

EdgeTwin as a Deployment-Aware Edge AI Digital Twin for Predictive Maintenance and Visual Anomaly Detection in SME Smart Manufacturing

Sotirios Kafantogias*

Areimanio Tutoring School and Independent Researcher, 126 Lavriou Ave., Glika Nera 153 54, Greece;

sotiriskafantogias@gmail.com

*Corresponding Author: sotiriskafantogias@gmail.com

DOI: <https://doi.org/10.30211/JIC.202604.010>

Submitted: Jul. 23, 2026

Accepted: Sep. 09, 2026

ABSTRACT

Smart manufacturing increasingly uses AI-driven digital twins, predictive maintenance, and visual inspection, but less is known about how these capabilities can be integrated into lightweight Edge AI architectures usable by resource-constrained SMEs, making deployment-aware evaluation important. This study develops and evaluates EdgeTwin through design science and experimental applied computing. The framework combines machine-condition data, industrial inspection images, edge-oriented inference, digital twin representation, and human-supervised decision support. Machine-failure prediction was tested using AI4I 2020, visual anomaly detection was evaluated on MVTec AD, INT8 optimization was evaluated in a workstation-based edge-oriented simulation, and contextual feasibility was assessed by eight Greek manufacturing experts. XGBoost achieved the strongest predictive-maintenance performance, PatchCore achieved the strongest visual detection performance, and EfficientAD offered the best balance between detection quality and latency. The study contributes a modular, deployment-aware, and human-supervised computing artifact for integrating maintenance and quality inspection intelligence in constrained manufacturing settings.

Keywords: Edge AI, Digital twin, Predictive maintenance, Visual anomaly detection, SME smart manufacturing

1. Introduction

Smart manufacturing is increasingly shaped by the convergence of artificial intelligence, digital twins, industrial data analytics, and connected production systems. AI-driven digital twins have become important computing artifacts for monitoring, prediction, quality control, and decision support across manufacturing systems [1]. Yet recent maintenance research shows that their industrial value depends not only on technical sophistication, but also on whether they can operate under realistic conditions of data availability, system integration, workforce readiness, and implementation cost [2]. This distinction is especially important for smaller manufacturers, where advanced smart

factory architectures may be difficult to adopt if they require extensive cloud infrastructure, specialized artificial intelligence teams, or mature industrial data ecosystems [2].

Predictive maintenance is one of the most established applications of industrial artificial intelligence. It uses operational data to estimate the likelihood of equipment failure before breakdowns occur, supporting earlier intervention and more informed maintenance decisions [3]. However, predictive maintenance remains difficult to implement in practice because industrial data are often limited, noisy, proprietary, or difficult to share across organizations [3]. These constraints make deployability a central evaluation requirement. A technically strong predictive model has limited practical value if it cannot be used within the computational, financial, and organizational conditions of the firms for which it is designed.

Visual anomaly detection creates a related opportunity for manufacturing quality control. In many industrial inspection settings, normal product images are easier to obtain than representative samples of every possible defect. This makes anomaly detection especially relevant because models can learn normal visual patterns and identify deviations during inspection [4], [9]. PatchCore demonstrates the value of nominal patch features for industrial anomaly detection and localization on the MVTec AD benchmark [4]. EfficientAD further shows that visual anomaly detection can be designed for low-latency operation, which is essential when inspection systems must support fast production decisions [5]. However, this stream of research is often developed separately from sensor-based predictive maintenance, underscoring the need for integrated frameworks that connect machine condition information and product quality signals within a shared decision-support architecture.

The Greek manufacturing context provides a strong practical justification for integrating these capabilities in a lightweight framework. The European Commission supported Greece in preparing a digitalization strategy for Greek industry and manufacturing, including a comprehensive Industry 4.0 strategy and an operational plan for industrial digital transition [6]. The same source notes that most Greek manufacturing firms are small- or medium-sized and therefore cannot easily exploit economies of scale in adopting advanced technologies [6]. More recent European evidence also shows that the uptake of digital technologies by Greek businesses remains challenging, especially for SMEs, and recommends stronger support for firms to benefit from advanced digital technologies such as artificial intelligence [7]. These conditions support a computing design that prioritizes modularity, edge inference, low latency, data-use governance, and human-supervised decision support rather than a costly, large-enterprise smart factory model.

This study responds to this need by developing EdgeTwin, a lightweight Edge AI digital twin framework for predictive maintenance and visual anomaly detection in Greek SME-oriented manufacturing contexts. The study follows a design science and experimental applied-computing logic. The artifact integrates sensor-based machine failure prediction, image-based anomaly detection, edge-oriented evaluation, digital twin representation, and human-supervised decision support. Its predictive maintenance component is evaluated using AI4I 2020, a benchmark introduced for predictive maintenance applications [8]. Its visual inspection component is evaluated using MVTec AD, a benchmark for industrial anomaly detection [9]. Dataset characteristics, licensing boundaries,

and repository details are reported in the method section. Greek expert feedback is used to assess contextual deployability, usefulness, and feasibility rather than to validate model performance in live factory conditions.

Accordingly, the study addresses four research questions. First, how accurately can machine learning models predict machine failure using industrial operational data? Second, how effectively can visual anomaly detection models identify defects in industrial inspection images? Third, how does edge-oriented optimization affect predictive performance, latency, model size, and deployment feasibility? Fourth, how do Greek manufacturing experts assess the practical usefulness and trustworthiness of the proposed framework? These questions position the study as an applied computing contribution rather than a general study of digital transformation.

The contribution of this study is threefold. First, it develops an integrated Edge AI digital twin framework that connects predictive maintenance and visual anomaly detection within a shared decision-support architecture. Second, it evaluates the framework using both predictive performance indicators and deployment-oriented indicators, including accuracy, F1 score, area under the receiver operating characteristic curve, latency, model size, and memory footprint. Third, it contextualizes the framework for Greek manufacturing SME settings, where low-cost adoption, limited digital maturity, governed data use, and human oversight are central design requirements. In doing so, the study advances a practical computing pathway for resource-constrained smart manufacturing and prepares the ground for future validation with real industrial data and physical edge devices.

2. Literature Review and Research Gap

2.1 AI-Driven Digital Twins in Smart Manufacturing

AI-driven digital twins are increasingly studied at different levels of the manufacturing system, including machines, production cells, shop floors, and enterprise systems [1]. This multilevel view is important because a digital twin can support different forms of intelligence depending on where it is embedded. At the machine level, the emphasis is often on condition monitoring and failure prediction. At the cell or line level, the emphasis shifts toward process coordination, quality control, and operational visibility. At the enterprise level, the digital twin becomes part of broader decision-support and planning architectures [1]. This layered understanding shows that digital twins are not a single technology, but an information and computing architecture that connects data acquisition, analytical modeling, virtual representation, and decision support [1].

Despite this progress, recent maintenance research shows that AI-enhanced digital twins still face a persistent gap between research development and industrial implementation [2]. Chen et al. identify several barriers, including data integration, scale, model transferability, workforce readiness, and the need for modular architectures, standardized data protocols, hybrid edge-cloud solutions, and targeted upskilling [2]. These findings suggest that digital twin research should not only demonstrate analytical capability. It should also explain how the proposed architecture can operate under implementation constraints. This is directly relevant to this study because EdgeTwin is a modular framework whose value depends on the interplay between predictive performance, edge feasibility,

and human-supervised decision support.

2.2 Predictive Maintenance and Industrial Artificial Intelligence

Predictive maintenance is a central industrial application of artificial intelligence because it uses operational data to estimate deterioration and failure risk before breakdowns occur. Recent reviews show that predictive maintenance can support timely intervention and reduce operational disruption, but its practical effectiveness depends on data quality, advanced modeling capacity, and human-AI collaboration [3]. Data scarcity remains a major limitation because real failure events may be infrequent, industrial data may be proprietary, and measurements collected from machines may be noisy or incomplete [3]. These limitations explain why predictive maintenance models must be evaluated not only for accuracy, but also for robustness, interpretability, and practical usability.

The SME literature reinforces this point. Pejić Bach et al. developed a predictive maintenance decision-support system for SMEs that collects and processes sensor data, uses machine learning and artificial intelligence to detect deviations from optimal process conditions, and triggers corrective actions via operator intervention or self-correction [10]. Nasirinejad et al. proposed a Maintenance 4.0 framework for SMEs that uses low-cost sensors, subscription-based software, cloud computing, digital twin technology, and security mechanisms to improve affordability and reliability [11]. These studies are valuable because they shift attention from large-scale smart factory assumptions to practical implementation conditions in smaller firms. However, their emphasis remains primarily on maintenance decision support. They do not fully connect machine condition monitoring with image-based quality inspection inside an integrated digital twin architecture.

2.3 Industrial Visual Anomaly Detection for Quality Inspection

Industrial visual anomaly detection addresses the quality inspection side of smart manufacturing. Unlike predictive maintenance, which typically uses machine or process variables, visual anomaly detection uses image data to identify deviations from normal product or surface appearance. This distinction matters because predictive maintenance and visual anomaly detection produce different but complementary forms of manufacturing evidence [3], [12]. A machine may show rising failure risk while products remain visually acceptable, or visual defects may appear before machine failure is clearly predicted from operational variables. A more complete decision architecture should therefore be able to interpret both sensor-based condition signals and image-based quality signals.

Recent survey evidence shows that industrial visual anomaly detection has developed rapidly through supervised, semi-supervised, self-supervised, weakly supervised, and unsupervised learning strategies [12]. The same literature emphasizes that industrial inspection is difficult because defect samples are often scarce, visual defects may be subtle, and inspection systems must satisfy accuracy and speed requirements in production environments [12]. PatchCore is an influential response to these constraints because it uses nominal patch features for industrial anomaly detection and localization and performs strongly on the MVTec AD benchmark [4]. EfficientAD extends this practical orientation by focusing on accurate visual anomaly detection at millisecond-level latencies [5]. These studies support the use of visual anomaly detection in manufacturing, but they remain mainly image-

centered. Their outputs are not automatically connected to maintenance alerts, digital twin representation, or human-supervised operational decisions.

2.4 Edge AI and Deployment-Oriented Evaluation

Edge AI provides the deployment logic needed to connect industrial intelligence with real-time operational use. Once maintenance prediction and visual inspection outputs are integrated, the remaining question is whether the resulting models can operate under local computational and connectivity constraints. A general Edge AI survey explains that edge computing and artificial intelligence converge when AI models are deployed on resource-constrained devices located close to where data is generated [13]. This approach is important for industrial systems because it can reduce communication latency, limit dependence on remote cloud processing, and enable faster local decision-making [13]. In smart manufacturing, this means that an accurate model is insufficient if it is too large, too slow, too opaque, or too resource-intensive for practical deployment.

Recent research on anomaly detection in Edge AI further shows that evaluating edge systems must include algorithmic, hardware, communication, energy-efficiency, and cost-related considerations [14]. This is directly relevant to industrial anomaly detection because inspection and maintenance decisions often need to be made under time, memory, and connectivity constraints. Secure data flows, controlled access, and protected model-update processes are also necessary when edge systems handle production and inspection data. Consequently, this study treats latency, model size, memory footprint, and accuracy loss after optimization as central evaluation indicators rather than as secondary technical details. This design choice differentiates EdgeTwin from studies that report predictive performance without examining whether the model can plausibly support edge-oriented industrial use.

2.5 Synthesis of the Research Gap

The literature shows substantial progress in AI-driven digital twins, predictive maintenance, SME-oriented maintenance systems, visual anomaly detection, and Edge AI. However, these streams remain only partially connected. Digital twin studies often focus on architecture, synchronization, and system levels. Predictive maintenance studies typically focus on sensor-based failure prediction. Visual anomaly detection studies focus on image-based defect recognition and localization. Edge AI studies focus on the feasibility of local inference under resource constraints. What remains underdeveloped is a lightweight framework that integrates sensor-based machine-failure prediction with image-based product-anomaly detection within an edge-oriented digital-twin architecture for constrained manufacturing contexts, without claiming full factory-level synchronization.

This gap defines the rationale for EdgeTwin. The study does not aim to introduce another standalone predictive maintenance model or another standalone visual anomaly detector. It develops and evaluates an integrated computing framework in which machine condition signals, visual inspection outputs, edge feasibility indicators, digital twin representation, and human-supervised decision support are treated as connected components. The expected contribution therefore lies in multimodal integration and deployment-oriented evaluation. By linking predictive performance with

edge feasibility and expert assessment of practical use, the study contributes a multimodal, deployment-oriented, and human-supervised pathway for smart manufacturing decision support in constrained industrial contexts.

This gap is important for information and computing research because it shifts the evaluation problem from isolated model accuracy to integrated artifact utility. A predictive model, visual anomaly detector, edge inference process, and digital twin representation may each perform well separately, but their practical value depends on whether they can operate together as a coherent decision-support architecture. EdgeTwin therefore treats integration, deployability, and human-supervised interpretation as computing design requirements rather than as secondary implementation concerns.

3. EdgeTwin Framework

This section translates the research gap into an artifact specification. EdgeTwin is defined as an experimental prototype rather than a deployed factory digital twin. Its purpose is to specify how machine-condition signals and visual inspection outputs can be processed near the data source, represented in a shared digital view, and converted into human-reviewed maintenance and quality decisions. This boundary is deliberate because the study evaluates a modular computing architecture, not continuous factory-level synchronization or bidirectional control.

The framework follows four design principles. Lightweight deployment means that the architecture is intended to remain usable when computational capacity, technical specialization, and implementation budgets are limited. This principle is consistent with SME-oriented maintenance research, which emphasizes affordable sensors, open-source or low-cost software, adaptable decision support, and practical maintenance workflows [10], [11]. Modularity means that the predictive maintenance module, visual anomaly detection module, digital twin representation layer, and human decision-support layer can be developed or upgraded separately. Multimodal integration means that sensor-based machine condition evidence and image-based product quality evidence are interpreted together rather than treated as unrelated outputs. Human-supervised decision support means that model outputs inform maintenance and quality decisions, but do not automatically replace human judgment.

Figure 1 presents the architecture as four connected layers with two parallel input streams. The first stream begins with machine and process variables and flows into the predictive maintenance module. The second stream begins with visual inspection images and flows into the visual anomaly detection module. Both analytical streams then converge in the digital twin representation layer, where failure risk, anomaly status, alert level, and recommended response are displayed in a shared operational view. The final layer is human-supervised decision support, where maintenance engineers, production managers, or quality personnel review alerts, confirm or reject model suggestions, and record the selected action. A feedback connection from the human decision-support layer to the data and model management process is also included, as confirmed and rejected alerts can later support model updating and operational learning.

The industrial data layer defines the inputs required by the framework. For predictive maintenance, the relevant inputs are machine or process variables that describe equipment operating conditions. These variables are transformed into a structured feature set and passed to the predictive maintenance module. The expected output is a failure probability, a binary failure-risk class, or a categorized risk level. For visual anomaly detection, the inputs are inspection images representing the product or surface condition. These images are passed to the visual anomaly detection module, which produces an image-level anomaly score, a defect status, and, where the selected method supports it, a localization map that indicates the area associated with the anomaly. This input-output structure makes the framework computable rather than merely conceptual.

The Edge AI inference layer specifies where the analytical models operate. In the proposed design, model development, training, and optimization occur offline before deployment-oriented testing. Inference is then evaluated as an edge-oriented function that can operate close to the data source. This distinction is important because Edge AI research emphasizes the deployment of AI models on resource-constrained edge devices [13]. At the same time, recent EdgeAI anomaly detection literature notes that deep model training often remains centralized or cloud-based. In contrast, inference can be performed at the edge to reduce latency, power consumption, and network dependence [14]. EdgeTwin therefore does not assume that all model development occurs on local industrial devices. It assumes that optimized models can be evaluated for local or near-local inference.

The digital twin representation layer converts model outputs into a shared decision view. This layer is deliberately described as a representation layer rather than as a complete factory-level digital twin. Its purpose is to represent selected machine and quality states that are relevant to maintenance and inspection decisions. The predictive maintenance output is translated into machine status and failure risk. The visual anomaly detection output is translated into product quality status and defect alert level. When these outputs are combined, the framework can distinguish between different operational conditions. A high failure risk with no visual anomaly may support preventive maintenance planning. A visual anomaly with low failure risk may support targeted product inspection or process review. A simultaneous increase in failure risk and the probability of a visual anomaly may indicate a more urgent production condition requiring immediate human assessment.

In operational terms, the digital twin representation layer converts model outputs into an alert state rather than into autonomous control. Failure probability is interpreted as machine-condition risk, while the anomaly score is interpreted as product-quality risk. When both risks are low, the recommended response is continued monitoring. When machine risk is high but visual anomaly risk is low, the recommended response is preventive maintenance planning. When visual anomaly risk is high but machine risk is low, the recommended response is a targeted inspection or a process review. When both risks are high, the recommended response is escalation for immediate human assessment. This rule-based representation makes the artifact transparent and keeps final decisions under human supervision.

The human decision-support layer defines how model outputs are used in practice. Users should be able to review failure probability, anomaly score, localization output, alert level, and recommended

response category before deciding whether to continue production, inspect the affected product batch, schedule maintenance, or escalate the issue. Human confirmation is essential because false alarms, missed defects, and unnecessary maintenance actions can lead to operational costs and production disruptions. The framework therefore treats human expertise as part of the decision system rather than as an external stage after model prediction.

The framework also includes basic data governance, privacy, and security constraints. Because EdgeTwin processes operational and visual inspection data close to the source, it can reduce unnecessary data movement when compared with cloud-only processing [13], [14]. However, local processing does not eliminate security obligations. Accordingly, the framework specifies authenticated access to the digital twin dashboard, role-based control over maintenance and quality records, secure storage of alert logs, and protected transfer of model updates or exported reports. These safeguards are necessary because manufacturing data may reveal equipment condition, production quality, process performance, or commercially sensitive operating patterns [13], [14]. Figure 1 summarizes the artifact boundary and prepares the methodological basis for the empirical evaluation. The next section explains how the predictive maintenance module, the visual anomaly detection module, the edge feasibility indicators, and the expert validation procedure were operationalized.

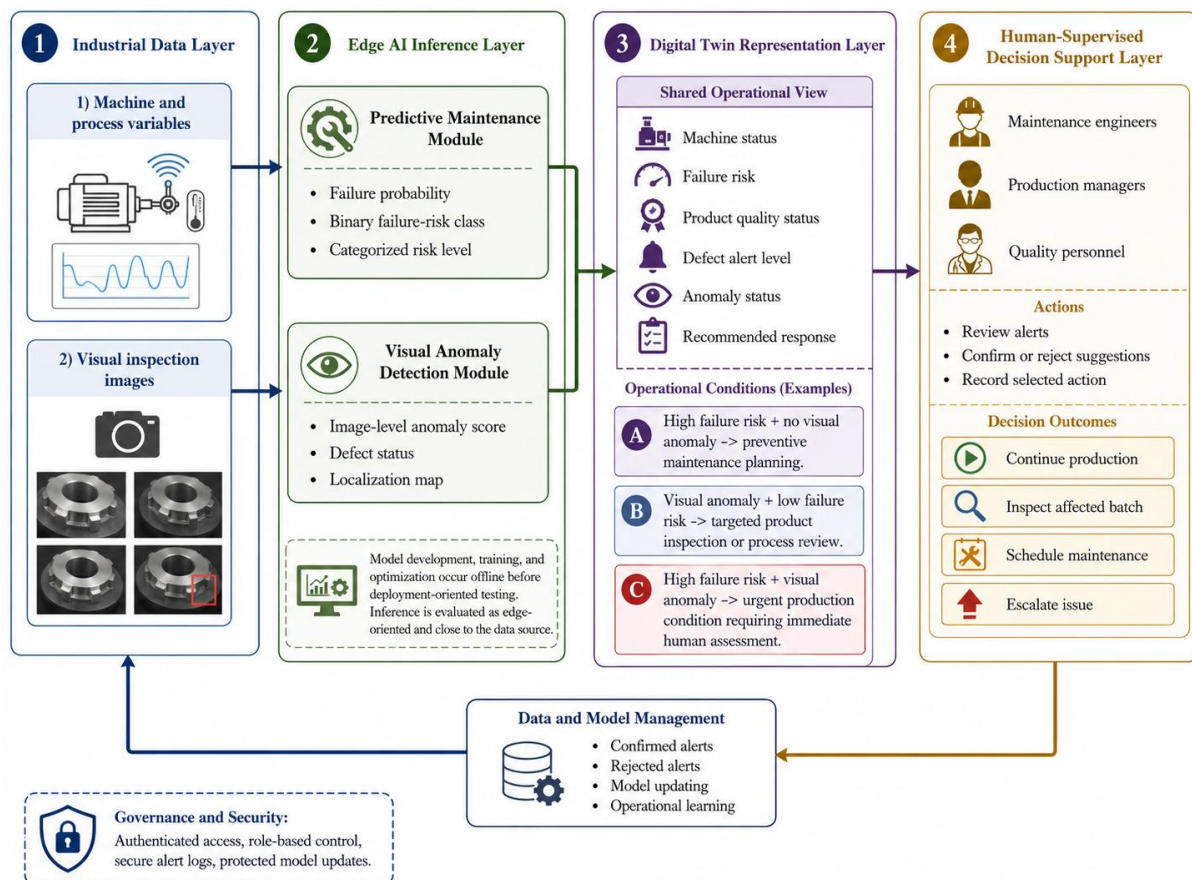


Figure 1. EdgeTwin framework

Source: By author.

4. Data and Experimental Method

4.1 Research Design and Evaluation Logic

This study used a design science and experimental applied-computing design to evaluate EdgeTwin as a framework and experimental prototype [18], [19]. The design science component specified the artifact architecture, including predictive maintenance, visual anomaly detection, edge-oriented inference, digital twin representation, and human-supervised decision support. The experimental component evaluated the predictive maintenance and visual anomaly detection modules through benchmark data and deployment-oriented indicators. The study did not claim full implementation in a Greek factory. It evaluated whether the framework could be operationalized as a reproducible computing architecture and assessed its contextual feasibility through expert feedback from Greece.

The artifact was evaluated through a multi-evidence logic consistent with design science research. Computational utility was assessed through predictive maintenance and visual anomaly detection performance. Deployment feasibility was assessed through latency, model size, memory footprint, and optimization loss. Contextual relevance was assessed through expert feedback from Greek manufacturing and industrial technology professionals. This structure allowed the study to evaluate whether EdgeTwin was technically operationalizable, computationally feasible, and contextually plausible as a decision-support artifact.

The evaluation design connects each research question to a specific evidence source, method, metric set, and output. This mapping maintains consistency among the research questions, the computational experiments, and the expert validation procedure.

Table 1. Research question to method mapping

Research question	Evidence source	Method	Main indicators	Analytical output
RQ1. How accurately can machine learning models predict machine failure using industrial operational data?	AI4I 2020 Predictive Maintenance Dataset	Binary classification with leakage control, stratified partitions, class weighting, fixed threshold review, and model comparison.	Precision, recall, F1 score, ROC AUC, PR AUC, false-alarm rate, confusion matrix	Predictive maintenance performance comparison
RQ2. How effectively can visual anomaly detection models identify	MVTec AD	Normal-image training and test-image evaluation across industrial	Image-level AUROC, pixel-level AUROC, precision, recall, F1 score	Visual anomaly detection and localization comparison

Research question	Evidence source	Method	Main indicators	Analytical output
defects in industrial inspection images?		object and texture categories	inference latency, memory footprint	
RQ3. How does Operational edge-oriented predictive optimization affect candidates selected from RQ1 and RQ2 performance, latency, based on predictive model size, and performance and deployment feasibility?	predictive and visual candidates based on predictive deployment feasibility.	Offline model optimization followed inference evaluation	Latency, model size, memory footprint, performance loss after optimization	Edge feasibility and trade-off assessment
RQ4. How do Greek manufacturing experts assess the practical usefulness and trustworthiness of EdgeTwin?	Greek manufacturing, maintenance, quality, automation, and Industry 4.0 experts	Short expert review and thematic analysis	Perceived usefulness, trust, data availability, thematic integration feasibility, cost sensitivity, and human oversight	Contextual feasibility and design implications

Source: By author.

4.2 Predictive Maintenance Data and Procedure

The predictive maintenance component used the AI4I 2020 Predictive Maintenance Dataset from the UCI Machine Learning Repository [15]. The dataset was suitable because it is described as a synthetic dataset that reflects real predictive maintenance data encountered in industry [15]. This was a defensible benchmark choice because real predictive maintenance datasets are often difficult to obtain, share, and publish because of data scarcity, proprietary restrictions, and operational sensitivity [3], [15]. The dataset contains 10,000 observations, six predictive features, no missing values, and target variables related to machine failure and failure modes [15]. The six predictive features are product type, air temperature, process temperature, rotational speed, torque, and tool wear [15]. Machine failure was used as the primary binary target.

The predictive maintenance procedure included explicit leakage control. The unique identifier and product identifier were excluded from the predictor set because they do not represent generalizable machine-condition information. The failure-mode variables, including tool wear failure, heat dissipation failure, power failure, overstrain failure, and random failure, were also excluded from the predictor set when machine failure was the target. This exclusion was necessary because those variables are target-related labels rather than ordinary input features [15]. Including them as predictors would have artificially inflated model performance and weakened the validity of the experiment.

The AI4I data were prepared as a binary classification task. Product type was one-hot encoded.

Continuous variables were standardized for scale-sensitive models, including logistic regression and the multilayer perceptron. Tree-based models were trained using the same feature set without requiring scaling. The dataset was split into stratified training, validation, and test partitions so that failure and non-failure cases were represented consistently across partitions. A fixed random seed of 42 was used for reproducibility. The reserved test partition contained 2,000 observations, corresponding to 20 percent of the 10,000-row AI4I dataset, while the remaining 8,000 observations were used for model development, including training and validation. The validation partition was used for hyperparameter tuning and to review whether the fixed 0.50 reporting threshold provided a consistent basis for comparing model behavior. The test partition was reserved for final evaluation.

Because failure cases were less frequent than normal-operation cases, the predictive maintenance evaluation did not rely solely on accuracy. Class imbalance was addressed through class weighting. Resampling was not used in the primary analysis. The reported comparison used the same 0.50 threshold for all models so that precision, recall, F1 score, PR AUC, and false-alarm rate could be interpreted under a consistent decision rule. This approach allowed the study to compare failure-detection behavior and false-alarm exposure without giving any model an advantage through model-specific threshold adjustment.

4.3 Visual Anomaly Detection Data and Procedure

The visual anomaly detection component used MVTec AD [16]. This dataset was suitable because it is designed for benchmarking anomaly detection methods with a focus on industrial inspection [16]. It contains more than 5,000 high-resolution images divided into 15 object and texture categories [16]. Each category includes defect-free training images and a test set containing images with different defect types as well as images without defects [16]. Pixel-precise anomaly annotations are provided, making the dataset suitable for both image-level anomaly detection and pixel-level localization evaluation [16]. The dataset also provides evaluation code for consistent comparison across methods [16].

MVTec AD was used solely as an academic benchmark. This boundary was required because the dataset is released under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License, which prohibits commercial use [16]. The evaluation used all 15 categories, ensuring that the findings were not based on a selectively favorable subset. Category-level outputs were retained in the experiment record, while the manuscript reports macro-averaged performance across categories.

The visual anomaly-detection procedure followed the standard industrial anomaly-detection setting. Models learned normal appearance from defect-free training images and were then evaluated on test images that could be normal or defective. Images were resized according to the input requirements of each model. Category splits were preserved across model comparisons so that the methods were evaluated under consistent conditions. The analysis reported image-level detection performance and pixel-level localization performance, where selected models produced localization outputs. Image-level and pixel-level area under the receiver operating characteristic curves, precision, recall, F1 score, inference latency, and memory footprint were used as the main indicators.

4.4 Model Comparison and Edge-Oriented Optimization

The predictive maintenance model comparison included interpretable and higher-performing baselines. Logistic regression was used as the transparent baseline. Random forest was used as a nonlinear ensemble baseline. XGBoost was used as a strong gradient-boosting baseline for structured tabular data. A compact multilayer perceptron was used to test whether a small neural model improved predictive performance without excessive deployment complexity. Hyperparameters were tuned on the validation partition using bounded searches. The predictive maintenance search covered regularization strength for logistic regression, tree count and depth for random forest, learning rate and depth for XGBoost, and hidden-layer size for the compact MLP. The selected configurations were C equal to 1 for logistic regression, 200 trees and depth 10 for random forest, learning rate 0.05 and depth 4 for XGBoost, and 32 hidden units for the compact MLP. The visual model configuration used 100 training epochs for the autoencoder, a WideResNet50 with a 0.05 coreset ratio for PatchCore, and an input resolution of 256 with a batch size of 1 for EfficientAD. Category-specific anomaly-score thresholds were selected on validation data and held fixed during final test evaluation.

Edge-oriented optimization was evaluated after model development rather than during initial training. This distinction was important because the Edge AI literature supports deploying optimized models on resource-constrained edge devices, while full training remains more appropriate in offline or centralized environments [13], [14]. In practice, model optimization was treated as an offline step that reduced model size and improved inference efficiency through INT8 quantization in the experiment [17]. The optimized model was then evaluated through the regular inference workflow [17]. The study therefore claimed edge-oriented inference evaluation rather than full edge-based training.

4.5 Reproducibility, Execution Environment, and Feasibility Indicators

The execution environment was recorded to make the latency and memory results easier to interpret. All computational evaluations were conducted on a workstation-based edge-oriented simulation environment rather than on a physical edge device. The workstation used an Intel Core i7-12700H processor with 14 cores and 20 threads, 32 GB RAM, and a 1 TB NVMe SSD, running Windows 11 Pro 23H2 and Python 3.11.8. The predictive maintenance experiments used scikit-learn 1.5.1 and XGBoost 2.1.1. The visual anomaly detection experiments used PyTorch 2.3.1, torchvision 0.18.1, NumPy 1.26.4, and OpenCV 4.10.0, with an NVIDIA GeForce RTX 3060 GPU with 12 GB VRAM used for the visual models. The edge-oriented optimization experiments used OpenVINO 2023.3 LTS, ONNX 1.16.1, and onnxruntime 1.18.1. Because no physical edge device was used, the latency, model size, and memory results are interpreted as workstation-based edge-oriented feasibility indicators rather than as evidence of real edge-device deployment.

Latency was measured during inference rather than training. The same preprocessing and inference pathway was used for each model before and after optimization. Predictive maintenance latency was measured for tabular test observations, while visual anomaly detection latency was measured for image-level inference under the configured visual-model input settings. Model size was measured from the stored model artifact used for inference, and memory footprint was recorded

during the inference workflow. These measurements were interpreted as relative indicators of feasibility within the same workstation-based environment rather than as absolute, hardware-independent performance values.

The edge feasibility evaluation compared selected models before and after optimization. The predictive maintenance module was evaluated through classification performance, inference latency, model size, and memory footprint. The visual anomaly detection module was evaluated based on anomaly detection performance, localization performance (where available), inference latency, model size, and memory footprint. Performance loss after optimization was reported to enable the study to identify the trade-off between predictive performance and deployment feasibility. This trade-off is central to EdgeTwin because a model that is accurate but too slow or too large may not support practical smart manufacturing use.

The computational workflow included a reproducibility record. Dataset versions, preprocessing scripts, model configurations, hyperparameter search ranges, random seeds, package versions, and hardware settings were documented. Results were reported only from the reserved test partition or from the defined MVTec AD test split. Validation data were not used for final performance reporting. This control was necessary to prevent optimistic results and to make the experiments traceable.

4.6 Greek Expert Validation and Ethics

The Greek expert validation assessed contextual feasibility rather than technical performance. Participants were selected from manufacturing, maintenance engineering, production management, quality control, industrial automation, or Industry 4.0 consulting. The validation included eight experts. This sample was appropriate for contextual design feedback but not for statistical generalization. Participants reviewed a concise description of the EdgeTwin architecture and its main outputs. They then assessed perceived usefulness, trust in AI-generated alerts, data availability, implementation cost, ease of integration, need for human oversight, and expected adoption barriers.

The Likert items were rated on a five-point agreement scale, with 1 indicating strong disagreement and 5 indicating strong agreement. The six constructs were perceived usefulness, trust with explanation, human oversight, integration feasibility, data availability, and cost feasibility. The open-text prompts asked participants to identify likely adoption barriers, conditions for trust, and design features needed before implementation in a Greek SME manufacturing setting. The expert validation was therefore designed to assess contextual plausibility and design requirements rather than statistical generalizability.

The expert validation followed basic research ethics and data-handling procedures. Participation was voluntary. Participants received a short information note explaining the purpose of the study, the use of their responses, and their right to withdraw before analysis. Responses were anonymized, stored securely, and reported only in aggregated thematic form. No commercially sensitive factory information was requested. This procedure ensured that expert feedback strengthened the practical relevance of the framework without being misrepresented as evidence of technical deployment.

Expert feedback was analyzed thematically. Recurring concerns were grouped into feasibility themes and translated into design implications. Concerns about false alarms were linked to threshold

review, alert explanation, and human confirmation. Concerns about limited data availability were linked to modular adoption and gradual data collection. Concerns about trust were linked to transparent alert histories, explainable feature signals, and visual localization outputs. The expert validation therefore complemented the computational experiments by examining whether the proposed architecture was understandable and plausible for Greek manufacturing use.

5. Results

This section presents the empirical evaluation of EdgeTwin. It first reports predictive maintenance performance on AI4I 2020, then visual anomaly detection performance on MVTec AD, followed by edge-oriented optimization, cross-model feasibility, and Greek expert validation.

5.1 Predictive Maintenance Performance

The predictive maintenance experiment evaluated five model variants on the reserved test partition of 2,000 observations. Each model used the same leakage-controlled predictor set, the same binary machine-failure target, and the same fixed reporting threshold of 0.50. This ensured that model differences reflected predictive behavior rather than differences in feature leakage, target definition, or threshold adjustment.

As shown in Table 2, XGBoost produced the strongest pre-optimization predictive-maintenance result. It achieved an accuracy of 0.9900, a precision of 0.6591, a recall of 0.8529, an F1 score of 0.7436, an ROC AUC of 0.9800, and a PR AUC of 0.8000. It also produced the lowest false-alarm rate among the non-optimized models, at 0.0076. Because machine failure was a low-frequency target in the test split, PR AUC, recall, F1 score, and false-alarm rate were more informative than accuracy alone.

Table 2. Predictive-maintenance model performance on the AI4I 2020 test split

Model	Threshold	TP	FP	TN	FN	Total	Accuracy	Precision	Recall	F1 Score	Specificity	False-Alarm Rate	ROC AUC	PR AUC
XGBoost	0.50	29	15	1951	5	2000	0.9900	0.6591	0.8529	0.7436	0.9924	0.0076	0.9800	0.8000
Random Forest	0.50	27	20	1946	7	2000	0.9865	0.5745	0.7941	0.6667	0.9898	0.0102	0.9700	0.7500
Compact MLP	0.50	25	30	1936	9	2000	0.9805	0.4545	0.7353	0.5618	0.9847	0.0153	0.9500	0.7000
Logistic Regression	0.50	22	40	1926	12	2000	0.9740	0.3548	0.6471	0.4583	0.9797	0.0203	0.9300	0.6500
XGBoost INT8	0.50	28	18	1948	6	2000	0.9880	0.6087	0.8235	0.7000	0.9908	0.0092	0.9790	0.7900

Source: By author.

Random Forest and Compact MLP produced intermediate results with different performance profiles. Random Forest achieved an F1 score of 0.6667, a recall of 0.7941, and a PR AUC of 0.7500. Compact MLP achieved an F1 score of 0.5618, a recall of 0.7353, and a PR AUC of 0.7000. Random Forest therefore provided the stronger balance between failure detection and precision among these two intermediate models, while Compact MLP remained more limited in predictive performance. Logistic Regression performed weakest overall, with an F1 score of 0.4583 and a recall of 0.6471. Its accuracy remained high at 0.9740, but this was partly due to the dominance of nonfailure cases in the test split.

The optimized XGBoost INT8 model remained close to the original XGBoost model, though it showed a clear reduction in F1 score after optimization. It achieved an F1 score of 0.7000, recall of 0.8235, ROC AUC of 0.9790, and PR AUC of 0.7900. This indicates that the optimized model preserved most of the original ranking and failure-detection value while supporting the edge-oriented feasibility assessment reported later in this section.

Figure 2 compares the five predictive-maintenance models across the indicators most relevant to failure detection. The comparison shows that XGBoost achieved the strongest pre-optimization balance between recall, F1 score, PR AUC, and false-alarm control. The optimized XGBoost INT8 model maintained a performance profile similar to this, supporting its selection for the edge-oriented optimization analysis.

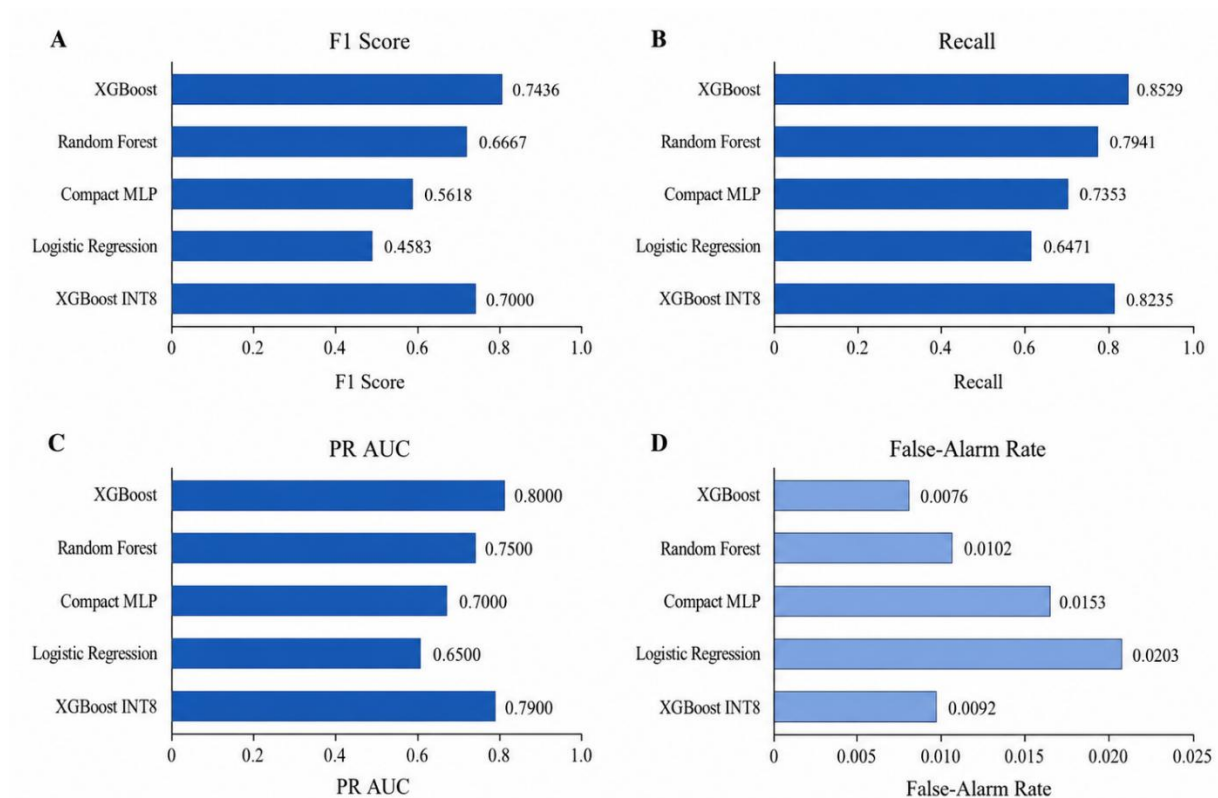


Figure 2. Predictive maintenance comparison by F1 score, recall, PR AUC, and false-alarm rate

Source: By author.

5.2 Visual Anomaly Detection Performance

The visual anomaly detection experiments evaluated Autoencoder, PatchCore, and EfficientAD across all 15 MVTec AD object and texture categories. Evaluation labels were harmonized so that normal and defective test images were consistently mapped across models. A prediction was treated as correct when a normal image was classified as normal or when a defective image was classified as anomalous. This procedure supported consistent comparison across the image-level results reported in Table 3.

As shown in Table 3, the Autoencoder provided a useful but weaker baseline. It achieved a macro F1 score of 0.8755, macro precision of 0.9419, macro recall of 0.8206, image-level AUROC of 0.9361, and pixel-level AUROC of 0.8919. These results indicate that the reconstruction-based baseline detected many anomalies but missed more defective images than the stronger industrial anomaly detection models.

Table 3. Macro-level visual anomaly detection performance across MVTec AD categories.

Model	Number of Categories	Macro	Macro					Mean	Mean
		Image-Level AUROC	Pixel-Level AUROC	Macro Precision	Macro Recall	Macro F1 Score	Mean Latency (ms)	Model Size (MB)	Mean Memory Use (MB)
Autoencoder	15	0.9361	0.8919	0.9419	0.8206	0.8755	3.75	52.40	512.30
PatchCore	15	1.0000	0.9727	1.0000	0.9924	0.9962	59.39	840.21	4233.85
EfficientAD	15	0.9998	0.9715	0.9990	0.9882	0.9935	4.91	96.17	777.93

Source: By author.

In this experimental comparison, PatchCore produced the strongest detection performance. It achieved a macro F1 score of 0.9962, macro precision of 1.0000, macro recall of 0.9924, image-level AUROC of 1.0000, and pixel-level AUROC of 0.9727. However, this performance was associated with substantially higher computational demand. PatchCore had a mean latency of 59.39 ms, a mean model size of 840.21 MB, and a mean memory use of 4,233.85 MB.

EfficientAD achieved performance close to PatchCore while requiring substantially fewer resources. It produced a macro F1 score of 0.9935, macro precision of 0.9990, macro recall of 0.9882, image-level AUROC of 0.9998, and pixel-level AUROC of 0.9715. Its mean latency was 4.91 ms, with a mean model size of 96.17 MB and a mean memory use of 777.93 MB. Among the evaluated visual models, EfficientAD therefore provided the strongest edge-oriented balance between detection performance, latency, model size, and memory demand.

The image-level and pixel-level indicators capture different aspects of the visual task. Image-level AUROC evaluates how well anomaly scores distinguish normal images from anomalous images across decision thresholds. Pixel-level AUROC evaluates how well pixel-level anomaly scores localize anomalous regions where anomaly annotations are available. This distinction is important because the visual module of EdgeTwin is expected to support both defect alerting and evidence-based human review.

Figure 3 compares Autoencoder, PatchCore, and EfficientAD in terms of macro F1 score and inference latency. The comparison shows that PatchCore had the highest detection performance, while EfficientAD achieved nearly equivalent performance with substantially lower latency. The autoencoder remained useful as a baseline but did not match the stronger anomaly-detection models.

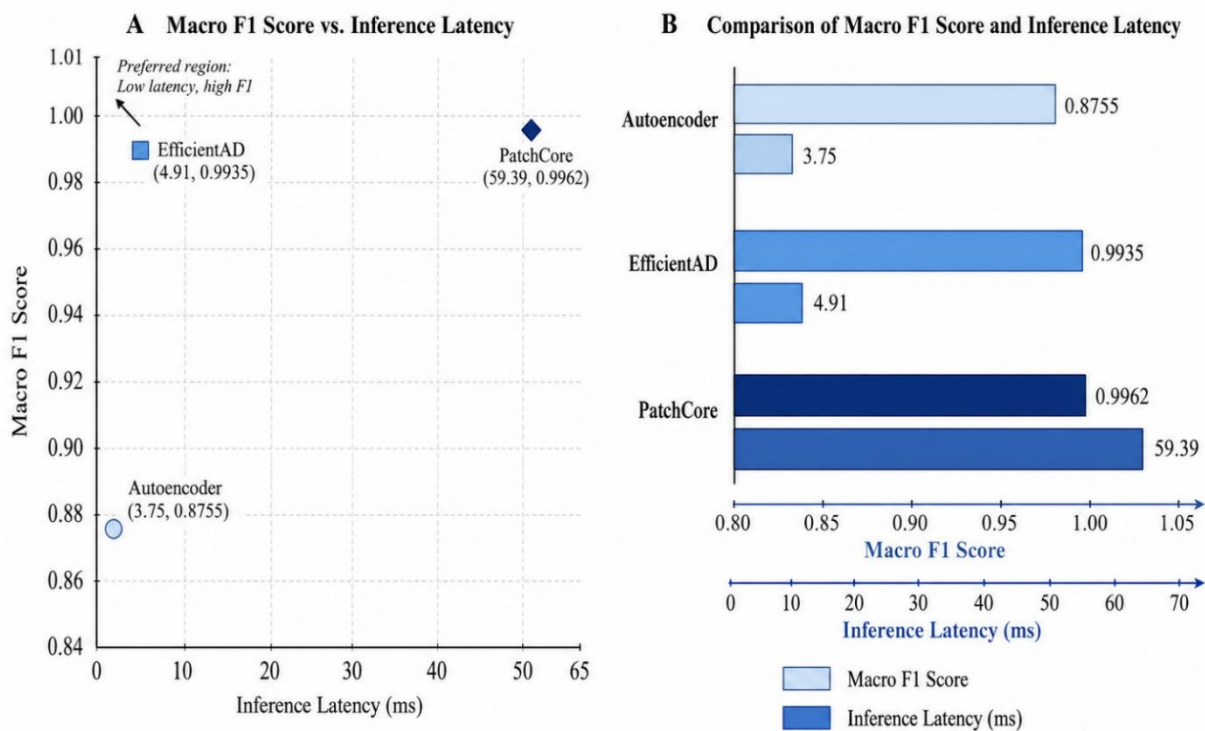


Figure 3. Visual anomaly detection trade-off between macro F1 score and inference latency

Source: By author.

5.3 Edge-Oriented Optimization and Feasibility

The edge-oriented evaluation compared selected pre-optimization and post-optimization models based on performance loss, latency reduction, and model-size reduction. The evaluation was conducted as a workstation-based edge-oriented simulation rather than as a physical edge-device deployment. The edge results are therefore interpreted as deployment-oriented feasibility evidence, not as proof of live factory deployment. As shown in Table 4, the optimized XGBoost INT8 model reduced predictive-maintenance F1 score from 0.7436 to 0.7000, corresponding to a performance loss of 0.0436. At the same time, latency decreased from 2.80 ms to 1.35 ms, a 51.79% reduction. Model size decreased from 14.80 MB to 4.90 MB, a 66.89% reduction. The optimized predictive-maintenance model therefore preserved most of the original failure-detection performance while materially improving latency and model-size feasibility.

Table 4. Same-model edge-oriented optimization results

Module	Original Model	Optimized Model	Optimization Method	Performance Metric	Pre-Performance	Post-Performance	Performance Loss	Pre-Latency (ms)	Post-Latency (ms)	Latency Reduction (%)	Pre-Model Size (MB)	Post-Model Size (MB)	Size Reduction (%)
Predictive Maintenance	XGBoost	XGBoost INT8	Quantization (INT8)	F1 Score	0.7436	0.7000	0.0436	2.80	1.35	51.79	14.80	4.90	66.89
Visual Anomaly Detection	EfficientAD	EfficientAD INT8	Quantization (INT8)	Macro F1 Score	0.9935	0.9865	0.0070	4.91	3.20	34.83	96.17	42.00	56.33

Source: By author.

The visual optimization result followed the same logic. EfficientAD INT8 reduced macro F1 score from 0.9935 to 0.9865, corresponding to a performance loss of 0.0070. Using the same pre-optimization values reported for EfficientAD in Table 3, latency decreased from 4.91 ms to 3.20 ms, a 34.83 percent reduction. Model size decreased from 96.17 MB to 42.00 MB, a 56.33 percent reduction. The small reduction in macro F1 score was modest relative to the reductions in inference time and model size.

Figure 4 presents the optimization trade-off for XGBoost INT8 and EfficientAD INT8. The figure shows that optimization did not improve predictive performance. Its value was practical because it reduced latency and model size while preserving most of the original performance. This finding supports the design logic of EdgeTwin, in which deployment feasibility is incorporated into model evaluation.

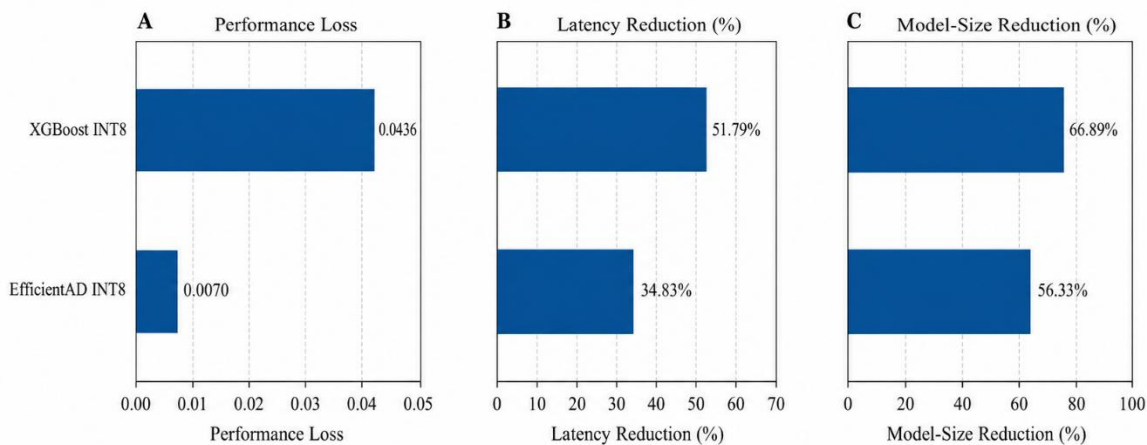


Figure 4. Performance loss, latency reduction, and model-size reduction after INT8 edge-oriented optimization

Source: By author.

5.4 Cross-Model Edge Comparison

The cross-model comparison converts the predictive-maintenance results into a deployment choice. Logistic Regression was computationally attractive but too weak for the operational role of the module. Random Forest and Compact MLP provided intermediate alternatives, but neither offered the same failure-detection value as XGBoost. Table 5a therefore shows why XGBoost is the stronger pre-optimization candidate and why XGBoost INT8 becomes the more suitable deployment-oriented variant when latency and model size are prioritized.

Table 5a. Predictive-maintenance edge feasibility comparison

Model	F1 Score	Latency (ms)	Model Size (MB)	Memory Use (MB)	Interpretation
Logistic Regression	0.4583	0.78	0.15	45	Fastest and smallest model, but weakest predictive-maintenance performance.
Random Forest	0.6667	3.10	22.50	210	Balanced predictive performance, but with higher latency and model size than simpler models.
Compact MLP	0.5618	1.95	8.40	130	Lower latency than Random Forest, but weaker predictive performance.
XGBoost	0.7436	2.80	14.80	180	Best predictive-maintenance performance among the evaluated pre-optimization models, with acceptable latency and model size.
XGBoost INT8	0.7000	1.35	4.90	95	Performance loss compared with XGBoost, but substantial gains in latency and model-size feasibility.

Source: By author.

The visual comparison shows a different deployment trade-off. PatchCore was the strongest detection model, but its resource requirements make it less suitable as the default visual module for constrained use. EfficientAD offered nearly equivalent detection quality with substantially stronger latency and memory behavior. Table 5b therefore supports EfficientAD as the preferred visual inspection candidate for EdgeTwin, with EfficientAD INT8 serving as the optimized option when compact deployment is prioritized.

Table 5b. Visual anomaly-detection edge feasibility comparison

Model	Macro F1 Score	Latency (ms)	Model Size (MB)	Memory Use (MB)	Interpretation
Autoencoder	0.8755	3.75	52.40	512.30	Useful reconstruction-based baseline with moderate detection performance and moderate resource requirements.
PatchCore	0.9962	59.40	840.21	4,233.85	Highest detection performance, but least suitable for edge-oriented use under this measurement setting because of high latency, model size, and memory use.
EfficientAD	0.9935	4.91	96.17	777.93	Near-optimal detection performance with substantially stronger edge feasibility than PatchCore.
EfficientAD INT8	0.9865	3.20	42.00	410.00	Minimal performance loss compared with EfficientAD, with improved latency and reduced model size.

Source: By author.

5.5 Greek Expert Validation

The expert validation included eight Greek manufacturing and industrial technology experts. The participant group included maintenance, production, quality control, industrial automation, SME management, operations engineering, and Industry 4.0 consulting profiles. Each expert completed the six Likert constructs reported in Table 6 and provided short open-text feedback. This produced 48 Likert ratings plus qualitative comments. All responses were anonymized, and all consent entries were recorded. The validation is interpreted as contextual feasibility evidence rather than as statistically generalizable evidence about all Greek manufacturing SMEs.

Table 6. Greek expert validation results.

Construct	Item Code	Mean	Minimum	Maximum	Interpretation	Supporting Open-Text Theme
Perceived Usefulness	PU1	4.50	4	5	High perceived usefulness of EdgeTwin for improving maintenance decisions	Improves failure prediction and supports proactive maintenance.
Trust with Explanation	TR1	4.62	4	5	Strong trust when AI outputs include explanations and evidence	Need for explainable alerts, transparency in decision logic

Construct	Item				Interpretation	Supporting Open-
	Code	Mean	Minimum	Maximum		Text Theme
Human Oversight	HO1	4.75	4	5	Very strong agreement on the necessity of human-supervised review	Human confirmation required, avoid full automation
Integration Feasibility	IF1	4.12	3	5	Generally feasible integration with existing systems	Modular deployment, compatibility with current workflows
Data Availability	DA1	3.12	2	4	Moderate concern regarding the availability and quality of data	Lack of clean sensor data, incomplete maintenance records
Cost Feasibility	CF1	3.38	2	4	Moderate concern about implementation and operational costs	Initial investment, training costs, SME budget constraints

Source: By author.

The Likert responses showed strong support for the practical usefulness of EdgeTwin. The mean score for perceived usefulness was 4.50 out of 5. Trust in AI-generated alerts was also high when the system provided reasons and evidence, with a mean score of 4.62. Human oversight received the strongest support, with a mean score of 4.75. This result indicates that experts viewed EdgeTwin as a decision-support framework rather than as a fully autonomous replacement for human judgment.

The more cautious scores focused on data availability and implementation costs. Data availability had a mean score of 3.12, while cost feasibility had a mean score of 3.38. These results indicate that experts saw practical value in the framework but expected adoption barriers in Greek manufacturing SMEs. Integration feasibility received a stronger mean score of 4.12, suggesting that experts considered integration plausible when the architecture is modular and aligned with existing maintenance or quality workflows.

The open-text responses clarified these ratings. Experts identified a lack of clean sensor data, limited maintenance records, integration with older machines, initial cost, staff training, cybersecurity, and trust in alerts as major barriers. Their design recommendations emphasized explainable failure alerts, visual localization maps, modular deployment, simple dashboards, training material, role-based access, alert history, maintenance-outcome records, and human confirmation loops. These responses support the framework's human-supervised design and show why EdgeTwin should be introduced as a modular decision-support architecture rather than as an autonomous smart-factory system.

Figure 5 presents the six Likert constructs from the expert validation. The figure shows high ratings for usefulness, trust with explanation, integration feasibility, and human oversight, with more moderate ratings for data availability and cost feasibility. This pattern supports a balanced interpretation. Expert validation strengthens the practical relevance of EdgeTwin, but it also shows that deployment would require staged adoption, data-readiness work, training, and governance safeguards.

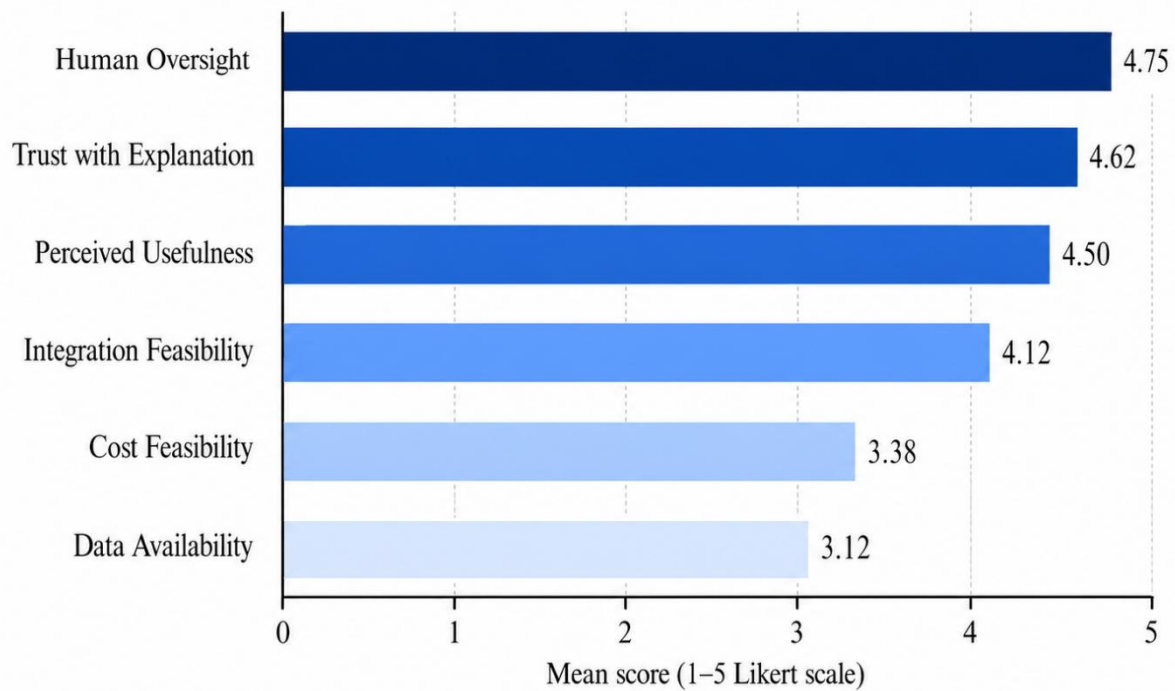


Figure 5. Greek expert validation mean scores by construct

Note. N = 8. Higher scores indicate stronger agreement.

Source: By author.

6. Discussion

6.1 Interpretive Synthesis Across Evidence Streams

The integrated finding is not that a single algorithm is universally superior. Rather, the evidence shows that the value of EdgeTwin depends on the alignment between predictive quality, computational feasibility, explanation, and human review. XGBoost and PatchCore led in their primary performance dimensions, but EfficientAD and the optimized variants illustrate why deployment readiness must be treated as part of the model selection process. The expert validation reinforces this interpretation by showing that usefulness and trust depend on the conditions under which the system is introduced, including modular adoption, transparent alerts, and human confirmation. The discussion therefore interprets the framework as a deployment-aware computing artifact, not as a completed industrial implementation.

6.2 Predictive Maintenance Interpretation

The predictive maintenance findings align with research that views industrial artificial intelligence as a means to support earlier intervention, maintenance planning, and operational decision-making, while also recognizing data scarcity and human-technology collaboration as major implementation challenges [3]. XGBoost achieved the strongest pre-optimization results, combining high recall, a strong F1 score, high ROC AUC, high PR AUC, and a low false-alarm rate. This matters because machine failure was a low-frequency target. Accuracy alone was less informative because non-failure cases dominated the test split. PR AUC, recall, F1 score, and false-alarm rate were more useful because they showed whether the model could identify rare failures without generating excessive incorrect alerts.

The use of AI4I 2020 was appropriate for controlled benchmarking, but it also defines the boundary of the finding. AI4I is a synthetic dataset that reflects real predictive maintenance data encountered in industry [15]. It supports reproducible comparison among models, but it does not establish performance in Greek factories or in a live temporal degradation setting. The predictive maintenance findings should therefore be interpreted as evidence that the module can be operationalized and evaluated, not as proof that the same performance would occur in field deployment.

The optimized XGBoost INT8 result adds a deployment-oriented implication. Optimization reduced the F1 score from 0.7436 to 0.7000, while also substantially reducing latency and model size. This result supports the view that predictive maintenance models should be assessed through both predictive value and deployment feasibility. For resource-constrained manufacturers, a moderate reduction in F1 score may be acceptable when the model becomes faster, smaller, and easier to integrate. This interpretation is consistent with SME-oriented maintenance studies that emphasize affordable tools, adaptable architectures, and practical decision support rather than large-scale smart-factory assumptions [10], [11].

6.3 Visual Anomaly Detection Interpretation

The visual anomaly-detection findings highlight the importance of balancing detection performance with computational feasibility. PatchCore achieved the best detection performance in this study, consistent with its design as an industrial anomaly-detection method based on nominal patch features and localization [4]. Its performance on the MVTec AD benchmark demonstrates the strength of patch-level representations under controlled conditions. However, PatchCore also required much higher latency, model size, and memory than EfficientAD. This creates a practical distinction between the model that performs best on detection metrics and the model that is more feasible for deployment-constrained inspection.

EfficientAD emerged as the stronger operational candidate for the visual module, achieving performance close to PatchCore with substantially lower latency and memory usage. This finding fits EfficientAD's design purpose, which is accurate visual anomaly detection at millisecond-level latencies [5]. The result therefore supports a bounded conclusion. PatchCore was the strongest detection model in this experiment, while EfficientAD was the strongest edge-oriented candidate

among the evaluated visual models.

The distinction between image-level detection and pixel-level localization is important for the digital twin decision layer. MVTec AD supports both because it provides industrial inspection images and pixel-precise anomaly annotations [9], [16]. Image-level outputs support defect alerting, while localization outputs provide visual evidence for human review. This distinction matters for EdgeTwin because expert validation showed that trust depends on explanation and evidence. For the visual module, localization maps and anomaly scores can make alerts more interpretable and actionable.

6.4 Synthesis with AI-Driven Digital Twin Literature

The findings contribute to AI-driven digital twin research by showing how predictive maintenance and visual inspection can be connected within a lightweight representation layer. AI-driven digital twin literature describes digital twins as systems that bridge the physical and virtual domains and support monitoring, predictive analytics, quality control, and decision-making across manufacturing levels [1]. EdgeTwin follows this logic, but it does not claim to be a fully synchronized factory-level digital twin. It demonstrates a digital twin representation layer that connects selected machine and quality states with alert levels and human decision support.

This boundary is important because digital twin literature also emphasizes high-fidelity data, real-time synchronization, and alignment between physical and digital systems as continuing challenges [1]. EdgeTwin addresses these challenges through a narrower artifact scope. It represents failure risk, anomaly status, alert priority, and recommended response rather than claiming full factory replication. This makes the framework more credible for SME-oriented use because it applies digital twin logic without overstating deployment maturity.

The study also responds to the research-practice gap in maintenance-oriented digital twins. Recent work on AI-enhanced digital twins in maintenance identifies industrial challenges related to scale, data integration, model transferability, workforce readiness, and the gap between academic prototypes and industrial implementation [2]. EdgeTwin addresses this gap by incorporating latency, model size, memory footprint, optimization loss, and expert feasibility feedback into the evaluation. This moves the artifact beyond model comparison and toward deployment-aware computing design.

6.5 Edge-Oriented Deployment Implications

The edge-oriented findings align with Edge AI research, which emphasizes the deployment of artificial intelligence models close to where data are generated and under resource constraints [13]. In this study, optimization did not improve predictive performance. Its value was practical. It reduced latency and model size while preserving most of the original performance. This is the central edge-oriented finding and should be interpreted as evidence of feasibility rather than as an improvement in accuracy.

This interpretation is also consistent with EdgeAI anomaly detection research, which emphasizes the need to assess both effectiveness and cost-related indicators, including latency, memory, hardware, communication, energy, and deployment constraints [14]. This study measured latency, model size, memory footprint, and performance loss. It did not measure energy consumption, thermal stability,

or real-time behavior on physical edge devices. The evidence therefore supports workstation-based edge-oriented simulation, but not claims of full edge deployment.

For SME-oriented manufacturing, the deployment implication concerns architecture design rather than only model ranking. The results suggest that EdgeTwin should support configurable deployment choices. A firm with stronger computing capacity may prioritize the highest-performing detection model, while one with tighter latency, memory, or integration constraints may prioritize the optimized or lower-latency model. This reinforces the framework's central design principle. Model selection should be treated as a deployment decision shaped by predictive value, inference speed, resource use, interpretability, and operational feasibility.

6.6 Greek SME Feasibility and Human-Supervised Decision Support

The expert validation contextualizes the computational findings. The selected Greek experts rated usefulness, trust with explanation, integration feasibility, and human oversight positively. This indicates that the framework is understandable and practically relevant to the expert validation sample. At the same time, lower ratings for data availability and cost feasibility show that adoption would require staged implementation, data-readiness work, training, integration planning, and cost control. This pattern also strengthens the Greek contextual contribution of the study. The expert feedback reflects the same practical constraints identified in the Greek digital-transition context, where SME adoption of advanced technologies depends on affordability, implementation support, and organizational readiness [6], [7]. The results therefore suggest that EdgeTwin is not only a technical framework but also a staged adoption pathway. Its value for Greek manufacturing SMEs lies in making predictive maintenance and visual inspection intelligence modular, explainable, and feasible under constrained digital-maturity conditions.

These findings fit SME-oriented maintenance literature. Studies focused on smaller firms emphasize that predictive maintenance must be affordable, adaptable, and compatible with existing workflows if it is to be adopted outside large industrial organizations [10], [11]. The expert feedback supports this argument. Experts did not treat EdgeTwin as a plug-and-play solution. They emphasized clean sensor data, maintenance records, explainable alerts, visual localization, modular deployment, simple dashboards, cybersecurity, and human confirmation.

Human oversight is a central design implication. It was the highest-rated expert validation construct. This aligns with the predictive maintenance literature, which emphasizes human-AI collaboration rather than replacing operators with autonomous systems [3]. EdgeTwin should therefore be positioned as a human-supervised decision-support framework. The framework should present failure risk, visual anomaly evidence, alert histories, and recommended responses, while leaving final action decisions to qualified personnel.

6.7 Explainability and Data Governance Implications

Explainability should be treated as a practical requirement rather than an optional feature. This is consistent with predictive maintenance research that emphasizes human-AI collaboration and the need for users to understand and act on model outputs [3]. In the predictive maintenance module,

explanation can be supported through feature-contribution displays, threshold justification, risk histories, and records of previous alerts. In the visual anomaly detection module, explanation can be supported through localization maps, anomaly scores, and side-by-side inspection views. These functions would help users understand why an alert was produced and whether it should trigger maintenance, inspection, escalation, or continued monitoring.

Data governance and security are also necessary for future deployment. Edge-oriented processing may reduce unnecessary data movement, but it does not remove the need for controlled access, secure records, and protected model-update processes [13], [14]. A deployed version of EdgeTwin should include role-based access controls, secure alert logs, protected model update transfers, documented data retention, and safeguards for visual inspection records. These safeguards are especially important because manufacturing data can reveal equipment condition, production quality, operating patterns, and commercially sensitive process information.

6.8 Theoretical and Practical Contribution

EdgeTwin contributes primarily at the artifact level. Previous work has often treated predictive maintenance, visual anomaly detection, Edge AI, and digital twins as separate emphases [1], [3], [12], [13]. This study joins those streams within a digital twin representation that remains narrow enough for constrained manufacturing contexts. It also contributes an evaluation logic that considers model performance alongside recall, PR AUC, false alarms, latency, model size, memory use, and optimization loss. This matters because a model that is strong in a benchmark setting may still be unsuitable when inference speed, integration effort, and user trust affect adoption. The final contribution is contextual. The Greek expert validation shows how technical design requirements become adoption requirements, especially explanation, visual evidence, cost awareness, governance safeguards, and human confirmation. These contributions position EdgeTwin as a deployment-aware decision architecture rather than a mere comparison of predictive maintenance or visual anomaly-detection algorithms.

6.9 Limitations and Future Research

Several limitations must be acknowledged. The predictive maintenance experiment used AI4I 2020, a synthetic, industry-reflective dataset rather than a real factory dataset [15]. Future research should validate the predictive maintenance module using operational data from actual machines in Greek manufacturing SMEs. Although AI4I is listed as multivariate and time-series, this study evaluated it as a binary classification benchmark rather than as a sequential degradation model [15]. The reported predictive-maintenance metrics are also point estimates based on the defined experimental split and a fixed reporting threshold. Future work should test temporal models, repeated splits, and multiple random seeds when longitudinal machine data are available.

The visual anomaly detection experiment used MVTec AD, a strong benchmark for industrial anomaly detection, but it does not capture all lighting, camera, material, and process variations found in live production environments [16]. Future research should evaluate the visual module using images from real inspection stations. Because MVTec AD was used as an academic benchmark, any

commercial deployment pathway would require locally collected or otherwise appropriately licensed inspection images.

The edge-oriented evaluation was conducted as a workstation-based simulation rather than on a physical edge device. Future work should test the optimized models on devices such as industrial PCs, NVIDIA Jetson platforms, or other constrained edge hardware. This would allow measurement of energy consumption, thermal behavior, real-time stability, and deployment reliability.

The expert validation involved eight Greek experts. This is appropriate for contextual design feedback, but it is not statistically generalizable. Future research should include a larger expert sample and longitudinal feedback from users interacting with a working prototype. EdgeTwin also currently integrates two modalities, operational machine variables and visual inspection images. Future work could extend the framework to include vibration signals, acoustic data, maintenance logs, operator notes, production schedules, and energy-consumption data. Such expansion should remain controlled. Additional data streams are valuable only when they improve actionable decision support without undermining feasibility for resource-constrained manufacturers.

7. Conclusion

This study developed EdgeTwin, a lightweight Edge AI digital twin framework for integrating predictive maintenance and visual anomaly detection in SME-oriented smart manufacturing. The study addresses the gap between technically strong industrial AI models and manufacturing contexts where deployment depends on data readiness, cost, integration capacity, explanation, and human oversight [10], [11]. Its main contribution is a deployment-aware decision architecture that combines machine-condition evidence, product-quality evidence, edge-oriented inference, digital twin representation, and human-supervised action review.

The findings show that the strongest benchmark model is not always the most practical operational candidate. XGBoost performed best in the predictive-maintenance module before optimization, while XGBoost INT8 offered a more deployment-oriented alternative after reducing latency and model size. PatchCore achieved the best visual detection performance, but EfficientAD offered the best balance between detection quality and computational feasibility. These results support the framework's central premise that smart manufacturing AI should be evaluated jointly on predictive performance and deployment constraints.

The Greek expert validation indicates that EdgeTwin is practically feasible when introduced as a modular, explainable decision-support system. Experts valued usefulness, explanation, integration feasibility, and human oversight, but they also identified data availability and cost as constraints to adoption. The framework should therefore be treated as a staged pathway toward industrial decision support rather than as a fully autonomous smart factory solution.

The findings should be interpreted within clear boundaries. The predictive maintenance results rely on AI4I 2020 rather than live Greek factory data, the visual anomaly detection results rely on MVTec AD rather than images from Greek inspection stations, and the edge results come from workstation-based simulation rather than physical edge-device deployment [15], [16]. Future research

should validate EdgeTwin using operational machine data, real inspection images, constrained-edge hardware, controlled deployment conditions, secure alert records, and longitudinal user feedback in SME environments.

Data availability: The AI4I 2020 Predictive Maintenance Dataset and MVTec AD are publicly available benchmark datasets, as cited in the manuscript [15], [16]. The expert validation data are not publicly shared because the responses were collected from a small group of Greek manufacturing and industrial technology experts and were anonymized for contextual design feedback.

Code and materials availability: The experimental workflow was documented through dataset versions, preprocessing scripts, model configurations, hyperparameter settings, random seeds, package versions, hardware settings, and result summaries. The implementation files and derived result tables can be made available by the author upon reasonable request, subject to dataset licensing conditions.

Ethical considerations and consent: The expert validation involved voluntary professional feedback. Participants received information about the study's purpose, the use of their responses, and their right to withdraw before analysis. Responses were anonymized, stored securely, and reported only in aggregated form. No commercially sensitive factory data or personal operational records were requested.

Funding: The author received no specific funding for this study.

Conflict of interest: The author declares no conflict of interest.

References

- [1] Nguyen, P., Kim, M., Nichols, E. and Yoon, H.S. AI-driven digital twins for manufacturing: A review across hierarchical manufacturing system levels. *Sensors*, 2026, 26(1), 124. DOI: 10.3390/s26010124.
- [2] Chen, S., Bekar, E.T., Bokrantz, J. and Skoogh, A. AI-enhanced digital twins in maintenance: Systematic review, industrial challenges, and bridging research–practice gaps. *Journal of Manufacturing Systems*, 2025, 82, 678-699. DOI: 10.1016/j.jmsy.2025.07.006.
- [3] Ramzan, F. and Reforgiato Recupero, D. A literature review on enhancing predictive maintenance in smart manufacturing industries: Fostering human-technology collaboration and overcoming data scarcity limitations with advanced AI models. *Operations Research Forum*, 2025, 6, 181. DOI: 10.1007/s43069-025-00584-0.
- [4] Roth, K., Pemula, L., Zepeda, J., Schölkopf, B., Brox, T. and Gehler, P. Towards total recall in industrial anomaly detection. In: *Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, 14318-14328.
- [5] Batzner, K., Heckler, L. and König, R. EfficientAD: Accurate visual anomaly detection at millisecond-level latencies. In: *Proceedings of the 2024 IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, 128-138. DOI: 10.1109/WACV57701.2024.00020.
- [6] European Commission. *Digital Transformation of the Greek Industry*. European Commission, 2019.

- [7] European Commission. Greece 2025 Digital Decade Country Report. European Commission, 2025.
- [8] Matzka, S. Explainable artificial intelligence for predictive maintenance applications. In: Proceedings of the 2020 Third International Conference on Artificial Intelligence for Industries, 2020, 69-74. DOI: 10.1109/AI4I49448.2020.00023.
- [9] Bergmann, P., Fauser, M., Sattlegger, D. and Steger, C. MVTec AD: A comprehensive real-world dataset for unsupervised anomaly detection. In: Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, 9584-9592. DOI: 10.1109/CVPR.2019.00982.
- [10] Pejić Bach, M., Topalović, A., Krstić, Ž. and Iveć, A. Predictive maintenance in Industry 4.0 for the SMEs: A decision support system case study using open-source software. *Designs*, 2023, 7(4), 98. DOI: 10.3390/designs7040098.
- [11] Nasirinejad, M., Afshari, H. and Sampalli, S. Implementing digital twin for Maintenance 4.0 in SMEs: A framework for affordable and secure solutions. *IFAC-PapersOnLine*, 2025, 59(10), 130-135. DOI: 10.1016/j.ifacol.2025.09.024.
- [12] Li, Z., Yan, Y., Wang, X., Ge, Y. and Meng, L. A survey of deep learning for industrial visual anomaly detection. *Artificial Intelligence Review*, 2025, 58, 279. DOI: 10.1007/s10462-025-11287-7.
- [13] Singh, R. and Gill, S.S. Edge AI: A survey. *Internet of Things and Cyber-Physical Systems*, 2023, 3, 71-92. DOI: 10.1016/j.iotcps.2023.02.004.
- [14] dos Santos Oliveira, A., Schäfer Tesch da Silva, F., Lopes Ferreira, D., Roveda, M., da Costa, C.A., de Oliveira Ramos, G. and da Rosa Righi, R. Looking at anomaly detection in EdgeAI through the up-to-date lens of academic and market perspectives. *Cluster Computing*, 2026, 29, 251. DOI: 10.1007/s10586-026-05954-9.
- [15] UCI Machine Learning Repository. AI4I 2020 Predictive Maintenance Dataset. UCI Machine Learning Repository, 2020. DOI: 10.24432/C5HS5C.
- [16] MVTec Software GmbH. MVTec AD: Industrial Anomaly Detection Benchmark Dataset. MVTec Software GmbH.
- [17] OpenVINO. Model Optimization Guide. OpenVINO Documentation, Version 2023.3.
- [18] Hevner, A.R., March, S.T., Park, J. and Ram, S. Design science in information systems research. *MIS Quarterly*, 2004, 28(1), 75-105. DOI: 10.2307/25148625.
- [19] Peffers, K., Tuunanen, T., Rothenberger, M.A. and Chatterjee, S. A design science research methodology for information systems research. *Journal of Management Information Systems*, 2007, 24(3), 45-77. DOI: 10.2753/MIS0742-1222240302.