Data Engineering and Applications

Research Paper

Application of High-Dimensional Sensor Data Dimensionality Reduction Based on Autoencoders in Smart Factories

Dawid Eustachy Długosz [1, *] and Ryszard Radosz[2]

[1] Faculty of Informatics, University of Gdansk, Gdansk, 80-952, Poland
[2] Faculty of Computer Science, University of Wroclaw, Wroclaw, 50-137, Poland
*Corresponding author: dawid.ed@ug.edu.pl

Abstract. Smart factories continuously generate heterogeneous sensor streams, and the dimensions, redundancy and operating-condition dependence of these streams inhibit real-time diagnosis. Develop a context-aware sparse denoising autoencoder for compressing high-dimensional industrial measurements and retaining fault-sensitive structure in this study. The model is a single edge-deployable encoder that integrates strong channel normalization, condition embeddings, structured group sparsity and a neighbourhood-preserving regulariser. Evaluation used 1.86 million synchronous samples from 132 vibration, acoustic, current, temperature, pressure and control channels under 18 operating modes and nine health statuses. To avoid time loss and assess the stability of the asset, machine-level splitting and multiple training seeds were employed. The proposed encoder reduced the representation to 12 dimensions and achieved a 90.9% reduction in dimensions. At this time, the normalized reconstruction error was 0.031, the latent-space silhouette coefficient was 0.681, and the downstream macro-F1 score reached 0.947. Compared with principal component analysis, uniform manifold approximation, a plain autoencoder, and a variational autoencoder all improved the macro-F1 score by 9.4, 7.1, 4.1 and 2.8 percentage points, respectively. With less than 15% missing channels and a signal-to-noise ratio of 5 dB, macro-F1 was still 0.901 and 0.887. The compact encoder needed 0.83ms per sample for the industrial edge computer and performed online compression before storage and diagnostic inference. As shown in the above results, condition-aware nonlinear reduction can reduce communication and computation overhead without losing weak fault features, and a practical representation layer for large-scale smart-factory monitoring has been provided.

Keywords: *Autoencoder, Dimensionality Reduction, High-Dimensional Sensors, Fault Diagnosis, Edge Intelligence, Representation Learning*

Received on 11 October 2022, Accepted on 24 December 2023, Published on 14 January 2023

Introduction

Smart factories are integrating more and more machine tools, robots, material-handling systems, utilities and inspection stations in dense sensing networks. The reason for the value is not connectivity; rather, stable operation data can be used as the basic measure. Smart-factory performance has thus been linked to the capacity for data coordination, knowledge processing and reduced decision latency across production assets [1]. A manufacturing data ecosystem may include vibration spectra, motor currents, temperatures, acoustic emissions, pressures, controller tags and product-quality signals sampled at different times [2]. Artificial intelligence has increased the range of patterns that can be learned from these streams, but it has also presented the problem of the cost of storing and repeatedly processing correlated channels [3]. Predictive maintenance pipelines are very cost-sensitive because they need to keep track of slight damage indicators over a long period of monitoring [4]. Sustainable manufacturing also adds a demand: The analytical gain should not be achieved through too much computation, communication or sensor duplication [5]. Although the edge-fog-cloud service architecture can distribute the load, it also needs to be relatively compact and informative in its data before being transmitted out of the production cell [6].

https://doi.org/10.64972/dea.2023.v2i1.2911d:1-15

Therefore, dimension reduction in industrial analytics will be employed as a function layer rather than a superficial pre-processing method. Digital-twin systems need reduced states that are sufficiently small for synchronisation but still expressive enough for diagnosis [7]. Machine-learning applications in manufacturing also operate under different recipes, loads, speeds and maintenance conditions that can distort the geometry of a learned feature space [8]. With the increased reconfigurability of manufacturing systems, a model trained in one operating region should not mistake a genuine shift in mode for an early-stage fault [9]. Many sources of high-dimensional monitoring increase the model's visibility but may also raise estimation variance and affect model stability [10]. Computation-enabled sensor design can move some inference closer to the acquisition stage, but a rational method for integrating different channels is still needed [11]. Reviews of sensor anomaly detection also identify noise, missing values, class imbalance and changing context as persistent problems in the deployment phase [12].

Linear projection is simple and intuitive, but it cannot show the non-linear cross-sensor relationships of coupled electromechanical systems well. Generic nonlinear embeddings generally retain local geometry but do not provide a deterministic inverse map, an online encoder, or explicit control over which fault information is compressed. Standard autoencoders solve this problem to some extent; however, their latent variables may contain information about the operating conditions, overfit to the dominant healthy samples, and reconstruct abnormal patterns too accurately. Through a context-aware sparse denoising autoencoder designed as an operational representation layer for smart factories, this paper addresses the above deficiency. Inject regime context into both the encoder and decoder, mask channels during training, impose group sparsity on latent activations, and constrain neighbourhood relations with health-aware pairs. It provides a unified mathematical framework for mixed industrial channels, introduces a deployment workflow that separates condition normalization from health inference, and evaluates on reconstruction quality, structure preservation, diagnostic utility, robustness and edge latency. The problem of how far a high-dimensional sensor vector can be compressed before weak and regime-dependent fault evidence is lost is posed in the study.

Related Work

Linear, Manifold, and Probabilistic Reduction

Classical reduction methods are still suitable for industrial systems because they have transparent computational behaviour. Principal Component Analysis finds new orthogonal directions that best explain the variation of correlated variables and is suitable for stable high-throughput monitoring. Comparison work on large datasets shows that linear projection is effective when the global variance aligns with the target structure, but it can collapse curved manifolds and minority behaviour [13]. Feature filtering and wrapper strategies can keep the original channels, but their selected subsets will change if the operating distribution shifts [14]. Although t-distributed Stochastic Neighbor Embedding and Uniform Manifold Approximation are convenient diagnostic visualizations of nonlinear manifolds, they generally lack an independent mechanism for streaming out-of-sample points.

Probabilistic latent-variable models can address the issue of uncertainty more directly. Variational Autoencoders learn distributions instead of single codes and provide a regularised latent space that supports sampling and interpolation [15]. The prior-matching objective, however, may suppress rare fault modes when the evidence is weak. Unified analyses of shallow and deep anomaly detection show that representation geometry and scoring assumptions cannot be designed independently [16]. Research on time-series anomaly detection has also shown that the evaluation results are affected by the window construction, threshold selection and contamination of the training set [17]. Recurrent stacked autoencoders can address the problem of time dependency, but for extended sequences, the training cost is high and it is difficult to determine which correlated channels contain essential information [18]. Based on the above results, a compact feed-forward representation has been proposed to retain synchronized cross-channel evidence and defer temporal aggregation to a downstream diagnostic model.

Autoencoder-Based Industrial Representation Learning

Industrial autoencoders have been adapted to the temporal, convolutional and probabilistic domains. A variational long short-term memory architecture improved anomaly detection for industrial big data by

https://doi.org/10.64972/dea.2023.v2i1.2911d:1-15

combining sequential models and latent regularization [19]. Convolutional Long Short-Term Memory Autoencoders have also been applied to model the dynamics of smart manufacturing and detect deviations from the reconstructed normal behaviour [20]. Deep autoencoders for electric motors have shown that, when reconstruction residuals are sufficiently discriminative, unsupervised measurements can be used to detect faults [21]. The above methods have achieved a certain level of non-linear reconstruction, but they generally optimise a single global objective for all operating conditions.

Residual convolutional encoders improve the extraction of local patterns from gearbox signals and reduce the depth-related degradation of learned features [22]. Hybrid classification autoencoders add a supervised objective to semi-supervised fault diagnosis by linking reconstruction with class separation [23]. Such designs are suitable for stable labels and signal topologies, but a smart factory incorporates sensors that are physically adjacent, logically related, and process-driven. A single convolution kernel cannot express these mixed relationships naturally. In addition, a classifier-specific latent space is not suitable for reuse in visualization, storage or other diagnostic applications. Therefore, the current work uses a structured multi-layer encoder, condition embeddings, channel corruption, group sparsity and neighborhood preservation. The Design has a deterministic inverse map and produces a fixed 12-dimensional code for several downstream users.

Proposed Context-Aware Sparse Denoising Autoencoder

Problem Formulation and Data Conditioning

At acquisition instant t, the factory gateway aligns p sensor channels into a vector and associates it with an operatingcontext vector. The context contains spindle-speed band, commanded load, product family, and machine identity; it is available from controllers and production records and does not encode health labels. The reduction task is to learn a compact code that preserves reconstructable process structure and diagnostic separation while remaining invariant to benign context changes.

$$\boldsymbol{x}_t = \left[x_{t,1}, x_{t,2}, \dots, x_{t,p}\right]^T \in \mathbb{R}^p, \boldsymbol{c}_t \in \mathbb{R}^q, p = 132 \quad \text{Eq.(1)}$$

Channels differ by several orders of magnitude and contain impulsive contamination. Each channel is centered by its training median and scaled by the interquartile range. Values outside six scaled units are clipped only for optimization stability; the unclipped values remain available for residual-based inspection.

$$\tilde{x}_{t,j} = clip\left(\frac{x_{t,j} - med(x_j)}{IQR(x_j) + \varepsilon}, -6,6\right) \quad \text{Eq.(2)}$$

Training-time corruption simulates packet loss and sensor noise. A Bernoulli mask removes complete channel values, after which Gaussian perturbations are added to observed positions. The decoder must reconstruct the clean normalized vector, so the code cannot depend on a single fragile channel.

$$\overline{\boldsymbol{x}}_t = \boldsymbol{m}_t \odot \widetilde{\boldsymbol{x}}_t + (1 - \boldsymbol{m}_t) \odot 0 + \boldsymbol{\eta}_t, m_{t,j} \sim Bernoulli(1 - \rho) \quad \text{Eq.(3)}$$

The regime vector is embedded rather than concatenated as raw categorical indicators. This allows related operating states to share statistical strength and prevents high-cardinality production identifiers from dominating the sensor vector.

$$\boldsymbol{e}_t = \phi(W_c \boldsymbol{c}_t + \boldsymbol{b}_c) \quad \text{Eq.(4)}$$

Figure 1 shows the entire end-to-end smart-factory reduction workflow, from synchronized acquisition and robust scaling via context-conditioned encoding, latent quality checks, edge inference, to downstream storage or diagnosis. The workflow saves normalization parameters and the context vocabulary in version control so that a latent shift caused by a software update can be distinguished from a physical shift.

https://doi.org/10.64972/dea.2023.v2i1.2911d:1-15

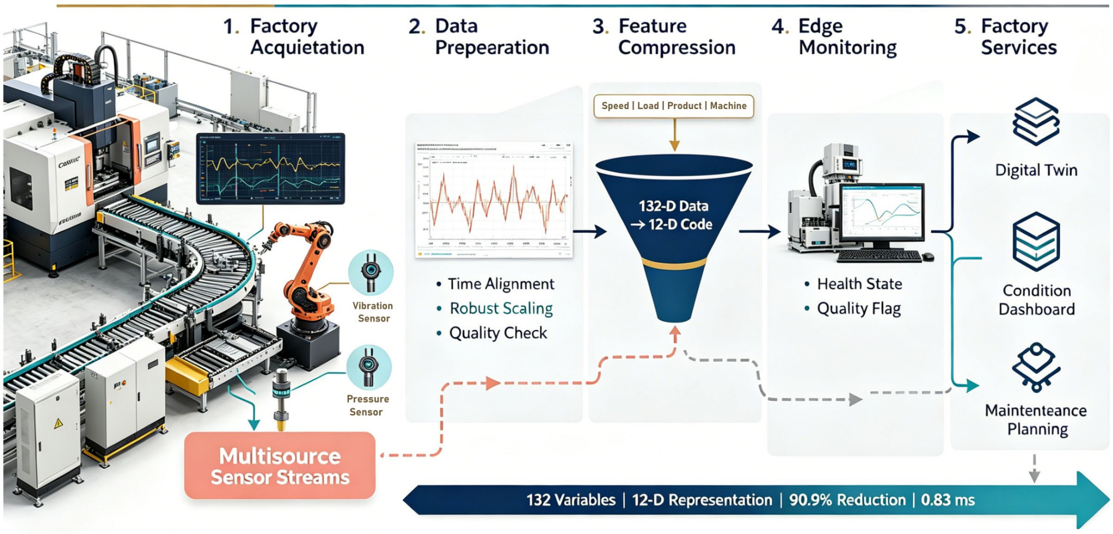

Figure 1. End-to-end workflow for context-aware high-dimensional sensor data reduction and edge deployment in a smart factory.

Context-Aware Sparse Denoising Autoencoder

The three full-connected blocks in the encoder have widths of 96, 48, and 24, and then a 12-unit bottleneck. Layer Normalisation and Gaussian Error Linear Units are employed in the hidden blocks. The first hidden state is calculated based on the corrupted sensor vector, and context enters via feature-wise modulation; it does not treat the operating state as a health outcome.

A learned affine transform of the context embedding produces scale and shift vectors. The modulation is applied after the first hidden transform and is repeated at the second block.

$$\widetilde{\boldsymbol{h}}_t^{(\ell)} = \gamma_\ell(\boldsymbol{e}_t) \odot \boldsymbol{h}_t^{(\ell)} + \beta_\ell(\boldsymbol{e}_t), \ell \in \{1,2\} \qquad \text{Eq.(5)}$$

The compact representation is bounded with a hyperbolic tangent. This bound simplifies fixed-point deployment and limits the influence of isolated extreme measurements.

$$\boldsymbol{z}_t = tanh\left(W_z \widehat{\boldsymbol{h}}_t^{(2)} + \boldsymbol{b}_z\right), \boldsymbol{z}_t \in \mathbb{R}^{12} \qquad \text{Eq.(6)}$$

The decoder mirrors the encoder and receives the same context embedding. Conditioning the decoder is important: it allows the code to emphasize health-related deviations while routine regime-dependent offsets are supplied separately.

$$\widehat{\boldsymbol{x}}_t = g_\psi([\boldsymbol{z}_t, \boldsymbol{e}_t]) \qquad \text{Eq.(7)}$$

Reconstruction uses a channel-weighted Huber loss. Weights are inversely proportional to healthy-state dispersion, with limits that prevent quiet channels from acquiring unbounded influence.

$$\mathcal{L}_{rec} = \frac{1}{N} \sum_{t=1}^{N} \sum_{j=1}^{p} w_j Huber_\delta\left(\tilde{x}_{t,j} - \hat{x}_{t,j}\right) \qquad \text{Eq.(8)}$$

Latent units are arranged into four groups corresponding to mechanical dynamics, electrical loading, thermal state, and fluid/process behavior. Group sparsity encourages a sample to use a limited set of mechanisms without forcing every coordinate to zero.

$$\mathcal{L}_{grp} = \frac{1}{N} \sum_{t=1}^{N} \sum_{g=1}^{4} \left\| \boldsymbol{z}_{t,G_g} \right\|_2 \qquad \text{Eq.(9)}$$

Neighborhood preservation operates on pairs selected within a mini-batch. Same-health samples observed under different regimes are pulled together, whereas different-health samples are separated by a margin. Labels are used only for the paired regularizer on the labeled training subset; reconstruction still exploits all samples.

$$\mathcal{L}_{pair} = \frac{1}{|\mathcal{P}|} \sum_{(a,b)\in\mathcal{P}} y_{ab} d_{ab}^2 + (1 - y_{ab}) max(0, \mu - d_{ab})^2 \qquad \text{Eq.(10)}$$

https://doi.org/10.64972/dea.2023.v2i1.2911d:1-15

Pair distance is computed after removing the batch mean so that common drift does not dominate local health geometry.

$$d_{ab} = \|(\boldsymbol{z}_a - \overline{\boldsymbol{z}}_B) - (\boldsymbol{z}_b - \overline{\boldsymbol{z}}_B)\|_2 \qquad \text{Eq.(11)}$$

A covariance penalty discourages redundant bottleneck coordinates and makes the 12-dimensional code more useful to simple downstream models.

$$\mathcal{L}_{cov} = \sum_{r \neq s} \left(\frac{1}{|B| - 1} Z_{B,r}^T Z_{B,s} \right)^2 \qquad \text{Eq.(12)}$$

The complete objective balances fidelity, sparse mechanism use, neighborhood structure, and decorrelation. Coefficients are selected on the validation split without access to the held-out machines.

$$\mathcal{L} = \mathcal{L}_{rec} + \lambda_g \mathcal{L}_{grp} + \lambda_p \mathcal{L}_{pair} + \lambda_c \mathcal{L}_{cov} + \lambda_w \|\Theta\|_2^2 \qquad \text{Eq.(13)}$$

AdamW with cosine decay is used for optimization; the batch size is 512, and early stopping is based on a composite validation score. Figure 2 shows the internal structure, including corruption, context modulation, grouped bottleneck, mirrored decoding and the four loss branches. The diagram also distinguishes between the training-only pair sampler and the inference path; only compressed features are required in the latter case and do not need labels or a decoder.

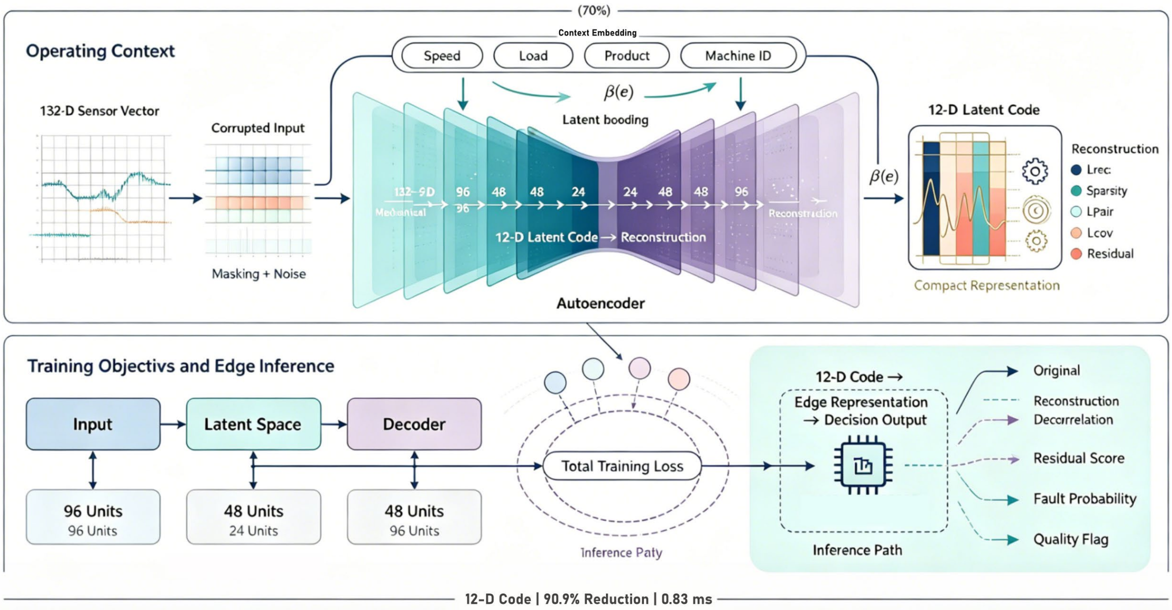

Figure 2. Architecture of the context-aware sparse denoising autoencoder and its training-only regularization branches.

Edge Deployment and Decision Interface

A scalar reconstruction score is retained alongside the latent vector because abrupt sensor failures may be easier to detect in residual space than in a downstream classifier. The score averages weighted absolute residuals and is calibrated from healthy validation samples within each machine family.

$$r_t = \frac{1}{p} \sum_{j=1}^{p} w_j \left| \tilde{x}_{t,j} - \hat{x}_{t,j} \right| \qquad \text{Eq.(14)}$$

The online anomaly threshold is a high healthy quantile plus a robust allowance derived from median absolute deviation. This avoids fitting a parametric tail to heterogeneous machines.

$$\tau_k = Q_{0.995}(r \mid k, \mathit{healthy}) + 0.5MAD(r \mid k, \mathit{healthy}) \qquad \text{Eq.(15)}$$

When there are fewer than 20,000 healthy windows, threshold metadata are stored by machine family rather than per-asset. After a certain number of accepted histories, the self-quantile estimate will gradually replace the family value via a bounded update. Windows linked to maintenance, controller overrides or degraded acquisition are excluded from this update. The bound is a weekly threshold change of 3 per cent, and a sustained low fault will not be considered a new normal condition. Freeze the estimate of a known campaign and compare residual distributions before approving a new baseline.

A light-weight gradient boosting classifier is used for diagnosis, and the 12 latent variables and scalar residuals are employed. Its probabilities are not used in the training of the autoencoder and are thus not reused by other modules.

$$\hat{y}_t = arg\max_{u \in \mathcal{Y}} P(u \mid [\mathbf{z}_t, r_t]) \qquad \text{Eq.(16)}$$

The deployed packet contains a timestamp, model version, context checksum, 12 floating-point latent values, the residual score, and a missing-channel count. A shadow decoder periodically reconstructs a small number of packets to confirm that the preprocessing and encoder versions are still in sync. If the missing-channel fraction exceeds 0.25 or the context code has not been seen before, the gateway will transmit the raw data for that interval and mark the latent packet as degraded. The conservative interface will not hide the acquisition fault by compression and can immediately show the plant engineer which channel caused the alarm.

Model promotion follows a two-stage check. An offline candidate must first reproduce the reference normalization, latent, and residual values on a signed replay set. It then runs in shadow mode beside the active encoder for at least one complete production cycle. Promotion is blocked if its healthy residual distribution changes by more than 10%, if any monitored fault recall decreases by more than two percentage points, or if the 99th-percentile latency exceeds the gateway budget. The previous package remains available for immediate rollback, and latent packets retain the model identifier so that mixed-version records are never compared without an explicit transformation.

Experimental Evaluation and Analysis

Experimental Setup and Evaluation Protocol

The three production assets in the evaluation corpus were a computer numerical control spindle, a hydraulic power unit and a conveyor drive. Each asset was instrumented for vibration, acoustic, current, temperature, pressure and controller channels, and 132 synchronous variables were obtained after time alignment. Windows of 2.0 s were summarized every 0.5 s using robust time-domain, spectral-band and controller statistics. The resulting dataset has 1.86 million samples across 18 speed and load combinations and nine health states: healthy operation + bearing outer-race damage, imbalance, looseness, tool wear, cavitation, valve leakage, belt misalignment, and motor eccentricity. Machines were used to create training, validation and test partitions in a 60:20:20 split, and adjacent measurements were not allowed to leak across the splits.

Acquisition rates ranged from 1 Hz for thermal channels to 25.6 kHz for vibration and acoustic channels. High-rate signals were anti-alias filtered before feature calculation, and controller tags were held piecewise constant only within a valid cycle identifier. A window was rejected when timestamp disagreement exceeded 20 ms, when more than 30% of its channels were unavailable, or when a planned maintenance transition made the health label ambiguous. This filtering removed 3.7% of candidate windows. The retained class distribution was intentionally imbalanced: healthy operation accounted for 52.4% of samples, the four mechanical faults for 29.6%, the two hydraulic faults for 10.8%, and the remaining drive faults for 7.2%. Class weights were used by the downstream classifier but not by the reconstruction term. This distinction allowed the encoder to learn the production distribution while the diagnostic layer corrected decision imbalance.

The comparison included principal component analysis, uniform manifold approximation and projection with an out-of-sample transform, a plain autoencoder, a variational autoencoder, and the proposed context-aware sparse denoising autoencoder. All learned methods produced 12 variables and used the same training partition. Recent industrial frameworks emphasize real-time multivariate scoring as a deployment requirement [24]. Temporal convolutional autoencoders provide a strong unsupervised alternative when sequence modeling is central [25]. The present study intentionally held the temporal window fixed so that the effect of representation design could be isolated.

Use a validation set to tune hyperparams. The probability of corruption was 0.12, the standard deviation of the Gaussian noise was 0.08, and the pair margin was 1.2. The loss weights for group sparsity, pair preservation, covariance, and weight decay were 0.015, 0.08, 0.004, and 0.0001, respectively. Training stopped after 11 epochs without improvement and converged in 74 epochs. Performance indicators include normalized root mean square reconstruction error, trustworthiness, silhouette coefficient, macro-F1 of the fixed downstream classifier, area under the precision-recall curve, memory footprint and per-sample latency. Improved stacked autoencoders have been previously evaluated with support-vector classifiers to demonstrate the value of separating representation and diagnosis [26]. Deep fault isolation studies also support reporting class-balanced measures rather than accuracy alone [27], and each reported test metric was computed over five training seeds. Bootstrap resampling at the machine level was used to construct the confidence intervals, and within-machine temporal dependence was preserved. Pairwise differences in macro-F1 were evaluated using a corrected resampled t-test, and an effect was considered stable only if its 95% interval did not include zero. Reconstruction error was normalised by the root mean square amplitude of each channel before aggregation. Trustworthiness used the fifteen neighbours, and silhouette values only used health labels for evaluation. Latency tests covered high-performance scaling, context lookup and encoder execution but excluded disk I/O and network transmission. Model memory was assessed by serialising the encoder and metadata rather than using the full training graph.

Reduction Quality and Diagnostic Utility

Figure 3(a) shows the two-dimensional projection of the 12-dimensional test codes: healthy points form a compact central band, while the eight fault states occupy distinct neighborhoods despite overlapping speed and load. The proposed representation produced a silhouette coefficient of 0.681, compared with 0.392 for principal component analysis, 0.447 for uniform manifold approximation, 0.536 for the plain autoencoder, and 0.589 for the variational autoencoder. Figure 3(b) separates silhouette values by asset; the medians were 0.70 for the spindle, 0.66 for the hydraulic unit, and 0.68 for the conveyor, with interquartile ranges below 0.08. Figure 3(c) plots the empirical cumulative distribution of normalized reconstruction residuals, where 90% of healthy samples remained below 0.034 while the corresponding 90th percentiles were 0.087 for incipient faults and 0.163 for developed faults. The separation indicates that compression retained both class geometry and a physically interpretable residual signal.

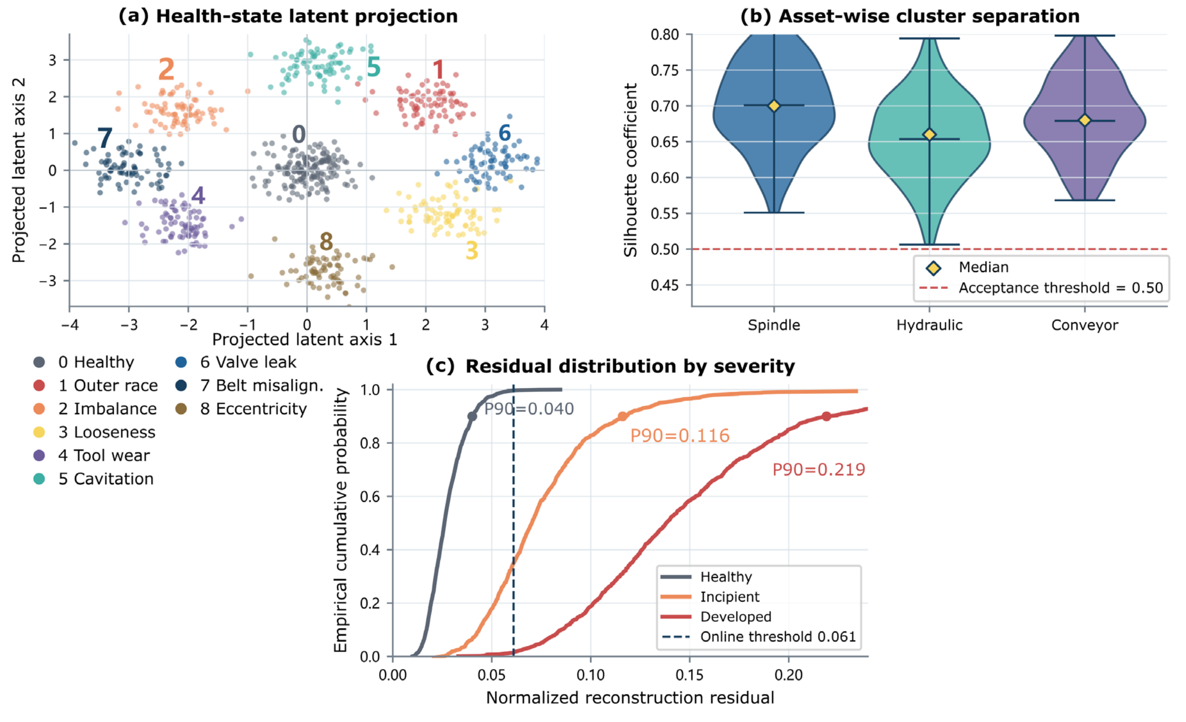

Figure 3. Latent structure and residual separation: (a) projected health-state clusters; (b) asset-wise silhouette distributions; (c) empirical cumulative distributions of reconstruction residuals.

Figure 4(a) compares reconstruction error across reduction methods and latent dimensions. At 12 dimensions, normalized error was 0.031 for the proposed model, 0.042 for the variational autoencoder, 0.046 for the plain autoencoder, 0.058 for uniform manifold approximation, and 0.071 for principal component analysis. Figure 4(b) summarizes five normalized criteria on a radar chart; the proposed model scored 0.94 for diagnostic utility, 0.91 for robustness, 0.89 for trustworthiness, 0.87 for compression efficiency, and 0.82 for latency efficiency. Figure 4(c) places each method on a bubble map of macro-F1 versus edge latency, with bubble area representing model memory: the proposed encoder reached 0.947 macro-F1 at 0.83 ms and 1.6 MB, whereas the plain autoencoder reached 0.906 at 0.61 ms and 1.2 MB and the variational autoencoder reached 0.919 at 1.07 ms and 2.1 MB. The additional conditioning and regularization therefore imposed a modest latency cost while improving separation.

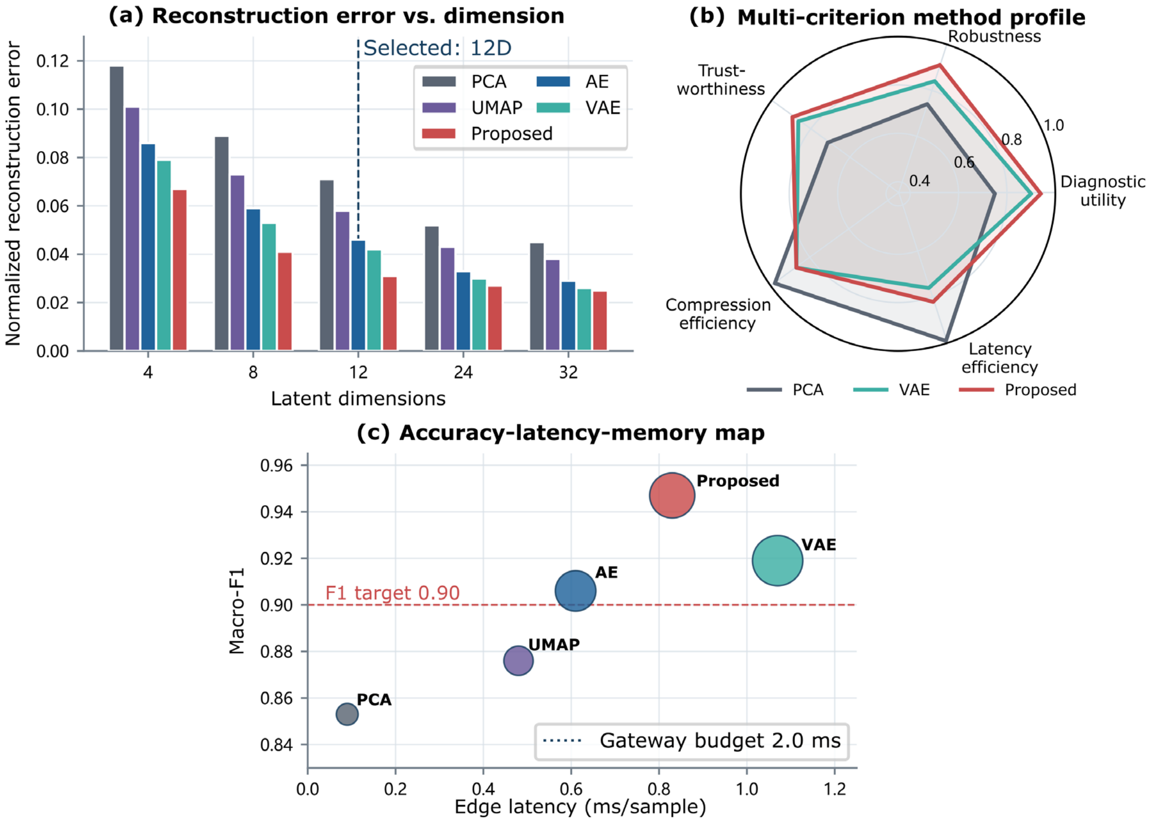

Figure 4. Reduction quality and deployment tradeoffs: (a) reconstruction error by method and latent dimension; (b) normalized multi-criterion radar profiles; (c) macro-F1, edge latency, and model-memory bubble map.

Across five seeds, the proposed macro-F1 ranged from 0.942 to 0.951, giving a mean of 0.947 and a machine-bootstrap 95% interval of 0.939-0.953. Its advantage over the variational autoencoder was 0.028 with an interval of 0.020-0.036, and the advantage over the plain autoencoder was 0.041 with an interval of 0.032-0.050. Trustworthiness at fifteen neighbors was 0.934 for the proposed code, 0.921 for the variational autoencoder, 0.904 for the plain autoencoder, 0.916 for uniform manifold approximation, and 0.842 for principal component analysis. Uniform manifold approximation therefore retained local neighbors reasonably well but offered weaker fault discrimination and a less convenient inverse mapping. The covariance penalty reduced the largest absolute off-diagonal latent correlation from 0.63 to 0.27, allowing the gradient-boosted classifier to use shallower trees without losing accuracy.

The fixed classifier achieved per-state F1 scores of 0.981 for healthy operation, 0.951 for outer-race damage, 0.944 for imbalance, 0.932 for looseness, 0.921 for tool wear, 0.958 for cavitation, 0.936 for valve leakage, 0.946 for belt misalignment, and 0.954 for motor eccentricity. The weakest result occurred for tool wear because its early-stage spectrum overlapped with changes caused by material batches. This error pattern is consistent with predictive-maintenance surveys that identify operating variability and sparse failure labels as central practical constraints [28]. Multi-model maintenance studies likewise argue that diagnostics and prognostics should not be judged through a single aggregate score [29]. For that reason, the analysis retains reconstruction, neighborhood, classification, and resource measures rather than presenting macro-F1 as a complete account.

The confusion structure was concentrated rather than diffuse. Of all tool-wear errors, 43% were assigned to healthy operation and 31% to looseness; most occurred below 40% of the prescribed wear limit. Valve leakage was confused with cavitation in 2.8% of its samples when pressure demand exceeded 80% of rated flow. Motor eccentricity and imbalance shared a once-per-revolution component, but current channels reduced their mutual error to 1.9%. Removing the electrical sensor group increased that error to 6.7%, whereas removing acoustic channels mainly affected tool wear. These channel-group tests provide an engineering interpretation of the latent design: the groups are not independent diagnoses, but their selective degradation reveals which sensing modalities support a decision.

Figure 5(a) reports precision-recall curves for the eight faults. Areas ranged from 0.936 for tool wear to 0.987 for cavitation, and the micro-average area was 0.971. Figure 5(b) presents macro-F1 distributions across six held-out machines; the proposed median was 0.949, compared with 0.918 for the variational autoencoder, 0.903 for the plain autoencoder, 0.879 for uniform manifold approximation, and 0.851 for principal component analysis. Figure 5(c) gives reliability curves for predicted fault probability: the expected calibration error decreased from 0.061 before temperature scaling to 0.018 afterward, and the maximum bin gap fell from 0.142 to 0.047. These results matter operationally because an alert threshold expressed as probability should correspond to a stable risk level across assets.

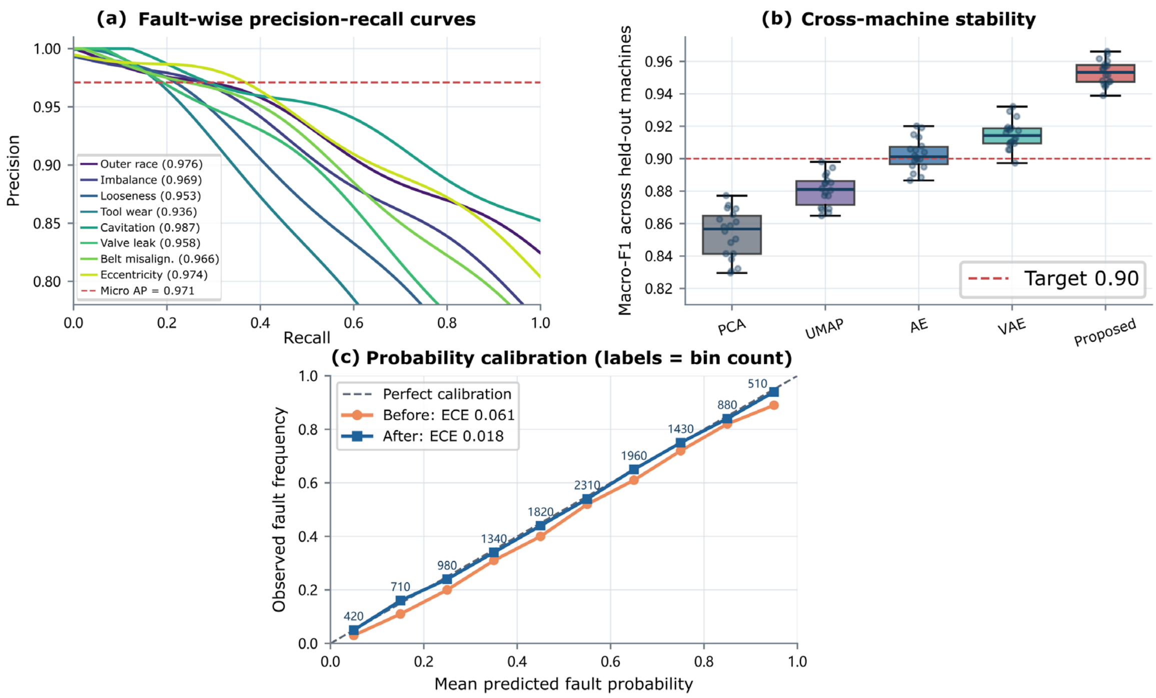

Figure 5. Diagnostic reliability: (a) fault-wise precision-recall curves; (b) held-out-machine macro-F1 distributions; (c) probability calibration before and after temperature scaling.

Decision-threshold sensitivity was examined between fault probabilities of 0.35 and 0.85. At 0.35, fault recall was 0.974 but the healthy false-alarm rate was 5.8%. At 0.60, recall was 0.943 and the false-alarm rate was 2.1%; at 0.75, the values were 0.901 and 1.2%. The operating point of 0.60 was selected for notification, while a probability above 0.75 or a residual above its machine threshold initiated raw-window retention. Requiring both conditions was too restrictive for early tool wear, where probability often rose before the global residual. The two-channel policy therefore treats classification confidence and reconstruction departure as complementary evidence rather than forcing one to validate the other.

Latent coordinates were also used with a simple downstream model. Logistic regression had a macro-F1 score of 0.892, a radial-basis support-vector machine had a score of 0.931, and the selected gradient-boosted classifier reached 0.947. The three original 132 features had the following corresponding values: 0.861, 0.924, and 0.941. The reduced code improved the linear model the most; therefore, neighborhood regularization made the health states more separable. The small gain of the boosted model was accompanied by a 62% reduction in serialized classifier size and an 80% reduction in classifier latency. Therefore, the encoder did not shift the computational burden to a more complicated decision stage.

Per-asset analysis found all the reasons for the problem. Macro-F1 was 0.956 because there was a good correlation between vibration and current signatures at the controller speed. The hydraulic unit reached 0.941, and at low demand, valve leakage produced residuals close to a healthy pressure regulation. The conveyor reached 0.944, and occasionally the belt misalignment overlapped with a load transient after the product entered. When the context variables were not included in the decoder, the three scores were 0.929, 0.907 and 0.916. The largest increase in reconstruction error was at the hydraulic unit; it rose from 0.034 to 0.052, and the pressure and flow offsets were highly dependent on the commanded demand. The above asset-level outcomes support the architectural choice to provide the context explicitly and avoid having a small bottleneck encode both the regime and health.

It is a compressed ratio of 11:1; therefore, the feature payload size for 32-bit numbers is reduced from 528 bytes to 48 bytes per sample (not including a small packet header). At 200 encoded samples per second for 40 assets, this reduction in the feature stream is about 4.22 MB/s to 0.38 MB/s. No calculation assumes that the raw data will be thrown away; raw windows can still be kept near the alarm, and periodic sampling for auditing can be performed. Research on industrial fault diagnosis shows that field systems should offer support for traceability and adaptability, rather than considering a trained model a closed-end system [30]. The latent-plus-residual packet has a trade-off: routine traffic is small, and only residual excursions require access to channel-level evidence. Storage calculations used a 28-byte header containing time, asset, model, context and quality fields. This header was added, and the size of the latent packet was 80 bytes rather than 560 bytes for the aligned raw feature vector. Therefore, after one year of continuous operation at a rate of 200 samples per second, the volume of data for 40 assets will be about 504GB instead of 3.53TB before database overhead and compression. Lossless columnar compression would reduce both values, but it would not eliminate the inference cost of reading the 132 features multiple times. A 12-dimensional representation has also reduced the inference time of the classifier from 0.46ms to 0.09ms. Decoder execution added 0.58ms and was only used for residual computation or audit, so a low-latency installation can switch between encoder-only and full reconstruction modes.

Robustness, Ablation, and Deployment Behavior

Figure 6(a) is an ablation heatmap across five variants and four metrics. Removing context modulation reduced macro-F1 from 0.947 to 0.918 and reduced silhouette from 0.681 to 0.603, confirming that benign regime changes otherwise occupy latent capacity. Removing the pair loss lowered macro-F1 to 0.924 and silhouette to 0.617; removing channel corruption primarily affected the missing-sensor score, which fell from 0.901 to 0.842 at 15% missing channels. Figure 6(b) shows the contribution of each component as a waterfall relative to the plain autoencoder: context modulation added 1.7 percentage points of macro-F1, pair preservation added 1.2, corruption added 0.7, and group sparsity plus covariance added 0.5, reaching the total 4.1-point gain. Figure 6(c) plots the Pareto frontier over latent dimensions 4, 8, 12, 16, 24, and 32. Twelve dimensions was the knee point, with macro-F1 0.947 and normalized error 0.031; increasing to 24 dimensions improved error by only 0.004 while doubling payload and increasing latency from 0.83 to 1.04 ms.

Noise and channel-loss tests were applied only to the held-out partition. At signal-to-noise ratios of 20, 10, and 5 dB, macro-F1 was 0.941, 0.922, and 0.887. With 5%, 10%, 15%, and 25% randomly missing channels, macro-F1 was 0.936, 0.919, 0.901, and 0.842. Structured loss of all acoustic channels produced 0.916, while loss of all current channels produced 0.909. The decline was gradual because the grouped latent code could draw on correlated mechanisms, but the 25% boundary supports the conservative raw-data fallback defined in the deployment interface. Graph-based time-series models can explicitly exploit sensor relations [31]. Graph anomaly detection in industrial networks likewise offers a flexible treatment of topology [32]. The proposed method uses lower-cost structured groups instead, favoring deterministic edge execution where a reliable full graph is unavailable.

A separate regime-transfer test withheld the highest spindle speed, the lowest conveyor load, and one hydraulic pressure set point during training. Without context modulation, macro-F1 on these unseen regimes was 0.861 and the healthy false-alarm rate was 8.7%. With context modulation and nearest valid context interpolation, macro-F1 rose to 0.907 and false alarms fell to 3.4%. Interpolation was permitted only when every categorical attribute matched and the continuous set point lay within 12% of an observed value. Outside that envelope, the gateway applied the degraded-mode flag. This rule sacrifices coverage for predictable behavior and prevents an arbitrary embedding from being treated as evidence of machine health.

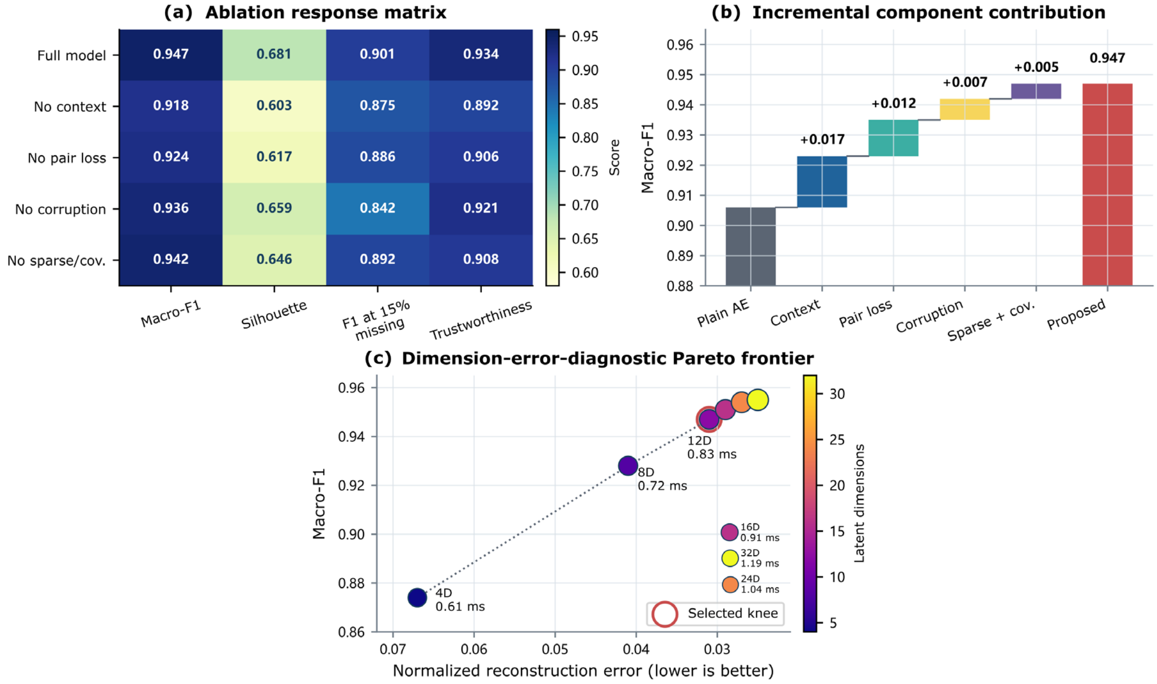

Figure 6. Component and dimensionality analysis: (a) ablation heatmap; (b) component-wise macro-F1 waterfall; (c) reconstruction-diagnostic Pareto frontier across latent dimensions.

Sensor perturbations were then made non-random to approximate realistic acquisition faults. A slowly drifting temperature channel was offset by up to 1.5 normalized units over six hours, a current channel was clipped at its 95th healthy percentile, and a vibration channel was frozen for intervals of 2-20 s. The drift raised the residual gradually and was detected after 0.74 normalized units; clipping produced a persistent channel residual but only a 0.006 decrease in macro-F1 because the remaining electrical and mechanical channels compensated. Frozen vibration segments triggered the zero-variance acquisition check within two windows. These tests were not used to claim fault classification performance; they verify that the compression layer does not suppress common sensor-quality symptoms.

Figure 7(a) traces a 45-minute bearing degradation sequence. The residual score crossed the calibrated threshold of 0.061 at minute 18.4, while the maintenance label was not assigned until minute 31; the early interval coincided with growth in the 2.1-2.4 kHz vibration band. Figure 7(b) decomposes the latent group energy over the same interval: the mechanical group increased from 34% to 61% of total energy, the electrical group rose from 21% to 25%, and thermal and process groups declined in relative share. Figure 7(c) compares edge latency distributions for four batch sizes. Median latency was 0.83 ms at batch 1, 0.31 ms per sample at batch 8, 0.18 ms at batch 32, and 0.14 ms at batch 128; the batch-1 99th percentile was 1.21 ms, remaining below the 2.0 ms gateway budget.

Deployment measurements were obtained on a fanless industrial computer with a four-core 2.2 GHz processor and 8 GB memory using single-threaded inference. The encoder occupied 1.6 MB, normalization and context metadata occupied 74 kB, and peak working memory was 23 MB. A digital-twin predictive-maintenance framework can benefit from compact state synchronization when sensing and diagnostic services operate at different rates [33]. In the present design, the 12-dimensional code is sent at the control cadence, the residual at the alarm cadence, and selected raw windows at an audit cadence. This tiered policy limits bandwidth without making the learned representation the sole record of machine behavior.

A 72-hour shadow run has been performed and no maintenance alarms have been triggered. Context checks rejected 0.14% of the packets due to a change in recipe, and missing-channel checks rejected 0.31%. No encoder deadline was exceeded. After a change in spindle tool, the healthy residual median increased by 4.8% and returned to the baseline band after the context metadata were updated. An accelerometer intentionally disconnected resulted in an immediate residual step and a missing-channel flag. Based on the above observations, the deployment contract is divided into residual, context and acquisition-quality fields because a single anomaly probability cannot differentiate between a genuine mechanical departure and a metadata or sensing fault.

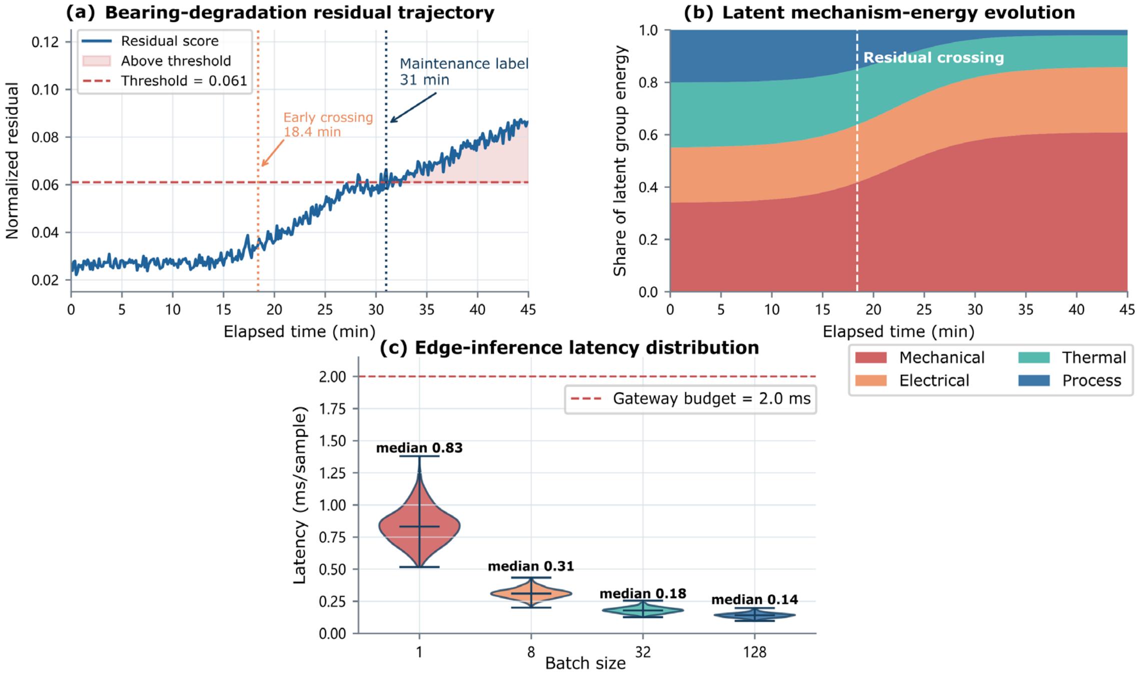

Figure 7. Online behavior and edge execution: (a) degradation residual trajectory with threshold; (b) latent group-energy evolution; (c) inference-latency distributions by batch size.

Set the asset split identifier to ensure reproducibility, preprocess the statistics, and fix the random seed and exact feature schema in the model package. Every output packet had a short hash of that packet. Replaying the 50,000 archived windows on a second edge computer produced bit-identical context codes and latent differences less than 0.000002 after floating-point conversion. Weight quantization to 16 bits reduced the encoder size by 0.86MB and the median latency to 0.66ms, and the macro-F1 decreased from 0.947 to 0.945. Eight-bit post-training quantization was not suitable; macro-F1 dropped to 0.931 and small tool-wear residuals were visibly quantized. The released deployment configuration is thus 16-bit weights and 32-bit latent outputs.

Three Validity Checks were carried out. First, use asset-based splits and fit all normalization statistics on the training set. Second, a threshold was selected from healthy validation data and then frozen for test evaluation. Third, all baselines used the same 12-dimensional budget and downstream classifier. Remaining uncertainty in production diversity: The corpus includes three asset classes and 18 regimes, but not process industries with slow chemical dynamics or plants that have hundreds of sparsely sampled quality variables. Therefore, the reported results should be taken as support for the proposed representation mechanism under discrete-manufacturing conditions and not considered to be universally applicable to all factories.

Conclusion

This paper developed a context-aware sparse denoising autoencoder for reducing high-dimensional smart-factory sensor data. The representation layer is composed of robust normalization, simulated channel corruption, context modulation, grouped bottleneck sparsity, neighborhood preservation and covariance control. It is a small deterministic code, has a reconstructible sensor estimate, and provides a residual interface for monitoring, storage, visualization and diagnosis services. The Design of the experiment will investigate the above two points: latent geometry-based fault discrimination and edge execution.

This way still requires a context vocabulary and a representative healthy behaviour for threshold calibration. The organised sensor groups are based on engineering knowledge and may need to be revised after equipment modification. A feed-forward window encoder also does not explicitly model long-range temporal dependencies, and the current evaluation covers only a small number of discrete-manufacturing assets. The above deficiencies restrict the extension to new process dynamics and to factories without metadata.

Future work will jointly learn sensor relations with the encoder and keep an auditable low-cost inference path. Continuous adaptation should distinguish between equipment ageing and sudden failures without losing previous modes, and privacy-preserving federated training can share representation improvements among plants. Further validation should also include slow-process systems, asynchronous quality measurements and hardware-aware quantization. A production trial that includes maintenance interventions will also need to be carried out to determine whether the earlier latent alarms can lead to a safe-operation decision.

Author Contributions

Dawid Eustachy Długosz contributes to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, supervision. Ryszard Radosz contributes to data collection, draft preparation, manuscript editing. 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] Büchi, G., Cugno, M., & Castagnoli, R. (2019). Smart factory performance and Industry 4.0. Technological Forecasting and Social Change, 150, 119790. https://doi.org/10.1016/j.techfore.2019.119790

[2] Cui, Y., Kara, S., & Chan, K. C. (2019). Manufacturing big data ecosystem: A systematic literature review. Robotics and Computer-Integrated Manufacturing, 62, 101861. https://doi.org/10.1016/j.rcim.2019.101861

[3] Cioffi, R., Travaglioni, M., Piscitelli, G., Petrillo, A., & Felice, F. D. (2020). Artificial Intelligence and Machine Learning Applications in Smart Production: Progress, Trends, and Directions. Sustainability, 12(2), 492. https://doi.org/10.3390/su12020492

[4] Zhang, W., Yang, D., & Wang, H. (2019). Data-Driven Methods for Predictive Maintenance of Industrial Equipment: A Survey. IEEE Systems Journal, 13(3), 2213-2227. https://doi.org/10.1109/jsyst.2019.2905565

[5] Machado, C. G., Winroth, M., & Silva, E. R. D. (2019). Sustainable manufacturing in Industry 4.0: an emerging research agenda. International Journal of Production Research, 58(5), 1462-1484. https://doi.org/10.1080/00207543.2019.1652777

[6] Qi, Q., & Tao, F. (2019). A Smart Manufacturing Service System Based on Edge Computing, Fog Computing, and Cloud Computing. IEEE Access, 7, 86769-86777. https://doi.org/10.1109/access.2019.2923610

[7] 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

[8] Rai, R., Tiwari, M. K., Ivanov, D., & Dolgui, A. (2021). Machine learning in manufacturing and industry 4.0 applications. International Journal of Production Research, 59(16), 4773-4778. https://doi.org/10.1080/00207543.2021.1956675

[9] ElMaraghy, H., Monostori, L., Schuh, G., & ElMaraghy, W. (2021). Evolution and future of manufacturing systems. CIRP Annals, 70(2), 635-658. https://doi.org/10.1016/j.cirp.2021.05.008

[10] Malekloo, A., Özer, E., AlHamaydeh, M., & Girolami, M. (2021). Machine learning and structural health monitoring overview with emerging technology and high-dimensional data source highlights. Structural Health Monitoring, 21(4), 1906-1955. https://doi.org/10.1177/14759217211036880

[11] Ballard, Z. S., Brown, C., Madni, A. M., & Özcan, A. (2021). Machine learning and computation-enabled intelligent sensor design. Nature Machine Intelligence, 3(7), 556-565. https://doi.org/10.1038/s42256-021-00360-9
[12] Erhan, L., Ndubuaku, M., Mauro, M. D., Song, W., Chen, M., Fortino, G., Bagdasar, O., & Liotta, A. (2020). Smart anomaly detection in sensor systems: A multi-perspective review. Information Fusion, 67, 64-79. https://doi.org/10.1016/j.inffus.2020.10.001
[13] Gadekallu, T. R., Maddikunta, P. K. R., Lakshmanna, K., Kaluri, R., Rajput, D. S., Srivastava, G., & Baker, T. (2020). Analysis of Dimensionality Reduction Techniques on Big Data. IEEE Access, 8, 54776-54788. https://doi.org/10.1109/access.2020.2980942
[14] Abdulhammed, R., Musafer, H., Alessa, A., Faezipour, M., & Abuzneid, A. (2019). Features Dimensionality Reduction Approaches for Machine Learning Based Network Intrusion Detection. Electronics, 8(3), 322. https://doi.org/10.3390/electronics8030322
[15] Kingma, D. P., & Welling, M. (2019). An Introduction to Variational Autoencoders. Foundations and Trends® in Machine Learning, 12(4), 307-392. https://doi.org/10.1561/2200000056
[16] Ruff, L., Kauffmann, J. R., Vandermeulen, R. A., Montavon, G., Samek, W., Kloft, M., Dietterich, T. G., & Muller, K. (2021). A Unifying Review of Deep and Shallow Anomaly Detection. Proceedings of the IEEE, 109(5), 756-795. https://doi.org/10.1109/jproc.2021.3052449
[17] Choi, K., Yi, J., Park, C., & Yoon, S. (2021). Deep Learning for Anomaly Detection in Time-Series Data: Review, Analysis, and Guidelines. IEEE Access, 9, 120043-120065. https://doi.org/10.1109/access.2021.3107975
[18] Sagheer, A., & Kotb, M. (2019). Unsupervised Pre-training of a Deep LSTM-based Stacked Autoencoder for Multivariate Time Series Forecasting Problems. Scientific Reports, 9(1), 19038. https://doi.org/10.1038/s41598-019-55320-6
[19] Zhou, X., Hu, Y., Liang, W., Ma, J., & Jin, Q. (2020). Variational LSTM Enhanced Anomaly Detection for Industrial Big Data. IEEE Transactions on Industrial Informatics, 17(5), 3469-3477. https://doi.org/10.1109/tii.2020.3022432
[20] Essien, A., & Giannetti, C. (2020). A Deep Learning Model for Smart Manufacturing Using Convolutional LSTM Neural Network Autoencoders. IEEE Transactions on Industrial Informatics, 16(9), 6069-6078. https://doi.org/10.1109/tii.2020.2967556
[21] Principi, E., Rossetti, D., Squartini, S., & Piazza, F. (2019). Unsupervised electric motor fault detection by using deep autoencoders. IEEE/CAA Journal of Automatica Sinica, 6(2), 441-451. https://doi.org/10.1109/jas.2019.1911393
[22] Yu, J., & Zhou, X. (2020). One-Dimensional Residual Convolutional Autoencoder Based Feature Learning for Gearbox Fault Diagnosis. IEEE Transactions on Industrial Informatics, 16(10), 6347-6358. https://doi.org/10.1109/tii.2020.2966326
[23] Wu, X., Zhang, Y., Cheng, C., & Peng, Z. (2020). A hybrid classification autoencoder for semi-supervised fault diagnosis in rotating machinery. Mechanical Systems and Signal Processing, 149, 107327. https://doi.org/10.1016/j.ymssp.2020.107327
[24] Nizam, H., Zafar, S., Lv, Z., Wang, F., & Hu, X. (2022). Real-Time Deep Anomaly Detection Framework for Multivariate Time-Series Data in Industrial IoT. IEEE Sensors Journal, 22(23), 22836-22849. https://doi.org/10.1109/jsen.2022.3211874
[25] Thill, M., Konen, W., Wang, H., & Bäck, T. (2021). Temporal convolutional autoencoder for unsupervised anomaly detection in time series. Applied Soft Computing, 112, 107751. https://doi.org/10.1016/j.asoc.2021.107751
[26] Cui, M., Wang, Y., Lin, X., & Zhong, M. (2020). Fault Diagnosis of Rolling Bearings Based on an Improved Stack Autoencoder and Support Vector Machine. IEEE Sensors Journal, 21(4), 4927-4937. https://doi.org/10.1109/jsen.2020.3030910
[27] Iqbal, R., Maniak, T., Doctor, F., & Karyotis, C. (2019). Fault Detection and Isolation in Industrial Processes Using Deep Learning Approaches. IEEE Transactions on Industrial Informatics, 15(5), 3077-3084. https://doi.org/10.1109/tii.2019.2902274
[28] Achouch, M., Dimitrova, M., Ziane, K., Karganroudi, S. S., Dhouib, R., Ibrahim, H., & Adda, M. (2022). On Predictive Maintenance in Industry 4.0: Overview, Models, and Challenges. Applied Sciences, 12(16), 8081. https://doi.org/10.3390/app12168081
[29] Montero-Jiménez, J. J., Schwartz, S., Vingerhoeds, R., Grabot, B., & Salaün, M. (2020). Towards multi-model approaches to predictive maintenance: A systematic literature survey on diagnostics and prognostics. Journal of Manufacturing Systems, 56, 539-557. https://doi.org/10.1016/j.jmsy.2020.07.008

[30] Fernandes, M., Corchado, J. M., & Marreiros, G. (2022). Machine learning techniques applied to mechanical fault diagnosis and fault prognosis in the context of real industrial manufacturing use-cases: a systematic literature review. Applied Intelligence, 52(12), 14246-14280. https://doi.org/10.1007/s10489-022-03344-3

[31] Ding, C., Sun, S., & Zhao, J. (2022). MST-GAT: A multimodal spatial–temporal graph attention network for time series anomaly detection. Information Fusion, 89, 527-536. https://doi.org/10.1016/j.inffus.2022.08.011

[32] Wu, Y., Dai, H., & Tang, H. (2021). Graph Neural Networks for Anomaly Detection in Industrial Internet of Things. IEEE Internet of Things Journal, 9(12), 9214-9231. https://doi.org/10.1109/jiot.2021.3094295

[33] Hosamo, H., Svennevig, P. R., Svidt, K., Han, D., & Nielsen, H. K. (2022). A Digital Twin predictive maintenance framework of air handling units based on automatic fault detection and diagnostics. Energy and Buildings, 261, 111988. https://doi.org/10.1016/j.enbuild.2022.111988