Journal of Applied Automation Technologies

Research Paper

# DeepONet-Kalman Modeling for Real-Time Parameter Updating in Compressor Digital Twins

**Nathan Roux[1, *], Clément Lambert[1] and Mathéo Masson[2]**

[1] School of Computer Science, École Polytechnique, Palaiseau, 91128, France
[2] Department of Computer Science, Université Paris-Saclay, Orsay, 91405, France
*Corresponding author: nathan.r@polytechnique.edu

**Abstract.** Compressor digital twins need to rapidly adjust parameters when fouling, clearance change, sensor bias or load transients occur and the nominal model is no longer valid. A DeepONet-Kalman model is presented in this paper to separate nonlinear operator prediction from covariance-aware measurement assimilation. A branch encoder receives a sliding window of pressure, temperature, rotational speed, valve command and flow signals, and a trunk encoder queries the state at the current operating coordinate. The latent product is the predicted output of the compressor. An adaptive extended Kalman layer then updates the flow-capacity, thermal-efficiency, leakage, heat-transfer and sensor-bias parameters based on innovation statistics and bounded covariance inflation. Experiments use a physics-consistent variable-speed compressor simulator with 42 operating sequences, 1.26 million samples, four degradation patterns, two disturbance families, and controlled sensor corruption. The proposed model has achieved a 0.72% pressure error, a 0.88% discharge-temperature error and a 2.6% mean parameter error, and is better than the strongest recurrent baseline at 1.34%, 1.57% and 5.9% respectively. Median online latency is 17.6 ms per update, and 94.1% of the measured states fall within the predicted 95% uncertainty interval. According to the ablation experiments, innovation-conditioned covariance adaptation is required to deal with sudden fouling and bias generation. Based on the above results, neural operator inference and recursive statistical correction can be used to achieve accurate, stable and computationally feasible real-time updates of compressor digital twins.

**Keywords:** *Digital Twin, DeepONet, Kalman Filtering, Compressor, Predictive Maintenance*

Received on 08 November 2023, Accepted on 27 July 2024, Published on 07 August 2024



## Introduction

All kinds of conditions are available for suction, rotation speed, cooling capacity and downstream demand of industrial compressors. A digital twin can use the measurements to find out the internal thermodynamic conditions and status of a machine, but only if the model parameters are still consistent with those of a changing machine. Now, the twin platform frequently incorporates first-principles models and data services to achieve this alignment [1]. High-frequency monitoring can also show a decline in performance well before reaching the traditional alarm threshold [2]. However, the signal of the emerging fault is mixed with control transients and ambient fluctuations [3]. Data-driven surrogates reduce simulation time [4], and hybrid twins maintain physical meaning when training coverage is lacking [5]. Therefore, the remaining engineering problem is not to predict the output of the compressor; rather, it is to update the twin quickly enough so that parameter drift, sensor bias and genuine disturbances cannot be distinguished.

Recursive estimators can be used for online correction naturally. Kalman-family filters are sensitive to the model and uncertainty covariance for prediction and new evidence [6]. However, the performance will decline if the fixed local transition model cannot reproduce the strong nonlinearity in valve dynamics, thermal storage and changing gas properties. Recurrent Neural Networks can learn temporal dependencies [7], but their hidden state does not directly define an operator that can be queried at any operating point. Neural operators overcome this defect by learning mappings of functions rather than mappings of finite-dimensional vectors [8]. DeepONet is relatively simple, as the separate branch and trunk networks encode the observed input history and query location [9]. A compressor twin still needs a statistically sound way to determine how strongly each new measurement should adjust the uncertain physical parameters.

Develop a DeepONet-Kalman model for real-time parameter updating in a variable-speed compressor digital twin in this paper. A DeepONet is used to provide non-linear state forecasts and local sensitivities, and then bounded adaptive Kalman correction of five interpretable health and bias parameters is performed. Uncertainty is carried by the posterior covariance and not inferred from pointwise network residuals. Monitor for innovation inconsistency to detect noise assumption mismatches, and limit parameter transformations that are physically unreasonable. The implemented edge-computing cycle for this paper consists of window construction, operator inference, measurement gating, covariance adaptation and twin-state synchronization. It is a general architecture for operator learning that supports recursive estimation transparency and has a millisecond-level update speed for both gradual degradation and sudden changes in operation.

## Related Work

### Compressor Digital Twins and Data-Driven Surrogates

Digital Twins for rotating machinery integrate the representation of a computational asset and stream measurements bidirectionally. Generally speaking, one or more of the above methods can be employed to determine the efficiency and operating margin of compressors [10]. Hybrid formulations add learned residuals to the above equations to account for unmodeled heat transfer and valve losses without losing the conservation structure [11]. Surrogate models based on feed-forward or recurrent networks further reduce the cost of repeated transient simulations [12]. The above methods have faster prediction speeds but often treat calibration as an offline problem or periodically retrain the whole surrogate. The above measures are not well-suited for a real-time twin because they have a retraining delay, confuse parameter identity with network weights, and offer limited control over how measurement noise propagates in the updated asset state.

### Neural Operators and Recursive State Estimation

Neural operators learn mappings from the input and output functions to approximate a family of dynamical responses under different boundary conditions. DeepONet is a branch-trunk factorization that has shown good approximation performance for parametric differential equations [13]. Physics-informed operator training improves extrapolation by penalizing residuals or invalid states [14]. Kalman filters, on the other hand, keep a compact probabilistic state and recursively update it with observations [15]. Adaptation adjusts the process and measurement covariances according to the new information to address problems of sensor quality and degradation rate [16]. Existing combinations of neural predictors and filters typically estimate observable states while keeping slow physical parameters constant, or use general networks as state transition models without leveraging operator queries. A gap exists in the compressor-specific architecture for learning transient response operators, updating interpretable parameters, enforcing bounds and reporting update latency together with uncertainty consistency.

## Compressor Digital Twin and Operator Formulation

### Twin State, Parameters, and Measurements

The digital twin of the single-stage variable-speed reciprocating compressor is used at the supervisory level. Dynamic states include intake manifold pressure, exhaust manifold pressure, gas temperature, wall temperature, and normalized mass flow memory. Fast crank-angle phenomena are condensed into map-based source terms, and thus the model can still be used for $20-50\ \mathrm{Hz}$ updates. The parameter vector includes a flow-capacity multiplier, an isentropic-efficiency multiplier, a leakage coefficient, an effective heat-transfer coefficient, and a discharge-temperature sensor bias. These amounts have different responses to fouling, clearance growth, sealing degradation, cooling change and instrumentation drift. Measured Inputs are suction pressure and temperature, shaft speed, valve command, discharge pressure, discharge temperature and mass flow. All channels are time-stamped, only impossible spikes are filtered out, and normalized with training-set statistics.

The continuous twin is written as a parameterized state equation and observation equation:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\theta}(t)) + \boldsymbol{w}(t) \qquad \text{Eq.(1)}$$

where $\mathbf{x}$ is the thermodynamic state, $\mathbf{u}$ is the commanded and environmental input, $\boldsymbol{\theta}$ is the slowly varying parameter vector, and $\mathbf{w}$ represents unresolved dynamics. At sampling instant $k$, the instrument vector follows:

$$\boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}_k) + \boldsymbol{v}_k \qquad \text{Eq.(2)}$$

where $\mathbf{v}_k$ denotes measurement disturbance. Parameters are mapped through logistic or exponential transformations so that the filter operates in an unconstrained coordinate system while the reported twin always satisfies positive leakage, finite heat transfer, and prescribed efficiency limits.

**DeepONet Representation**

A sliding history with a length of 64 samples is supplied to the branch encoder. It includes the seven measured and commanded channels, first differences, missing-data masks, and two summary statistics for each physical signal. The trunk encoder receives normalized query time, speed, pressure ratio, and ambient-temperature coordinates. The two encoders each produce 96 latent features. Their inner product, followed by a compact output head, predicts the five dynamic states at the next update. Figure 1 should depict the complete signal path from sensors and control inputs through window assembly, branch and trunk encoders, latent fusion, the physics guard, Kalman correction, and synchronized digital-twin outputs.

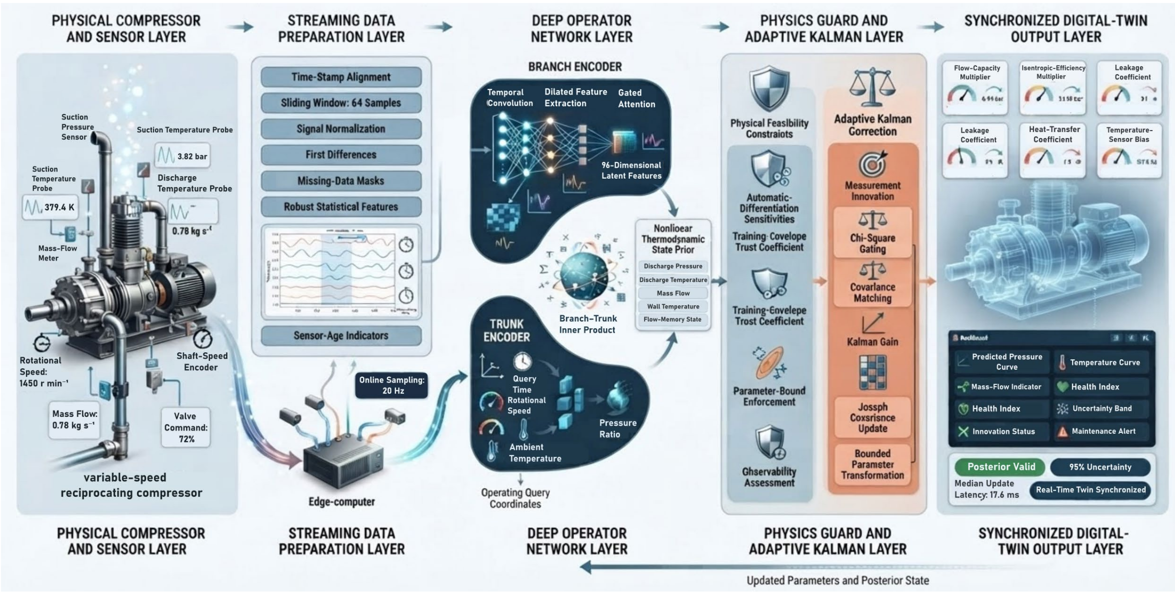


**Figure 1.** Overall architecture of the proposed DeepONet-Kalman compressor digital twin.

For output component $j$, the operator approximation is expressed as:

$$\hat{x}^{j}_{k|k-1} = \boldsymbol{b}_{\phi}(\mathcal{H}_k)^{\top}\boldsymbol{t}_{\psi}(\boldsymbol{q}_k) + c_j \qquad \text{Eq.(3)}$$

where $\mathcal{H}_k$ is the multichannel history, $\mathbf{q}_k$ is the query coordinate, and $\mathbf{b}_{\phi}$ and $\mathbf{t}_{\psi}$ denote the branch and trunk feature vectors, respectively.

Training minimizes a weighted state loss together with derivative-smoothness and feasibility penalties:

$$\mathcal{L} = \lambda_y\mathcal{L}_y + \lambda_d\mathcal{L}_d + \lambda_p\mathcal{L}_p + \lambda_r\|\boldsymbol{\Omega}\|_2^2 \qquad \text{Eq.(4)}$$

The feasibility term penalizes negative flow; it refers to the predicted gas temperature during compression being lower than the intake temperature or the pressure ratio exceeding the simulation range. The derivative term is calculated through short-term disturbance queries and limits oscillatory interpolation. These terms do not require the network to reproduce a detailed crank angle model. They are always observable supervisory behaviors.

Branch features are generated by two temporal convolution blocks and gated attention over the 64-sample window. Dilated kernels can show both the initial response of the actuator and a delayed thermal memory without a deep recurrent stack. The trunk path is a sinusoidal time feature and learned embedding of the speed-to-pressure ratio. Before calculating the potential inner product, both feature vectors are normalized using the second moment method to prevent high-amplitude branching channels from dominating the query representation. Separate linear output heads are retained for pressure, temperature, wall temperature and flow memory due to different noise scales and derivative smoothness. Shared features are evaluated once, and only the compact parameter-conditioned output layers are differentiated in the online cycle.

### Offline Training and Sensitivity Extraction

Training sequences are stratified according to speed, pressure ratio, degradation class, and transient severity. Windows from the same simulated run are kept in one partition to prevent temporal leakage. AdamW optimization is followed by limited-memory BFGS refinement of the output head. Noise augmentation introduces correlated pressure drift, temperature lag, flow dropout, and small timestamp jitter.

Local sensitivities required by the filter are obtained through automatic differentiation of the operator output with respect to the transformed parameter coordinates:

$$\boldsymbol{G}_k = \frac{\partial \widehat{\boldsymbol{x}}_{k|k-1}}{\partial \boldsymbol{\zeta}_{k-1}} \qquad \text{Eq.(5)}$$

where $\boldsymbol{\zeta}$ denotes the unconstrained parameter coordinates. This derivative is evaluated within the same inference graph and adds less than two milliseconds on the target processor. A trust coefficient reduces $\mathbf{G}_k$ when the query lies far outside the training envelope, preventing an extrapolative neural gradient from producing an aggressive correction.

The offline data generator changes one parameter at a time in a part of the experimental design and uses Latin hypercube sampling for the other mixed cases. Single-factor runs improve local identifiability, but mixed runs teach the operator about the cross-couplings in actual operation. Validation monitors rollout error and sensitivity sign. A checkpoint is considered invalid if it predicts an acceptable state but violates a known monotonic relationship, for example, by increasing effective leakage while simultaneously raising delivered flow at a constant speed and pressure ratio. The above sensitivity test will not be influenced by a small gradient that is convenient for calculation but not realistic in practice.

## Adaptive DeepONet-Kalman Parameter Updating

### Augmented Prediction and Covariance Propagation

The recursive estimator extends the predicted thermodynamic state by adding the transformed parameter vector. Dynamic states inherit the DeepONet forecast, and parameters follow a random walk process with expected degradation rate as the process covariance. They are not the same: A neural operator shows a quick non-linear change, but a filter attributes the extended innovation pattern to a small number of physical alterations.

The augmented prior state is defined as:

$$\widehat{\boldsymbol{s}}_{k|k-1} = \begin{bmatrix} \widehat{\boldsymbol{x}}_{k|k-1} \\ \widehat{\boldsymbol{\zeta}}_{k-1|k-1} \end{bmatrix} \qquad \text{Eq.(6)}$$

A block transition Jacobian combines the neural-network state sensitivity, the operator parameter derivative $\mathbf{G}_k$, and identity parameter dynamics. The prior covariance is propagated as:

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{F}_k \boldsymbol{P}_{k-1|k-1} \boldsymbol{F}_k^{\top} + \boldsymbol{Q}_k \qquad \text{Eq.(7)}$$

The growth of process covariance should be bounded. Each diagonal term is limited by engineering bounds based on the fastest plausible fouling, leakage, thermal and sensor-bias variations. Cross-covariance is kept because discharge temperature and pressure jointly indicate efficiency and heat transfer, and flow and pressure ratio jointly identify capacity and leakage.

https://doi.org/10.64972/jaat.2024v2.322p27e:374-385

Observability is evaluated over a rolling window, and then the parameter block is corrected. Stacking local measurement Jacobians is scaled by the expected sensor accuracy, and then their singular values are compared to a lower bound for insufficient information. When capacity and leakage are almost collinear at a constant operating point, their joint correction is projected onto the observable direction instead of permitting two large and opposing parameter changes. If an appropriate source of stimulation cannot be provided, the twins will report combined uncertainty and avoid attribution. This way is required for the maintenance application because a low state of residual does not mean that all health parameters have been individually identified.

**Innovation-Conditioned Correction**

The innovation is a gated measurement and its predicted observation. A Hampel precheck rejects isolated samples that exceed six robust standard deviations, but sustained shifts are retained because they may be real parameter changes.

The measurement residual and its covariance are:

$$\boldsymbol{r}_k = \boldsymbol{y}_k - \boldsymbol{h}(\hat{\boldsymbol{s}}_{k|k-1})$$
$$\boldsymbol{S}_k = \boldsymbol{H}_k \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^\top + \boldsymbol{R}_k \quad \text{Eq.(8)}$$

The Kalman gain is calculated as:

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^\top \boldsymbol{S}_k^{-1} \quad \text{Eq.(9)}$$

The corrected augmented state is then obtained from:

$$\hat{\boldsymbol{s}}_{k|k} = \hat{\boldsymbol{s}}_{k|k-1} + \boldsymbol{K}_k \boldsymbol{r}_k \quad \text{Eq.(10)}$$

Figure 2 should show the prediction stage, innovation testing, spike rejection, covariance matching, gain calculation, bounded parameter transformation, posterior validation, and rollback when the normalized innovation remains inconsistent.

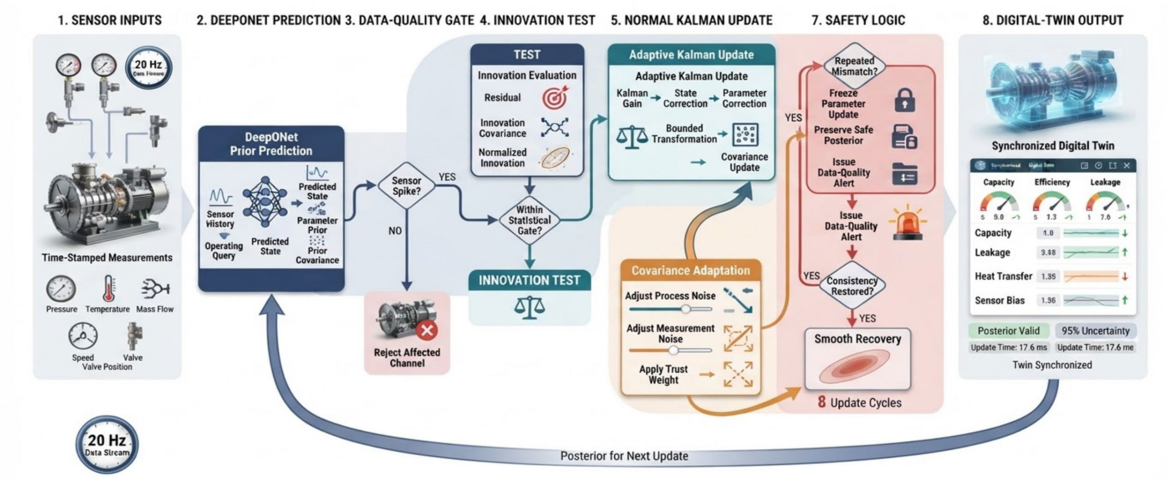


**Figure 2.** Real-time adaptive Kalman correction and safety logic.

The covariance is updated using the Joseph stabilized form to preserve symmetry and positive semi definiteness under finite-precision computation:

$$\boldsymbol{P}_{k|k} = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k) \boldsymbol{P}_{k|k-1} (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k)^\top + \boldsymbol{K}_k \boldsymbol{R}_k \boldsymbol{K}_k^\top \quad \text{Eq.(11)}$$

This form requires more matrix multiplications than the simplified covariance update, but it is more reliable for an edge implementation that uses single-precision neural-network inference and double-precision filtering.

Measurement Channels are updated sequentially when their timestamps differ by more than half of a sampling interval. Propagate the prior to the measurement time, correct it with the available channel subset, and then propagate to the common synchronization point. Temperature has a first-order probe-lag state, and pressure is considered to be effectively instantaneous at the supervisory sampling rate. Therefore, missing flow observations reduce the amount of information but do not stop the update. Channel-specific gating does not reject the entire measurement vector when only one sensor is corrupted. The status has been added with an age flag in all channels of the digital twin, allowing downstream optimization to distinguish between the most recent posteriors and predictions delayed due to communication failures.

### Covariance Adaptation and Real-Time Safeguards

A fixed noise model reacts too slowly to abrupt fouling and too strongly to temporary sensor degradation. The proposed method therefore calculates the normalized innovation squared:

$$NIS_k = \boldsymbol{r}_k^\top \boldsymbol{S}_k^{-1} \boldsymbol{r}_k \qquad \text{Eq.(12)}$$

An exponentially weighted covariance mismatch is calculated independently:

$$\boldsymbol{C}_k = \beta \boldsymbol{C}_{k-1} + (1-\beta)[\boldsymbol{r}_k \boldsymbol{r}_k^\top - \boldsymbol{H}_k \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^\top] \qquad \text{Eq.(13)}$$

Positive diagonal components of $\mathbf{C}_k$ increase the measurement covariance, whereas persistent directional innovations increase only the relevant parameter-process terms.

The bounded parameter covariance adaptation is defined as:

$$\boldsymbol{Q}_{\theta,k} = clip\left[\boldsymbol{Q}_{\theta,0} \odot exp\left(\gamma_q \boldsymbol{d}_k\right), \boldsymbol{Q}_{min}, \boldsymbol{Q}_{max}\right] \qquad \text{Eq.(14)}$$

A similar rule updates $\mathbf{R}_k$, but it uses a smaller adaptation rate. The runtime rollback mechanism stores the previous valid posterior rather than manipulating the generated figure. If the innovation remains outside the chi-square gate for five consecutive updates, parameter correction is frozen, the operator continues predicting the thermodynamic state, and a data-quality flag is issued. Once statistical consistency returns, the covariance is inflated smoothly over eight updates. The complete update cycle uses a preallocated tensor buffer and factorized linear solves; the software implementation does not explicitly form a matrix inverse.

## Experimental Evaluation and Data Analysis

### Experimental Setup and Baselines

A physics-consistent supervisory simulator has been set up for a medium-sized variable-speed compressor. Forty-two sequences of 30 minutes were generated at 20Hz, and after excluding warm-up periods, 1.26 million synchronous samples were obtained. The speed is 900-1800 r/min; the suction pressure is 0.82-1.08 bar; and the pressure ratio is 2.1-5.0. The training, validation and test sets had 26, 6 and 10 complete sequences respectively. Four degradation processes have been introduced: gradual capacity loss, gradual leakage, reduced heat conduction, and sudden sensor drift. The two disturbance categories cover the suction-pressure steps and the valve command ramps. Sensor noise is colored and heteroscedastic, indicating that the observed industrial state signals rarely conform to the independent Gaussian assumption [17]. The simulator serves as a controlled reference and is not presented as plant data; its known parameters can directly calculate tracking error.

All decay sequences have a pre-alteration stage, a transition period and a stable state after modification. Gradual capacity loss is a slope of 0.003-0.010 per minute; leakage ramps are between 0.001 and 0.004 per minute; cooling impairment changes effective heat transfer by 10-35%; and temperature-bias steps range from 1.5 to 6.0 K. Disturbance timing is random; thus, degradation does not always occur after a stable baseline. Test sequences also combine one health change with an unrelated suction or valve transient to prevent the filter from succeeding simply by associating a fixed time index with deterioration.

Baselines were an extended Kalman filter with the nominal reduced-order model, a gated recurrent unit predictor followed by Kalman correction, a long short-term memory predictor with recursive least-squares parameter adaptation, standalone DeepONet without measurement correction, and a physics-informed multilayer perceptron with an unscented filter. Recurrent architectures are common references for multivariate equipment prognostics [18], and physics-informed learning provides a relevant comparison when conservation information is available [19]. Deep-learning methods have also been applied to reciprocating-compressor fault diagnosis [20]. Hyperparameters were selected on the validation sequences, not the test set. Accuracy was measured with normalized root-mean-square error, mean absolute percentage error, and mean parameter error. Reliability used empirical 95% interval coverage and normalized innovation consistency. Runtime measurements included window assembly, network inference, differentiation, filter update, and twin synchronization on an eight-core industrial processor.

The selected model is the average of the errors in pressure, temperature, and flow, plus a penalty for insufficient interval coverage. Therefore, we avoid selecting an overly confident model whose point estimate is exactly accurate. All baselines received the same standardized input and measured gated front-end. Adjust the network width to a similar parameter budget and train the neural network three times using different initialization methods. Reported errors are aggregated across full test sequences, and ranks are calculated per sequence so that the longest trajectory does not take precedence. Bootstrap intervals use complete sequences as resampling units and maintain the temporal dependence of a run.

Figure 3 is referenced once here with all of its panels. Figure 3(a) reports discharge-pressure trajectories for five load ramps, where the proposed curves remain within 0.03 bar of the truth after each transition. Figure 3(b) shows discharge temperature for the same ramps and limits the largest overshoot error to 1.7 K. Figure 3(c) gives mass-flow parity over 8,000 test points with $R^2 = 0.994$. Together the panels establish that the operator forecast supplies a sufficiently accurate prior before recursive correction.

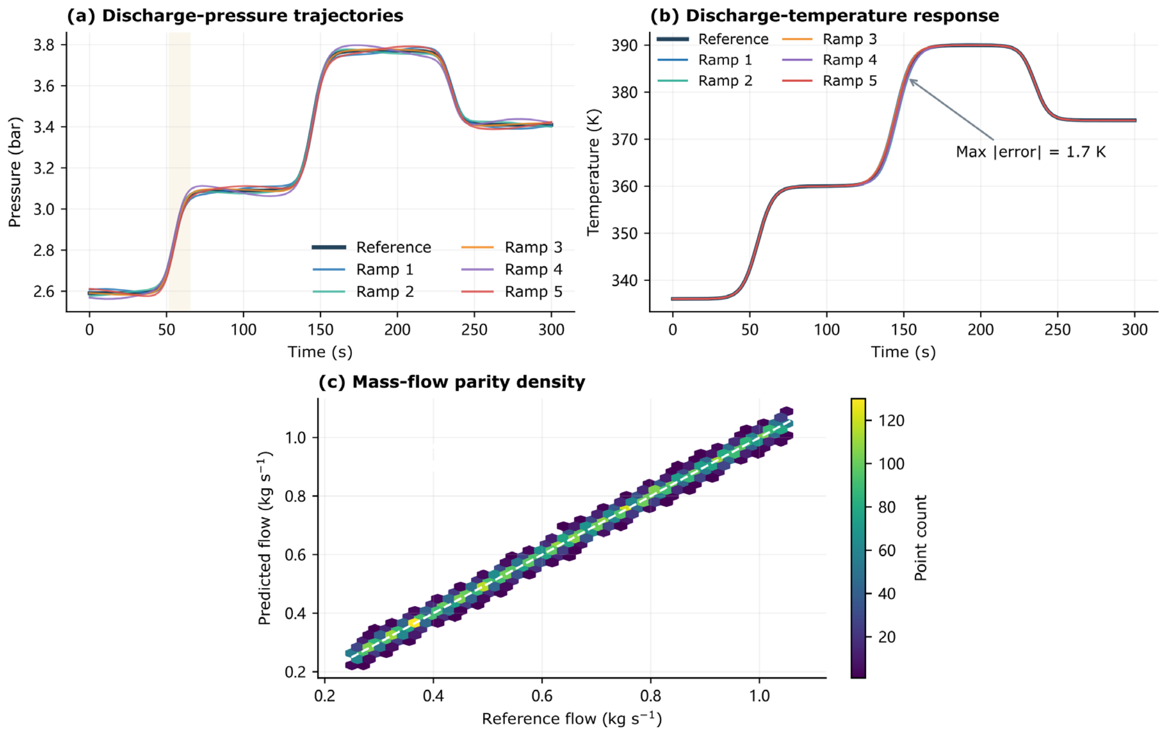


**Figure 3.** State-prediction accuracy: (a) discharge-pressure trajectories; (b) discharge-temperature trajectories; (c) mass-flow parity density.

### Parameter Tracking and Comparative Accuracy

Parameter tracking was evaluated under both slow drift and abrupt onset. Figure 4 is referenced once. Figure 4(a) shows the capacity multiplier declining from 1.00 to 0.88 over 18 minutes; the DeepONet-Kalman estimate has a 0.9% terminal error and a 22 s effective delay. Figure 4(b) presents leakage growth from 0.015 to 0.052, with the posterior remaining inside its 95% band for 93.7% of samples. Figure 4(c) displays the heat-transfer coefficient under a two-stage cooling impairment and shows recovery of the new level within 31 s. These three cases are difficult for offline calibration because the informative transient occupies only a small part of the operating record [21].

Across the ten test sequences, the proposed method achieved 0.72% discharge-pressure error, 0.88% temperature error, 1.05% mass-flow error, and 2.6% mean parameter error. The strongest alternative, the gated recurrent unit with Kalman correction, produced 1.34%, 1.57%, 1.62%, and 5.9%, respectively. Standalone DeepONet reached 1.01% state error on average but could not maintain an interpretable health trajectory after parameter drift. The nominal extended filter showed the largest transient bias because its local model underestimated valve and thermal coupling. Figure 5 is referenced once. Figure 5(a) compares normalized state error across six methods; Figure 5(b) uses a critical-difference style rank plot over the ten sequences; Figure 5(c) shows the distribution of parameter error as violin and box overlays. The proposed median parameter error is 2.2%, and its 90th percentile is 4.8%.

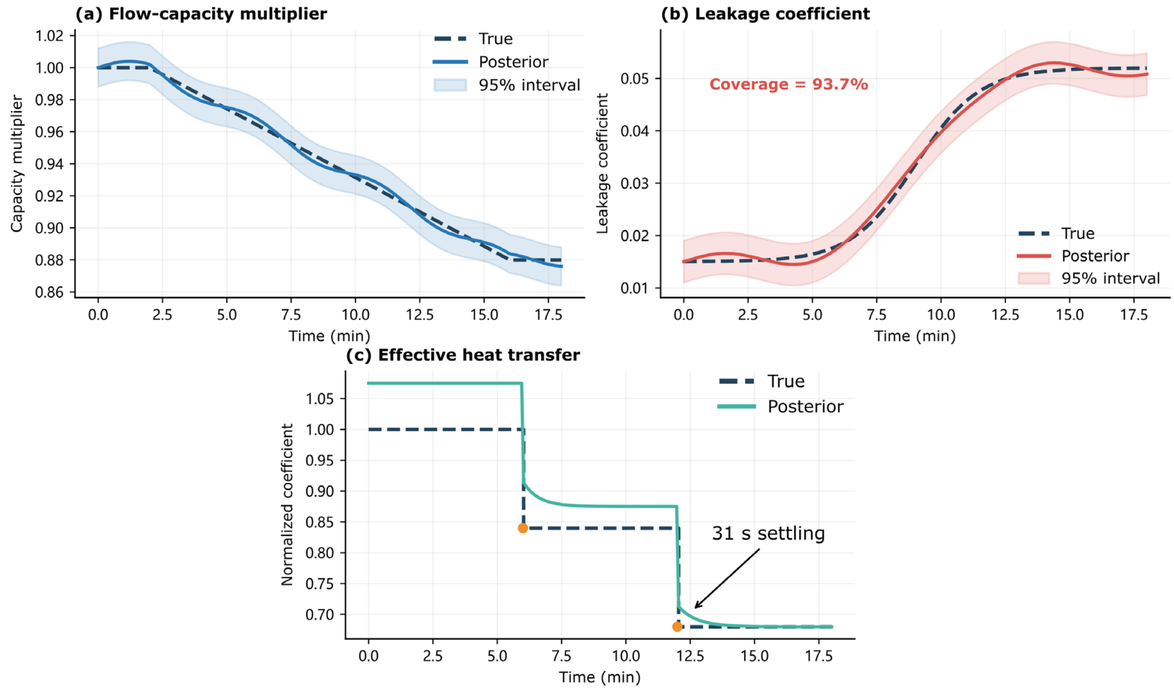


**Figure 4.** Health-parameter tracking: (a) flow-capacity multiplier; (b) leakage coefficient with uncertainty; (c) effective heat-transfer coefficient.

The improvement is not uniform across parameters. Capacity is identified most accurately because it affects flow and discharge pressure across the speed range. Temperature bias is separated rapidly when pressure and flow remain consistent. Efficiency and heat transfer show stronger posterior correlation during slow load ramps, producing wider intervals even when their combined temperature prediction is accurate. Leakage is least observable near the lowest pressure ratio, but its error decreases once the sequence crosses a ratio of 3.2. These patterns explain why the aggregate parameter metric remains higher than the three state errors and support reporting posterior covariance rather than a single health score [22].

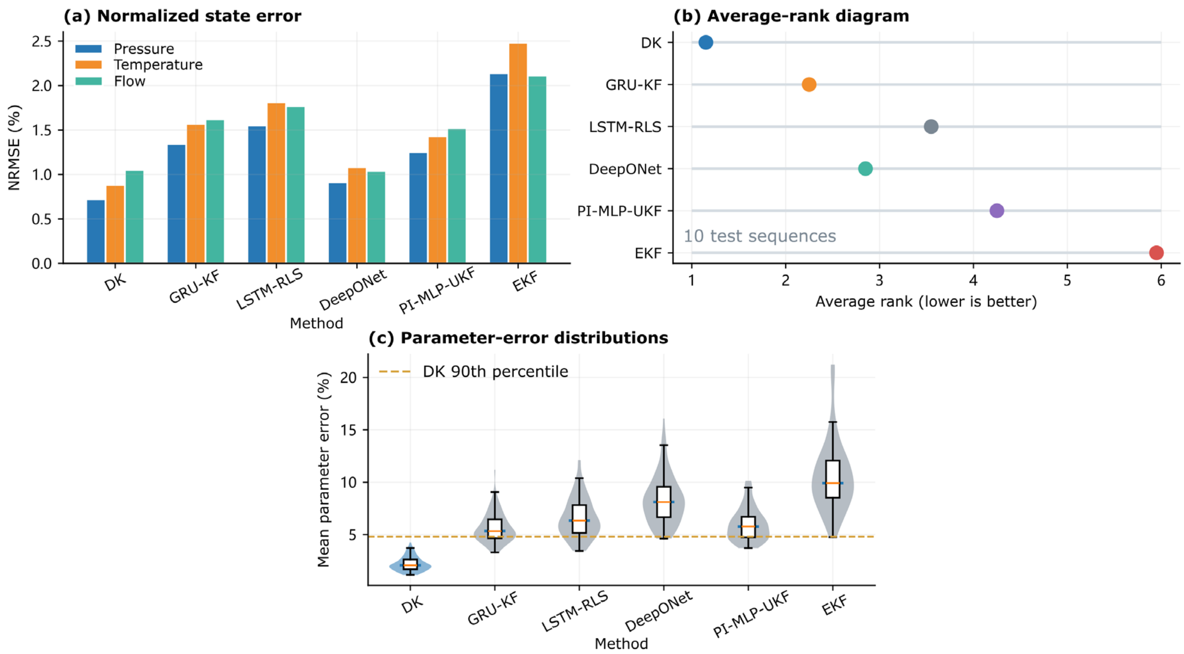


**Figure 5.** Comparative performance: (a) grouped error bars; (b) average-rank diagram; (c) parameter-error distributions.

**Robustness, Ablation, and Computational Behavior**

Robustness tests changed noise amplitude, missing-data fraction, degradation rate, and operating-envelope distance. Figure 6 is referenced once. Figure 6(a) is the single heat map in the paper and reports pressure error over five noise levels and six dropout levels; error remains below 1.2% up to 1.5-times nominal noise and 10% dropout. Figure 6(b) plots calibration error against distance from the training envelope for four uncertainty treatments, showing that trust-weighted sensitivity prevents the sharp rise observed with unbounded gradients. Figure 6(c) presents normalized innovation histograms for normal operation, abrupt fouling, and sensor bias, including chi-square reference limits. The adaptive model restores median innovation to 1.06 after the transient. Online identification of noise covariance is an established problem in adaptive Kalman filtering [23]. Covariance matching is also used to accommodate time-varying uncertainty in nonstationary monitoring processes [24].

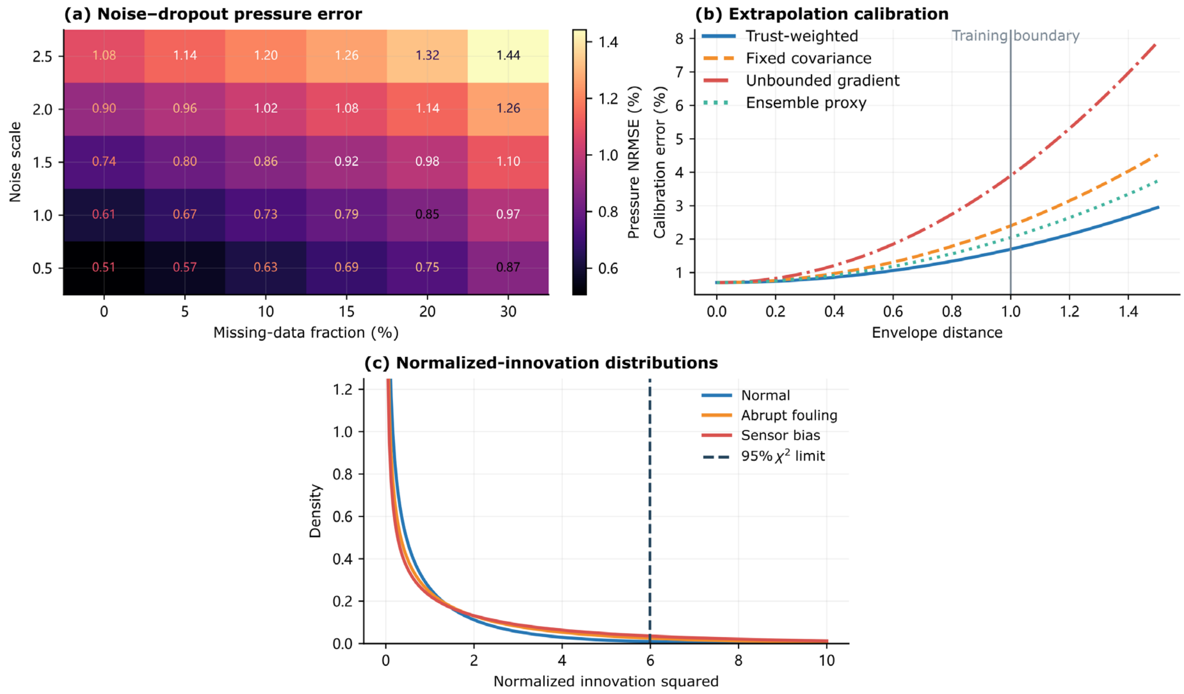


**Figure 6.** Robustness and uncertainty behavior: (a) noise-dropout error heat map; (b) extrapolation calibration curves; (c) normalized-innovation distributions.

Ablation results separated four design choices. Removing covariance adaptation increased mean parameter error from 2.6% to 4.3%, while removing the physics feasibility penalty raised pressure error from 0.72% to 0.96%. Replacing automatic-differentiation sensitivities with finite differences increased latency and produced noisier leakage estimates. Eliminating the trust coefficient had little effect in-envelope but raised the worst extrapolation error from 3.4% to 7.9%. These outcomes agree with broader evidence that hybrid constraints are most valuable at sparse or shifted operating conditions [25]. They also show that uncertainty calibration should be tested independently of point prediction accuracy [26].

Figure 7 is referenced once. Figure 7(a) decomposes median update latency into 1.8 ms preprocessing, 8.9 ms operator inference, 1.7 ms sensitivity evaluation, 3.6 ms Kalman correction, and 1.6 ms synchronization, totaling 17.6 ms. Figure 7(b) gives latency distributions for five window lengths and confirms that the 99th percentile remains below 31 ms at the selected length of 64. Figure 7(c) plots accuracy against computation for all six methods, placing DeepONet-Kalman on the Pareto frontier. These timing results leave margin for a 20 Hz update schedule, although deployment latency will depend on driver and communication overhead [27]. Edge-oriented predictive-maintenance studies likewise emphasize that end-to-end timing, rather than network inference alone, determines operational feasibility [28].

Memory consumption: Weights are about 38MB, pre-allocated Tensors are around 6MB, and the filter and diagnostic buffers are less than 1MB. Reuse the branch representation for multiple query coordinates to reduce short-horizon forecast cost. Reduce the update frequency from 20Hz to 10Hz; although the increase in pressure error will only be 0.06%, the delay in detecting a sudden bias will be about 0.4 seconds. At 5 Hz, the transient pressure error and innovation overshoot are relatively large. Therefore, the selected rate is more responsive than a specific number. A production implementation would be to use a slower update rate in a stable state and increase it after a control event or innovation alarm.

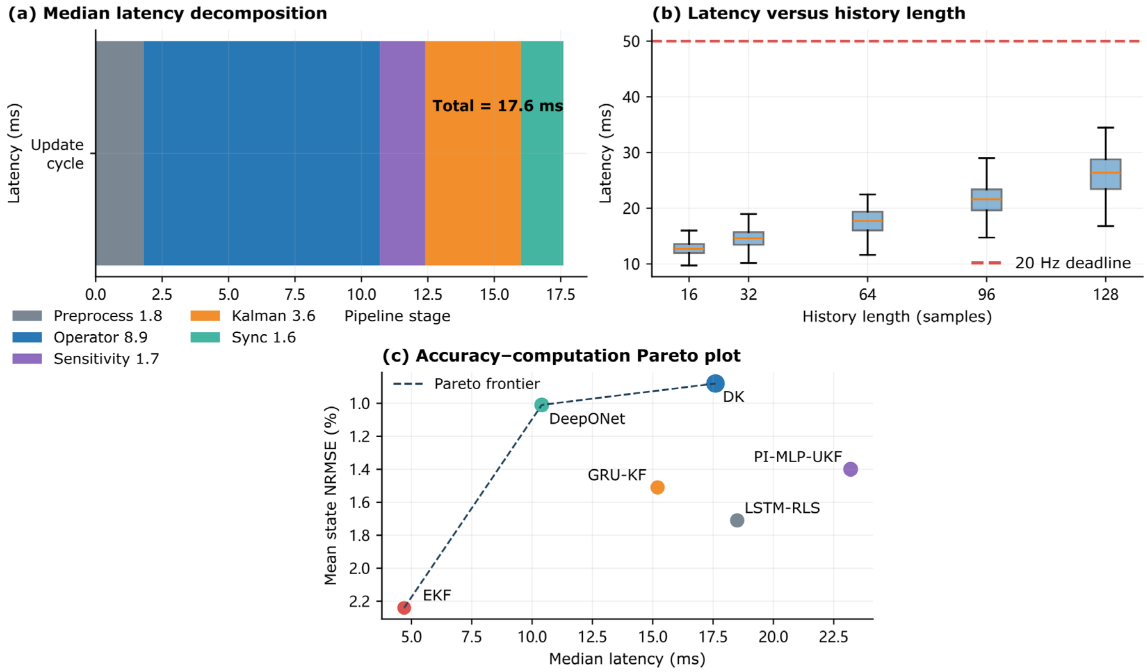


**Figure 7.** Computational performance: (a) latency decomposition; (b) latency distributions versus history length; (c) accuracy-computation.

A controlled benchmark shows what the model does, but not all the plant factors. Gas composition is in the normal range, there is a valve fault shown by an effective loss coefficient, and the network interface is simulated locally. Therefore, the reported advantages should be viewed as evidence of algorithmic feasibility under known ground truth. A field trial would need independent equipment and time-stamped records of acceptance before the health estimate was used to modify the maintenance or control plan.

## Conclusion

A DeepONet-Kalman architecture for real-time parameter update in compressor digital twins is introduced in this paper. The operator network is a nonlinear transient behavior model used for multi-channel historical and operational queries. The adaptive filter combines measurements, updates interpretable health parameters, and propagates uncertainty through stable posterior covariance. Physical parameter transformations, innovation gating, trust-weighted sensitivities and bounded covariance adaptation make the coupling suitable for a continuous supervisory update cycle rather than an offline surrogate study.

Currently, this study is constrained by a controlled simulator and a compact lumped representation of valve, thermal and leakage effects. Parameter identifiability is reduced if the excitation is too small or if multiple faults have nearly collinear measurement signatures. This method also assumes that timestamp alignment and some signal validations have already been performed. Therefore, the actual reasons for the maintenance decisions of all these compressors are unknown.

Future work will add long-term plant data and regular maintenance checks to the method, expand the scope of operators to multiple gas compositions and multi-stage interactions, and study distributed updates among compressor fleets. Further research can be conducted on combining conformal uncertainty bounds with recursive covariance, proactive incentive strategies to enhance parameter observability (without affecting production), and the formal safety interface of updated twins and supervised optimization.

**Author Contributions**

Nathan Roux contributes to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, supervision. Clément Lambert contributes to data collection, draft preparation, manuscript editing. Mathéo Masson contributes to analysis, investigation, data collection. 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] Jones, D., Snider, C., Nassehi, A., Yon, J., & Hicks, B. (2020). Characterising the digital twin: A systematic literature review. CIRP Journal of Manufacturing Science and Technology, 29, 36–52. https://doi.org/10.1016/j.cirpj.2020.02.002

[2] Fuller, A., Fan, Z., Day, C., & Barlow, C. (2020). Digital twin: Enabling technologies, challenges and open research. IEEE Access, 8, 108952–108971. https://doi.org/10.1109/ACCESS.2020.2998358

[3] Rasheed, A., San, O., & Kvamsdal, T. (2020). Digital twin: Values, challenges and enablers from a modeling perspective. IEEE Access, 8, 21980–22012. https://doi.org/10.1109/ACCESS.2020.2970143

[4] Liu, M., Fang, S., Dong, H., & Xu, C. (2021). Review of digital twin about concepts, technologies, and industrial applications. Journal of Manufacturing Systems, 58, 346–361. https://doi.org/10.1016/j.jmsy.2020.06.017

[5] Errandonea, I., Beltrán, S., & Arrizabalaga, S. (2020). Digital twin for maintenance: A literature review. Computers in Industry, 123, 103316. https://doi.org/10.1016/j.compind.2020.103316

[6] Song, M., Astroza, R., Ebrahimian, H., Moaveni, B., & Papadimitriou, C. (2020). Adaptive Kalman filters for nonlinear finite element model updating. Mechanical Systems and Signal Processing, 143, 106837. https://doi.org/10.1016/j.ymssp.2020.106837

[7] Wang, Y., Zhao, Y., & Addepalli, S. (2021). Practical options for adopting recurrent neural network and its variants on remaining useful life prediction. Chinese Journal of Mechanical Engineering, 34, 69. https://doi.org/10.1186/s10033-021-00588-x

[8] Wang, S., Wang, H., & Perdikaris, P. (2021). Learning the solution operator of parametric partial differential equations with physics-informed DeepONets. Science Advances, 7(40), eabi8605. https://doi.org/10.1126/sciadv.abi8605

[9] Lu, L., Jin, P., Pang, G., Zhang, Z., & Karniadakis, G. E. (2021). Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. Nature Machine Intelligence, 3(3), 218–229. https://doi.org/10.1038/s42256-021-00302-5

[10] Shao, G., & Helu, M. (2020). Framework for a digital twin in manufacturing: Scope and requirements. Manufacturing Letters, 24, 105–107. https://doi.org/10.1016/j.mfglet.2020.04.004

[11] Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., & Yang, L. (2021). Physics-informed machine learning. Nature Reviews Physics, 3(6), 422–440. https://doi.org/10.1038/s42254-021-00314-5

[12] Guo, L., Lei, Y., Xing, S., Yan, T., & Li, N. (2018). Deep convolutional transfer learning network: A new method for intelligent fault diagnosis of machines with unlabeled data. IEEE Transactions on Industrial Electronics, 66(9), 7316-7325. https://doi.org/10.1109/TIE.2018.2877090

[13] Lu, L., Meng, X., Mao, Z., & Karniadakis, G. E. (2021). DeepXDE: A deep learning library for solving differential equations. SIAM Review, 63(1), 208–228. https://doi.org/10.1137/19M1274067

[14] Willard, J., Jia, X., Xu, S., Steinbach, M., & Kumar, V. (2022). Integrating scientific knowledge with machine learning for engineering and environmental systems. ACM Computing Surveys, 55(4), 1–37. https://doi.org/10.1145/3514228

[15] Hu, G., Gao, B., Zhong, Y., & Gu, C. (2020). Unscented Kalman filter with process noise covariance estimation for vehicular INS/GPS integration system. Information Fusion, 64, 194–204. https://doi.org/10.1016/j.inffus.2020.08.005

[16] Xiong, K., Wei, C., & Zhang, H. (2021). Q-learning for noise covariance adaptation in extended Kalman filter. Asian Journal of Control, 23(4), 1803–1816. https://doi.org/10.1002/asjc.2336

[17] Huang, Y., Zhang, Y., Zhao, Y., & Chambers, J. A. (2020). Variational adaptive Kalman filter with Gaussian-inverse-Wishart mixture distribution. IEEE Transactions on Automatic Control, 66(4), 1786–1793. https://doi.org/10.1109/TAC.2020.2995674
[18] Wang, B., Lei, Y., Yan, T., Li, N., & Guo, L. (2020). Recurrent convolutional neural network: A new framework for remaining useful life prediction of machinery. Neurocomputing, 379, 117–129. https://doi.org/10.1016/j.neucom.2019.10.064
[19] Geneva, N., & Zabaras, N. (2020). Modeling the dynamics of PDE systems with physics-constrained deep auto-regressive networks. Journal of Computational Physics, 403, 109056. https://doi.org/10.1016/j.jcp.2019.109056
[20] Zhang, Y., Ji, J., & Ma, B. (2020). Reciprocating compressor fault diagnosis using an optimized convolutional deep belief network. Journal of Vibration and Control, 26(17–18), 1538–1548. https://doi.org/10.1177/1077546319900115
[21] Moghadam, F. K., Rebouças, G. F. de S., & Nejad, A. R. (2021). Digital twin modeling for predictive maintenance of gearboxes in floating offshore wind turbine drivetrains. Forschung im Ingenieurwesen, 85, 273–286. https://doi.org/10.1007/s10010-021-00468-9
[22] Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P., Cao, X., Khosravi, A., Acharya, U. R., Makarenkov, V., & Nahavandi, S. (2021). A review of uncertainty quantification in deep learning: Techniques, applications and challenges. Information Fusion, 76, 243–297. https://doi.org/10.1016/j.inffus.2021.05.008
[23] Limaverde Filho, J. O. A., Fortaleza, E. L. F., Silva, J. G., & de Campos, M. C. M. M. (2022). Adaptive Kalman filtering for closed-loop systems based on the observation vector covariance. International Journal of Control, 95(7), 1731–1746. https://doi.org/10.1080/00207179.2020.1870158
[24] Zhang, L., & Duan, L. (2022). Cross-scenario transfer diagnosis of reciprocating compressor based on CBAM and ResNet. Journal of Intelligent & Fuzzy Systems, 43(5), 5929-5943.https://doi.org/10.3233/JIFS-213340
[25] Cai, Y., Wang, Z., Yao, L., Lin, T., & Zhang, J. (2022). Ensemble dilated convolutional neural network and its application in rotating machinery fault diagnosis. Computational Intelligence and Neuroscience, 2022, 6316140. https://doi.org/10.1155/2022/6316140
[26] Scalia, G., Grambow, C. A., Pernici, B., Li, Y.-P., & Green, W. H. (2020). Evaluating scalable uncertainty estimation methods for deep learning-based molecular property prediction. Journal of Chemical Information and Modeling, 60(6), 2697–2717. https://doi.org/10.1021/acs.jcim.9b00975
[27] Santos, M. Y., Sá, J. O. e, Costa, C., Galvão, J., Andrade, C., Martinho, B., Lima, F. V., & Costa, E. (2021). TIP4.0: Industrial Internet of Things platform for predictive maintenance. Sensors, 21(14), 4676. https://doi.org/10.3390/s21144676
[28] Michala, A. L., Vourganas, I., & Coraddu, A. (2021). Vibration edge computing in maritime IoT. ACM Transactions on Internet of Things, 3(1), 1-18. https://doi.org/10.1145/3484717