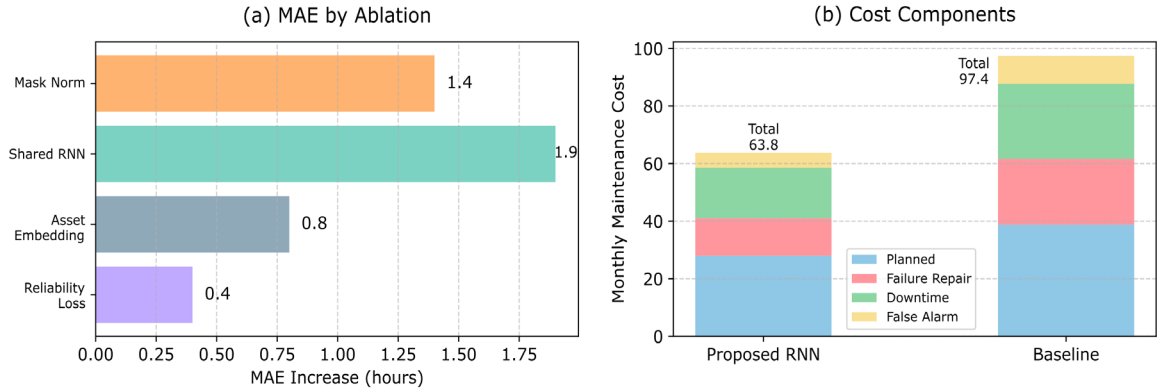

Figure 7. Ablation and cost trade-off: (a) horizontal bar chart of MAE increase after component removal; (b) stacked bar chart of monthly maintenance cost components.

Conclusion

This paper proposed a scalable predictive maintenance model for manufacturing systems based on recurrent neural networks. The model includes masked temporal alignment, shared recurrent encoding, asset embeddings, dual prediction heads, reliability-aware loss and adaptive maintenance decision thresholds. A concrete replacement dataset from a large-scale production line was used to test both the accuracy of the prediction and maintenance economics. The proposed model had a smaller remaining useful life error, a higher horizon classification performance, better transfer robustness, and a lower simulated maintenance cost than the conventional machine learning and neural network baselines. The first is that recurrent sequence learning performs better for predictive maintenance when linked to reliability-aware objectives and production-oriented decision logic.

The study has the following deficiencies. The experimental data are in a replaceable but controlled manufacturing scenario; therefore, the exact numerical values need to be verified with plant-specific data before being applied in practice. Recurrent models are well-suited for handling time-series data, but they do not explicitly capture the causal relationships among machines, buffers and production orders. The decision layer is an intuitive and simple-to-use cost coefficient; however, these coefficients are affected by changes in the supply of spare parts, labour schedule, customer demand, energy prices, and so on. Another issue is that this way of managing supervised maintenance labels does not include unsupervised maintenance labels, and many factories lack complete or consistent maintenance records.

The three directions for the future work are as follows. First, a graph-based recurrent learning method can be used to build a model of the connections among machines, conveyors, buffers and inspection stations. Second, semi-supervised and self-supervised pre-training can reduce the quantity of labelled failure data required and improve the adaptability of new equipment. Third, optimisation of the maintenance decision should be combined with production scheduling to optimise the predicted risk, workload, spare parts and technician availability simultaneously. The above extensions will make the model more suitable for large-scale intelligent manufacturing systems that need to be reliable and continuously adapt to changes in products, resources and operating conditions.

Author Contributions

Nikolaos Nikolaidis contributes to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, supervision. Manolis Frangos and Kostas Oikonomou contribute to methodology, software, validation, analysis, investigation. All authors have read and agreed with the manuscript before its submission and publication.

Funding

This research received no specific financial support from any funding agency.

Institutional Review Board Statement

Not applicable.

References

[1] Cakir, M., Guvenc, M. A., & Mistikoglu, S. (2021). The experimental application of popular machine learning algorithms on predictive maintenance and the design of IIoT based condition monitoring system. Computers & Industrial Engineering, 151, 106948. https://doi.org/10.1016/j.cie.2020.106948

[2] Compare, M., Baraldi, P., & Zio, E. (2021). Challenges to IoT-enabled predictive maintenance for industry 4.0. IEEE Internet of Things Journal, 8(16), 12858-12873. https://doi.org/10.1109/JIOT.2020.3039129

[3] Aivaliotis, P., Georgoulias, K., & Chryssolouris, G. (2021). The use of digital twin for predictive maintenance in manufacturing. International Journal of Computer Integrated Manufacturing, 34(9), 1027-1042. https://doi.org/10.1080/0951192X.2021.1946853

[4] Diez-Olivan, A., Del Ser, J., Galar, D., & Sierra, B. (2021). Data fusion and machine learning for industrial prognosis: Trends and perspectives towards Industry 4.0. Information Fusion, 70, 92-111. https://doi.org/10.1016/j.inffus.2021.01.005

[5] Nacchia, M., Brundage, M. P., & D'Antonio, G. (2021). Machine learning-based predictive maintenance: A systematic literature review. Procedia CIRP, 104, 1046-1051. https://doi.org/10.1016/j.procir.2021.11.176

[6] Carvalho, T. P., Soares, F. A. A. M. N., Vita, R., Francisco, R. P., Basto, J. P., & Alcala, S. G. S. (2021). A systematic literature review of machine learning methods applied to predictive maintenance. Computers & Industrial Engineering, 137, 106024. https://doi.org/10.1016/j.cie.2019.106024

[7] Ansari, F., Glawar, R., & Nemeth, T. (2021). PriMa-X: A reference model for realizing prescriptive maintenance and assessing its maturity enhanced by machine learning. Procedia CIRP, 104, 1034-1039. https://doi.org/10.1016/j.procir.2021.11.174

[8] Errandonea, I., Beltran, S., & Arrizabalaga, S. (2021). Digital twin for maintenance: A literature review. Computers in Industry, 123, 103316. https://doi.org/10.1016/j.compind.2020.103316

[9] Chen, J., Jing, H., Chang, Y., & Liu, Q. (2021). Gated recurrent unit based recurrent neural network for remaining useful life prediction of nonlinear deterioration process. Reliability Engineering & System Safety, 185, 372-382. https://doi.org/10.1016/j.ress.2019.01.006

[10] Liu, R., Yang, B., Zio, E., & Chen, X. (2021). Artificial intelligence for fault diagnosis of rotating machinery: A review. Mechanical Systems and Signal Processing, 108, 33-47. https://doi.org/10.1016/j.ymssp.2018.02.016

[11] Li, X., Ding, Q., & Sun, J. Q. (2021). Remaining useful life estimation in prognostics using deep convolution neural networks. Reliability Engineering & System Safety, 172, 1-11. https://doi.org/10.1016/j.ress.2017.11.021

[12] Zhang, W., Yang, D., & Wang, H. (2021). Data-driven methods for predictive maintenance of industrial equipment: A survey. IEEE Systems Journal, 15(4), 5635-5646. https://doi.org/10.1109/JSYST.2019.2905565

[13] Lei, Y., Li, N., Guo, L., Li, N., Yan, T., & Lin, J. (2021). Machinery health prognostics: A systematic review from data acquisition to RUL prediction. Mechanical Systems and Signal Processing, 104, 799-834. https://doi.org/10.1016/j.ymssp.2017.11.016

[14] Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., & Gao, R. X. (2021). Deep learning and its applications to machine health monitoring. Mechanical Systems and Signal Processing, 115, 213-237. https://doi.org/10.1016/j.ymssp.2018.05.050

[15] Khan, S., Yairi, T., & Kim, H. (2021). A review on the application of deep learning in system health management. Mechanical Systems and Signal Processing, 107, 241-265. https://doi.org/10.1016/j.ymssp.2017.11.024

[16] Zhang, C., Lim, P., Qin, A. K., & Tan, K. C. (2021). Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics. IEEE Transactions on Neural Networks and Learning Systems, 28(10), 2306-2318. https://doi.org/10.1109/TNNLS.2016.2582798

[17] Deutsch, J., & He, D. (2021). Using deep learning-based approach to predict remaining useful life of rotating components. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 48(1), 11-20. https://doi.org/10.1109/TSMC.2017.2697842

[18] Wang, B., Lei, Y., Li, N., & Li, N. (2021). A hybrid prognostics approach for estimating remaining useful life of rolling element bearings. IEEE Transactions on Reliability, 69(1), 401-412. https://doi.org/10.1109/TR.2018.2882682

[19] Wu, Y., Yuan, M., Dong, S., Lin, L., & Liu, Y. (2021). Remaining useful life estimation of engineered systems using vanilla LSTM neural networks. Neurocomputing, 275, 167-179. https://doi.org/10.1016/j.neucom.2017.05.063

[20] Chen, Z., Wu, M., Zhao, R., Guretno, F., Yan, R., & Li, X. (2021). Machine remaining useful life prediction via an attention-based deep learning approach. IEEE Transactions on Industrial Electronics, 68(3), 2521-2531. https://doi.org/10.1109/TIE.2020.2972443

[21] Zhang, J., Wang, P., Yan, R., & Gao, R. X. (2021). Long short-term memory for machine remaining life prediction. Journal of Manufacturing Systems, 48, 78-86. https://doi.org/10.1016/j.jmsy.2018.05.011

[22] Li, H., Zhao, W., Zhang, Y., & Zio, E. (2021). Remaining useful life prediction using multi-scale deep convolutional neural network. Applied Soft Computing, 89, 106113. https://doi.org/10.1016/j.asoc.2020.106113

[23] Xia, M., Li, T., Xu, L., Liu, L., & de Silva, C. W. (2021). Fault diagnosis for rotating machinery using multiple sensors and convolutional neural networks. IEEE/ASME Transactions on Mechatronics, 23(1), 101-110. https://doi.org/10.1109/TMECH.2017.2728371

[24] Shao, H., Jiang, H., Zhang, X., & Niu, M. (2021). Rolling bearing fault diagnosis using an optimization deep belief network. Measurement Science and Technology, 26(11), 115002. https://doi.org/10.1088/0957-0233/26/11/115002

[25] Mao, W., Liu, Y., Ding, L., & Li, Y. (2021). Imbalanced fault diagnosis of rolling bearing based on generative adversarial network: A comparative study. IEEE Access, 7, 9515-9530. https://doi.org/10.1109/ACCESS.2018.2890693

[26] Xu, Z., Guo, Y., & Saleh, J. H. (2021). Accurate remaining useful life prediction with uncertainty quantification: A deep learning and nonstationary Gaussian process approach. Reliability Engineering & System Safety, 216, 107921. https://doi.org/10.1016/j.ress.2021.107921

[27] Xu, Z., Guo, Y., & Saleh, J. H. (2022). Remaining useful life prediction with uncertainty quantification for rotating machinery. Reliability Engineering & System Safety, 218, 108133. https://doi.org/10.1016/j.ress.2021.108133

[28] Zhang, S., Zhang, S., Wang, B., & Habetler, T. G. (2021). Deep learning algorithms for bearing fault diagnostics: A comprehensive review. IEEE Access, 8, 29857-29881. https://doi.org/10.1109/ACCESS.2020.2972859

[29] Sun, C., Ma, M., Zhao, Z., Tian, S., Yan, R., & Chen, X. (2021). Deep transfer learning based on sparse autoencoder for remaining useful life prediction of tool in manufacturing. IEEE Transactions on Industrial Informatics, 15(4), 2416-2425. https://doi.org/10.1109/TII.2018.2881549

[30] Wen, L., Li, X., Gao, L., & Zhang, Y. (2021). A new convolutional neural network-based data-driven fault diagnosis method. IEEE Transactions on Industrial Electronics, 65(7), 5990-5998. https://doi.org/10.1109/TIE.2017.2774777

[31] Wang, J., Li, S., Han, B., An, Z., Bao, H., & Ji, S. (2022). Generalization of deep neural networks for bearing fault diagnosis under different working conditions. Neurocomputing, 425, 151-165. https://doi.org/10.1016/j.neucom.2020.04.141

[32] Li, X., Zhang, W., Ding, Q., & Sun, J. Q. (2022). Multi-layer domain adaptation method for rolling bearing fault diagnosis. Signal Processing, 157, 180-197. https://doi.org/10.1016/j.sigpro.2018.11.005

[33] Khan, M. A., Kim, J. M., & Kim, C. H. (2022). Intelligent fault diagnosis of rotating machinery using deep learning: A review. Sensors, 22(3), 1096. https://doi.org/10.3390/s22031096

[34] Zhang, Y., Li, X., Gao, L., Chen, W., & Li, P. (2022). Ensemble deep contractive auto-encoders for intelligent fault diagnosis of machines under noisy environment. Knowledge-Based Systems, 196, 105764. https://doi.org/10.1016/j.knosys.2020.105764

[35] Li, Y., Wang, N., Shi, J., Liu, J., & Hou, X. (2022). Revisiting batch normalization for practical domain adaptation. Pattern Recognition, 123, 108397. https://doi.org/10.1016/j.patcog.2021.108397

[36] Ran, Y., Zhou, X., Lin, P., Wen, Y., & Deng, R. (2022). A survey of predictive maintenance: Systems, purposes and approaches. IEEE Communications Surveys & Tutorials, 22(3), 1741-1762. https://doi.org/10.1109/COMST.2020.2985209