International Journal of Advances in Applied Computational Intelligence

♦ ISSN: 2833-5600 Vol: 08, No. 02, 2026 ♦ PP. 19–27

Article DOI: https://doi.org/10.54216/IJAACI.080203

FULL LENGTH ARTICLE

Metaheuristic-Optimized Conditional Diffusion Networks for Industrial Sensor Data Synthesis: An Applied Computational Intelligence Benchmark

Khaled Sh. Gaber[1] **Mahmoud Elshabrawy Mohamed**[1,*]

[1] Computer Science and Intelligent Systems Research Center, Blacksburg 24060, Virginia, USA

Emails: khsherif@jcsis.org · mshabrawy@jcsis.org

Received: March 26, 2026 Revised: May 31, 2026 Accepted: July 14, 2026 ⋆ Corresponding author

ABSTRACT

Synthetic industrial telemetry can alleviate scarcity and confidentiality constraints, but its value depends on more than distributional similarity. Generated sequences must support downstream engineering analysis, limit disclosure risk, and satisfy operational relations. This paper presents a reproducible three-axis benchmark and a differential-evolution calibration layer for conditional denoising diffusion. Six generators–bootstrap resampling, a shrinkage-Gaussian model, Fourier surrogates, a conditional variational autoencoder, a conditional diffusion model, and its calibrated counterpart–are evaluated on chronologically partitioned manufacturing telemetry. The protocol covers 18,000 generated 80-minute windows and combines train-synthetic/test-real alarm classification, marginal and temporal fidelity measures, record-proximity and membership-inference tests, and five engineering-rule audits. Differential evolution selects four post-generation controls using validation data only. Relative to the uncalibrated diffusion model, calibration reduces the aggregate physical-violation rate by 94.7% and the Wasserstein error by 29.2%, while increasing mean downstream ROC–AUC by 0.009. The improvement is accompanied by a 0.073 increase in membership-inference AUC and a small deterioration in autocorrelation error. Bootstrap resampling provides the strongest mean predictive utility but exactly reproduces 77.9% of its outputs; the conditional variational autoencoder attains the lowest Wasserstein error (0.046) with a 0.020% physical-violation rate. No generator dominates utility, privacy, and plausibility simultaneously. Synthetic industrial data should therefore be selected through deployment-specific acceptance regions rather than a single realism score.

Keywords: Synthetic industrial data ▪ Conditional diffusion ▪ Differential evolution ▪ Industrial sensor data ▪ Privacy auditing ▪ Physical plausibility ▪ Predictive maintenance ▪ Benchmark

1. INTRODUCTION

Industrial analytics often develops under a mismatch between abundant routine telemetry and limited fault, alarm, or transition records. Data sharing is further constrained by commercial confidentiality, plant cybersecurity, and the possibility that multivariate operating traces reveal production schedules or equipment behavior. Synthetic data can support predictive maintenance, quality monitoring, process simulation, and model validation, but only when its intended use and associated risks are examined explicitly [1, 2, 3, 4]. The availability of a recent anonymized discrete-manufacturing corpus with machine-level operational variables enables this question to be studied with reproducible measurements rather than illustrative samples [5].

Diffusion models have become a major generative paradigm

because iterative denoising can represent complex distributions without adversarial training [6, 7]. Extensions to time series include conditional imputation, interpretable generation, and industrial multivariate synthesis [8, 9, 10, 11]. Recent turbine-engine work also illustrates the appeal of conditioning synthetic sensor trajectories on operational context [12]. Nevertheless, an industrially credible benchmark must go beyond visual similarity. A model can achieve favorable marginal distances yet violate simple process relations; it can preserve downstream classification while copying training windows; or it can appear private while suppressing the alarm signatures required for engineering use.

This paper therefore treats synthetic-data evaluation as a constrained, multi-criteria problem. The principal contributions are:

1. a chronological benchmark that separates downstream utility, statistical/temporal fidelity, disclosure risk, and engineering-rule compliance;
2. a conditional diffusion generator augmented by a four-variable differential-evolution output calibration, avoiding retraining during optimization;
3. a comparison with five heterogeneous baselines under three repeated seeds and identical synthetic sample budgets;
4. a deployment-oriented interpretation that reports trade-offs and failure modes instead of declaring a universal winner; and
5. a complete reproducibility package containing the source data, processed windows, model checkpoints, generated data, code, results, and PNG figures.

The study design is summarized in figure 1. Unlike benchmark studies that collapse heterogeneous properties into one score, the reported evidence remains decomposed. This decision follows recent warnings that synthetic data can reproduce analytical errors or imply anonymity without providing it [13, 14].

2. RELATED WORK

2.1 Synthetic data in manufacturing and engineering

Manufacturing applications require synthetic data to represent operating regimes, defects, transitions, and multivariate dependencies. Broad reviews have catalogued generator families and evaluation criteria while also identifying the absence of a universally sufficient quality measure [4, 15]. [1] proposed a systematic framework organized around collection, preprocessing, generation, and evaluation, while [16] emphasized parameterization, verification, and segmentation for general time-series synthesis. For industrial control telemetry, [17] showed that an attention-based variational recurrent autoencoder can capture multivariate temporal dependencies, but its evaluation remained centered on statistical and visual fidelity. These frameworks establish the need for utility and fidelity testing, but neither resolves how privacy and physical process rules should be balanced within a common benchmark. Physics-informed machine learning offers a broader principle: scientific or engineering knowledge should constrain learning when unconstrained statistical fit can produce inadmissible states [18, 19].

2.2 Diffusion models for structured and temporal data

The denoising diffusion probabilistic model constructs a tractable forward corruption process and learns a reverse denoiser [6]. Score-based formulations generalize the process in continuous time [7]. Conditional score models have been successful for multivariate time-series imputation [8]; TabDDPM demonstrated diffusion for mixed tabular data [20]; Diffusion-TS developed interpretable time-series generation [9]; and Diff-MTS introduced temporal augmentation for industrial multivariate series [10]. The survey by [11] identifies fidelity, efficiency, conditioning, and evaluation as persistent challenges. The present study differs by optimizing a post-generation control layer using a metaheuristic and auditing the resulting sequences across utility, privacy, and explicit engineering constraints.

2.3 Synthetic-data auditing and metaheuristic calibration

Sample-level fidelity measures can reveal whether a generator covers the training distribution or merely creates plausible-looking averages [21]. Privacy must also be measured directly. Synthetic records can reproduce training observations [14], diffusion models can emit memorized samples [22], and loss- or likelihood-based membership attacks can distinguish training participation [23]. Accordingly, the benchmark includes exact-copy, near-duplicate, distance-to-closest-record, and membership-inference measures.

Differential evolution (DE) is a population-based optimizer that remains effective for continuous, non-convex engineering search and has accumulated more than two decades of methodological development [24]. Evolutionary methods have also been integrated with generative-model learning [25], while multi-objective hyperparameter optimization literature cautions that scalarization expresses a preference rather than an objective truth [26]. Here, DE is used only to calibrate diffusion outputs against a validation objective; final conclusions are based on the disaggregated test metrics.

As shown in table 1, the novelty lies less in proposing another denoiser than in coupling a transparent metaheuristic calibration with a multi-axis industrial audit.

3. BENCHMARK FORMULATION

3.1 Data representation and chronological protocol

The source is Company A in the SME Manufacturing Dataset, released with the 2023 study by [5]. It contains 1,520,330 timestamped records from nine assets at a nominal five-minute cadence. Four variables were retained: item count, status duration, average power, and cycle duration. Status code 3 was treated as an alarm indicator, consistent with the source labeling. Obvious integer sentinels were controlled through training-period robust caps before logarithmic scaling of item and power variables.

Let a window be $\boldsymbol{X}_i \in \mathbb{R}^{T\times d}$ with $T = 16$ and $d = 4$, corresponding to 80 minutes. The binary condition $y_i \in \{0,1\}$ denotes whether any record in the window contains an alarm. Windows were formed only within uninterrupted five-minute asset runs. Chronological boundaries created training, valida-

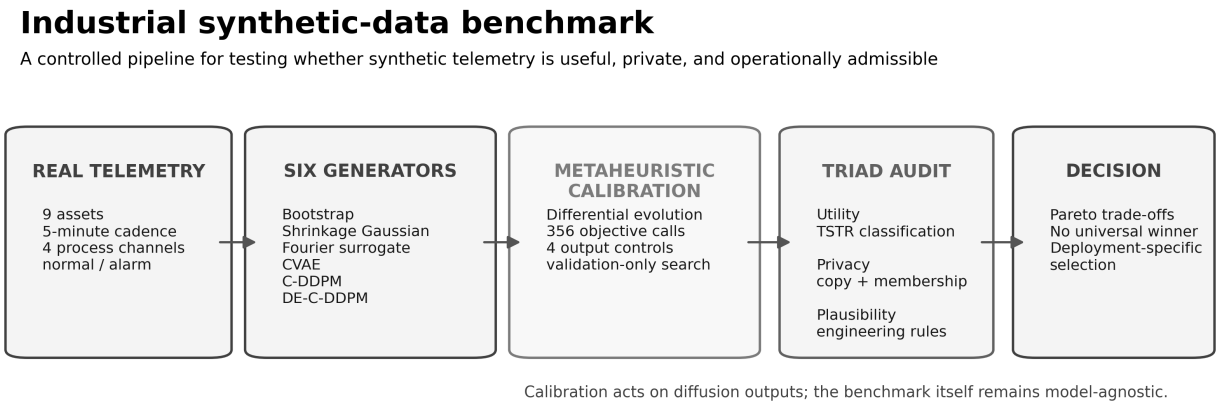

Figure 1. Benchmark architecture. Differential evolution calibrates the conditional diffusion output, whereas the comparison protocol remains model-agnostic and evaluates utility, privacy, and engineering plausibility separately.

Table 1. Positioning of the proposed benchmark relative to recent work.

Study	Industrial time series	Diffusion generator	Privacy audit	Explicit physical-rule audit
[1]	Yes	Not central	Conceptual	Limited
[16]	General time series	No	No	Verification-oriented
[10]	Yes	Yes	No	No
[12]	Turbine sensors	Yes	No	Application metrics
[21]	General synthetic data	Model-agnostic	Sample fidelity	No
This study	Yes	Conditional DDPM + DE	Copy, proximity, membership	Five process constraints

tion, and test periods; each split was balanced by condition after window formation. Table 2 reports the protocol, and figure 2 shows the operating distributions after cleaning.

The training split contained 2,455 unique exact windows among 8,000 balanced samples. This repetition is operationally plausible because assets can remain in steady states, but it makes naive nearest-neighbor privacy thresholds misleading. Consequently, proximity thresholds were estimated from unique training windows only.

3.2 Compared generators

Six generators were selected to span direct resampling, parametric modeling, signal-domain synthesis, latent-variable learning, and diffusion:

1. **Bootstrap**: condition-specific sampling with replacement, included as a high-utility but disclosure-prone reference.
2. **Shrinkage Gaussian**: class-specific mean and Ledoit–Wolf covariance over flattened windows.
3. **Fourier surrogate**: amplitude-preserving temporal surrogates with randomized phases and condition-specific source windows.
4. **CVAE**: a conditional variational autoencoder with a 12-dimensional latent representation.
5. **C-DDPM**: a label-conditioned denoising diffusion probabilistic model with 24 diffusion steps.
6. **DE-C-DDPM**: the same trained C-DDPM followed by a DE-selected four-parameter calibration transform.

Table 3 summarizes the controlled configuration. The benchmark focuses on comparative behavior under a common compute budget, not on claiming that each architecture is exhaustively optimized.

3.3 Conditional diffusion model

For a scaled clean window $\boldsymbol{x}_0$, the forward Markov chain adds Gaussian noise:

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}) = \mathcal{N}\left(\sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t \boldsymbol{I}\right), \tag{1}$$

with $\alpha_t = 1-\beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$. Direct sampling gives

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}). \tag{2}$$

The direct representation in equation (2) supports sampling at arbitrary diffusion steps. A denoiser $\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t, y)$ receives the noisy window, sinusoidal time embedding, and alarm condition. Training minimizes

$$\mathcal{L}_{\mathrm{DDPM}}(\theta) = \mathbb{E}_{\boldsymbol{x}_0, t, y, \boldsymbol{\varepsilon}}\left[\left\|\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t, y)\right\|_2^2\right]. \tag{3}$$

The training criterion in equation (3) estimates the injected noise. The reverse sampler iteratively applies the standard posterior mean using the learned noise estimate [6]. Conditioning enables equal synthetic budgets for normal and alarm windows without post-hoc rejection.

3.4 Differential-evolution output calibration

The calibration operates on a generated scaled window $\boldsymbol{Z}$ and class mean $\boldsymbol{\mu}_y$. Four controls are optimized: mean shrinkage

Table 2. Dataset and chronological benchmark design.

Component	Period / description	Total windows	Normal	Alarm
Source telemetry	30 March 2021–27 October 2022; 9 assets; 5-minute nominal cadence	1,520,330 records	–	–
Training	Before 1 May 2022	8,000	4,000	4,000
Validation	1 May–31 July 2022	2,000	1,000	1,000
Test	1 August–27 October 2022	2,000	1,000	1,000
Synthetic evaluation	6 generators × 3 seeds × 1,000 windows	18,000	9,000	9,000

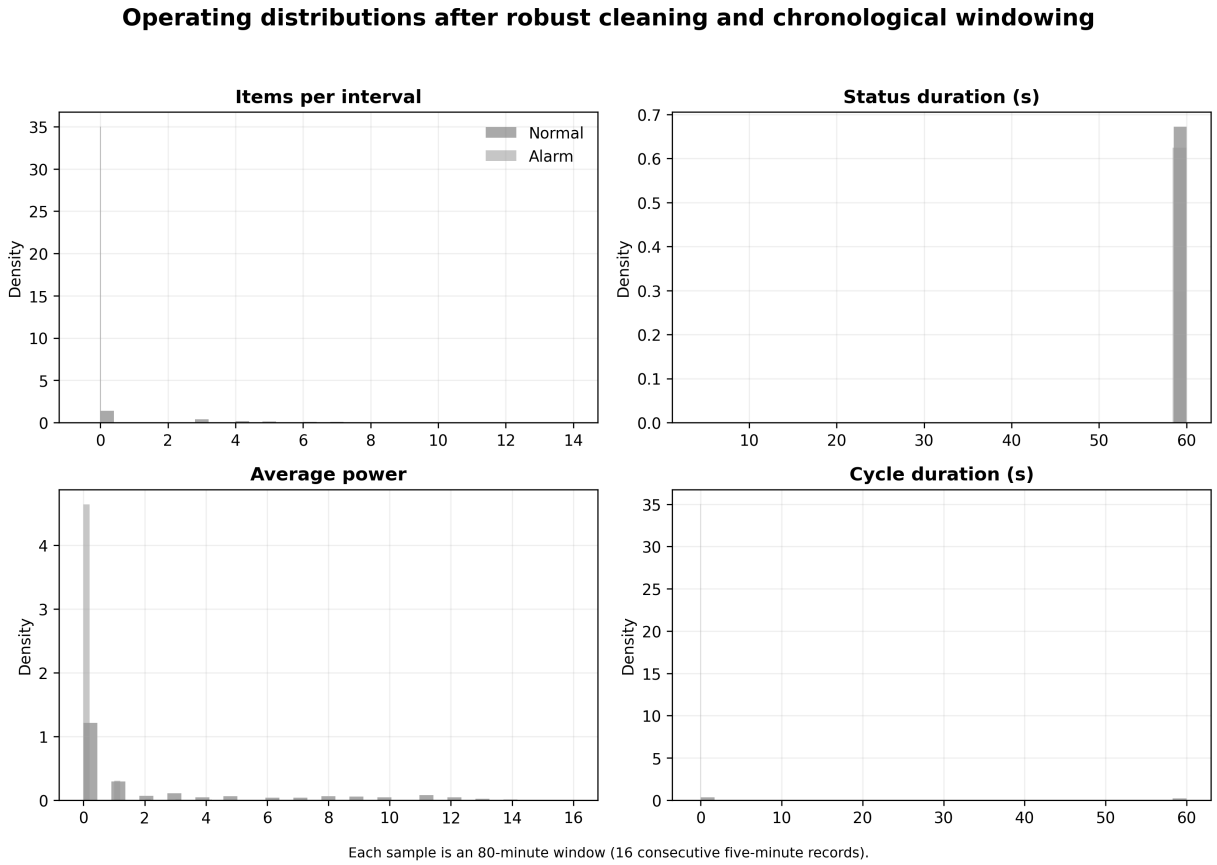

Figure 2. Training-period distributions for normal and alarm-conditioned windows. The plots reveal regime overlap, making downstream discrimination a non-trivial utility test.

Table 3. Generator and evaluation configuration.

Element	Configuration
Window shape	16×4 scaled measurements; condition $y \in \{0,1\}$
CVAE	Fully connected encoder/decoder; latent dimension 12; hidden widths 192 and 128; 16 epochs; KL weight 0.02
C-DDPM	Conditional MLP denoiser; 24 steps; linear β_t schedule from 10^{-4} to 0.02; 20 epochs
DE calibration	Four variables; population multiplier 4; 5 generations; 356 objective calls including polishing; seed 2026
Evaluation repeats	Seeds 11, 37, and 71; 500 windows per condition and generator per seed
Utility learner	Random forest; 80 trees; maximum depth 10; balanced class weighting; selected as a strong, transparent tabular baseline [27]

λ_m, temporal smoothing λ_s, additive privacy noise σ_p, and clipping quantile q_c. First,

$$\boldsymbol{Z}^{(1)} = (1-\lambda_m)\boldsymbol{Z} + \lambda_m \boldsymbol{\mu}_y. \tag{4}$$

Interior time points are smoothed as

$$\boldsymbol{Z}_t^{(2)} = (1-\lambda_s)\boldsymbol{Z}_t^{(1)} + \frac{\lambda_s}{2}\left(\boldsymbol{Z}_{t-1}^{(1)} + \boldsymbol{Z}_{t+1}^{(1)}\right), \tag{5}$$

then $\mathcal{N}(0,\sigma_p^2)$ noise is added and each channel is clipped to the generated batch's q_c and $1-q_c$ quantiles.

For a DE population vector $\boldsymbol{p}_i^{(g)}$, mutation and crossover follow

$$\boldsymbol{v}_i^{(g)} = \boldsymbol{p}_{r_1}^{(g)} + F\left(\boldsymbol{p}_{r_2}^{(g)} - \boldsymbol{p}_{r_3}^{(g)}\right), \tag{6}$$

$$u_{ij}^{(g)} = \begin{cases} v_{ij}^{(g)}, & r_j \le C_R \text{ or } j = j_{\text{rand}}, \\ p_{ij}^{(g)}, & \text{otherwise.} \end{cases} \tag{7}$$

followed by greedy selection. The fixed validation objective was

$$J = \frac{1}{2}\sum_{y=0}^{1}\big(0.30W_y + 0.22A_y + 0.12C_y + 0.23P_y + 0.13R_y\big), \tag{8}$$

where W is normalized marginal Wasserstein error, A is autocorrelation error, C is correlation-matrix error, P is physical-rule violation, and R penalizes median distance below the unique-training reference separation. The scalarization in equation (8) is an engineering preference used to find one operating point; it is not used as the final benchmark score.

3.5 Evaluation dimensions

3.5.1 Utility

Train-synthetic/test-real (TSTR) evaluation trains an alarm classifier on generated windows and tests it on the untouched chronological real test split. The primary measure is ROC–AUC, supplemented by average precision, F1, and balanced accuracy. A train-real/test-real (TRTR) classifier provides a contextual baseline rather than a strict upper bound.

3.5.2 Fidelity

Marginal Wasserstein distances are averaged over channels and conditions. Temporal behavior is measured through lag autocorrelation and power spectral density errors; cross-variable structure is measured through correlation-matrix error. These complementary measurements avoid equating marginal similarity with sequence fidelity.

3.5.3 Privacy

For each synthetic window $\tilde{\boldsymbol{x}}$, distance to the closest unique training record is

$$D_{\min}(\tilde{\boldsymbol{x}}) = \min_{\boldsymbol{x}\in\mathscr{D}_{\text{train}}^{\text{unique}}} \|\text{vec}(\tilde{\boldsymbol{x}}) - \text{vec}(\boldsymbol{x})\|_2 . \tag{9}$$

Exact copies use a numerical tolerance of 10^{-10}. Near duplicates use a threshold estimated from non-self nearest-neighbor separation among unique training windows. A distance-based membership classifier distinguishes training from held-out records; because attack orientation may reverse, the reported AUC is $\max(\text{AUC}, 1-\text{AUC})$.

3.5.4 Physical plausibility

Five normalized rule checks were measured after decoding: hard range-projection checks; cycle duration exceeding status duration; positive item production without a positive cycle duration; positive power without status duration; and abrupt inter-step jumps beyond robust training limits. Their unweighted mean is the aggregate physical-violation rate. The range check is zero by construction because decoding applies hard engineering bounds; it is retained to verify that this protection is active for every generator. These rules are intentionally simple and auditable; they test whether generators respect basic operating semantics rather than claiming a complete digital twin.

4. RESULTS

4.1 Downstream utility

The TRTR ROC–AUC was 0.601. As reported in table 4 and visualized in figure 3, bootstrap data produced the largest mean TSTR ROC–AUC (0.610), followed by shrinkage Gaussian (0.594) and CVAE (0.588). Bootstrap slightly exceeded the TRTR estimate, which is plausible under resampling and classifier variance and should not be interpreted as synthetic data being intrinsically superior. The uncalibrated C-DDPM had the largest variation across seeds (0.551 ± 0.126). DE calibration increased its mean AUC to 0.561 and reduced the standard deviation to 0.037. Its balanced accuracy also rose from 0.489 to 0.592, while F1 increased from 0.224 to 0.710.

4.2 Distributional and temporal fidelity

The CVAE achieved the smallest average Wasserstein distance (0.046), but its autocorrelation and correlation-matrix errors were the largest among the non-diffusion models. Fourier surrogates preserved spectral behavior well but had the worst marginal Wasserstein error (0.651). These results, detailed in table 5 and figure 4, demonstrate why a single fidelity statistic is insufficient. The DE calibration reduced diffusion Wasserstein error from 0.401 to 0.284 (29.2%), with nearly unchanged spectral and correlation errors; autocorrelation error increased from 0.302 to 0.310.

Representative alarm trajectories in figure 5 make the metric decomposition tangible. Bootstrap retains realistic local patterns by construction. CVAE smooths some transitions, whereas raw diffusion can produce large incompatible changes. The calibrated diffusion reduces these extremes without reproducing an observed window exactly.

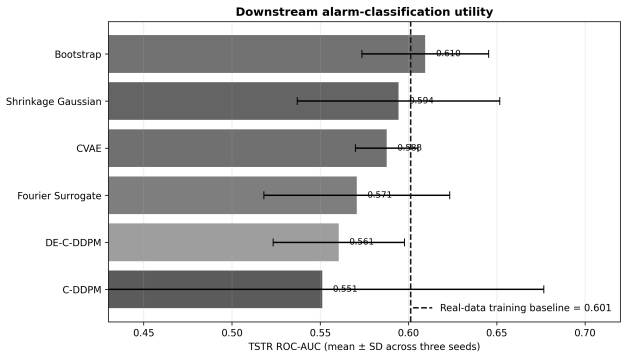

Figure 3. Repeated TSTR alarm-classification utility. The dashed line is the TRTR reference obtained from real training windows.

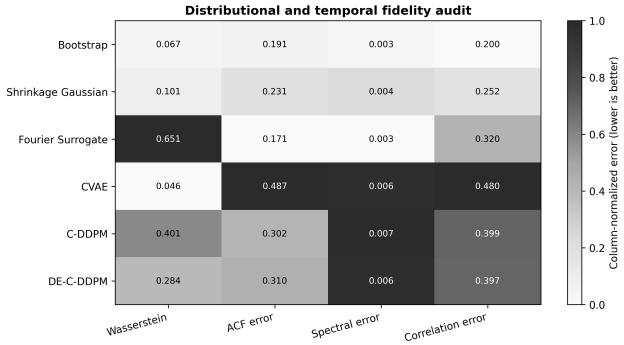

Figure 4. Fidelity audit across marginal, temporal, spectral, and dependency measures. Cell values are raw errors; shading is normalized within each column.

Table 4. Main benchmark results (mean ± standard deviation across three seeds). Lower is better for errors and violation/copy rates; membership AUC is best near 0.5.

Generator	TSTR ROC–AUC	Wasserstein	Physical violation	Exact copies	Membership AUC
Bootstrap	0.610 ± 0.036	0.067 ± 0.006	0.013%	77.9%	0.530 ± 0.003
Shrinkage Gaussian	0.594 ± 0.057	0.101 ± 0.005	0.831%	0.0%	0.529 ± 0.002
Fourier surrogate	0.571 ± 0.053	0.651 ± 0.010	0.211%	0.0%	0.518 ± 0.003
CVAE	0.588 ± 0.018	$\mathbf{0.046} \pm 0.003$	0.020%	0.0%	0.540 ± 0.005
C-DDPM	0.551 ± 0.126	0.401 ± 0.012	8.966%	0.0%	0.503 ± 0.002
DE-C-DDPM	0.561 ± 0.037	0.284 ± 0.011	0.473%	0.0%	0.575 ± 0.005

Table 5. Fidelity decomposition averaged across conditions and seeds.

Generator	Wasserstein	ACF error	Spectral error	Correlation error
Bootstrap	0.0669	0.1907	0.00273	0.2003
Shrinkage Gaussian	0.1011	0.2306	0.00354	0.2516
Fourier surrogate	0.6512	**0.1713**	**0.00264**	0.3197
CVAE	**0.0460**	0.4867	0.00645	0.4804
C-DDPM	0.4015	0.3022	0.00656	0.3993
DE-C-DDPM	0.2842	0.3100	0.00645	0.3967

4.3 Privacy audit

The most consequential privacy result is the contrast between predictive utility and record disclosure. Bootstrap retained the highest utility but exactly copied 77.9% of generated windows and produced 79.1% near duplicates. No other method generated exact copies. CVAE produced 9.23% near duplicates despite a zero exact-copy rate, showing why exact matching alone is inadequate. Membership AUC remained relatively close to chance for most methods, but DE-C-DDPM reached 0.575. Figure 6 reports both proximity and attack-based views. The result aligns with prior evidence that synthetic generation does not itself guarantee anonymization [14, 22, 23].

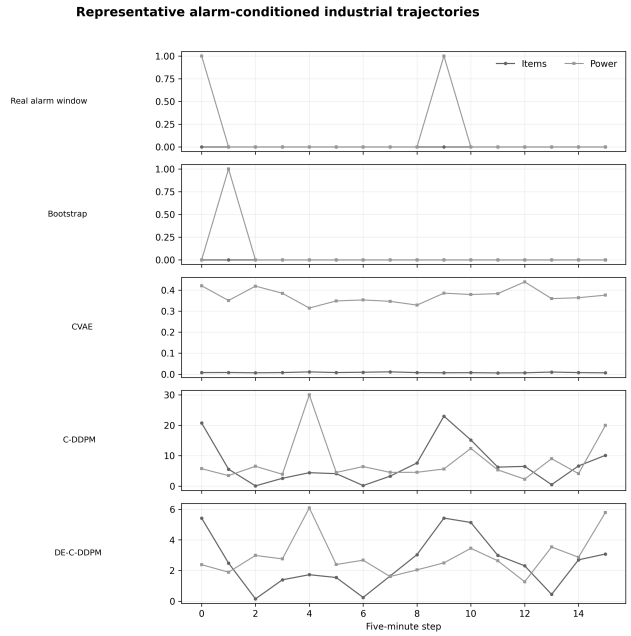

Figure 5. Representative alarm-conditioned trajectories for item count and average power. The examples are illustrative; all conclusions use repeated aggregate metrics.

4.4 Physical plausibility

The real test split had an aggregate rule-violation rate of 0.0131%. Bootstrap and CVAE were closest to this level. The raw C-DDPM violated 8.966% of rule checks, driven by cycle/status inconsistencies, abrupt jumps, and item/cycle conflicts. DE calibration reduced the rate to 0.473%, a 94.7% relative reduction. The detailed components in table 6 and figure 7 show that calibration eliminated observed bounds, item/cycle, power/status, and abrupt-jump violations in the generated evaluation samples; residual error was concentrated in cycle duration exceeding status duration.

4.5 Metaheuristic calibration and trade-offs

The fixed-budget DE search selected $\lambda_m = 0.350$, $\lambda_s = 0.180$, $\sigma_p = 0.061$, and $q_c = 0.054$. The iteration limit was reached after five generations; therefore, the solution is described as the best observed validation point, not a proof of global convergence. Figure 8 shows the objective history and selected controls. The comparison in table 7 confirms that the largest gains concern physical plausibility and marginal fidelity, whereas privacy attack exposure worsens and ACF error does not improve.

The utility–fidelity map in figure 9 reinforces the absence of a universal winner. Bootstrap lies near the high-utility/low-Wasserstein region but fails record-level privacy. CVAE attains excellent marginal fidelity and physical compliance but weaker temporal dependency preservation. Shrinkage Gaussian provides a comparatively stable compromise without direct copies. DE-C-DDPM moves substantially toward the feasible physical region but does not overtake the simpler generators on utility or privacy.

5. DISCUSSION

5.1 Implications for industrial engineering

Three practical lessons follow. First, high TSTR performance can arise from copying: bootstrap is a useful diagnostic baseline but is unsuitable for external release. Second, low marginal error does not guarantee valid dynamics. The CVAE's favorable Wasserstein score coexists with comparatively high ACF and correlation errors. Third, output calibration can be an efficient engineering control when retraining is costly. The DE layer changed only generated sequences, allowing process-rule penalties and disclosure-distance terms to be introduced without modifying the diffusion network.

A deployment decision should therefore specify acceptance regions. For internal model debugging, a generator may prioritize utility and physical validity while access controls man-

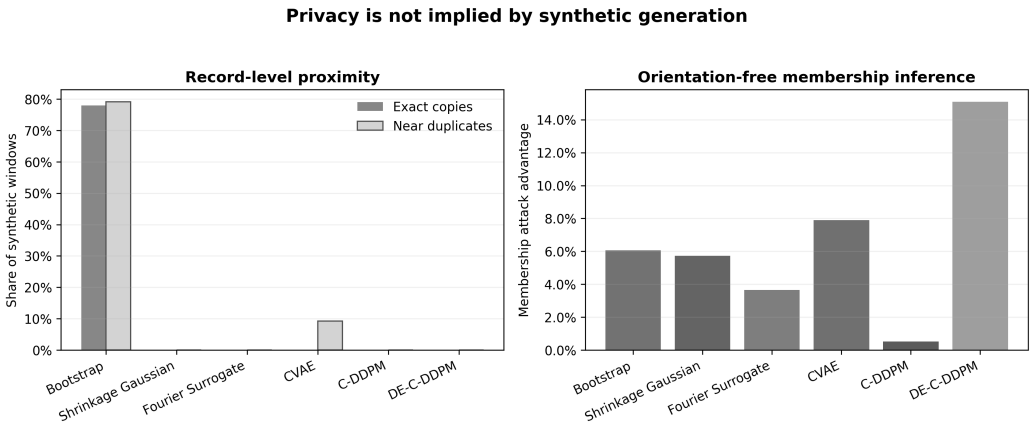

Figure 6. Privacy audit. Left: exact-copy and near-duplicate shares. Right: orientation-free membership attack advantage $2(\mathrm{AUC} - 0.5)$.

Table 6. Physical-rule violation rates (percent).

Generator	Bounds	Cycle > status	Items/no cycle	Power/no status	Jumps	Mean
Bootstrap	0.000	0.002	0.065	0.000	0.000	0.013
Shrinkage Gaussian	0.000	3.063	1.094	0.000	0.000	0.831
Fourier surrogate	0.000	0.915	0.008	0.000	0.133	0.211
CVAE	0.000	0.098	0.002	0.000	0.000	0.020
C-DDPM	0.000	25.977	3.435	0.871	14.547	8.966
DE-C-DDPM	0.000	2.365	0.000	0.000	0.000	0.473

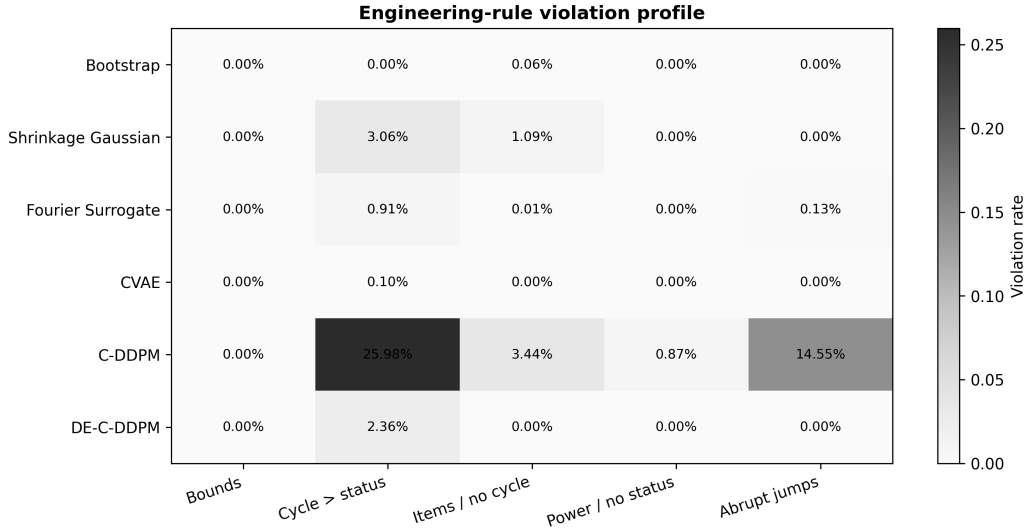

Figure 7. Engineering-rule violation profile. The uncalibrated diffusion model produces the broadest set of inadmissible relations; DE calibration sharply contracts that profile.

Table 7. Effect of DE calibration relative to the same uncalibrated C-DDPM.

Measure	C-DDPM	DE-C-DDPM	Change
TSTR ROC–AUC	0.5512	0.5605	+0.0093
Wasserstein error	0.4015	0.2842	−29.2%
Physical violation	8.966%	0.473%	−94.7%
ACF error	0.3022	0.3100	+2.6%
Membership AUC	0.5026	0.5754	+0.0728

age disclosure. For cross-organization release, exact-copy and membership-risk thresholds become mandatory, even if utility decreases. For simulator initialization or digital-twin stress tests, constraint compliance and temporal structure should dominate marginal similarity. The same principle extends to energy and electrical-engineering telemetry, where synthetic-data trustworthiness has been framed jointly in terms of robustness, privacy, reproducibility, and sustainability [28]. These choices reflect the multi-objective perspective described by [26]; they should not be obscured by a weighted overall rank.

5.2 Why the calibrated diffusion is still useful

The benchmark does not show that diffusion dominates simpler approaches. Instead, it identifies a repairable weakness: raw C-DDPM outputs violated engineering relations at an unacceptable rate. DE calibration reduced these violations by nearly an order of magnitude and improved marginal fidelity without retraining. This makes the method relevant

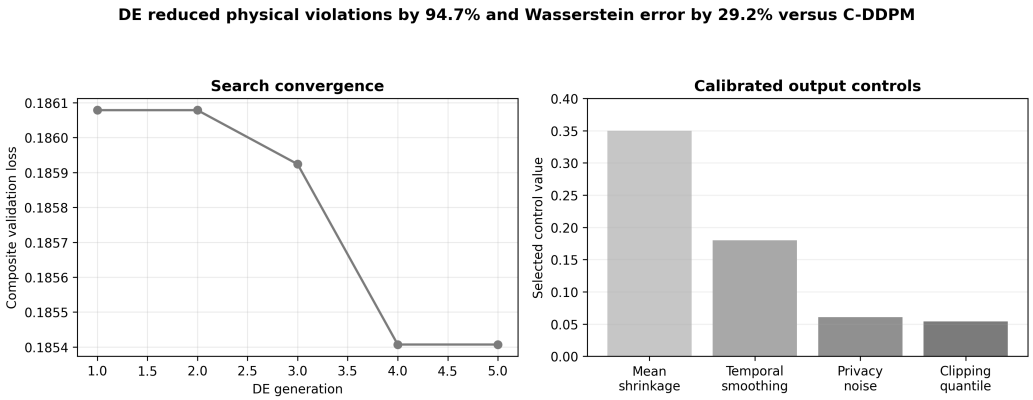

Figure 8. DE calibration. The objective improved within a deliberately small search budget; selected controls prioritize mean contraction and moderate temporal smoothing.

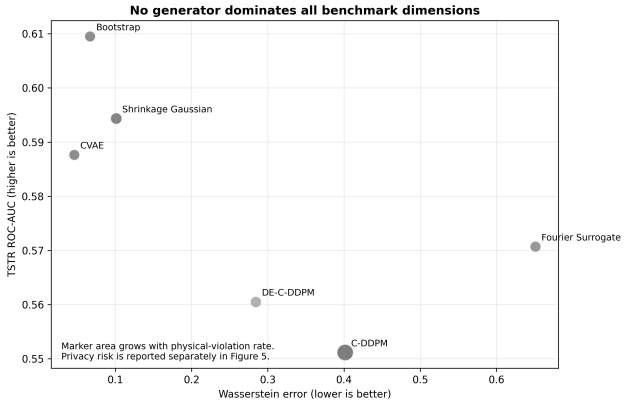

Figure 9. Utility–fidelity operating points. Marker area increases with physical-violation rate; privacy is intentionally shown separately in figure 6 rather than hidden in a composite score.

when conditional diffusion is required for extensible context control, rare-event synthesis, or integration with richer text and asset embeddings. The present four-variable transform is deliberately lightweight; future work can place DE or a multi-objective evolutionary algorithm inside training, optimize the noise schedule, or evolve rule-specific projection operators.

6. LIMITATIONS AND THREATS TO VALIDITY

The study uses one anonymized company and four retained channels. The alarm label is derived from source status coding and does not identify failure mechanisms. Windows are balanced for controlled evaluation, so prevalence-sensitive deployment performance must be re-estimated under site-specific alarm frequencies. The physical rules are necessary but not sufficient; they do not encode asset-specific thermodynamics, control logic, conservation laws, or maintenance states. The neural models use moderate training budgets, and the DE search is intentionally small. Results should therefore be interpreted as a reproducible benchmark of trade-offs, not a definitive architecture ranking.

Privacy results are empirical rather than formal. A low attack AUC does not provide differential privacy, and stronger white-box attacks may reveal additional leakage. Finally, the source contains repeated steady states; although unique-window thresholds reduce distortion, record similarity can still reflect legitimate process recurrence rather than memorization. These limitations are documented to encourage additional datasets, constraints, and attacks rather than to weaken the need for multi-axis evaluation.

7. CONCLUSION

This study introduced a synthetic industrial time-series benchmark that evaluates utility, privacy, and physical plausibility as separate requirements. Across six generators and 18,000 generated windows, no method dominated all dimensions. Bootstrap retained strong downstream utility but directly reproduced training records. CVAE offered the best marginal fidelity and low rule violation, while Fourier surrogates preserved selected temporal properties at the expense of distributional fit. The uncalibrated conditional diffusion model exhibited substantial engineering-rule violations. A differential-evolution output calibration reduced those violations by 94.7% and Wasserstein error by 29.2%, with a small utility gain, but increased membership-inference exposure. The central conclusion is therefore methodological: synthetic industrial data should be accepted through task-specific, measurable constraints rather than realism claims or a single aggregate score.

REFERENCES

[1] V. Buggineni, C. Chen, and J. Camelio, "Enhancing manufacturing operations with synthetic data: A systematic framework for data generation, accuracy, and utility," *Frontiers in Manufacturing Technology*, vol. 4, p. 1320166, 2024.

[2] F. K. Dankar and M. Ibrahim, "Fake it till you make it: Guidelines for effective synthetic data generation," *Applied Sciences*, vol. 11, no. 5, p. 2158, 2021.

[3] M. Goyal and Q. H. Mahmoud, "A systematic review of synthetic data generation techniques using generative AI," *Electronics*, vol. 13, no. 17, p. 3509, 2024.

[4] A. Figueira and B. Vaz, "Survey on synthetic data generation, evaluation methods and GANs," *Mathematics*, vol. 10, no. 15, p. 2733, 2022.

[5] D. Atzeni, R. Ramjattan, R. Figliè, G. Baldi, and D. Mazzei, "Data-driven insights through industrial retrofitting: An anonymized dataset with machine learning use cases," *Sensors*, vol. 23, no. 13, p. 6078, 2023.

[6] J. Ho, A. N. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 6840–6851.

[7] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through

stochastic differential equations," in *International Conference on Learning Representations*, 2021.

[8] Y. Tashiro, J. Song, Y. Song, and S. Ermon, "CSDI: Conditional score-based diffusion models for probabilistic time series imputation," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 24 804–24 816.

[9] X. Yuan and Y. Qiao, "Diffusion-TS: Interpretable diffusion for general time series generation," in *International Conference on Learning Representations*, 2024.

[10] L. Ren, H. Wang, and Y. Laili, "Diff-MTS: Temporal-augmented conditional diffusion-based AIGC for industrial time series toward the large model era," *IEEE Transactions on Cybernetics*, vol. 54, no. 12, pp. 7187–7197, 2024.

[11] L. Lin, Z. Li, R. Li, X. Li, and J. Gao, "Diffusion models for time-series applications: A survey," *Frontiers of Information Technology & Electronic Engineering*, vol. 25, no. 1, pp. 19–41, 2024.

[12] L. P. Mora-de León, D. Solís-Martín, J. Galán-Páez, and J. Borrego-Díaz, "Text-conditioned diffusion-based synthetic data generation for turbine engine sensor analysis and RUL estimation," *Machines*, vol. 13, no. 5, p. 374, 2025.

[13] B. Van Breugel, Z. Qian, and M. van der Schaar, "Synthetic data, real errors: How (not) to publish and use synthetic data," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 202, 2023, pp. 34 793–34 808.

[14] T. Stadler, B. Oprisanu, and C. Troncoso, "Synthetic data–anonymisation groundhog day," in *31st USENIX Security Symposium*, 2022, pp. 1451–1468.

[15] H. Murtaza, M. Ahmed, N. F. Khan, G. Murtaza, S. Zafar, and A. Bano, "Synthetic data generation: State of the art in health care domain," *Computer Science Review*, vol. 48, p. 100546, 2023.

[16] K.-M. Kim and J. W. Kwak, "PVS-GEN: Systematic approach for universal synthetic data generation involving parameterization, verification, and segmentation," *Sensors*, vol. 24, no. 1, p. 266, 2024.

[17] S. Jeon and J. T. Seo, "A synthetic time-series generation using a variational recurrent autoencoder with an attention mechanism in an industrial control system," *Sensors*, vol. 24, no. 1, p. 128, 2024.

[18] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, 2021.

[19] J. Willard, X. Jia, S. Xu, M. Steinbach, and V. Kumar, "Integrating scientific knowledge with machine learning for engineering and environmental systems," *ACM Computing Surveys*, vol. 55, no. 4, pp. 1–37, 2022.

[20] A. Kotelnikov, D. Baranchuk, I. Rubachev, and A. Babenko, "TabDDPM: Modelling tabular data with diffusion models," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 202, 2023, pp. 17 564–17 579.

[21] A. Alaa, B. Van Breugel, E. S. Saveliev, and M. van der Schaar, "How faithful is your synthetic data? sample-level metrics for evaluating and auditing generative models," in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 162, 2022, pp. 290–306, .

[22] N. Carlini, J. Hayes, M. Nasr, M. Jagielski, V. Sehwag, F. Tramèr, B. Balle, D. Ippolito, and E. Wallace, "Extracting training data from diffusion models," in *32nd USENIX Security Symposium*, 2023, pp. 5253–5270.

[23] H. Hu and J. Pang, "Loss and likelihood based membership inference of diffusion models," in *Information Security*, ser. Lecture Notes in Computer Science. Springer, 2023, vol. 14411, pp. 121–141.

[24] Bilal, M. Pant, H. Zaheer, L. Garcia-Hernandez, and A. Abraham, "Differential evolution: A review of more than two decades of research," *Engineering Applications of Artificial Intelligence*, vol. 90, p. 103479, 2020.

[25] J. Drefs, E. Guiraud, and J. Lücke, "Evolutionary variational optimization of generative models," *Journal of Machine Learning Research*, vol. 23, no. 21, pp. 1–51, 2022.

[26] F. Karl, T. Pielok, J. Moosbauer, F. Pfisterer, S. Coors, M. Binder, L. Schneider, J. Thomas, J. Richter, M. Lang, E. C. Garrido-Merchán, J. Branke, and B. Bischl, "Multi-objective hyperparameter optimization in machine learning–an overview," *ACM Transactions on Evolutionary Learning and Optimization*, vol. 3, no. 4, pp. 1–50, 2023.

[27] R. Shwartz-Ziv and A. Armon, "Tabular data: Deep learning is not all you need," *Information Fusion*, vol. 81, pp. 84–90, 2022.

[28] M. Meiser and I. Zinnikus, "A survey on the use of synthetic data for enhancing key aspects of trustworthy AI in the energy domain: Challenges and opportunities," *Energies*, vol. 17, no. 9, p. 1992, 2024.