

Article

A Unity-Based Multi-Asset Industrial Digital Shadow Environment with Embedded Machine Learning for Interactive Fault Diagnosis

Jose E. Naranjo^{*}, Segundo Á. Cevallos Betún, Jessica N. Castillo

Facultad de Ciencias de la Ingeniería y Aplicadas, Universidad Técnica de Cotopaxi, Latacunga 050102, Ecuador

^{*} Corresponding author: Jose E. Naranjo, jose.naranjo0463@utc.edu.ec

CITATION

Naranjo JE, Cevallos Betún SA, Castillo JN. A Unity-Based Multi-Asset Industrial Digital Shadow Environment with Embedded Machine Learning for Interactive Fault Diagnosis. *Metaverse*. 2026; 7(4): m260049.
<https://doi.org/10.54517/m260049>

ARTICLE INFO

Received: 28 August 2026

Accepted: 14 September 2026

Available online: 30 September 2026

COPYRIGHT



Copyright © 2026 by author(s).

Metaverse is published by Asia Pacific Academy of Science Pte. Ltd. This work is licensed under the Creative Commons Attribution (CC BY) license.

<https://creativecommons.org/licenses/by/4.0/>

Abstract: Industrial metaverse and intelligent virtual-environment research increasingly combines immersive three-dimensional representation, digital-twin/shadow concepts, and artificial intelligence, yet reproducible examples that embed asset-specific diagnostic models directly inside a multi-asset game-engine environment remain limited. This paper presents a Unity-based, Digital Shadow-oriented virtual industrial environment containing six heterogeneous manufacturing assets: an industrial pump, furnace, CNC milling machine, CNC lathe, laser cutting station, and packaging/quality-control line. Each asset is represented by asset-specific operational variables and a dedicated local diagnostic workflow deployed through ONNX and Unity Sentis. The pump pilot uses an original controlled synthetic dataset of 800 records across four operating conditions and six variables. Four baseline classifiers were compared on an 80/20 stratified split: K-Nearest Neighbors, Support Vector Machine (SVM), Random Forest, and an artificial neural network (ANN). SVM achieved the highest held-out accuracy (98.12%), while the ANN achieved 97.50%. To strengthen the multi-asset evidence, controlled synthetic benchmark datasets were also quantitatively evaluated for the five additional assets (800 records and four classes per asset); the best held-out accuracies ranged from 94.38% to 98.12%. Robustness was further examined across five independently sampled datasets per asset using fixed documented seeds. A matched Unity runtime experiment compared the deployed ANN and SVM over 30 end-to-end executions: the ANN required 22.00 ± 1.66 ms and the SVM 25.00 ± 3.28 ms, revealing a measurable accuracy-runtime trade-off rather than a universally superior classifier. Unity output percentages are therefore interpreted as uncalibrated model output probabilities, not guaranteed diagnostic confidence. The platform is explicitly scoped as a Digital Shadow-oriented proof of concept because it uses synthetic surrogate inputs and lacks continuous physical-sensor connectivity or bidirectional actuation. The complete reproducibility package includes datasets, generation specifications, scripts, models, robustness outputs, and raw Unity latency measurements.

Keywords: Digital Shadow; Industrial metaverse; Intelligent virtual environment; Unity; Fault diagnosis; ONNX; Unity Sentis; Embedded machine learning; Synthetic data; On-device inference

1. Introduction

Digital representation of manufacturing assets within interactive virtual environments has become a central theme of Industry 4.0 and the emerging Industrial Metaverse, enabling remote inspection, operator training, data-driven visualization, and interactive access to digital counterparts of machines and production systems [1–6]. Contemporary Industrial Metaverse frameworks emphasize the convergence of digital representations, extended-reality interfaces, artificial intelligence, and persistent or data-connected virtual environments to support smart-manufacturing

activities across design, operation, maintenance, and training [4–6]. Within this context, the specific problem addressed in this paper is narrower and engineering-oriented: embedding intelligent, asset-specific analytics directly inside the virtual environment so that a user interacting with a machine representation can trigger and receive an on-device diagnostic assessment of its current operating condition. The resulting platform should therefore be understood as an intelligent industrial virtual environment and a technical building block toward richer Industrial Metaverse implementations, rather than as a complete metaverse ecosystem.

1.1. Digital Twin and Digital Shadow Concepts

The Digital Twin concept has been extensively discussed as a virtual, continuously updated representation of a physical asset, process, or system connected through bidirectional data exchange that allows the virtual model to reflect and, in some implementations, influence its physical counterpart [7,8]. The term is nevertheless used inconsistently. Kritzinger et al. distinguish Digital Models, Digital Shadows, and Digital Twins according to the degree and direction of automated data exchange [9], while Wright and Davidson caution against labeling partially automated or static representations as full Digital Twins [10]. Recent manufacturing reviews likewise show that many maintenance-oriented implementations are closer to Digital Shadows because operational data update the virtual domain without closed-loop actuation [11,12]. This distinction is particularly relevant to the Industrial Metaverse, where immersive interaction does not by itself establish bidirectional physical synchronization [4–6]. Applied examples span reinforcement-learning training environments [13], human-robot collaboration [14], factory resilience [15], and ML-enabled simulated digital twins for fault diagnosis [16]. The present study does not have continuously instrumented physical assets; it therefore implements a Digital Shadow-oriented architecture in which controlled synthetic operational data act as a surrogate data source. This terminology is maintained deliberately throughout the paper.

1.2. Game Engines and Unity in Industrial Virtual Environments

Game engines have transitioned from entertainment-oriented software into real-time three-dimensional platforms suitable for industrial training, simulation, remote monitoring, human-machine interaction, and virtual commissioning [2,3,17–21]. This trajectory is directly relevant to Industrial Metaverse research, which treats immersive interaction, digital representations, AI-enabled services, and connectivity between physical and virtual spaces as complementary technical layers [4–6]. Unity has been widely used to implement interactive industrial scenes because it combines real-time rendering, first-person or immersive navigation, user-interface controls, and extensible software integration. Previous work by the authors and collaborators has also demonstrated 3D interactive environments for assistive human-machine interaction [22], immersive fault-oriented training for induction motors with user evaluation [23], and remote robotic teleoperation architectures for industrial processes [24]. These studies motivate the interaction and industrial-connectivity aspects of the present work, but the current contribution differs by embedding asset-specific ML diagnostic inference locally inside a multi-asset Unity environment. More recent work has coupled Unity-based VR/AR environments directly with digital-twin architectures for production-line control and multi-user interaction [21]. Unity Sentis further enables ONNX models to be executed inside the game-engine runtime without an external inference server [25,26], closing the gap between offline model training and local interactive inference.

1.3. Embedded Machine Learning for Industrial Fault Diagnosis

Beyond digital representation and visualization, embedding intelligent analytics inside a virtual industrial environment requires an underlying ML component capable of translating operational variables into an interpretable diagnostic output. ML-based fault diagnosis has been applied to a wide range of rotating and reciprocating industrial equipment, including pumps, motors and compressors, typically framed as a multiclass classification problem over discretized health states [27,28]. Liu et al. provide a broad survey of artificial-intelligence methods for rotating-machinery fault diagnosis, noting the continued relevance of classical, feature-based classifiers alongside emerging deep-learning approaches [29]. Focusing specifically on pumps, Kumar and Kumar used time–frequency analysis combined with a Support Vector Machine to detect defects from vibration signals of a centrifugal pump [30], while Ranawat et al. compared Support Vector Machine and Artificial Neural Network classifiers for centrifugal-pump fault diagnosis under multiple operating conditions [31]. Tama et al. reviewed recent deep-learning approaches for vibration-based rotating-machinery diagnosis, highlighting the trade-off between model complexity and the amount of labeled data required for reliable training [32], and Schwendemann et al. surveyed machine-learning techniques for condition monitoring and predictive maintenance of bearings, reporting that classical classifiers such as K-Nearest Neighbors (KNN), SVM and Random Forest remain competitive baselines on small and moderately sized tabular datasets [33]. Focusing specifically on the KNN family, Lu et al. proposed an enhanced KNN variant for intelligent fault diagnosis of rotating machinery, reporting improved classification accuracy over standard KNN on benchmark vibration datasets [34], supporting the use of KNN as a legitimate supervised baseline in the present study. Wang et al. combined vibro-acoustic data fusion with a one-dimensional convolutional neural network for bearing diagnosis, illustrating how deep architectures can be favored when richer, higher-dimensional sensor signals are available [35], a data regime distinct from the low-dimensional tabular feature set used in the present environment. Closer to the digital-representation focus of this paper, Alshathri et al. proposed a Digital Twin-based fault-diagnosis framework for a triplex pump, combining a simulation model with genetic-algorithm-optimized classical classifiers [36]; unlike the present study, their framework does not couple the diagnostic model with a game-engine visualization layer or an ONNX-based embedded-deployment pathway. Within this broader landscape, “predictive maintenance” is understood as the general industrial application context [37,38], while the specific technical problem addressed and embedded within the proposed virtual environment is fault diagnosis, understood as the classification of the current operating condition of an asset from its instantaneous operational variables, rather than the temporal forecasting of a future failure event or the estimation of remaining useful life (RUL); this distinction is maintained consistently throughout the paper.

A further practical obstacle to embedding intelligent analytics inside virtual industrial environments is the limited availability of real, densely instrumented industrial datasets. Access to production-line sensor data is frequently restricted by confidentiality agreements, safety regulations and the cost of retrofitting legacy machinery with sensing infrastructure [39]. Controlled synthetic datasets, generated to reproduce plausible statistical patterns of normal and faulty operating regimes, provide a pragmatic alternative for developing, testing and demonstrating an embedded diagnostic pipeline while a real-data acquisition campaign is planned or before industrial partnerships are formalized [40]. Selcuk et al. proposed a workflow

for generating synthetic vibration data specifically for predictive-maintenance applications [41], Buggineni et al. proposed a systematic framework for generating, evaluating and applying synthetic data across manufacturing operations [42], Chan et al. used discrete-event simulation to generate synthetic manufacturing datasets for ML training [43], and generative approaches have also been explored for synthetic data augmentation in industrial visual inspection tasks [44]. This paper explicitly adopts a controlled synthetic-data approach: all operational inputs used to develop and evaluate the embedded diagnostic pipeline are synthetic, and this choice is treated as a methodological boundary condition rather than a limitation to be concealed.

1.4. Research Gap and Contributions

Taken together, the literature shows substantial progress in Digital Twin/Digital Shadow architectures, Industrial Metaverse concepts, immersive industrial interaction, ML-based fault diagnosis, and game-engine deployment, but these dimensions are still often investigated separately. Industrial Metaverse studies define broad architectures and enabling technologies [4–6], while Unity/VR studies emphasize interaction, training, or synchronized visualization [20,21,23]. Fault-diagnosis studies often provide stronger classifier validation but no embedded game-engine interaction [30,31,36]. **Table 1** positions the present work against representative closely related systems using the dimensions specifically relevant to this study: asset scope, virtual environment, physical connectivity, ML functionality, embedded inference, and real-time interaction. The novelty claim is therefore intentionally narrow: a reproducible Unity-based multi-asset environment in which heterogeneous assets expose their own operational variables and locally executed diagnostic workflows through a common ONNX/Sentis interaction pattern. This claim concerns architectural integration and reproducibility, not superior diagnostic accuracy on real machinery.

Table 1. Comparison of representative closely related systems and the positioning of the proposed multi-asset Digital Shadow-oriented environment.

Reference	Assets/domain	Virtual environment	Physical connectivity	ML functionality	Embedded inference	Real-time interaction
Xia et al. [13]	Smart manufacturing plant	Custom digital-twin environment	Simulated	Deep reinforcement learning	Not ONNX/Sentis	Interactive training/simulation
Dewberry et al. [21]	Production-line / multi-user Industry 4.0	Unity VR	Real-time synchronized	Not centered on classifier comparison	No reported ONNX/Sentis pipeline	Yes, multi-user
Alshathri et al. [36]	Triplex pump	No game-engine layer	Simulated/IIoT-oriented DT	GA-optimized LR/NB/SVM	No embedded Unity inference	No
Naranjo et al. [23]	Induction motor training	Immersive VR	Training demonstrator	Fault-oriented guidance	No ONNX/Sentis	Yes; usability evaluated
Ren et al. [4]	Smart manufacturing, conceptual	Industrial Metaverse architecture	Conceptual physical-virtual integration	AI as enabling layer	Architecture-level	Conceptual
Tu et al. [6]	Smart factory/material flow	Industrial Metaverse architecture	Physical-factory / virtual-landscape bridge	Data/semantic intelligence	Architecture-level	Yes, architecture/case study

Continuation Table:

Reference	Assets/domain	Virtual environment	Physical connectivity	ML functionality	Embedded inference	Real-time interaction
This study	6 assets: pump, furnace, CNC mill, CNC lathe, laser, packaging/QC	Unity first-person industrial environment	No continuous sensor link; Digital Shadow-oriented	Asset-specific fault classification; KNN/SVM/RF/ANN benchmarks	ONNX + Unity Sentis, local execution	Yes; in-scene Analizar control

Building on this gap, the contributions of this work are: (i) a Unity-based virtual industrial environment representing six heterogeneous manufacturing assets through a common first-person interaction layer; (ii) a deliberately scoped Digital Shadow-oriented architecture integrating asset-specific operational variables, virtual representation, and local diagnostic functionality without claiming bidirectional physical synchronization; (iii) a reproducible ONNX/Unity Sentis deployment pathway for local, on-device fault-diagnosis inference; (iv) quantitative pump evaluation using four baseline classifiers together with repeated stratified cross-validation, matched ANN-versus-SVM Unity runtime measurements, and explicit interpretation of the deployment trade-off; (v) quantitative controlled synthetic benchmarks for the five additional assets, including dataset size, class structure, input variables, baseline model performance, and independent-generator robustness; and (vi) a public-ready reproducibility package containing the controlled datasets, generation specifications, scripts, trained models, robustness outputs, and raw runtime measurements. The quantitative results are interpreted strictly as controlled synthetic-data evidence and not as proof of field diagnostic performance.

The remainder of this paper is organized as follows. Section 2 describes the proposed architecture, the multi-asset virtual environment, the synthetic dataset, the ML models and the Unity/ONNX deployment pipeline. Section 3 reports the comparative performance of the evaluated classifiers and the multi-asset runtime behavior of the environment. Section 4 discusses the implications, scope and limitations of the proposed environment, and Section 5 concludes the paper.

2. Materials and Methods

2.1. Proposed Digital Shadow-Oriented Architecture

The proposed environment follows an eight-stage pipeline, illustrated conceptually in **Figure 1**: (1) controlled synthetic operational data definition; (2) preprocessing and training-only normalization where required; (3) baseline comparison of supervised classifiers on a stratified 80/20 split; (4) deployment-oriented model selection; (5) export of the selected model to ONNX with explicit input/output ordering; (6) execution through Unity Sentis; (7) propagation of the classification output to the corresponding digital representation; and (8) presentation of the predicted condition and model output probability on the in-scene panel. The pump is the fully documented pilot asset. The same architectural pattern is instantiated for the other five assets, while the controlled dataset-level benchmark is reported separately in Section 3.3. Because the current environment consumes offline synthetic surrogate inputs, has no continuous physical-sensor stream, and

does not actuate a physical machine, it is classified as a Digital Shadow-oriented proof of concept rather than a synchronized Digital Twin [9–12]. The repository accompanying the study records the data, scripts, models, generation specifications, and runtime measurements required to reproduce the reported computational analyses.

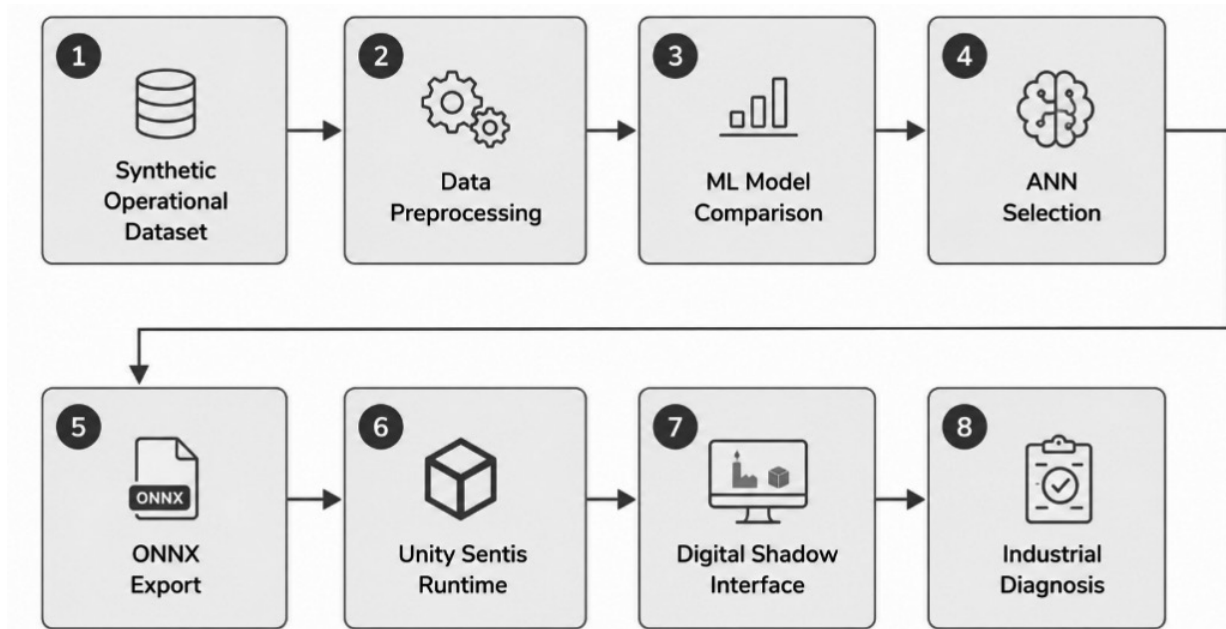


Figure 1. Overview of the proposed eight-stage Digital Shadow-oriented pipeline, from synthetic dataset generation to real-time industrial diagnosis inside the Unity environment. This pipeline is documented in full quantitative detail for the pump pilot asset and is replicated with asset-specific data and models for each additional digitally represented asset. Step 4 reflects the original pump deployment path in which the ANN was selected; the evaluation additionally deploys the top-performing SVM for the matched runtime comparison reported in Section 3.4.

2.2. Multi-Asset Virtual Industrial Environment

The Unity-based environment was developed in two sequential stages, summarized in **Figure 2**. Stage 1 established and fully documented the ML diagnostic workflow using the industrial pump as the initial pilot asset. Stage 2 extended the environment to five additional heterogeneous industrial assets, an industrial furnace, a CNC milling machine, a CNC lathe, a laser cutting station and a packaging and quality-control line, each digitally represented with its own asset-specific operational variables and coupled with a dedicated diagnostic model, enabling asset-specific in-scene inference through the same Unity/Sentis deployment mechanism used for the pump. Presenting the development in two explicit stages clarifies the chronology of the implementation and identifies the pump as the asset for which the ML pipeline is documented in full quantitative detail in this paper (dataset generation, classifier comparison, cross-validation and runtime latency, Sections 2.3–2.6), while all six digitally represented assets, pump included, are equipped with their own dedicated model, dataset and in-scene interface, verified through direct runtime execution, with representative runtime examples presented later in this section.

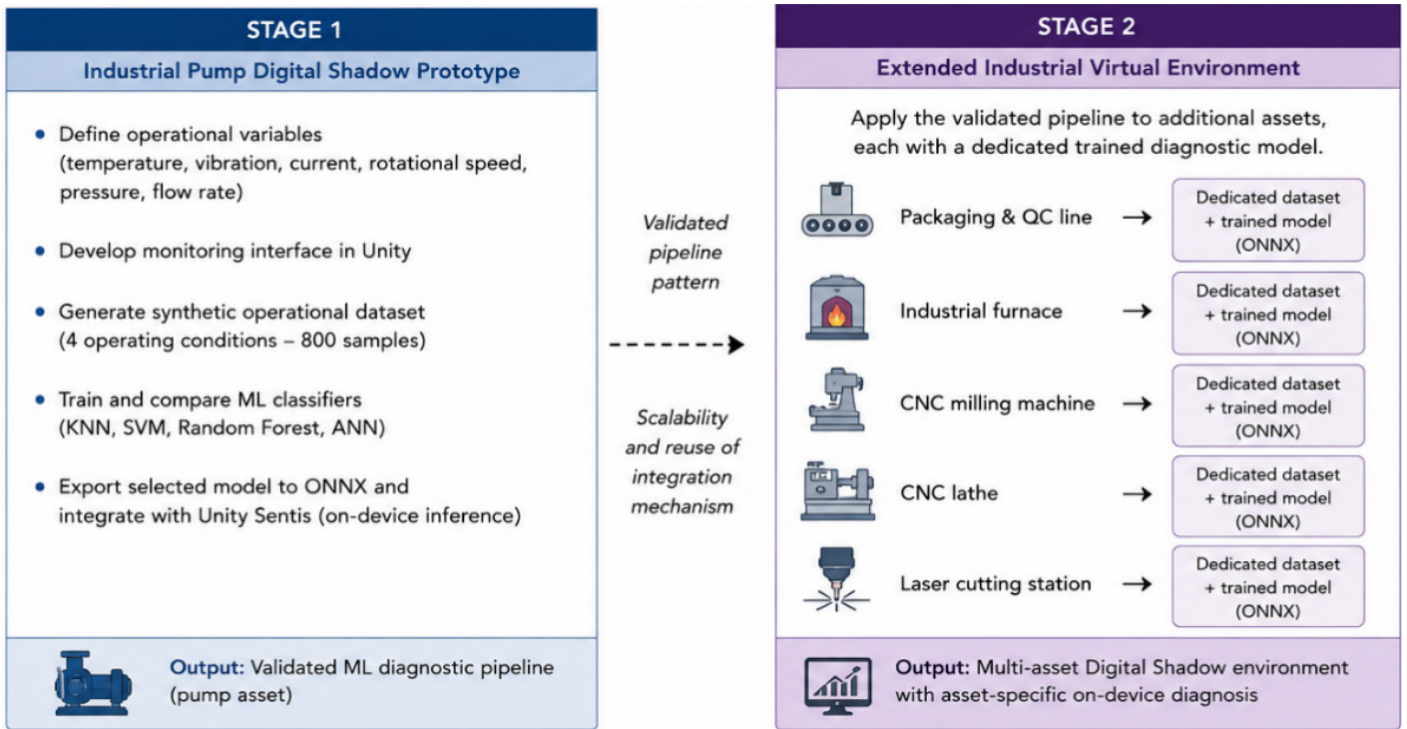


Figure 2. Two-stage development strategy of the proposed Digital Shadow-oriented environment. Stage 1 establishes and fully documents the ML diagnostic pipeline on the pump pilot asset; Stage 2 extends the environment to five additional heterogeneous industrial assets, each digitally represented and supported by its own dedicated trained diagnostic model.

2.2.1. Stage 1: Industrial Pump Digital Representation

The industrial pump was selected as the initial pilot asset for the complete workflow, and its initial digital representation and monitoring interface are shown in **Figure 3**. This first stage comprised: defining the six operational variables used for diagnosis (temperature, vibration, current, rotational speed, pressure and flow rate); developing the in-scene digital representation and monitoring interface for the pump; generating the synthetic operational dataset described in Section 2.3; training and comparatively evaluating the four ML classifiers described in Section 2.4; exporting the selected model to ONNX; and integrating the resulting model as a local inference component inside Unity through Sentis, as described in Section 2.5. The pump asset is the subject of full quantitative documentation in this paper: the classifier comparison, cross-validation robustness check, confusion matrix and runtime latency measurement reported in Section 3 correspond specifically to this pilot asset, and it is used throughout the paper as the worked example through which the complete dataset-to-deployment pipeline is described end to end.



Figure 3. Initial digital representation of the industrial pump used for validating the embedded machine-learning diagnosis workflow. The in-scene panel displays the six operational variables used as model inputs (temperature, vibration, current, RPM, pressure and flow rate) and an idle “Diagnóstico ML: Sin analizar” status prior to inference, following the interaction pattern later reused, with asset-specific variables and models, throughout Stage 2.

2.2.2. Stage 2: Extended Multi-Asset Digital Representation

After the pump-based workflow was validated end to end, the Unity project was extended into a larger, explorable, first-person industrial facility (“Recorrido Virtual”) representing an industrial furnace, CNC milling machine, CNC lathe, laser cutting station, and packaging/quality-control line. Each asset exposes an asset-specific set of operational variables and is coupled to a dedicated diagnostic component reachable through the same in-scene Analizar control. For example, the furnace panel includes furnace and refractory temperatures, chamber pressure, energy consumption, gas flow, oxygen percentage, and vibration; the CNC assets include spindle speed, vibration, current, temperature, feed-related variables, coolant pressure, cutting force, and tool-wear indicators; the laser station includes cutting speed, gas pressure, laser power, and focus deviation; and the packaging line includes product weight, line speed, sealing temperature/pressure, vibration, detected defects, and ambient humidity. In addition to functional Unity execution (Figures 4 and 5), Section 3.3 now provides a concise controlled quantitative benchmark for all six assets. The benchmark is reported separately from the Unity screenshots so that dataset-level classification evidence is not conflated with physical validation or per-asset latency measurement.



(a)



(b)



(c)



(d)

Figure 4. First-person digital representations of the Stage 2 extended Unity-based virtual industrial facility. (a) Packaging and quality-control line. (b) Industrial furnace. (c) CNC milling machine. (d) Laser cutting station. Each digitally represented asset is coupled with a dedicated diagnostic model and displays asset-specific operational variables together with an Analizar control that triggers on-device inference for that asset.

Figure 5 shows the CNC lathe asset before and after the diagnostic control is triggered, illustrating the transition from the idle panel state (“Sin analizar”) to a populated diagnosis with a predicted class and a model output probability; this is the same interaction pattern originally established for the pump asset in Stage 1 and described in Section 2.5, here applied to the CNC lathe's own dedicated model. Direct runtime execution of the remaining Stage 2 assets provides further evidence of asset-specific, embedded inference: the packaging line returns a Normal diagnosis at 99.9% model output probability, the furnace returns Normal at 89.9% model output probability, the milling machine returns Sobrecalentamiento (Overheating) at 100.0% model output probability, the CNC lathe returns Normal at 89.6% model output probability, and the laser cutter returns Normal at 100.0% model output probability. The varying predicted classes and model output probabilities across assets, together with the asset-specific variable sets visible on each panel, provide runtime evidence that each asset executes its corresponding embedded diagnostic model rather than serving as a static visual interface: for a given asset, the operational variables shown on its panel are passed to that asset's model upon triggering Analizar, producing a predicted class and an associated model output probability specific to that asset.



(a)



(b)

Figure 5. Diagnostic panel interaction pattern, shown here for the CNC lathe asset in the Stage 2 environment. (a) Idle state of the monitoring panel prior to inference. (b) Panel state after the Analizar control is triggered, displaying the predicted class and model output probability returned by the CNC lathe's own dedicated on-device inference model.

2.2.3. Interaction Design and Intended Users

Interaction follows a task-oriented sequence: the user navigates the facility in first person, approaches an asset, inspects its asset-specific operational panel, triggers the Analizar control, and receives the predicted operating-condition label together with an uncalibrated model output probability. The intended users are engineering students, researchers, maintenance trainees, and technical personnel who need to explore diagnostic logic in a controlled virtual setting. This interaction design is consistent with prior industrial VR work showing the value of immersive guidance and fault-oriented training [20,23], while the present study concentrates on technical integration rather than formal usability validation. No SUS, task-completion, workload, or user-performance study was conducted for this multi-asset prototype; therefore, usability effectiveness is not claimed and remains a separate future evaluation objective.

2.3. Controlled Synthetic Operational Dataset

The retained pump dataset contains 800 controlled synthetic records representing four operating conditions: Normal, Mechanical Failure, Electrical Failure, and Overheating (200 records per class). Each record contains six continuous variables: temperature (°C), vibration (mm/s), electrical current (A), rotational speed (RPM), discharge pressure (bar), and flow rate (m³/h), with no missing values. **Table 2** reports the realized class-conditional descriptive statistics and **Figure 6** shows the observed distributions. The dataset file itself is preserved in the reproducibility repository and is the canonical source for the original pump results reported in this paper.

Table 2. Descriptive statistics (mean $\pm$ standard deviation) of the six operational variables, by class, for the synthetic pump dataset (n = 800; 200 records per class).

Class	Temp. (°C)	Vibration (mm/s)	Current (A)	RPM	Pressure (bar)	Flow (m ³ /h)
Normal	62.16 $\pm$ 5.91	2.43 $\pm$ 0.72	12.50 $\pm$ 1.65	1471.42 $\pm$ 37.70	5.81 $\pm$ 0.46	41.97 $\pm$ 4.69
Mechanical Failure	74.76 $\pm$ 8.56	8.05 $\pm$ 1.70	15.03 $\pm$ 2.02	1321.68 $\pm$ 92.42	4.67 $\pm$ 0.66	30.45 $\pm$ 5.73
Electrical Failure	76.79 $\pm$ 8.68	3.63 $\pm$ 0.93	23.94 $\pm$ 3.52	1389.12 $\pm$ 98.81	5.21 $\pm$ 0.69	35.13 $\pm$ 5.39
Overheating	100.86 $\pm$ 6.89	4.50 $\pm$ 1.25	17.69 $\pm$ 2.95	1407.49 $\pm$ 60.90	5.39 $\pm$ 0.67	37.27 $\pm$ 5.21

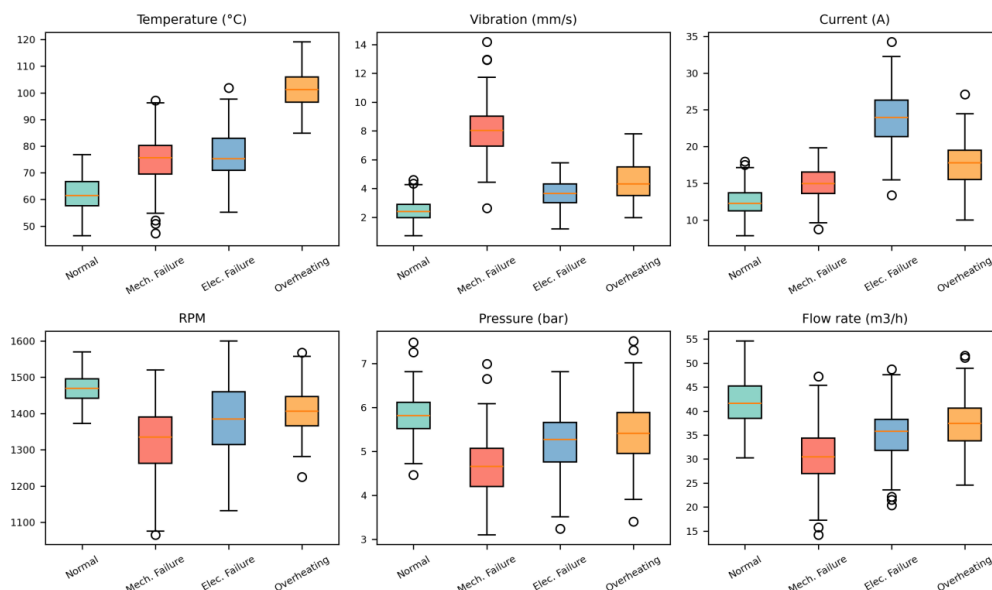


Figure 6. Distribution of the six operational variables of the synthetic pump dataset, grouped by health-state class.

2.3.1. Synthetic Dataset Generation Procedure

The retained pump dataset follows a class-conditional Gaussian design of the general form shown in Equation (1), in which operational-condition-specific centers and dispersions define distinguishable synthetic regimes. For reproducibility, the retained pump CSV is provided in the repository as the canonical artifact for the primary pump classification results. A fully parameterized controlled generator was created for the independent-generator robustness analysis in Section 2.4.2 using documented class-conditional values, explicit lower/upper bounds, rounding rules, and fixed seeds. The full generator specification is supplied in machine-readable JSON and CSV form in the accompanying repository.

$$xc = \mu c + \sigma c \varepsilon, \quad \varepsilon \sim N(0, I) \quad (1)$$

where μc denotes the class-conditional reference mean, σc denotes the corresponding dispersion, and ε is sampled from a standard Gaussian distribution. Equation (1) describes the synthetic-generation family used in the controlled benchmarks. The observed means and standard deviations of the retained pump dataset are reported in **Table 2**, while the complete parameterization used for the reproducible independent-generator experiments is provided in the accompanying repository.

The retained pump data exhibit intentionally distinguishable signatures: Mechanical Failure is associated with elevated vibration and reduced speed/flow, Electrical Failure with increased current, and Overheating with elevated temperature. Such separability is useful for testing the software and deployment pipeline, but it also makes high classification accuracy expected. Accordingly, all accuracy and F1 values in this work quantify discrimination within controlled synthetic distributions and must not be interpreted as validated field diagnostic capability. Real industrial data would introduce sensor noise, drift, mixed faults, operating-point changes, class imbalance, maintenance history, and domain shift that are not represented by the present generator.

2.4. Machine Learning Model Development and Evaluation

Four supervised classifiers were evaluated as baseline models: K-Nearest Neighbors (KNN), radial-basis-function Support Vector Machine (SVM), Random Forest, and a feedforward Artificial Neural Network (ANN). For the canonical pump evaluation, the retained dataset was split into 640 training records (80%) and 160 test records (20%) using stratified sampling with `random_state = 42`. KNN and SVM inputs were standardized using statistics fitted on the training partition only; Random Forest used the unscaled variables. The purpose of this comparison is baseline screening under identical data conditions, not exhaustive hyperparameter optimization. KNN used $k = 5$, SVM used an RBF kernel with $C = 1$ and $\gamma = \text{scale}$, and Random Forest used 100 trees. The ANN architecture and training settings are reported below. This framing avoids interpreting differences between baseline settings as evidence of globally optimal classifier selection.

The pump ANN was implemented as a fully connected PyTorch multilayer perceptron with six inputs, hidden layers of 16 and 8 units with ReLU activations, and four output logits (**Figure 7**). Training-set normalization was embedded inside the network as non-trainable buffers so that raw input values could be passed from Unity. The network was trained for 250 epochs using Adam (learning rate = 0.01), cross-entropy loss, and `torch.manual_seed(42)`. The model was exported to ONNX opset 17. The historical held-out performance reported for this deployed ANN is preserved

from the retained experiment artifacts (97.50% accuracy; 97.48% macro-F1). This deployed PyTorch ANN is distinct from the scikit-learn MLPClassifier used as the uniform ANN baseline in the reproducible multi-asset benchmark; the latter yields 95.625% accuracy for the pump and should not be interpreted as a repeated evaluation of the same trained ANN.

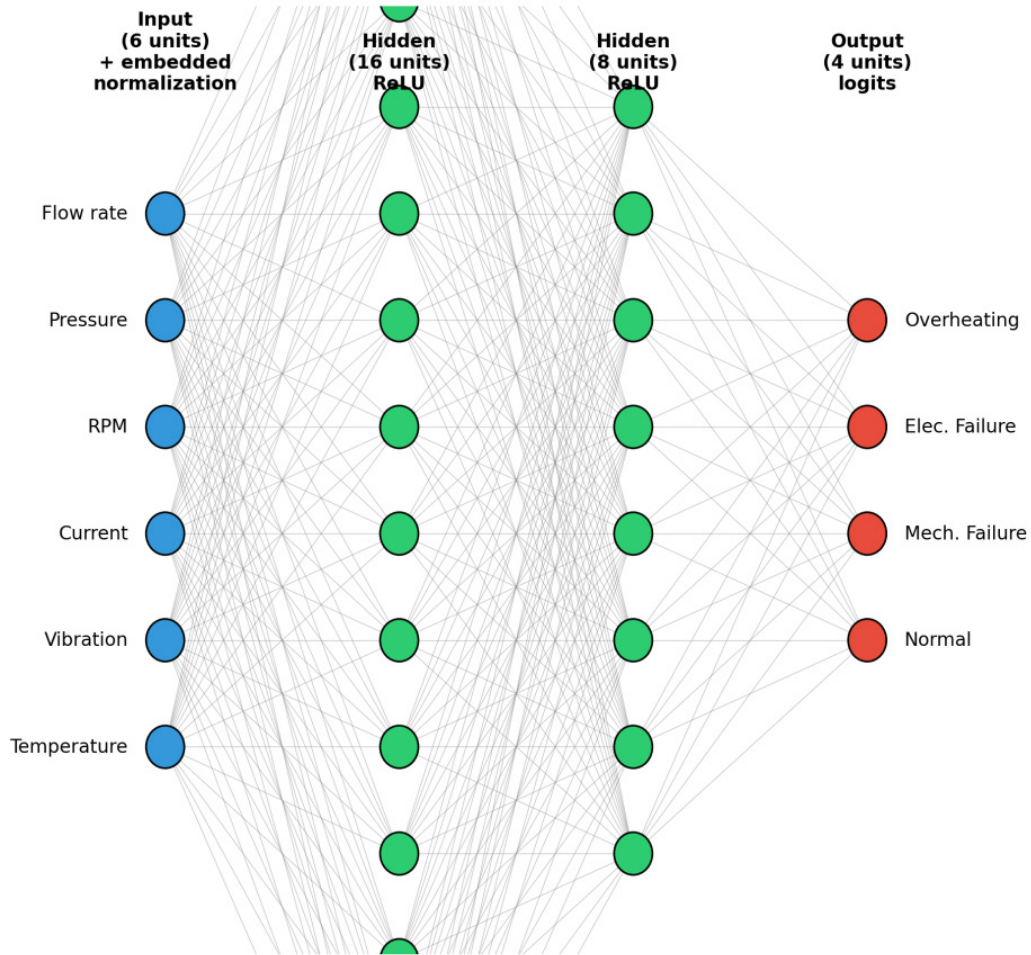


Figure 7. Architecture of the feedforward artificial neural network used for pump operating-condition classification, with input normalization embedded inside the model graph.

KNN, SVM, and Random Forest were evaluated on the same stratified split as the ANN using the baseline settings stated above. Standardization was fitted on the training data only for KNN and SVM, and no feature scaling was applied to Random Forest. Because the comparison is intentionally a baseline screening experiment, no claim is made that any of the four configurations is globally hyperparameter-optimal. This choice is revisited in the limitations.

It is important to note that the accuracy, precision, recall and F1-score values reported for all four classifiers in Section 3 quantify each model's ability to separate the statistical patterns embedded in the controlled synthetic dataset described in Section 2.3; they do not constitute empirical validation of diagnostic performance under real industrial operating conditions, a distinction maintained throughout this paper and discussed further in Section 4.

2.4.1. Cross-Validation Robustness Check

The single 80/20 split provides a directly interpretable held-out comparison

but remains sensitive to partition choice. A complementary repeated stratified 5-fold cross-validation was therefore performed on the retained pump dataset using five random seeds (25 folds total) with the same feature set and per-model preprocessing. **Table 3** reports mean $\pm$ sample standard deviation across folds. This resampling analysis evaluates split sensitivity within the retained dataset; it does not evaluate uncertainty in the synthetic data-generation process itself. Generator-level uncertainty is addressed separately in Section 2.4.2.

Table 3. Stratified 5-fold cross-validation results (5 random seeds, 25 folds total) for the four evaluated classifiers, reported as mean $\pm$ sample standard deviation across folds.

Model	CV Accuracy (%)	CV F1-score (%)
K-Nearest Neighbors	96.42 $\pm$ 1.22	96.42 $\pm$ 1.22
Support Vector Machine	96.75 $\pm$ 1.21	96.74 $\pm$ 1.22
Random Forest	95.92 $\pm$ 1.40	95.91 $\pm$ 1.41
Artificial Neural Network	95.92 $\pm$ 1.48	95.92 $\pm$ 1.48

2.4.2. Independent-Generator Robustness Analysis

To assess uncertainty associated with synthetic-data generation rather than only train/test resampling, five independently sampled controlled datasets were created for each asset using seeds 101, 202, 303, 404, and 505. Each dataset contained 800 records (200 per class), and the train/test split used the corresponding fixed seed. For the pump robustness analysis, the documented parameterized generator described above was used, while the retained pump CSV remained the basis of the primary historical pump results. For the five additional assets, the documented asset-specific generator specifications were used to create the independent datasets. Accuracy and macro-F1 mean $\pm$ sample standard deviation across the five independent datasets are reported in Section 3.5 for the strongest mean-performing baseline per asset; full per-model/per-seed results are available in the repository.

2.5. ONNX Export and Unity Sentis Integration

Following training, the pump ANN was exported to ONNX (opset 17) with a single input tensor of shape (batch, 6) and four output logits. A companion metadata file fixes the ordering of the six input variables and four output classes. The graph was checked before deployment. To directly test the deployment trade-off identified by the baseline classification results, the best-performing pump SVM was also converted to ONNX and executed in Unity under the same end-to-end runtime protocol used for the ANN. The purpose of this additional deployment experiment was not to replace the ANN implementation, but to quantify whether the SVM's modest accuracy advantage translated into an equal or better interactive runtime profile.

The validated ONNX models were imported into Unity and executed locally through Sentis on the CPU backend. For each inference request, a 1×6 tensor is constructed from the current pump variables, the corresponding model is executed, and the output is mapped to the predicted operating-condition label. For the ANN, logits are transformed with softmax in C#. The displayed percentage is referred to throughout the revised manuscript as a model output probability rather than

diagnostic confidence because no calibration procedure was applied to the deployed Unity models. High output probabilities therefore indicate model decisiveness within the controlled synthetic distribution, not empirically calibrated diagnostic reliability on physical machinery. Supplementary ECE and multiclass Brier diagnostics for the reproducible baseline models are included in the repository but are not used to claim calibration of the historical deployed ONNX ANN.

2.6. Computational Environment and Runtime Latency Measurement

The Unity Digital Shadow-oriented environment was developed and executed on a single laptop workstation, whose specifications are summarized in **Table 4**. The system combines a mobile high-core-count CPU with a discrete GPU, representative of the kind of workstation available in a university research laboratory rather than dedicated industrial edge hardware, which is relevant when interpreting the runtime figures reported below.

End-to-end latency was measured inside Unity under a matched protocol of 30 executions per deployed pump model, from inference request to diagnostic visualization. For the ANN, the 30 measurements were 19, 23, 22, 23, 24, 20, 24, 23, 22, 24, 18, 20, 22, 24, 22, 22, 20, 22, 21, 20, 21, 24, 22, 24, 23, 22, 22, 24, 23, and 20 ms, yielding 22.00 ± 1.66 ms. For the converted SVM, the 30 measurements were 23, 27, 21, 29, 24, 26, 22, 28, 25, 25, 20, 30, 19, 31, 24, 26, 22, 28, 23, 27, 21, 29, 25, 25, 20, 30, 24, 26, 22, and 28 ms, yielding 25.00 ± 3.28 ms. These measurements include tensor construction, model execution, output retrieval/post-processing, and UI update rather than isolated mathematical inference cost. **Table 4** summarizes the workstation; the matched ANN-SVM runtime comparison is reported later in Section 3.4. In the present click-triggered training/monitoring prototype, both measured latencies support responsive local interaction; however, future sensor acquisition, preprocessing, networking, protocol translation, database access, and physical-system synchronization could add substantially more latency and are outside the present measurement. **Figure 8** visualizes the distribution of the 30 ANN end-to-end response-time measurements summarized above.

Table 4. Computational environment used for the Unity Digital Shadow-oriented environment. Runtime results for both deployed pump models are compared separately in Section 3.4.

Parameter	Value
System	Alienware m16 R2
Operating system	Windows 11 Home (64-bit)
CPU	Intel Core Ultra 9 185H (22 logical processors)
RAM	32 GB
GPU	NVIDIA GeForce RTX 4070 Laptop GPU (approx. 8 GB VRAM)
Integrated GPU	Intel Arc Graphics
Graphics API	DirectX 12
Model format	ONNX (opset 17)
Inference framework	Unity Sentis
Runtime protocol	Matched end-to-end Unity measurements; 30 executions per model

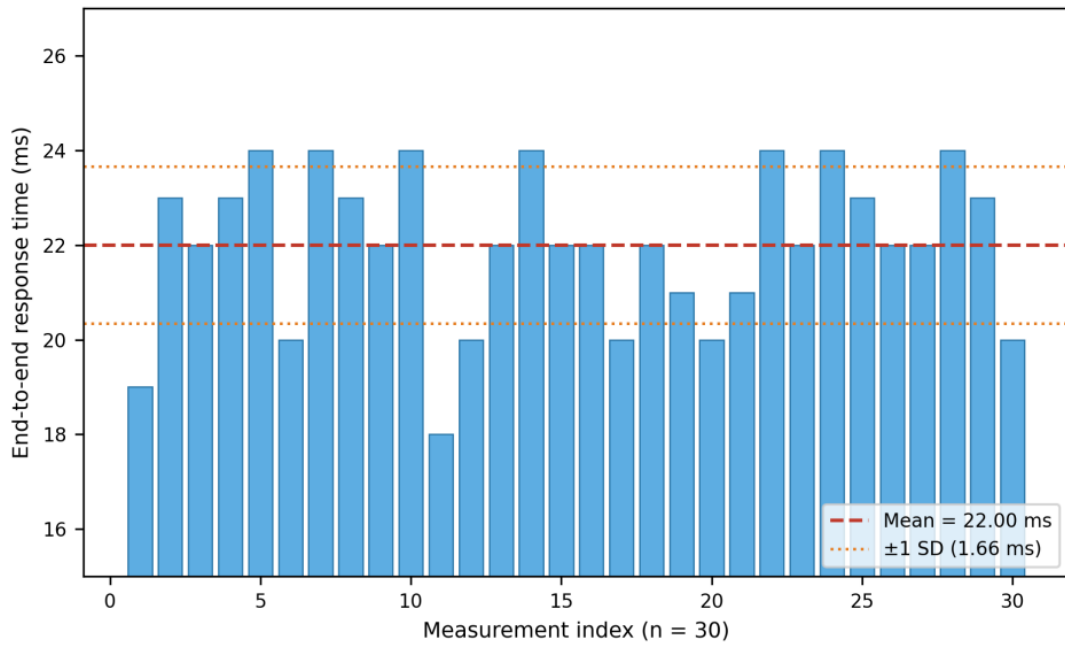


Figure 8. Distribution of the 30 independent end-to-end response-time measurements of the Unity diagnostic-inference cycle, with the sample mean (22.00 ms) and ± 1 standard deviation band (1.66 ms) indicated.

3. Results and Discussion

This section reports the controlled quantitative results of the pump baseline comparison, the multi-asset benchmark, independent-generator robustness, and matched Unity runtime deployment experiment. The results demonstrate software-pipeline behavior and discrimination within defined synthetic distributions. They are not interpreted as evidence of diagnostic performance on physical industrial machinery.

3.1. Comparative Performance of the Evaluated Classifiers

Table 5 summarizes the held-out pump results. SVM achieved the highest baseline accuracy (98.12%) and macro-F1 (98.12%), followed by the historically deployed ANN (97.50% accuracy; 97.48% F1), while KNN and Random Forest each reached 96.25% accuracy. Because the class-conditional synthetic distributions were intentionally distinguishable, these high values are expected and should be read as controlled pipeline-separation results. Repeated stratified cross-validation (**Table 3**) preserves SVM as the strongest mean performer on the retained pump dataset, while Section 3.4 examines whether that advantage persists as a deployment advantage inside Unity. The comparative held-out metrics are visualized in **Figure 9**.

Table 5. Baseline performance of the four pump classifiers on the retained held-out test set (160 records). Hyperparameters were not exhaustively optimized.

Model	Acc. (%)	Prec. (%)	Rec. (%)	F1 (%)	Baseline deployment note	ONNX status
K-Nearest Neighbors	96.25	96.31	96.25	96.26	Baseline classifier; full training set retained at inference	Convertible; not deployed in matched test
Support Vector Machine	98.12	98.17	98.12	98.12	Highest pump baseline accuracy; kernel-based inference	Converted and deployed for matched runtime test

Continuation Table:

Model	Acc. (%)	Prec. (%)	Rec. (%)	F1 (%)	Baseline deployment note	ONNX status
Random Forest	96.25	96.28	96.25	96.22	Tree ensemble baseline	Convertible; not deployed in matched test
Artificial Neural Network	97.50	97.51	97.50	97.48	Compact fixed-size neural inference; original deployment	Native export; deployed via Sentsis

Note: Training time is not reported quantitatively, since wall-clock training time was not systematically benchmarked in this study.

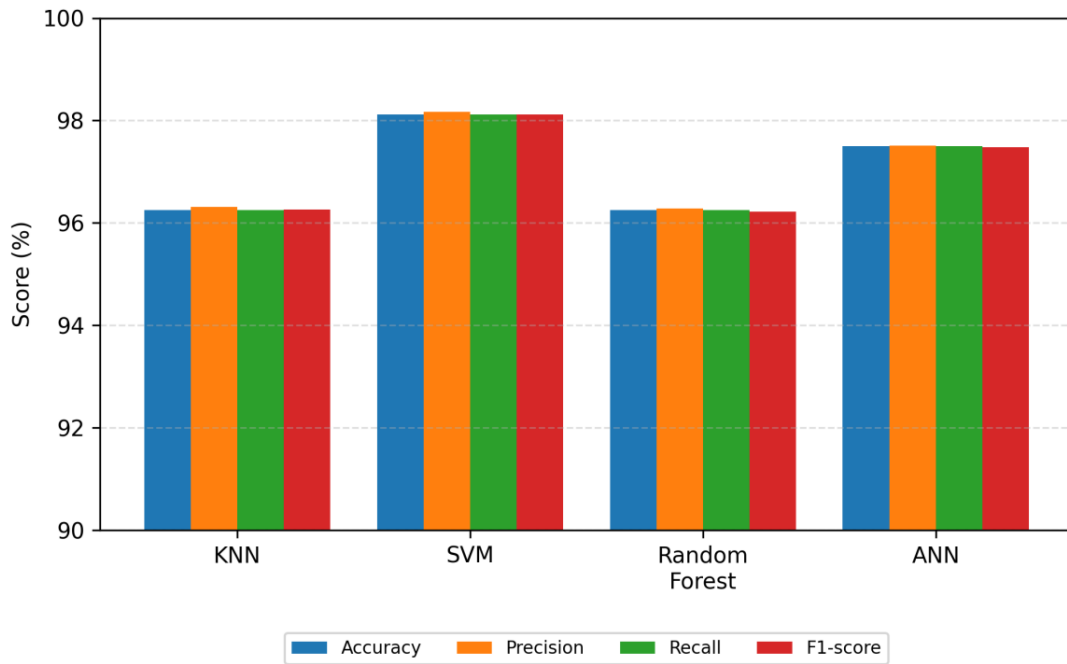


Figure 9. Comparative accuracy, precision, recall and F1-score of the four evaluated classifiers on the held-out test set.

Although SVM attained the highest pump baseline classification metrics, model selection cannot be justified from accuracy alone. The revised study therefore includes a direct matched Unity deployment comparison. SVM retained a 0.62 percentage-point accuracy advantage over the ANN (98.12% versus 97.50%), but its mean end-to-end Unity latency was 25.00 ± 3.28 ms compared with 22.00 ± 1.66 ms for the ANN, as quantified later in Section 3.4. Thus, the ANN reduced mean response time by 12% relative to the SVM and showed lower execution-time variability, while the SVM provided slightly higher classification accuracy. The result supports an explicit accuracy-runtime trade-off; it does not establish that the ANN is universally superior. Native ONNX export and normalization embedded in the ANN graph remain practical implementation advantages, but the deployment choice is now supported by measured runtime evidence rather than convenience alone.

3.2. Confusion Matrix of the Deployed Model

Figure 10 shows the confusion matrix of the deployed ANN classifier on the test partition. Of the 160 test records, 156 were correctly classified. The Mechanical Failure and Electrical Failure classes were classified without error (40/40 each), while three Overheating records were misclassified, two as Normal and one as Mechanical Failure, and one Normal record was misclassified as Overheating. This pattern is

consistent with the partial statistical overlap between the Normal and Overheating classes observed in the temperature and flow-rate distributions in **Figure 6**.

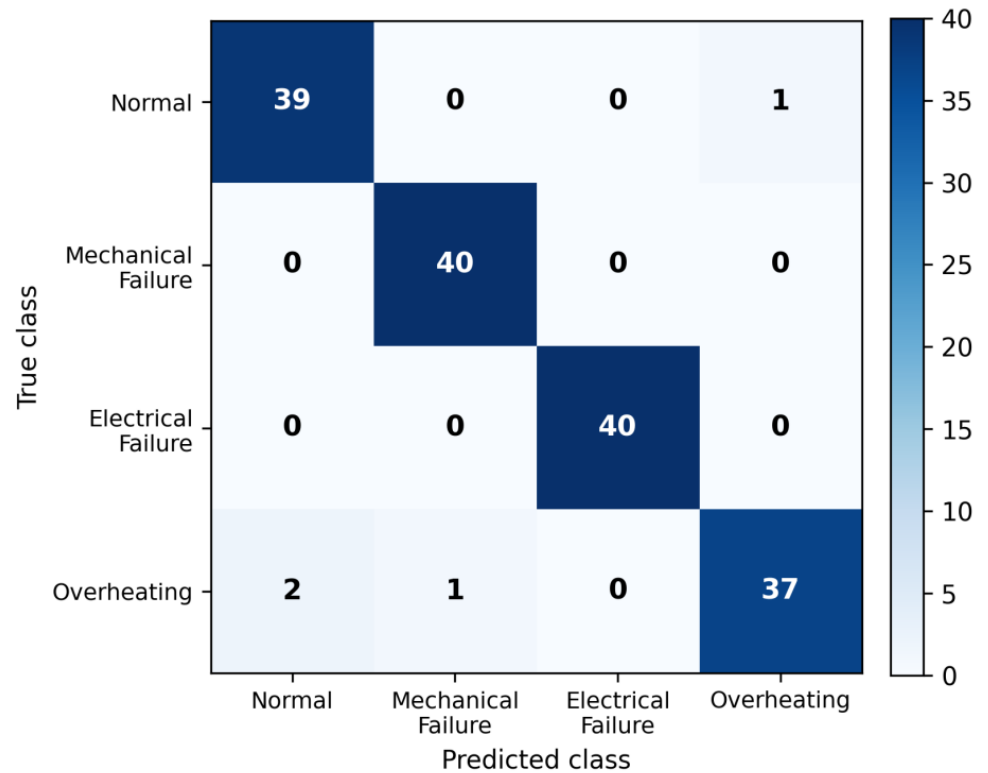


Figure 10. Confusion matrix of the deployed artificial neural network classifier on the held-out test set (160 records).

3.3. Multi-Asset Quantitative Benchmark and Functional Unity Execution

To address the imbalance between the detailed pump case and the five additional assets, each asset was evaluated using a controlled synthetic benchmark of 800 records (200 per class) with an 80/20 stratified holdout and the same four baseline model families. **Table 6** reports the complete asset scope, input variables, class count, strongest held-out baseline, accuracy/F1, and the status of Unity execution. Across the five additional assets, the strongest held-out accuracy ranged from 94.38% (furnace and packaging/QC) to 98.12% (CNC lathe). These values provide quantitative evidence that the asset-specific pipelines can discriminate their defined synthetic operating regimes; they do not demonstrate field accuracy or cross-machine generalization.

Table 6. Controlled multi-asset benchmark. “Best baseline” refers to the strongest held-out classifier in the reproducible dataset benchmark and should not be conflated with physical validation or with the exact model shown in every Unity screenshot.

Asset	Input variables	n / classes	Best baseline	Acc. / F1 (%)	Unity status / latency
Pump	Temp.; vibration; current; RPM; pressure; flow	800 / 4	SVM	98.12 / 98.12	ANN and SVM deployed; matched latency in Table 7
Furnace	Furnace temp.; refractory temp.; chamber pressure; energy; gas flow; O2; vibration	800 / 4	KNN	94.38 / 94.37	Dedicated model functionally verified; latency not separately benchmarked

Continuation Table:

Asset	Input variables	n / classes	Best baseline	Acc. / F1 (%)	Unity status / latency
CNC milling	Spindle RPM; vibration; current; temp.; feed; depth; coolant pressure; cutting force; tool wear	800 / 4	KNN	95.00 / 94.98	Dedicated model functionally verified; latency not separately benchmarked
CNC lathe	Temp.; vibration; current; spindle RPM; feed; coolant pressure; cutting force; tool wear	800 / 4	Random Forest	98.12 / 98.12	Dedicated model functionally verified; latency not separately benchmarked
Laser cutter	Temp.; vibration; current; cutting speed; gas pressure; laser power; focus deviation	800 / 4	Random Forest	97.50 / 97.50	Dedicated model functionally verified; latency not separately benchmarked
Packaging/QC	Product weight; line speed; seal temp.; seal pressure; vibration; detected defects; humidity	800 / 4	SVM	94.38 / 94.40	Dedicated model functionally verified; latency not separately benchmarked

Functional Unity execution was separately verified through the in-scene Analizar control for all represented assets. The screenshots show asset-specific variable sets and distinct diagnostic outputs, confirming that the panels are not static mock-ups. The displayed percentages are now described as model output probabilities because the deployed outputs were not formally calibrated. For the five additional assets, these runtime demonstrations establish implementation functionality only; per-asset end-to-end latency was not separately benchmarked. This distinction between quantitative dataset benchmarking, functional runtime demonstration, and physical validation is maintained throughout the Discussion.

3.4. Matched ANN-SVM Unity Runtime Comparison

The matched pump deployment experiment shows that the SVM's classification advantage did not translate into lower Unity response time. **Table 7** summarizes the 30-run end-to-end distributions. The ANN was faster on average by 3.00 ms (12% relative reduction from the SVM mean) and exhibited approximately half the sample standard deviation. Both models remained fast enough for the present click-triggered interaction, but these values reflect only the current local workstation pipeline and exclude future sensor/network delays.

Table 7. Matched end-to-end Unity runtime comparison for the deployed pump ANN and SVM (30 executions per model).

Model	Pump accuracy (%)	Pump F1 (%)	n	Mean $\pm$ SD (ms)	Median (ms)	Range (ms)
ANN	97.50	97.48	30	22.00 $\pm$ 1.66	22.00	18-24
SVM	98.12	98.12	30	25.00 $\pm$ 3.28	25.00	19-31

3.5. Robustness Across Independently Sampled Synthetic Datasets

Table 8 summarizes the strongest mean-performing baseline for each asset across five independently sampled datasets. The between-dataset standard deviations show that model performance varies when the synthetic draw changes, even with

a fixed generator specification. This analysis is stronger than repeated folds on a single dataset because it explicitly samples generator uncertainty; nevertheless, it still quantifies robustness only within the defined synthetic distributions.

Table 8. Independent-generator robustness summary across five independently created controlled datasets per asset.

Asset	Strongest mean baseline	Accuracy mean $\pm$ SD (%)	F1 mean $\pm$ SD (%)	Independent seeds
Pump	SVM	96.13 $\pm$ 1.35	96.11 $\pm$ 1.37	101, 202, 303, 404, 505
Furnace	SVM	95.25 $\pm$ 1.05	95.24 $\pm$ 1.05	101, 202, 303, 404, 505
CNC milling	SVM	95.13 $\pm$ 1.28	95.13 $\pm$ 1.29	101, 202, 303, 404, 505
CNC lathe	Random Forest	95.50 $\pm$ 1.62	95.50 $\pm$ 1.62	101, 202, 303, 404, 505
Laser cutter	SVM	96.50 $\pm$ 2.36	96.49 $\pm$ 2.37	101, 202, 303, 404, 505
Packaging/QC	SVM	94.13 $\pm$ 2.10	94.13 $\pm$ 2.09	101, 202, 303, 404, 505

4. Discussion

4.1. Integration of Intelligent Analytics into a Multi-Asset Virtual Industrial Environment

Taken together, the results support an architectural integration claim rather than a field-diagnostic superiority claim. The system combines six asset representations, asset-specific operational inputs, embedded diagnostic logic, ONNX/Sentis execution, and a common first-person interaction pattern. Quantitative evidence is now available for all six controlled datasets (**Table 6**), while runtime deployment is directly benchmarked for the pump ANN and SVM (**Table 7**). This positioning aligns the work more closely with Industrial Metaverse and intelligent virtual-environment research, where virtual representation, interaction, AI, and physical-virtual data pathways are complementary components [4–6]. The present prototype implements several of these components, but it lacks persistence across distributed users, continuous physical connectivity, and bidirectional actuation; it is therefore better viewed as a reproducible intelligent industrial virtual-environment building block than as a complete Industrial Metaverse implementation.

4.2. Why Digital Shadow and Not a Synchronized Digital Twin

The environment is deliberately positioned as a Digital Shadow-oriented proof of concept rather than a synchronized Digital Twin. Under the taxonomy of Kritzinger et al. [9], the absence of continuous automated physical-to-virtual sensing and virtual-to-physical actuation precludes a full Digital Twin claim. Synthetic records currently stand in for the upstream operational data source. A future physical deployment could replace this surrogate source with sensor streams while preserving much of the software pipeline, but equivalent classifier accuracy cannot be assumed because real data would introduce domain shift, noise, drift, mixed operating states, missing values, asynchronous sampling, and class imbalance. Thus, migration to sensors is a proposed capability pathway, not a validated performance transfer.

4.3. Advantages of Local Deployment via ONNX and Unity Sentis

Local ONNX/Sentis execution removes dependency on an external inference server and supports deterministic integration with the Unity interaction loop. The matched experiment now provides quantitative evidence for the model-selection trade-

off: the SVM produced slightly higher pump baseline accuracy, whereas the ANN produced lower and less variable end-to-end Unity latency. Because the difference is small in absolute terms, the engineering conclusion is not that neural inference is universally faster; rather, both models are viable for the present click-triggered scenario, and deployment choice depends on the relative importance of accuracy, serialization/toolchain simplicity, normalization handling, model complexity, and runtime stability. Future comparisons should additionally report ONNX model size, backend-specific CPU/GPU behavior, memory use, and performance on dedicated edge hardware.

4.4. Modularity, Extensibility, and Pathway Toward Physical Sensor Integration

The environment has practical value as a teaching, research, and maintenance-training demonstrator. Users can move through the virtual facility, inspect machine-specific variables, trigger a diagnostic analysis, and observe how changes in operational inputs map to model outputs. Prior immersive industrial-training research, including induction-motor fault guidance, suggests that task-oriented VR can support procedural learning and user engagement [20,23], while industrial teleoperation research illustrates how distributed control architectures can later connect virtual interfaces to physical processes [24]. In the present work, however, interaction was not evaluated with a formal user study. A next-stage evaluation should define representative tasks, target user groups, completion time/error metrics, usability scales such as SUS, and workload or learning measures. Physical sensor integration would additionally require acquisition protocols, timestamp synchronization, preprocessing, missing-data handling, safety constraints, and communication layers such as OPC UA or MQTT; these components may add delays that are absent from the reported local runtime measurements.

4.5. Limitations

Several limitations bound the interpretation of this work. First, all quantitative datasets are controlled synthetic data; the high accuracies reflect separability within designed statistical regimes and do not establish diagnostic validity on physical machinery. Second, the retained pump CSV is used as the canonical artifact for the primary pump results, while a fully documented parameterized generator is used for the reproducible multi-asset and independent-dataset analyses. Third, the classifier comparison is a baseline screening rather than exhaustive hyperparameter optimization. Fourth, model output probabilities shown in Unity are not calibrated probabilities and should not be interpreted as empirical diagnostic confidence. Fifth, only the pump ANN and SVM were subjected to matched end-to-end latency benchmarking; the other five assets were functionally executed in Unity but were not separately timed. Sixth, no formal usability study was conducted. Seventh, the current environment has no continuous sensor connection, physical actuation, or closed-loop synchronization. Finally, performance under real sensor noise, drift, unseen fault modes, different machines, network delays, and edge hardware remains unknown. These limitations distinguish what has been quantitatively evaluated, what has been functionally demonstrated, and what remains future work.

5. Conclusion

This paper presented a Unity-based, Digital Shadow-oriented multi-asset industrial virtual environment with local ML fault-diagnosis functionality through

ONNX and Unity Sentis. The evidence extends beyond the pump pilot through controlled quantitative benchmarks for the furnace, CNC milling machine, CNC lathe, laser cutter, and packaging/QC line, together with robustness analysis across independently sampled synthetic datasets. For the pump, SVM achieved the highest retained held-out accuracy (98.12%), whereas the deployed ANN achieved 97.50%. A matched 30-run Unity comparison showed 22.00 ± 1.66 ms for the ANN and 25.00 ± 3.28 ms for the SVM, demonstrating a concrete accuracy-runtime trade-off. The contribution is therefore the reproducible integration of heterogeneous asset representations, controlled diagnostics, embedded inference, and interactive visualization, not proof of industrial fault-diagnosis accuracy. The platform remains a Digital Shadow-oriented proof of concept because its inputs are synthetic surrogates and no bidirectional physical connection is present. Future work should validate the complete pipeline on continuously acquired sensor data, calibrate probabilistic outputs, evaluate users and task performance, compare deployment on dedicated edge hardware, and quantify communication/synchronization delays in a sensor-connected implementation. A reproducibility package containing the data, parameterized generators, scripts, models, robustness outputs, and raw Unity latency measurements accompanies the study.

Author contributions: Conceptualization, JEN; methodology, JEN; software, JEN; validation, JEN; formal analysis, SACB; investigation, SACB; resources, JEN; data curation, JNC; writing—original draft preparation, JEN; writing—review and editing, JEN, SACB, JNC; visualization, SACB, JNC; supervision, JEN; project administration, JEN. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: The authors gratefully acknowledge the Universidad Técnica de Cotopaxi (UTC) and the AIOLOS Research Group for their institutional, academic, and research support. This study was conducted within the framework of the research project “Transformación de la Capacitación en Seguridad y Producción Industrial con Realidad Mixta e Inteligencia Artificial en el Marco de la Industria 4.0 y los ODS”, approved by the Universidad Técnica de Cotopaxi under the UTC-2025 Call for Academic Scientific Research, Outreach and Educational Innovation Projects.

Conflicts of interest: The authors declare no conflicts of interest.

Data availability: The reproducibility materials supporting this study are publicly available at <https://github.com/josenaranjo199423-bot/unity-multiasset-digital-shadow-ml>. The repository contains the controlled synthetic datasets for all six assets, documented generator specifications (including means, standard deviations, bounds, rounding rules, and seeds for the reproducible benchmarks), evaluation scripts, trained model artifacts, retained pump workflow files, independent-dataset robustness outputs, and raw ANN/SVM Unity latency measurements. The repository supports computational reproducibility but does not contain measurements from physical industrial machines.

References

1. Kagermann H, Wahlster W, Helbig J. Recommendations for Implementing the Strategic Initiative Industrie 4.0: Final Report of the Industrie 4.0 Working Group. Acatech: Frankfurt, Germany, 2013.
2. Berg LP, Vance JM. Industry use of virtual reality in product design and manufacturing: A survey. *Virtual Reality*. 2017, 21:

- 1–17. <https://doi.org/10.1007/s10055-016-0293-9>
3. Choi S, Jung K, Noh SD. Virtual reality applications in manufacturing industries: Past research, present findings, and future directions. *Concurrent Engineering*. 2015, 23: 40–63. <https://doi.org/10.1177/1063293X14568814>
4. Ren L, Dong J, Zhang L, et al. Industrial Metaverse for Smart Manufacturing: Model, Architecture, and Applications. *IEEE Transactions on Cybernetics*. 2024, 54(5): 2683–2695. <https://doi.org/10.1109/TCYB.2024.3372591>.
5. Guo J, Leng J, Zhao JL, et al. Industrial metaverse towards Industry 5.0: Connotation, architecture, enablers, and challenges. *Journal of Manufacturing Systems*. 2024, 76: 25–42. <https://doi.org/10.1016/j.jmsy.2024.07.007>
6. Tu X, Ala-Laurinaho R, Yang C, et al. Architecture for data-centric and semantic-enhanced industrial metaverse: Bridging physical factories and virtual landscape. *Journal of Manufacturing Systems*. 2024, 74: 965–979. <https://doi.org/10.1016/j.jmsy.2024.05.016>.
7. Tao F, Zhang H, Liu A, et al. Digital twin in industry: State-of-the-art. *IEEE Transactions on Industrial Informatics*. 2018, 15: 2405–2415. <https://doi.org/10.1109/TII.2018.2873186>
8. Grieves M, Vickers J. Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. In: *Transdisciplinary Perspectives on Complex Systems*. Kahlen FJ, Flumerfelt S, Alves A (editors). Springer: Cham, Switzerland, 2017, 85–113. https://doi.org/10.1007/978-3-319-38756-7_4
9. Kritzinger W, Karner M, Traar G, et al. Digital twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine*. 2018, 51: 1016–1022. <https://doi.org/10.1016/j.ifacol.2018.08.474>
10. Wright L, Davidson S. How to tell the difference between a model and a digital twin. *Advanced Modeling and Simulation in Engineering Sciences*. 2020, 7, 13. <https://doi.org/10.1186/s40323-020-00147-4>
11. van Dinter R, Tekinerdogan B, Catal C. Predictive maintenance using digital twins: A systematic literature review. *Information and Software Technology*. 2022, 151, 107008. <https://doi.org/10.1016/j.infsof.2022.107008>
12. AboElHassan A, Sakr AH, Yacout S. General purpose digital twin framework using digital shadow and distributed system concepts. *Computers & Industrial Engineering*. 2023, 183, 109534. <https://doi.org/10.1016/j.cie.2023.109534>
13. Xia K, Sacco C, Kirkpatrick M, et al. A digital twin to train deep reinforcement learning agent for smart manufacturing plants: Environment, interfaces and intelligence. *Journal of Manufacturing Systems*. 2021, 58: 210–230. <https://doi.org/10.1016/j.jmsy.2020.06.012>
14. Malik AA, Bilberg A. Digital twins of human robot collaboration in a production setting. *Procedia Manufacturing*. 2018, 17: 278–285. <https://doi.org/10.1016/j.promfg.2018.10.047>
15. Bécue A, Maia E, Feeken L, et al. A new concept of digital twin supporting optimization and resilience of factories of the future. *Applied Sciences*. 2020, 10, 4482. <https://doi.org/10.3390/app10134482>.
16. Zayed SM, Attiya G, El-Sayed A, et al. An efficient fault diagnosis framework for digital twins using optimized machine learning models in smart industrial control systems. *International Journal of Computational Intelligence Systems*. 2023, 16, 69. <https://doi.org/10.1007/s44196-023-00241-6>.
17. Havard V, Jeanne B, Lacomblez M, et al. Digital twin and virtual reality: A co-simulation environment for design and assessment of industrial workstations. *Production & Manufacturing Research*. 2019, 7(1): 472–489. <https://doi.org/10.1080/21693277.2019.1660283>.
18. Striffler N, Voigt T. Concepts and trends of virtual commissioning—A comprehensive review. *Journal of Manufacturing Systems*. 2023, 71: 664–680. <https://doi.org/10.1016/j.jmsy.2023.10.013>
19. Korpai R, Szántó N, Csapó ÁB. A framework for effective virtual commissioning: Guiding principles for seamless system integration. *Journal of Manufacturing and Materials Processing*. 2024, 8, 165. <https://doi.org/10.3390/jmmp8040165>
20. Naranjo JE, Sanchez DG, Robalino-Lopez A, et al. A scoping review on virtual reality-based industrial training. *Applied Sciences*. 2020, 10(22), 8224. <https://doi.org/10.3390/app10228224>
21. Dewberry NK, AlHmoud I, Benton K, et al. A real-time VR-enabled digital twin framework for multi-user interaction in Industry 4.0. *Manufacturing Letters*. 2025, 44: 1486–1497. <https://doi.org/10.1016/j.mfglet.2025.06.168>.
22. Guzman D, Brito G, Naranjo JE, et al. Virtual Assistance Environment for Deaf People Based on an Electronic Gauntlet. 2018 IEEE 3rd Ecuador Technical Chapters Meeting (ETCM); 2018. Article 8580281. <https://doi.org/10.1109/ETCM.2018.8580281>.
23. Naranjo JE, Caiza G, Gallo CV, et al. Immersive Technology-Based Guidance Module for Induction Motor Diagnosing. In: Garcia MV, Gordon-Gallegos C, editors. *Lecture Notes in Networks and Systems*, vol. 678. Springer; 2023. pp. 478–493. https://doi.org/10.1007/978-3-031-30592-4_32.

24. Garcia CA, Naranjo JE, Campana LA, et al. Flexible Robotic Teleoperation Architecture Under IEC 61499 Standard for Oil & Gas Process. 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA); 2018. pp. 1269-1272. <https://doi.org/10.1109/ETFA.2018.8502520>.
25. Unity Technologies. Sentis Documentation (Unity Inference Engine). Available online: <https://docs.unity3d.com/Packages/com.unity.sentis@latest> (accessed on 20 August 2026).
26. ONNX Contributors. Open Neural Network Exchange (ONNX). GitHub repository. Available online: <https://github.com/onnx/onnx> (accessed on 10 September 2026).
27. Lee J, Ardakani HD, Yang S, et al. Industrial big data analytics and cyber-physical systems for future maintenance and service innovation. *Procedia CIRP*. 2015, 38: 3-7. <https://doi.org/10.1016/j.procir.2015.08.026>.
28. Carvalho TP, Soares FAAMN, Vita R, et al. A systematic literature review of machine learning methods applied to predictive maintenance. *Computers & Industrial Engineering*. 2019, 137, 106024. <https://doi.org/10.1016/j.cie.2019.106024>
29. Liu R, Yang B, Zio E, et al. Artificial intelligence for fault diagnosis of rotating machinery: A review. *Mechanical Systems and Signal Processing*. 2018, 108: 33–47. <https://doi.org/10.1016/j.ymssp.2018.02.016>
30. Kumar A, Kumar R. Time-frequency analysis and support vector machine in automatic detection of defect from vibration signal of centrifugal pump. *Measurement*. 2017, 108: 119–133. <https://doi.org/10.1016/j.measurement.2017.04.041>
31. Ranawat NS, Kankar PK, Miglani A. Fault diagnosis in centrifugal pump using support vector machine and artificial neural network. *Journal of Engineering Research*. 2021, 99 (EMSME Special Issue), 111. <https://doi.org/10.36909/jer.EMSME.13881>.
32. Tama BA, Vania M, Lee S, et al. Recent advances in the application of deep learning for fault diagnosis of rotating machinery using vibration signals. *Artificial Intelligence Review*. 2023, 56: 4667–4709. <https://doi.org/10.1007/s10462-022-10293-3>
33. Schwendemann S, Amjad Z, Sikora A. A survey of machine-learning techniques for condition monitoring and predictive maintenance of bearings in grinding machines. *Computers in Industry*. 2021, 125, 103380. <https://doi.org/10.1016/j.compind.2020.103380>
34. Lu J, Qian W, Li S, et al. Enhanced K-nearest neighbor for intelligent fault diagnosis of rotating machinery. *Applied Sciences*. 2021, 11, 919. <https://doi.org/10.3390/app11030919>.
35. Wang X, Mao D, Li X. Bearing fault diagnosis based on vibro-acoustic data fusion and 1D-CNN network. *Measurement*. 2021, 173, 108518. <https://doi.org/10.1016/j.measurement.2020.108518>
36. Alshathri S, Hemdan EE, El-Shafai W, et al. Digital twin-based automated fault diagnosis in industrial IoT applications. *Computers, Materials & Continua*. 2023, 75: 183–196. <https://doi.org/10.32604/cmc.2023.034048>.
37. Zonta T, da Costa CA, da Rosa Righi R, et al. Predictive maintenance in the Industry 4.0: A systematic literature review. *Computers & Industrial Engineering*. 2020, 150, 106889. <https://doi.org/10.1016/j.cie.2020.106889>
38. Theissler A, Pérez-Velázquez J, Kettelgerdes M, et al. Predictive maintenance enabled by machine learning: Use cases and challenges in the automotive industry. *Reliability Engineering & System Safety*. 2021, 215, 107864. <https://doi.org/10.1016/j.res.2021.107864>
39. Nikolakis N, Alexopoulos K, Xanthakis E, et al. The digital twin implementation for linking the virtual representation of human-based production tasks to their physical counterpart in the factory-floor. *International Journal of Computer Integrated Manufacturing*. 2019, 32: 1–12. <https://doi.org/10.1080/0951192X.2018.1529430>
40. Figueira A, Vaz B. Survey on synthetic data generation, evaluation methods and GANs. *Mathematics*. 2022, 10, 2733. <https://doi.org/10.3390/math10152733>
41. Selcuk SY, Unal P, Albayrak O, et al. A workflow for synthetic data generation and predictive maintenance for vibration data. *Information*. 2021, 12(10), 386. <https://doi.org/10.3390/info12100386>.
42. Buggineni V, Chen C, Camelio J. Enhancing manufacturing operations with synthetic data: A systematic framework for data generation, accuracy, and utility. *Frontiers in Manufacturing Technology*. 2024, 4, 1320166. <https://doi.org/10.3389/fmtec.2024.1320166>
43. Chan KC, Rabaev M, Pratama H. Generation of synthetic manufacturing datasets for machine learning using discrete-event simulation. *Production & Manufacturing Research*. 2022, 10: 337–353. <https://doi.org/10.1080/21693277.2022.2086642>
44. Rozanec JM, Zajec P, Theodoropoulos S, et al. Synthetic data augmentation using GAN for improved automated visual inspection. *IFAC-PapersOnLine*. 2023, 56: 11094–11099. <https://doi.org/10.1016/j.ifacol.2023.10.817>