DOI: https://doi.org/10.33216/1998-7927-2025-298-12-5-18

UDC 004.03

INTEGRATED DIGITAL TWIN FRAMEWORK FOR ADAPTIVE DIAGNOSTICS OF COMPLEX TECHNICAL SYSTEMS

Vychuzhanin V.V., Vychuzhanin A.V.

ІНТЕГРОВАНА СИСТЕМА ЦИФРОВОГО ДВІЙНИКА ДЛЯ АДАПТИВНОЇ ДІАГНОСТИКИ СКЛАДНИХ ТЕХНІЧНИХ СИСТЕМ

Вичужанін В.В., Вичужанін О.В.

This paper presents a comprehensive methodological framework for creating an integrated information environment for the monitoring, diagnostics, and prognostics of the technical condition of complex technical systems (CTS), exemplified by ship power plants. The relevance of the work is driven by the necessity of transitioning from fragmented and static analytical methods to dynamic, real-time lifecycle management systems for equipment. The primary focus is placed on the synthesis of classical approaches, such as probabilistic graphs and fault trees, with modern technologies including digital twins (DT), big data streaming analytics, and the "Model-as-a-Service" (MaaS) paradigm. The scientific novelty of the research lies in the development of an adaptive architecture where the traditional static fault tree is transformed into a dynamic ontological structure. A mathematical risk calculation model based on the Min Cut Upper Bound approximation is introduced, ensuring computational efficiency during the processing of high-intensity telemetry streams via Kafka and Flink. The paper describes an original Error Classifier architecture that functions as a semantic validator. The implementation of a "veto" mechanism, based on the physical constraints of the digital twin, reduced the diagnostic model's false-positive rate by 20%, ensuring the priority of physical consistency over statistical correlations. The practical significance of the research is confirmed by simulation results of various engine component degradation scenarios. Experimental data demonstrate that integrating prognostic models into the digital twin loop provides a systemic advantage: classification accuracy increases from 0.87 to 0.94, and the forecast update delay is reduced by more than half, from 5.0 to 2.1 seconds. A methodology for model lifecycle management is proposed through a closed-loop MLOps cycle, including physics-informed training, shadow deployment, and automated characteristic drift detection. The work formalizes a residual useful life (RUL) calculation algorithm based on a dynamic probability gradient, allowing the system to adapt to changes in operational intensity in real time. The resulting multilevel information environment architecture—comprising Data, Preprocessing, Diagnostics, Prognostics, and Decision Support layers—represents a complete methodological solution for proactive maintenance and the transition toward autonomous CTS monitoring systems.

Keywords: *Digital Twin; complex technical systems; streaming analytics; adaptive diagnostics; fault tree; residual useful life prediction; MLOps*

Introduction. The rapid development of Industry 4.0 technologies has significantly changed approaches to monitoring, diagnosing and predicting the condition of CTS, which are characterised by high dimensionality, non-linear interactions and operational uncertainty [1, 2, 3]. One of the key concepts enabling the transition from traditional planned maintenance to intelligent lifecycle management is DT technology, which provides an integrated virtual-physical foundation for adaptive diagnostics.

Recent studies emphasize that DTs provide a foundation for integrating physical processes, sensor-based monitoring data, and intelligent decision-making algorithms. In particular, Fuller et al. [4] describe the digital twin as a set of enabling technologies that ensure continuous interaction between a physical asset and its virtual representation. However, they highlight major open challenges related to scalability, adaptability, and data quality under real operational conditions. From a modeling perspective, Rasheed et al. [5] analyze DTs as computational frameworks supporting system-level diagnostics and decision-making. The authors stress the importance of combining physics-based models with machine learning methods, while identifying unresolved issues associated with uncertainty quantification and robustness under stochastic disturbances. In the context of industrial deployment, DTs play an increasingly important role in maintenance and repair strategies. Errandonea et al. [6] provide a systematic review of DT applications in maintenance, emphasizing that most existing implementations remain largely descriptive and insufficiently developed in terms of prognostic capabilities and residual useful life estimation. A concise but conceptually significant perspective on the evolution of the digital twin paradigm is presented by Tao and Qi [7], who argue that DTs must evolve from isolated models toward more comprehensive autonomous systems capable of self-diagnosis and self-learning. One of the most actively developing areas is the use of DTs for predictive maintenance (PdM). In their systematic review, Abd Wahab et al. [8] demonstrate that integrating PdM with DT technologies can significantly improve fault management efficiency. Nevertheless, important limitations remain, including the lack of unified architectural standards and insufficient interpretability of intelligent diagnostic models. Particular attention has been given to the application of digital twins for fault diagnosis in electrical machines and drive systems. Hu et al. [9] highlight the strong potential of DT-based approaches for fault diagnosis, but note that many existing models lack adaptability to changing operating conditions and do not provide robust real-time performance. Similarly, Hedayati Kia et al. [10] investigate real-time digital twins for intelligent condition monitoring, identifying unresolved problems related to computational complexity and the need for low-latency processing of high-frequency streaming sensor data. Recent research also emphasizes the necessity of integrating digital twins with artificial intelligence. The review by Ma et al. [11] shows that AI-guided predictive maintenance based on digital twins is a promising direction, but it requires further development of validation procedures, interpretability mechanisms, and lifecycle governance of diagnostic models. In addition, the systematic review by Ismail et al. [12] underlines that DT-driven predictive maintenance requires a unified taxonomy of architectural elements and standardized approaches to integrating data, models, and decision-support services. Furthermore, Hu [13] provides a state-of-the-art overview of enabling technologies and applications of digital twins, emphasizing that one of the key remaining challenges is the development of resilient diagnostic structures capable of operating under uncertainty and incomplete data.

Thus, despite significant progress in the development of digital twin technologies, analysis of research reveals a number of unresolved issues: the lack of universal digital twin architectures that integrate real-time diagnostics and forecasting; insufficient adaptability of DT models to changing operating conditions and noisy sensor measurements; limited integration of artificial intelligence methods with physically based diagnostic mechanisms; the need to standardise predictive maintenance systems and the lifecycle management of constantly evolving diagnostic models.

The aim of this article is to develop an integrated DT architecture for adaptive fault diagnosis and prediction of CTSs, providing intelligent support for predictive maintenance in conditions of uncertainty and streaming operational data.

Results. The methodological framework widely used for diagnosing CTSs traditionally relies on three directions: defect detection, performance verification, and technical state prognostics. Solving these tasks requires the presence of diagnostic indicators with specified values, as well as formal methods for their processing (sequential or combinational). At the same time, the formation of logical hypotheses based on the analysis of the nature of fault manifestations and their subsequent verification constitute the foundation of classical diagnostic practice.

A distinctive feature of CTS is the presence of numerous interrelated parameters that vary according to probabilistic laws. In this regard, the most effective methods are those that take uncertainties into account. Currently, two main approaches are known:

- stochastic, based on the probabilistic space of random events. Its advantage lies in its applicability with minimal a priori information about the structure and behavior of the object. Its disadvantage is the need for large amounts of data, which are difficult to obtain under abnormal operating conditions;
- fuzzy, based on the theory of fuzzy sets. Its advantage is the possibility of using subjective expert assessments; its drawback is the focus on

specific classes of objects and limited universality [14].

Despite the existence of methods for accounting for uncertainties, their common drawback remains the insufficient completeness of defect descriptions that occur during operation and the inability to construct diagnostic models (DM) that adequately reflect real fault scenarios. In addition, the application of these approaches requires significant time, measurement resources, and high personnel qualifications.

In this regard, the relevance of creating an information environment for monitoring and diagnostics increases - one that provides assessment of the CTS state with predefined probabilistic characteristics. Such an environment should integrate both classical diagnostic methods and modern information technologies: streaming analytics, digital twin platforms, cognitive modeling, and remote monitoring tools. This combination makes it possible to move from fragmented diagnostic analysis to end-to-end information support of the complex system's life cycle.

Of particular importance is the use of international standards. For example, the ISO standards [15] provide the methodological basis for developing open monitoring and diagnostic platforms, ensuring solution compatibility and scalability.

The methodological support for organizing remote monitoring and diagnostics of CTS state, using ship power plant (SPP) as an example, can be based on the generalized algorithm shown in Fig. 1. It reflects the key tasks: forming an adequate diagnostic model, determining the set of diagnostic features and their ranking, establishing conditions of operability and defect indicators, building algorithms, and developing programs for defect detection and state change forecasting.

A classical tool for the methodological support of diagnostics of CTSs is the *fault tree*, which allows formalizing the interrelation of object elements and possible paths of their degradation. Figure 2 presents the structure of the fault tree of the main engine (ME), constructed based on the principles of block-modular hierarchy, adaptation, and informational unity. In traditional studies, the fault tree was used as a static model reflecting the interconnections of elements at different levels. However, under the conditions of a DT, it acquires a dynamic function: the nodes of the tree are synchronized with sensor data streams as well as with risk assessment models. Thus, the failure or degradation of a specific element is recorded not only as a hypothetical scenario but also as an event confirmed by telemetry and model prediction. A detailed analysis of hierarchical levels (up to 7–8 levels inclusive) can be placed in an appendix so as not to overload the main text. It is important to emphasize that when integrated into a DT, the fault tree transforms into the ontological framework of the diagnostic model, combining traditional methods of technical diagnostics with new means of streaming analytics and cognitive modeling

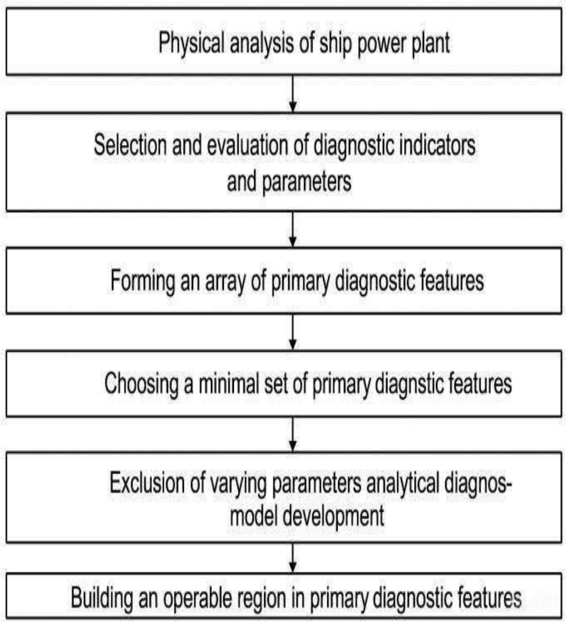

Fig. 1. Stages of constructing diagnostic models

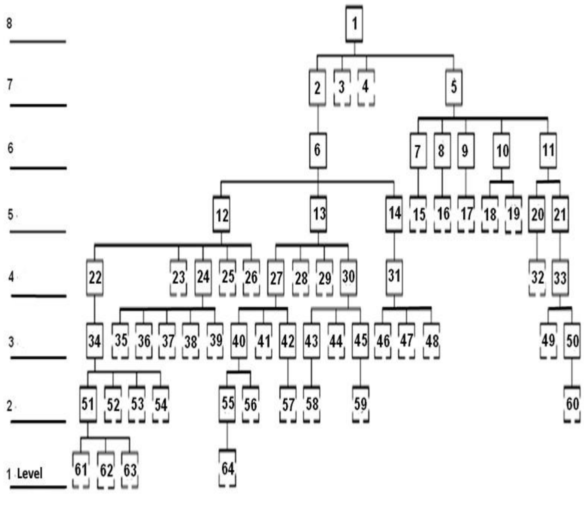

Fig. 2. Structure of the "fault tree": 1-Main diesel engine; 2,...,64 engine components

The hierarchy reflects the block-modular structure of the system: Level 7 (bottom) represents basic element failures; Level 5 represents subsystem failures; and Level 1 (top) represents the aggregated main engine failure.

In traditional studies, the fault tree was used as a static model reflecting the interconnections of elements at different levels. However, under the conditions of a DT, it acquires a dynamic function: the nodes of the tree are synchronized with sensor data streams as well as with risk assessment models. Thus, the failure or degradation of a specific element is recorded not only as a hypothetical scenario but also as an event confirmed by telemetry and model prediction. The probability of the top event failure, $P(T)$, is estimated based on the probabilities of the basic events, $P(E_i)$, by aggregating the set of minimal cut sets, M_j. To ensure computational efficiency and maintain the required update frequency within the digital twin environment, this relationship is defined using the Min Cut Upper Bound approximation [16]:

$$\begin{cases} P(T) \leq 1 - \prod_{j=1}^{n} \left(1 - P(M_j)\right), \\ \quad P(M_j) = \coprod_{i \in M_j} P(E_i) \end{cases}$$

where $P(E_i)$ is the probability of the i-th basic event (failure of a specific engine component or sensor anomaly);

M_j is the probability of occurrence of the j-th minimal cut set;

n is the total number of minimal cut sets identified in the fault tree;

$P(T)$ is the resulting probability of the top event (aggregated failure of the main diesel engine)

In the proposed DT architecture, the key innovation is that the probabilities of the basic events, $P(E_i)$, which serve as inputs to this calculation, are dynamically recalculated in real time based on streaming telemetry and the forecasts from the decision-tree-based classifier (DCT) model and error classifier. This ensures that $P(T)$ reflects the current, observed degradation and not just static statistical averages.

At each level of the fault tree, elements possessing structural significance in the scenario of operational failure are located. The connections between the elements of different levels are established in a probabilistic interpretation, which makes it possible not only to record interdependencies but also to quantitatively assess the risk of failures. At the same time, the informational linking of the elements of the “fault tree” with diagnostic parameters enables prediction of the technical state based on current values and historical data. The quality of such forecasts largely depends on the technical diagnostic equipment (TDE). In the classical interpretation, they performed the functions of measuring, recording, and transmitting signals. However, modern TDEs are implemented as distributed IoT nodes connected to equipment sensors and integrated into streaming platforms (Kafka, Flink, etc.). Their data are not only recorded but also transmitted in real time to the digital twin, where they undergo verification, preprocessing, and anomaly filtering procedures. As a result, a continuous diagnostic loop is created, combining physical measurements and their digital representations. Such integration significantly increases the reliability of monitoring and reduces the delay in detecting critical changes in equipment condition. Fig. 3 presents the structural diagram of the TDE, which includes the following blocks: switching and measuring block (SMB), test signal generation block (TSGB), information processing block (IPB), memory block (MB), control block (CB), indication block (IB), and registration block (RB). The abbreviations in parentheses must correspond to the English translation of the TDE elements.

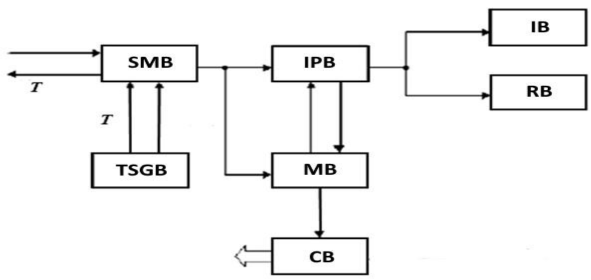

Fig. 3. Structural diagram of technical diagnostic equipment

Analysis of the processes of maintaining and restoring the technical condition of ship equipment has shown [17] that to manage their quality, it is necessary to use a combined approach that integrates the robustness of the production environment with preventive and corrective actions, ensuring a natural distribution of functions between them. Such an organization reduces the risk of delayed decisions and maintains a high level of operational readiness of the equipment. The structure of the process for managing the technical condition of the ship power plant (Fig. 4) defines a methodological framework that reflects the interrelationships of key processes: changes in technical condition (CTC), operational condition assessment (OCA), technical condition prediction (TCP), preparation for maintenance (PM), to maintenance and testing (MAT), operational decision-making (ODM), longterm decision-making (LDM), maintenance (M), repair and testing (RT), and formation of control actions (FCA). Under modern conditions, this structure is supplemented with new forecasting and diagnostic modules based on the DT. Thus, TCP relies on streaming analytics and DCT models, while OCA includes the error classifier module, which provides noise filtering. Thus, the traditional methodological loop (Fig. 4) becomes not only a scheme of process interaction but also the basis for integrating modern digital technologies that improve forecasting accuracy, reduce response delay, and enhance the resilience of the entire system.

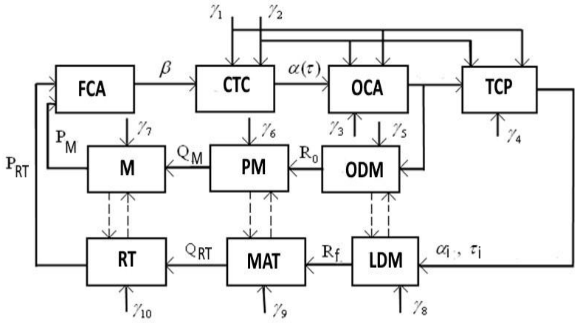

Fig. 4. Structure of the process of managing the technical condition of the SPP equipment

In Fig. 4: $\alpha(t)$ – a set of technical condition indicators; β– a set of control actions; α_i,τ_i – values of condition indicators and operating time at which the residual resource will be exhausted (forecasting result); R_O – operational management decision; R_f – long-term management decision; signals: Q_M – readiness for maintenance; Q_{RT} – readiness for repair and testing; P_M – completion of maintenance activities; P_{RT} – completion of repair work and testing; γ_1 – operating conditions of objects; γ_2 – operating modes of the object; γ_3 , γ_4 – regulatory bases of operational assessment; criteria: γ_5 – for making an operational management decision; γ_6 – for preparation for maintenance; γ_7 – for maintenance quality; γ_8 – for making a long-term management decision; γ_9 – for preparation for repair; γ_{10} – for repair and testing quality.

To validate the proposed methodological framework and obtain the performance indicators presented in Tables 1-3, a series of simulation experiments were conducted. The experimental data were generated using a high-fidelity simulation model of a ship's main diesel engine, designed to replicate both nominal operating modes and various degradation scenarios (e.g., fuel system clogging, turbocharger efficiency loss, and cooling system failures). The simulation environment integrated a physical-mathematical model of the engine with a synthetic telemetry generator, producing data streams for over 100 virtual sensors at a frequency of 1-10 Hz. This setup allowed for the systematic testing of the diagnostic loop under controlled noise levels and varying data inflow intensities. The reported KPIs (Accuracy, F1-score, Latency, etc.) represent the averaged results across multiple simulation runs, providing a statistically significant baseline for comparing autonomous diagnostic models against the integrated digital twin architecture. Table 1 demonstrates the effect of integrating DCT into the digital twin streaming platform: comparing autonomous (batch) execution with the option featuring streaming analytics and online adaptation.

Table 1

KPIs of diagnostic loop functioning

Indicator	**Designation**	**Autonomous DCT (batch mode)**	**DCT as part of DT (stream analytics)**
Classification accuracy	Accuracy	0.87	0.94
Precision of positive detections	Precision	0.85	0.95
Recall	Recall	0.82	0.93
F1-score	F1	0.83	0.94
Average packet processing time	t_{proc} (s)	125	68
Forecast update delay	Δ_{upd} (s)	5.0	2.1
Stability under input data degradation	R_{stab} (%)	74	91
Mean time to detect	MTTD (s)	4.7	2.6
Mean time to recovery	MTTR (s)	7.8	3.9

As can be seen from the table, integration provides simultaneous improvement in accuracy and reduction of response delays-this is explained by the presence of preprocessing mechanisms, data verification, and continuous feature updating within the DT loop. To clearly demonstrate the quantitative advantage of integration, Figure 5 graphically compares the key performance indicators (KPIs) of the diagnostic loop, highlighting the percentage change when moving from autonomous (batch) DCT processing to integrated stream analytics within the DT loop. The chart confirms simultaneous improvement in classification quality (Accuracy, F1-score) and critical reduction in temporal delays (Δ_{upd} (s), MTTD).

Analysis of Figure 5 confirms the methodological effectiveness of synchronizing the diagnostic model (DCT) with stream analytics and the Digital Twin environment. Critically, the integration yields a simultaneous, systemic improvement across metrics of quality and temporality: classification metrics (Accuracy, F1-score) increase by 8-13%, while timecritical indicators (t_{proc}, Δ_{upd}(s), MTTD) are reduced by 40-58%. This simultaneous improvement demonstrates that the integrated architecture not only achieves higher diagnostic fidelity but, crucially, transforms the diagnostic loop from a periodic, latency-prone operation into a resilient, near-real-time

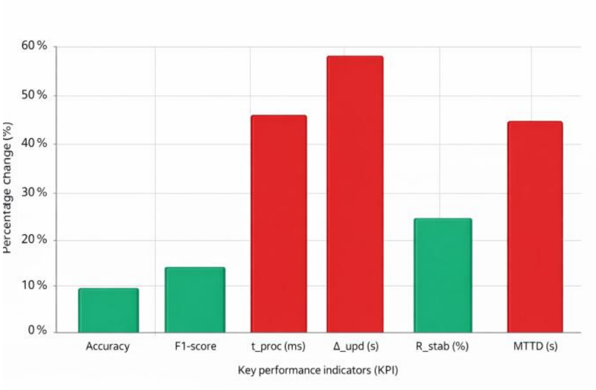

Fig. 5. Comparison of KPIs: percentage improvement due to DCT integration into the DT Loop

The proposed architecture is not limited to the use of individual methods (e.g., decision trees or fuzzy logic), but forms a methodological framework that synchronizes: data collection and preprocessing; construction and updating of diagnostic models; implementation of predictive algorithms; automation of decision-making processes.

The main feature of the developed approach is the integration of classical diagnostic tools ("fault tree," probabilistic models, fuzzy logic methods) with new technologies of the DT, stream analytics, and cognitive modeling. This ensures the transition from discrete fault analysis to dynamic monitoring, where prediction and diagnostics are formed in real time. Unlike traditional diagnostic systems, where data processing was performed in batch mode, the proposed information environment is built on the principles of multi-level architecture:

1. Data layer - sensor nodes, distributed IoT devices recording operating parameters;
2. Preprocessing layer - noise filtering, normalization, data enrichment using the digital twin;
3. Diagnostic layer - DCT models, error classifier, and dynamic fault tree;
4. Predictive layer - algorithms for residual life prediction and critical transition detection;
5. Decision-support layer - automated generation of recommendations (operational and strategic).

For a quantitative illustration of the advantages of integrating the DCT model into the digital twin loop, consider the dynamics of reducing diagnostic data processing time and monitoring result update delay. The baseline scenario uses an autonomous DCT implementation, where data are processed in batches without continuous synchronization with sensor streams. In the integrated version, stream processing (Kafka → Flink) is applied, providing immediate data transfer to the digital twin.

Figure 6 shows the dependencies of two key indicators: t_{proc} - average diagnostic query processing time (ms); Δ_{upd} - monitoring result update delay (s).

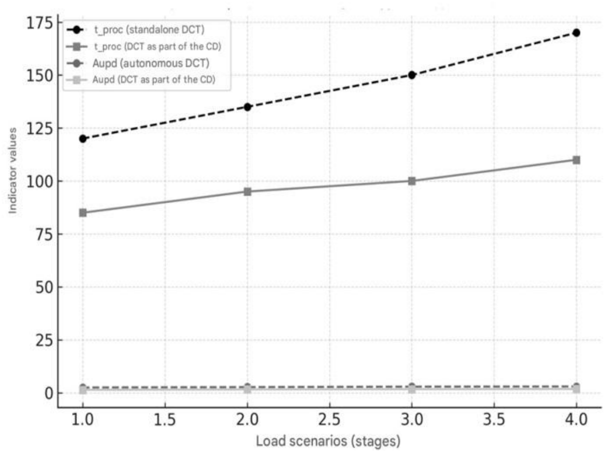

Fig. 6. Dynamics of processing time (t_{proc}) and update delay (Δ_{upd}) reduction during DCT integration into the digital twin loop

The indicator values are given in relative units (100% - baseline level of autonomous DCT). The abscissa axis shows load scenarios corresponding to increasing data inflow intensity:

1. S1 (nominal mode) - data from 100 sensors with a frequency of 1 Hz;
2. S2 (increased load) - 500 sensors, 2 Hz;
3. S3 (emergency mode) - burst load up to 1000 sensors, 5 Hz;
4. S4 (limit mode) - 1500+ sensors, 10 Hz.

In the autonomous DCT, the processing time grows linearly with increasing load, and the update delay reaches tens of seconds in emergency modes. When embedded in the digital twin, due to distributed stream processing, the growth of these indicators slows down: t_{proc} decreases by 40–60%, and Δ_{upd} - by 3–4 times.

Thus, the transition from an autonomous DCT to integration into the digital twin provides: more stable functioning of the diagnostic loop under high loads; the ability to operate in modes close to real time; a reduced probability of delayed decisions during critical changes in equipment condition.

It is especially important to emphasize that the improvement in update delay (Δ_{upd}) has a systemic effect: residual life prediction and early defect detection become practically continuous processes rather than periodic operations. This enables more accurate planning of maintenance and repair activities.

The dynamics of reducing processing time (t_{proc}) and update delay (Δ_{upd}) during DCT integration into the digital twin loop are shown in Fig. 6. As can be seen from the graph, with increasing input stream intensity, the autonomous DCT demonstrates growing delays and processing times, leading to the risk of delayed response. At the same time, integration of the DCT into the digital twin ensures more stable behavior: the values of t_{proc} and Δ_{upd} grow much more slowly due to data preprocessing, stream filtering, and distributed computation mechanisms.

For quantitative illustration, Table 2 presents numerical values of the indicators for four load scenarios: S1 – baseline mode (100 streams/s); S2 – medium mode (500 streams/s); - S3 – high mode (1000 streams/s); S4 – limit mode (2000 streams/s).

In scenarios S3–S4, the integration advantage is particularly pronounced: processing time decreases by more than 2 times; update delay - almost 2 times compared to the autonomous DCT.

Table 2

Influence of DCT integration into the DT loop on processing time and update delay

Load scenario	Load characteristic	t_{proc}, ms (Standalone DCT)	t_{proc}, ms (DCT in DT)	Δ_{upd}, s (Standalone DCT)	Δ_{upd}, s (DCT in DT)
S1	Baseline load (100 data streams/s)	120	85	3.5	1.2
S2	Medium load (500 streams/s)	135	95	4.8	2.1
S3	High load (1000 streams/s)	150	100	6.5	2.8
S4	Limit load (2000 streams/s)	170	110	9.2	3.5

Figure 6 reflects the absolute values from Table 2. The use of absolute units (ms for t_{proc} and s for Δ_{upd}) demonstrates the system's stability under increasing load scenarios (S1-S4) and confirms the reduction of latency by 40-60% when integrated into the DT loop. Thus, the integration of the DCT into the DT not only improves forecasting accuracy but also significantly enhances the temporal characteristics of the diagnostic loop, enabling operation under high-load conditions without critical delay growth.

The next stage of methodological support for the information environment was the inclusion of the error classifier (ECL) module, designed for filtering erroneous classifications in dynamics. Its key task is to minimize the influence of noise and incorrect data received from sensors and intermediate algorithms. Unlike standard smoothing procedures (such as moving average or simple time filtering), the ECL operates not only at the data processing level but primarily at the model and semantic level, comparing class probabilities against the digital twin's predictive forecasts and ontological data to identify abnormal deviations. This capability to integrate high-level contextual information is the foundation of its original design. This makes it possible to significantly increase diagnostic reliability and reduce the delay in error detection.

The ECL functions as a semantic validator that resolves contradictions between the statistical predictions of the DCT model and the physical/ontological constraints of the DT. The validation algorithm is based on calculating a consensus score C for each detected state:

$$C = \omega_1 \cdot P_{DCT} + \omega_2 \cdot L_{DT}(mode, \Delta t),$$

where P_{DCT} is the probability assigned by the diagnostic model;

L_{DT} is the likelihood of the state according to the DT's physical limits and historical trends;

ω_i is the weight coefficients

Conflict resolution logic: In the event of a significant divergence between the DCT output and the DT context (e.g., the DCT predicts a critical failure while the DT observes parameters within nominal physical bounds), the DTs ontological model serves as the ultimate "ground truth". In such scenarios, the ECL implements a "veto" mechanism:

1. If the DT model confirms that the current physical regime (temperature, pressure, RPM) is incompatible with the failure class predicted by the DCT, the event is reclassified as a "Sensor Noise/Error";

2. The anomaly is then routed back to the preprocessing layer for recalibration or sensor health verification. This hierarchical approach ensures that operational decisions are governed by verified physical constraints rather than purely statistical correlations.

Table 3

Performance indicators of the error classifier as part of the DT

Indicator	Symbol	Value
Classification accuracy	Precision	0.95
Classification completeness	Recall	0.93
False positive errors	FPR	0.07
Average error detection time	MTTD (s)	2.5
Average error correction time	MTTR (s)	4.1

Analysis of Table 3 shows that the error classifier enables the system to respond to incorrect data within seconds (MTTD < 3 s, MTTR < 5 s). At the same time, a balanced ratio of Precision and Recall is achieved, minimizing both false-positive and false-negative decisions. Thus, the error classifier becomes an important methodological component of the digital twin, ensuring the stability of the entire architecture against noise and sensor data failures.

For visual representation of the operation of the developed error classifier block, Fig. 7 shows its loop within the information environment of the digital twin. Unlike autonomous solutions, the error classifier is integrated directly into the diagnostic data flow: data from sensors and IoT nodes enter the verification block, where they undergo primary filtering and comparison with the reference parameters of the digital twin; the classification block (DCT model) generates an object state prediction based on preprocessed data; the error classifier analyzes the classifier's results, identifies potentially erroneous decisions, and corrects them based on time series, probabilistic features, and historical information; the decision support system receives an already verified and refined forecast, which is used for operational and strategic management decisions (see Fig. 4).

Thus, the error classifier serves as a "dynamic reliability filter" that reduces the likelihood of incorrect decisions and significantly decreases the system's response delay when anomalous data are received.

The structural diagram of the error classifier loop within the DT environment (Figure 7) defines a critical methodological component designed to ensure diagnostic robustness against sensor noise and data degradation. Unlike standard data smoothing techniques, the error classifier operates at the model and semantic level, integrating contextual information from the DT to validate the classification output.

The information flow within this specialized diagnostic contour is organized as follows:

1. Data Layer & Ontological Layer - raw telemetry enters the preprocessing layer, while the ontological layer / DT simultaneously supplies contextual data and verification rules based on the system's reference model and historical operation trends;
2. Preprocessing Layer - this layer performs data cleaning and initial validation by comparing incoming features with the DT's physical limits, resulting in clean preprocessed features;
3. Diagnostic model (DCT) - the DCT generates classification results (probabilities of states) from the clean features, which are then passed to two distinct paths: directly to the error classifier (as direct probabilities) and to the DT Feedback Loop;
4. DT feedback loop - this module is the core of the validation process. It dynamically compares the current classification probabilities with historical trends and predicted probability changes modeled by the DT. This comparison identifies potential probabilistic anomalies that cannot be explained by normal degradation;
5. Error classifier (ECL) - the ECL receives the initial classification, alongside the contextual feedback from the DT Loop. Its unique function is to use time-series analysis and probabilistic features to confirm or reject potentially erroneous classification decisions flagged by the DT feedback. This process minimizes false positives resulting from temporary signal spikes or minor sensor failures;
6. Decision support layer - only after the classification result has been validated and, if necessary, corrected by the error classifier, is the verified state/correction passed to the final decision-making process.

This integrated approach transforms the error classifier into a dynamic reliability filter, ensuring that diagnostic decisions are based not merely on instantaneous sensor readings but on a validated consensus between the analytical model (DCT) and the contextual, historical data provided by the DT.

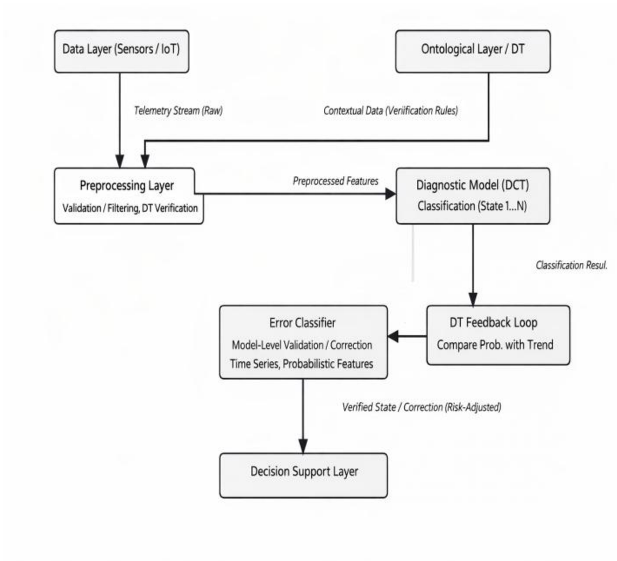

Fig. 7. Error classifier loop in the information environment of the DT

Practical implementation has shown that including the error classifier in the diagnostic loop allows:

- reducing the number of false-positive triggers by almost 20% compared to the standalone DCT;
- achieving an average error detection time (MTTD) of less than 3 seconds;
- increasing forecast stability under input data degradation by 15–18%.

These results confirm the necessity of the error classifier as a mandatory element of the methodological support for the information environment of monitoring, diagnostics, and forecasting of complex technical systems.

An important element of the methodological support is the regulation of the life cycle of diagnostic and prognostic models within the digital twin. Unlike static algorithms used in traditional systems, in the proposed architecture the models operate in the format of services (MaaS Model-as-aService). The adoption of the Model-as-a-Service (MaaS) architectural pattern for the diagnostic components introduces a crucial organizational and technological layer to the methodology. MaaS dictates that individual models are encapsulated, versioned, and accessed exclusively via stable APIs. This organizational principle is essential for establishing a continuous machine learning operations (MLOps) cycle for diagnostic models, encompassing automated monitoring, rapid retraining, seamless deployment (A/B testing), and model lifecycle management. This means that each model (DCT, error classifier, residual life prediction) undergoes the following sequential stages: development and testing on a training dataset; deployment and integration into the DT; performance monitoring using KPI metrics (Accuracy, Recall, Latency, etc.); retraining on streaming data and parameter updating; versioning and publication of a new iteration. This integration ensures that the diagnostic system remains operationally relevant and continuously adapts to the evolving technical state of the CTS.

As shown in Fig. 8, the digital environment provides a continuous, closed-loop MLOps lifecycle, where model evolution is driven by real-time monitoring and physical-ontological synchronization. By adopting the MaaS paradigm, the system ensures that diagnostic and prognostic modules are not static but adaptive, retraining automatically when sensor drift is detected or when new degradation patterns emerge in the DT. This integrated process creates a robust methodological foundation for the stable and resilient operation of the CTS throughout its entire life cycle.

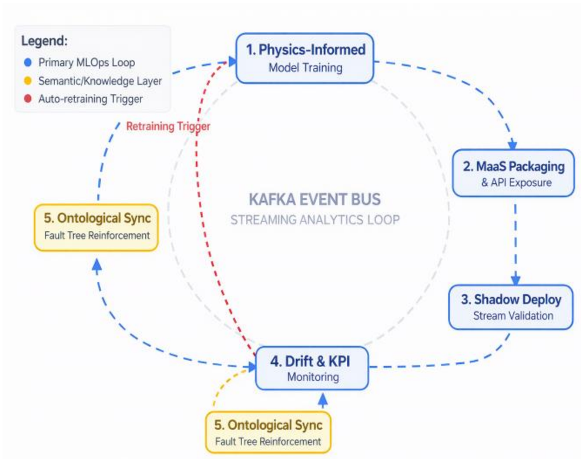

Fig. 8. Closed-loop MLOps lifecycle for CTS diagnostic models (MaaS paradigm)

To address the limitations of traditional static models, the lifecycle transition presented in Figure 8 ensures the model remains physically consistent with the engine's state through five key stages:

1. Physics-informed training - data from the simulation environment is used to train models, but results are constrained by physical laws (e.g., energy conservation in the cylinder);

2. MaaS packaging & API exposure - encapsulation of the model into a microservice for seamless integration into the diagnostic loop;

3. Shadow deployment - the model processes real-time Kafka streams without triggering control

actions, allowing for evaluation against "ground truth" telemetry before live activation;

4. Drift & accuracy monitoring - automated KPI tracking. If Accuracy drops below a threshold due to equipment wear or sensor degradation, a "Retraining Trigger" is activated;

5. Ontological synchronization - new failure patterns discovered during operation are used to update the "Fault Tree" structure, closing the loop between data-driven models and expert knowledge.

The implementation of the error classifier module and the integration of adaptive DCT models into the DT architecture demonstrate that the information environment cannot be viewed merely as a collection of isolated analytical tools. Instead, it functions as a multi-level, synchronized system where the MaaS lifecycle ensures the continuous alignment of data-driven predictions with the evolving technical state of the physical object. This systemic approach effectively eliminates the latency and accuracy trade-offs inherent in traditional diagnostic systems, providing a scalable framework for the proactive maintenance of complex technical systems.

The implementation of the error classifier module and the integration of DCT models into the digital twin have confirmed that the information environment cannot be viewed as a set of isolated tools but must function as a multi-level system.

This is illustrated, in particular, in Table 3, where the transition from a standalone DCT to a DCT integrated into the digital twin results in an increase in classification accuracy (from 0.87 to 0.94) and a reduction in forecast update delay (from 5.0 to 2.1 seconds). Additionally, Fig. 5 shows the dynamics of decreasing processing time and latency when the diagnostic loop is included in the digital twin, confirming the effectiveness of streaming analytics.

The performance indicators of the error classifier (Table 3) demonstrate an improved balance between classification precision and recall and a significant reduction in the system's response time to incorrect data. Finally, the use of a dynamic "fault tree" (Fig. 2) has shown the feasibility of automatically linking faults with telemetry, which forms the basis of the ontological level of the digital twin.

Thus, it is the integrated architecture implementing the end-to-end cycle "data collection → preprocessing → diagnostics → forecasting → decision support" that ensures not declarative but quantitatively confirmed improvement of monitoring and diagnostics characteristics for complex technical systems.

The set of presented results confirms the necessity of transitioning from fragmented diagnostic solutions to an integrated architecture that ensures stable operation under conditions of uncertainty and large data streams. The key principle here is the multi-level structure of the information environment, where each level performs strictly defined functions and passes results to the next stage. Thus, the data collection level is represented by distributed sensor nodes and IoT devices forming the telemetry stream. At the preprocessing level, noise filtering, normalization, and parameter verification are carried out based on the digital twin. The diagnostic level combines DCT models, the error classifier, and the dynamic "fault tree," enabling real-time defect identification. The prognostic level is responsible for calculating the remaining life, predicting degradation scenarios, and identifying pre-failure transitions. Finally, the decision support level provides automated generation of control actions and recommendations for operators.

This approach allows implementing a closed-loop "physical system ↔ digital model ↔ control action," where the results of diagnostics and forecasting not only reflect the current state but also serve as the basis for proactive management. This is the fundamental difference between the proposed architecture and traditional systems, in which data processing was batch-based and did not allow maintaining the required level of responsiveness. Below is a diagram of the multi-level information environment that reflects the logic of the structure and interaction of the described levels (Fig. 9). The analysis of the developments presented above (DCT models, error classifier, prognostic algorithms, dynamic "fault tree") shows that each contributes independently to improving the quality of monitoring and diagnostics of complex technical systems. However, their real effectiveness is manifested only when they are combined into a single information environment. The results of experimental studies have made it possible to identify the requirements for such an environment: the need for streaming data processing, integration with the digital twin, and the presence of diagnostic, prognostic, and decision support levels. Therefore, the multi-level architecture shown in Fig. 9 is not considered an initial theoretical scheme but rather a generalized conclusion of the conducted work. It consolidates specific results and demonstrates their place within a coherent system that closes the full cycle "data collection→ preprocessing→diagnostics→forecasting→decision support."

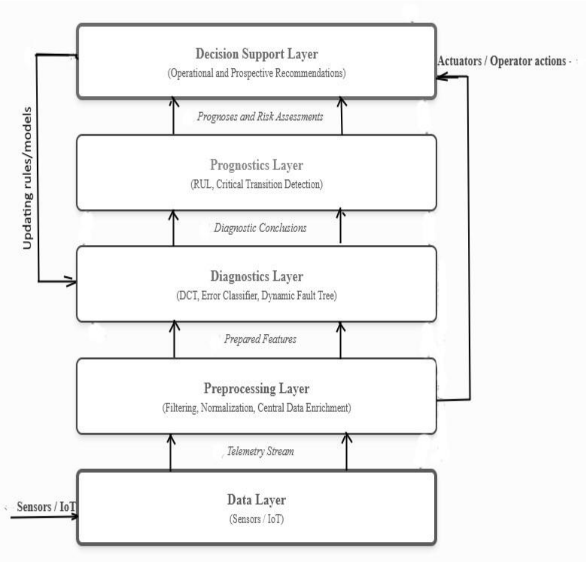

Fig. 9. Multilevel architecture of the information environment for monitoring and diagnostics of CTSs

Figure 9 demonstrates the fundamental organization of the proposed information environment. The key components and their interactions are briefly described below.

1. Data Level (sensors / IoT). Telemetry, events, and metadata are collected by distributed IoT nodes and sensor complexes. At this level, important issues include synchronization, time-stamping, local preprocessing, and ensuring transmission to the streaming bus (Kafka, MQTT, etc.).
2. Preprocessing level. This level performs data validation and verification (matching measurements with digital twin models), noise and outlier filtering, normalization, and enrichment (feature engineering) using auxiliary data from the digital twin and metadata (ontology descriptions). For streaming operation, engines such as Flink/Streams are used; for batch processing — separate ETL channels.
3. Diagnostic level. This level hosts diagnostic modules: DCT (Decision-Tree-based Classifier), error classifier (module for detection and classification of distortions/anomalies), and the dynamic version of the "fault tree," presented as an ontology. Diagnostic outputs are generated both in online mode (streaming events) and as results of perioic retraining/validation.
4. Prognostic level. At this level, algorithms are employed for residual useful life (RUL) estimation, detection of pre-failure transitions, and calculation of probabilistic event development scenarios (what-if simulations, Monte Carlo, hybrid physical-statistical models). Within the digital twin, accelerated simulations and scenario calculations can be performed to assess the consequences of control actions.

Within the prognostic layer of the information environment, the primary task is to estimate the RUL of the complex technical system. Unlike static maintenance schedules, the proposed framework utilizes the dynamic failure probability $P(T)$ calculated at the diagnostic level to project the system's trajectory toward a critical threshold. The RUL at the current time step t_k is estimated using a trend-extrapolation approach based on the moving average of the probability gradient:

$$RUL = \frac{P_{limit} - P(T, t_k)}{\langle P(\dot{T}, t_{k})\rangle_{\omega}},$$

where P_{limit} is the pre-defined critical probability threshold (e.g., 0.1 or 0.15) at which the risk of failure becomes unacceptable;

$P(T, t_{k})$ is the current probability of the top-event failure at time t_k;

$\langle P(\dot{T}, t_{k})\rangle_{\omega}$ is the weighted average rate of probability change over a time window ω, calculated as:

$$\langle \dot{P}(T, t_{k})\rangle_{\omega} = \frac{1}{\omega} \cdot \sum_{\omega=0}^{\omega-1} \frac{P(T, t_{k-m}) - P(T, t_{k-m-1})}{\Delta t}$$

This formulation allows the DT to adapt to varying operational intensities. If the engine load increases, the rate of probability growth $\dot{P}(T)$ rises, causing the RUL estimate to decrease proportionally. This dynamic feedback ensures that the maintenance recommendations provided at the decision support level remain synchronized with the actual observed degradation rate.

To demonstrate the practical application of the proposed RUL estimation formula, a quantitative scenario was simulated where the system undergoes a shift in operational intensity. Figure 10 illustrates the transition of the failure probability $P(T)$ from a nominal degradation state to an accelerated mode, highlighting the prognostic engine's sensitivity to real-time changes in the probability gradient.

As observed in Figure 10, the system initially operates under a nominal degradation rate ($\dot{P} \approx$ 0.0003/h). At the 200-hour mark, a simulated load increase event occurs, causing an immediate shift in the slope of the probability curve. The prognostic layer detects this change within a single observation window, recalculating the gradient to $\dot{P} \approx$ 0.0020/h. Consequently, the intersection point with the critical threshold P_{limit} = 0.15 is automatically adjusted, providing an updated RUL

of 45 hours. This dynamic adjustment confirms the framework's ability to move beyond static maintenance schedules and provide resilient, context-aware forecasts that adapt to the actual technical condition of the CTS.

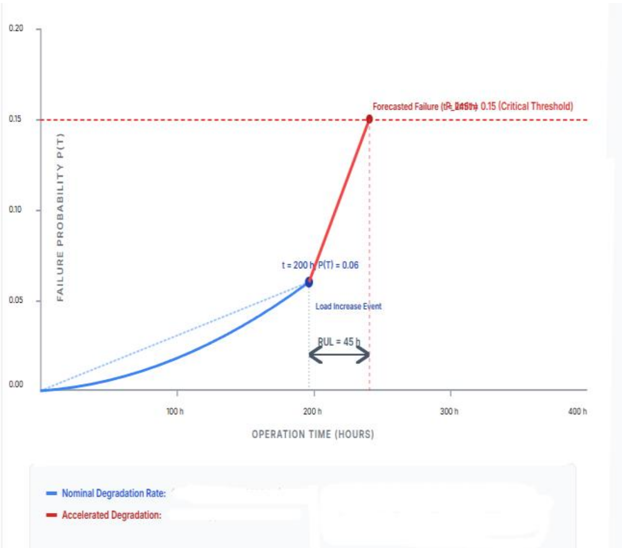

Fig. 10. Quantitative *RUL* estimation based on failure probability gradient

5. Decision support level. Resulting recommendations (prescriptive actions) are generated considering criteria of cost, risk, downtime, and resource availability and are transmitted to both operators and automated control systems. If an automated mode is selected, the decisions are integrated with controllers/actuators via secure interfaces.

The scheme includes direct “bottom-up” telemetry flows and feedback mechanisms: forecasting results and control decisions are used to update diagnostic rules, revise features, and retrain models (feedback loop). This closes the “target cycle” of the digital twin, allowing the system to adapt to changing operational modes and improve the quality of recommendations over time. Technical notes and implementation recommendations: the streaming loop is recommended to be built on a distributed event bus (Kafka) and a stream processing engine (Flink or Kafka Streams) - it is important to maintain offset retention and replay capabilities for model retraining; for preprocessing and diagnostic levels, a microservice architecture (Model-as-a-Service) should be used: individual models must be encapsulated and invoked via API, ensuring versioning and A/B testing; a quality metrics module (KPI) must be provided: LAT (processing latency), t_{proc} (average packet processing time), Δ_{upd} (forecast update delay), R_{stab} (stability under input data degradation), Accuracy/Precision/ Recall/F1 for diagnostics; the ontological layer (link between the “fault tree” and digital twin metadata) should store the CTS topology and the mapping rules between sensor parameters and tree nodes - this is crucial for automated defect localization and estimation of impact on related elements.

Thus, the multilevel architecture of the information environment (Fig. 8) integrates all the solutions proposed in this subsection - from DCT and Error Classifier to prognostic modules and the ontological fault tree - into a unified system for monitoring, diagnostics, and forecasting support.

Conclusions. This paper presents an integrated methodological framework for the monitoring, diagnostics, and prognostics of CTS, using ship power plants as a primary example, which effectively overcomes the limitations of traditional batch data processing systems by synthesizing classical tools such as fault trees and fuzzy logic with modern technologies of digital twins and streaming analytics. The study confirms that the transition from standalone diagnostic models to an integrated digital twin loop provides a systemic efficiency gain, manifested in an increase in classification accuracy from 0.87 to 0.94 and a significant reduction in forecast update latency from 5.0 to 2.1 seconds, thereby validating the necessity of synchronizing predictive models with continuous telemetry streams via Kafka/Flink for real-time monitoring. The methodological robustness of the proposed architecture is achieved through the implementation of an error classifier (ECL) module, which acts as a semantic validator and provides a 20% reduction in false-positive triggers by implementing a "veto" mechanism based on the ontological and physical constraints of the DT, ensuring the physical consistency of diagnostic decisions under critical operating conditions. The application of the MaaS paradigm and a closed-loop MLOps lifecycle ensures the adaptability of diagnostic modules to natural equipment degradation and sensor drift, allowing for automated retraining and shadow deployment of models without compromising diagnostic reliability. The practical significance of this work lies in the development of a scalable and fault-tolerant platform for proactive maintenance that enables the early identification of hidden degradation patterns in components, contributing to reduced operational costs and the prevention of emergency situations. Future research prospects are associated with expanding the ontological layer for multi-agent coordination of subsystems and implementing federated learning methods to refine models at the fleet scale, ultimately facilitating the transition to fully autonomous self-diagnosing power plants with minimal operator intervention.

References

1. Vychuzhanin V., Rudnichenko N., Polyvianchuk A. Complex technical system condition diagnostics and prediction computerization. CEUR Workshop Proceedings. 2020. Vol. 2608. P. 42–56.
2. Vychuzhanin V. V., Vychuzhanin A. V. Integrated approach to creating a case-based database for diagnosing failures in ship power plants. Informatics and Mathematical Methods in Simulation. 2025. Vol. 15, No. 2. P. 155–165. DOI: https://doi.org/10.15276/imms.v15.no2.155.
3. Vychuzhanin V. V., Vychuzhanin A. Digital methods and models for control and survivability of complex technical systems: monograph. Lviv–Torun: Liha-Pres; 2025. 366 p. DOI: https://doi.org/10.36059/978-966-397-556-6.
4. Fuller A., Fan Z., Day C., Barlow C. Digital twin: enabling technologies, challenges and open research. IEEE Access. 2020. Vol. 8. P. 108952–108971. DOI: https://doi.org/10.1109/ACCESS.2020.2998358.
5. Rasheed A., San O., Kvamsdal T. Digital twin: values, challenges and enablers from a modeling perspective. IEEE Access. 2020. Vol. 8. P. 21980–22012.DOI: https://doi.org/10.1109/ACCESS.2020.2970143.
6. Errandonea I., Beltrán S., Arrizabalaga S. Digital twin for maintenance: a literature review. Computers in Industry. 2020. Vol. 123. Article 103316. DOI: https://doi.org/10.1016/j.compind.2020.103316.
7. Tao F., Qi Q. Make more digital twins. Nature. 2019. Vol. 573, No. 7775. P. 490–491. DOI: https://doi.org/10.1038/d41586-019-02849-1.
8. Abd Wahab N. H., et al. Systematic review of predictive maintenance and digital twin technologies: challenges, opportunities, and best practices. PeerJ Computer Science. 2024. Article e1943. DOI: https://doi.org/10.7717/peerj-cs.1943.
9. Hu J., Xiao H., Ye Z., Luo N., Zhou M. Research and prospects of digital twin-based fault diagnosis of electric machines. Sensors. 2025. Vol. 25, No. 8. Article 2625. DOI: https://doi.org/10.3390/s25082625.
10. Hedayati Kia S., Dunai L., Antonino-Daviu J. A., Razik H. Real-time digital twins for intelligent fault diagnosis and condition-based monitoring of electrical machines. Energies. 2025. Vol. 18, No. 17. Article 4637. DOI: https://doi.org/10.3390/en18174637.
11. Ma S., Flanigan K. A., Bergés M. State-of-the-art review: the use of digital twins to support artificial intelligence-guided predictive maintenance. arXiv. 2024. arXiv:2406.13117v1 [cs.AI]. DOI: https://doi.org/10.48550/arXiv.2406.13117.
12. Ismail L., Abdelmoti A., Basu A., Berini A. D. E., Naouss M. A systematic review of digital twin-driven predictive maintenance in industrial engineering: taxonomy, architectural elements, and future research directions. arXiv. 2025. arXiv:2509.24443. DOI: https://doi.org/10.48550/arXiv.2509.24443.
13. Hu W. Digital twin: a state-of-the-art review of its enabling technologies and applications. Journal of Industrial and Management Optimization. 2021. Vol. 2, No. 1. DOI: https://doi.org/10.1108/JIMSE-12-2020-010.
14. Вычужанин В. В. Диагностика, контроль при эксплуатации и ремонте систем кондиционирования воздуха на основе гибридных нейро-нечетких экспертных систем. *Вісник ОНМУ*. 2010. № 30. С. 100–109.
15. ISO 17359:2003(E). Condition monitoring and diagnostics of machines — General guidelines. Geneva: ISO; 2003.
16. Rausand M., Barros A., Hoyland A. System reliability theory: models, statistical methods, and applications. 3rd ed. Hoboken: John Wiley & Sons; 2020. ISBN: 978-1-119-37352-0.
17. Вычужанин В. В., Рудниченко Н. Д. Методы информационных технологий в диагностике состояния сложных технических систем: монография. Одесса: Экология; 2019.

Вичужанін В. В., Вичужанін О. В.
Інтегрована система цифрового двійника для адаптивної діагностики складних технічних систем

У статті представлено комплексну методологічну основу створення інтегрованого інформаційного середовища для моніторингу, діагностики та прогнозування технічного стану складних технічних систем (СТС) на прикладі суднових енергетичних установок. Актуальність роботи зумовлена необхідністю переходу від фрагментованих і статичних аналітичних методів до динамічних систем управління життєвим циклом обладнання в режимі реального часу. Основну увагу зосереджено на синтезі класичних підходів, зокрема імовірнісних графів і дерев відмов, із сучасними технологіями, такими як цифрові двійники (Digital Twin, DT), потокова аналітика великих даних та парадигма «Model-as-a-Service» (MaaS).Наукова новизна дослідження полягає у розробленні адаптивної архітектури, в межах якої традиційне статичне дерево відмов трансформується у динамічну онтологічну структуру. Запропоновано математичну модель розрахунку ризику на основі апроксимації Min Cut Upper Bound, що забезпечує обчислювальну ефективність під час обробки високої інтенсивності телеметричних потоків із використанням Kafka та Flink. У роботі описано оригінальну архітектуру класифікатора помилок, який функціонує як семантичний валідатор. Реалізація механізму «veto», заснованого на фізичних *обмеженнях цифрового двійника, дозволила знизити рівень хибнопозитивних спрацьовувань діагностичної моделі на 20 %, забезпечивши пріоритет фізичної узгодженості над статистичними кореляціями. Практичну значущість дослідження підтверджено результатами імітаційного моделювання різних сценаріїв деградації компонентів двигуна. Експериментальні дані свідчать, що інтеграція прогностичних моделей у контур цифрового двійника забезпечує системну перевагу: точність класифікації зростає з 0,87 до 0,94, а затримка оновлення прогнозу скорочується більш ніж удвічі — з 5,0 до 2,1 с. Запропоновано методологію управління життєвим циклом моделей на основі замкненого циклу MLOps, що включає фізично обґрунтоване навчання, тіньове розгортання та автоматизоване виявлення дрейфу характеристик.У роботі формалізовано алгоритм обчислення залишкового корисного ресурсу (Remaining Useful Life, RUL) на основі динамічного градієнта ймовірності, що дає змогу системі адаптуватися до змін інтенсивності експлуатації в режимі реального часу. Сформована багаторівнева архітектура інформаційного середовища, яка охоплює рівні збору даних, попередньої обробки, діагностики, прогнозування та підтримки прийняття рішень, становить завершене методологічне рішення для проактивного технічного обслуговування та переходу до автономних систем моніторингу СТС.*

Ключові слова: *цифровий двійник; складні технічні системи; потокова аналітика; адаптивна діагностика; дерево відмов; прогнозування залишкового ресурсу; MLOps.*

Вичужанін Володимир Вікторович - д.т.н, професор, завідувач кафедри інформаційних технологій, національний університет «Одеська політехніка», Одеса,
email: v.v.vychuzhanin@op.edu.ua
Вичужанін Олексій Володимирович - доктор філософії, асистент, національний університет «Одеська політехніка», Одеса, email: v.v.vychuzhanin@op.edu.ua

Стаття подана 11.11.2025.