



Full Length Article

A multi-agent cognitive digital twin architecture with neurosymbolic diagnostic integration for railway maintenance

Wanting Mao^{a,*}, Sara Scheffer^b, John Ahmet Erkoyuncu^c, Xi Vincent Wang^d, Lihui Wang^d, Arnab Majumdar^a, Bernadin Namono^{c,*}^a Department of Civil and Environmental Engineering, Imperial College London, London, SW7 2AZ, United Kingdom^b TU Wien, Research Unit of Production and Maintenance Management, Institute of Management Science, Getreidemarkt 9, 1060, Wien, Austria^c Centre of Digital Engineering and Manufacturing, Cranfield University, Bedford, MK43 0AL, United Kingdom^d Department of Production Engineering, KTH Royal Institute of Technology, Stockholm, 10044, Sweden

ARTICLE INFO

Keywords:

Cognitive digital twin
Multi-agent system
Neurosymbolic reasoning
Human-in-the-loop
Railway fault diagnosis
Ontology

ABSTRACT

Modern railway maintenance increasingly requires diagnostic systems that not only detect anomalies but also align decisions with established engineering standards and remain consistent across inspection cycles; however, many existing solutions rely either on operational alarm thresholds or on purely data-driven detectors, with limited integration between numerical inference and domain semantics. This study proposes a Multi-Agent Cognitive Digital Twin (MA-CDT) architecture designed to distribute cognitive responsibilities across specialised agents linked through shared semantic representations to support traceable information flow between physical inspection processes and digital reasoning components. To provide an initial implementation of the proposed architecture, its diagnostic component was instantiated and empirically evaluated using real operational track-geometry inspection data. The implemented module integrates neural anomaly detection, engineering intervention thresholds and ontology-guided semantic validation within a single decision structure. Comparative experiments against an operational alarm baseline and a neural-only configuration show that the integrated design reduces unsupported alerts while preserving threshold-defined intervention detections under noisy and imbalanced sensing conditions. Ablation analysis isolates the contributions of neural inference, rule-based constraints and semantic validation, demonstrating how their structured interaction stabilises decision behaviour. The diagnostic case study illustrates how one cognitive agent can operate within the proposed semantic framework and exchange information through the architectural knowledge structures. The results provide an initial empirical validation of the diagnostic component as a proof of concept within the proposed MA-CDT architecture and suggest that embedding domain constraints into neurosymbolic diagnostic reasoning could enhance decision stability and traceability in threshold-driven maintenance systems.

1. Introduction

Reliable and sustainable railway operations depend upon the maintenance systems with the ability to do more than simply sense faults or forecast failures: they must learn from experience, interpret context, and adapt their behaviour over time [1,2]. Modern railways generate continuous streams of geometry, signalling and rolling-stock data [3], yet the operational intelligence built on these measurements remains highly fragmented. The main analytical stages commonly supporting maintenance decision-making, namely: Health State Assessment (HSA), Predictive Maintenance (PdM), and Fault Detection and Diagnosis

(FDD), are typically implemented as separate algorithms optimised for local tasks rather than coordinated system reasoning [4]. As a result, anomalies may be detected, faults diagnosed, and failures predicted, but these outputs are rarely transferred across stages or accumulated into reusable operational knowledge [5].

This fragmentation reflects a lack of cognitive continuity, defined here as the ability of a maintenance system to preserve and develop knowledge, reasoning, and interpretation across tasks, time, and operational contexts. Existing digital systems remain dominated by sensing and prediction, with limited support for contextual understanding, causal interpretation, or structured knowledge transfer [6,7]. Even

* Corresponding authors.

E-mail addresses: w.mao23@imperial.ac.uk (W. Mao), bernadin.namono@cranfield.ac.uk (B. Namono).<https://doi.org/10.1016/j.jii.2026.101165>

Available online 15 July 2026

2452-414X/© 2026 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Digital Twins (DTs), while effective for real-time state mirroring, often lack persistent memory, semantic grounding, and mechanisms for incorporating human or operational feedback [8]. Consequently, maintenance intelligence remains stage-specific and difficult to reuse across assets, inspections, and interventions [9].

The emerging paradigm of the Cognitive Digital Twin (CDT) provides a conceptual foundation for addressing these challenges by extending conventional DTs with human-inspired cognitive capabilities such as memory, reasoning and learning [10,11]. However, most existing CDT studies remain confined to isolated maintenance functions rather than continuous cross-stage cognition [9]. In addition, cognitive mechanisms are often described abstractly, without being translated into interacting system components, semantic structures, or human-machine workflows. The role of distributed agents, shared knowledge representations, and human feedback therefore remains insufficiently specified for operational deployment in railway maintenance environments [12].

To address these gaps, this study proposes a Multi-Agent Cognitive Digital Twin (MA-CDT) architecture for railway maintenance. The proposed architecture is designed to organise sensing, reasoning, knowledge representation, and human input through specialised diagnostic, predictive, coordination, and knowledge-oriented agents that communicate via a semantically grounded Model Context Protocol (MCP). Within this structure, shared semantic representations support traceable information exchange across agents and maintenance stages, while Augmented Reality (AR) and Virtual Reality (VR) interfaces enable engineers to inspect, validate, and refine system outputs during operational use. The MA-CDT formalises it as an architectural arrangement of distributed functions, semantic structures, and human-centred interaction. The interactions between these agents, including information exchange through the semantically grounded MCP and ontology, are specified at the architectural level in this study, while only the diagnostic agent is implemented and experimentally evaluated.

To empirically ground the architecture, this study instantiates and evaluates its diagnostic component using real track-geometry data from Network Rail, which manages over 32,000 km of railway infrastructure across England, Scotland, and Wales. Although the proposed MA-CDT comprises multiple interacting agents at the architectural level, the present study focuses on the implementation and empirical evaluation of its diagnostic component using operational railway inspection data. Accordingly, the experimental validation should be interpreted as proof of concept for the diagnostics stage within the broader architecture rather than as a validation of the complete multi-agent system (MAS). This implemented component provides an initial operational realisation of the broader MA-CDT architecture, while the remaining architectural components and their coordinated interactions are proposed for future implementation and evaluation. The instantiated module integrates neural anomaly detection, engineering thresholds derived from track standards, and ontology-guided semantic validation within a single decision structure. The case study examines how this diagnostic configuration operates under operational inspection data and whether the integration of probabilistic inference with formal engineering constraints can reduce unsupported alerts while preserving threshold-defined intervention detections. Given the scale of railway maintenance expenditure (£2290 million in total maintenance, of which £921 million relates specifically to track maintenance in 2023–24), improvements in diagnostic stability have direct operational relevance [13].

Accordingly, this study should be understood as proposing a comprehensive MA-CDT architecture while implementing and empirically validating its diagnostic component using operational railway inspection data. This research makes three principal contributions:

- i. Architectural formalisation of a MA-CDT for railway maintenance: The study proposes a layered architecture in which diagnostic, predictive, coordination, and knowledge agents interact through a semantically grounded MCP. The proposed

architecture explicitly defines the intended data flow, semantic alignment mechanisms, and human-in-the-loop interfaces to address cognitive discontinuity across maintenance stages.

- ii. Instantiation of a neurosymbolic diagnostic module within the MA-CDT: The study implements a diagnostic agent that integrates neural anomaly detection, engineering rule-based thresholds, and ontology-guided semantic validation through a structured two-level fusion mechanism. This module provides an operational implementation of the perception and reasoning functions defined within the proposed architecture.
- iii. Empirical evaluation under operational railway inspection data: Using real track-geometry records, the study compares operational alarm, neural-only, fusion-only, and full semantic validation configurations, showing that the integrated design reduces false alarms while preserving detection coverage. Iterative semantic refinement further illustrates how structured updates can be introduced without destabilising the diagnostic process.

The paper is structured as follows. Section 2 reviews research on railway diagnostics, neurosymbolic reasoning and CDT. Section 3 introduces the MA-CDT architecture. Section 4 presents the diagnostic case study. Section 5 reports the comparative evaluation. Section 6 discusses implications and limitations, and Section 7 concludes.

2. Literature review

2.1. Intelligent maintenance: from data to knowledge

Reliable railway operation hinges on detecting and addressing emerging faults before they compromise safety or disrupt services. Modern rail networks are densely instrumented: track recording coaches produce geometry profiles for every few feet of line, on-board equipment streams vibration and electrical signals, and signalling subsystems report states continuously. The challenge is no longer data scarcity but the lack of diagnostic intelligence with the ability to transform heterogeneous sensor evidence into coherent, defensible maintenance decisions. Operational variability, sensor drift, and environmental interference exacerbate this gap, eroding confidence in automated analytics and forcing engineers to rely on experience rather than evidence [14].

Data-driven approaches have delivered measurable gains in specific contexts. For track geometry, random-forest prediction of defect occurrence combined with Markov decision optimisation has been shown to reduce costs under condition-based policies [15]. For track-circuit health, Long Short-Term Memory (LSTM) models exploit spatio-temporal dependencies to achieve high classification accuracy with very low false positives [16]. Deep architectures have improved diagnostic performance for high-speed on-board equipment, with DBNs outperforming classical baselines on real operational data [17]. These studies confirm that large-scale pattern learning enhances sensitivity and reliability, yet they remain numerical rather than contextual, which can recognise correlations but are unable to reason about causes, priorities, or maintenance consequences.

A persistent manifestation of this limitation is the prevalence of false alarms and alarm flooding. Across industries, alarm rates often exceed recommended benchmarks by an order of magnitude due to threshold chattering, misconfiguration, and correlated breaches [18]. False, nuisance and chattering alarms erode trust and desensitise operators [19], and compliance with ISA-18.2-style practices is insufficient without socio-technical integration [20]. Recent approaches, denoising auto-encoders with adaptive thresholds [21], two-stage false-positive refinement [22], and serial/deadband filtering [23], improve statistical performance but fail to capture the semantic dimension of diagnosis: how a deviation relates to an asset, a condition state, and a maintenance rule. Under non-stationary conditions, these purely statistical pipelines degrade quickly [24].

To compensate, human-in-the-loop strategies embed expert judgement within FDD workflows. Early work adapted rule weights using labelled alerts [25]; more recent systems embed operator feedback in logs [26] or outlier selection [27], reduce labelling via active learning [28,29], or steer models with attention/spuriousness cues [30,31]. In these approaches, human interaction typically influences local model behaviour, for example, by adjusting parameters, refining training samples or validating specific detections, rather than shaping how agents reason, remember or generalise across tasks. Knowledge-centric approaches introduce fuzzy vocabularies or ontologies to align domain semantics with sensor patterns [32–34]. Yet these solutions remain episodic: expert rationale is rarely formalised into persistent knowledge structures that can guide reasoning or learning across maintenance stages.

Consequently, most current systems exhibit a cognitive discontinuity across the maintenance lifecycle. HSA, PdM, and FDD often operate as independent silos, each optimised for specific tasks but lacking semantic and temporal coherence. HSA modules quantify degradation, PdM predicts remaining useful life, and FDD isolates failure causes; yet, few frameworks integrate these insights into a continuous cognitive loop that learns from interventions and updates future inferences. This fragmentation traps the industry in a pattern-recognition paradigm, where data are abundant but contextual understanding remains shallow.

2.2. Cognitive digital twins and distributed reasoning

The concept of the CDT extends the conventional DT from monitoring and simulation towards interpretation, reasoning and decision support. Existing studies have explored this transition from different perspectives, including semantic modelling, lifecycle integration and cognitive capability augmentation.

Some review of the literature shows three main directions [4,35,11]. First, semantic CDT approaches use ontologies and knowledge graphs to connect heterogeneous data, engineering models and contextual information. These studies demonstrate how semantic representations can preserve meaning across data sources and support inference in complex systems. Second, architecture-oriented CDT studies focus on integrating multiple digital models across system components and lifecycle phases. These works emphasise the need to link data, models and processes beyond isolated system representations. Third, capability-oriented CDT definitions describe cognition in terms of functional capabilities such as perception, attention, memory, reasoning, problem-solving and learning. These definitions frame CDT as a system that can process data, interpret states and support decision-making. The definition of cognition adopted in this study is grounded in cognitive systems theory and AI literature, where cognition is commonly described as a set of information-processing capabilities, including perception, attention, memory, reasoning, problem-solving and learning [36]. These capabilities are widely used in both intelligent agent design and cognitive architectures to describe how systems transform data into actionable knowledge. The selection of these six capabilities in this work reflects their relevance to maintenance decision-making, where systems must sense asset conditions, prioritise relevant signals, interpret anomalies, support decisions under constraints and adapt based on feedback.

Building on these perspectives, a CDT is defined in this study as a digital representation of a physical system that integrates real-time condition data, semantic knowledge, reasoning mechanisms and feedback processes to support interpretable and adaptive maintenance decision-making. This definition focuses on operational functionality in maintenance contexts, where data must be transformed into actionable knowledge under engineering constraints. Based on this, CDT in this study is operationally defined as: a distributed, semantically grounded digital representation of a physical system that integrates data-driven perception, symbolic reasoning and human feedback to support adaptive and interpretable decision-making over time.

Existing CDT-related solutions demonstrate the feasibility of

individual components of this definition. Knowledge-graph-based DTs show how semantic models can structure and link data [37]. Lifecycle-oriented DT frameworks illustrate how multiple digital models can be connected across system phases [38]. Human-machine CDT approaches highlight the role of expert interaction and feedback in decision support [39]. However, most implementations focus on specific functions, such as monitoring, prediction or anomaly detection [10]. In industrial maintenance, and particularly in railway maintenance, fewer studies demonstrate how diagnostic reasoning, HSA, predictive reasoning and human feedback can be integrated within a single operational architecture [9]. However, these approaches typically address isolated capabilities rather than integrating semantic reasoning, distributed coordination and human feedback within a unified cognitive architecture.

Based on this definition, two design requirements arise for maintenance-oriented CDTs. These requirements are not arbitrary but emerge directly from the nature of maintenance cognition. Distributed structure is required because maintenance decision-making is inherently decentralised across assets, processes, and actors. Hybrid reasoning is required because neither purely data-driven models nor purely rule-based systems can adequately represent the combination of stochastic behaviour and engineering constraints present in railway systems. Recent studies in neurosymbolic AI similarly argue that integrating neural learning with symbolic reasoning is particularly important in domains requiring explainability, robustness and constraint-aware decision-making [40].

The first requirement is a distributed structure. Railway maintenance involves multiple assets, data streams, diagnostic tasks and decision actors. A centralised model cannot represent these distributed responsibilities effectively. MASs have long been recognised as a foundation for decentralised intelligence in industrial contexts, balancing autonomy with coordination through explicit communication protocols [41–43]. Existing studies confirm the potential of MAS to coordinate distributed maintenance and diagnostic functions [44–46], however, agent coordination remains largely syntactic, with data exchanged in real time but without shared interpretation.

For maintenance decision-making, agents must coordinate through shared semantic representations. This allows anomalies, fault hypotheses, severity levels and maintenance actions to be interpreted consistently across different processes. Ontologies and causal models provide this semantic layer by defining domain concepts, relations and constraints that can be reused across agents and over time. This enables knowledge to be accumulated and applied beyond individual tasks [47].

The second requirement, hybrid reasoning, is addressed through neurosymbolic AI, which integrates data-driven learning with symbolic inference. Within this paradigm, neural models function as black-box components that extract patterns from raw sensor data [48], while symbolic rules, ontologies and engineering thresholds constitute white-box knowledge that encodes domain constraints and causal structure [49]. Grey-box reasoning emerges from their interaction, where learned evidence is interpreted, verified and refined through symbolic reasoning, supporting explainability and robustness [50,51]. Although recent studies demonstrate the value of such neurosymbolic integration [52,53], most implementations remain centralised and lack mechanisms to distribute reasoning and knowledge across multiple agents and operational contexts [54].

Integrating neurosymbolic reasoning within a multi-agent architecture enables CDTs to realise distributed cognition, in which agents coordinate in real time through shared semantics while progressively accumulating knowledge across operational cycles. However, few existing systems jointly demonstrate multi-agent coordination, hybrid reasoning and human validation within a single cognitive framework. In particular, while MAS enable distributed coordination, they rarely support semantic-level alignment of interpreted knowledge across agents. As a result, distributed cognitive systems often lack semantic coherence, with agents exchanging data without maintaining a shared

understanding of system state. This fragmentation directly motivates the identified gap of limited semantic coherence: while data and models are distributed, the meaning of exchanged information is not consistently aligned, preventing the emergence of shared system understanding across agents. This limitation directly constrains the effectiveness of distributed reasoning in maintenance contexts and motivates the need for architectures that integrate shared semantic representations across HSA, PdM and FDD.

2.3. Human knowledge and cognitive interaction

While CDT can perceive, reason and learn, its long-term effectiveness depends on how human knowledge is embedded within these cognitive processes. Industrial maintenance remains a socio-technical activity, where tacit expertise, such as contextual understanding, procedural adaptation and experiential reasoning, cannot be fully captured by data alone. This perspective aligns closely with the Operator 4.0 paradigm, which emphasises the augmentation of human cognitive and decision-making capabilities through intelligent, adaptive technologies [55]. Within this paradigm, digital systems are expected to act as cognitive partners that support human sensemaking, judgement and learning in complex operational environments.

A growing body of research demonstrates that immersive interfaces enhance human-AI collaboration in operational contexts. Liang et al. [56] developed an AR-based intelligent human-machine interaction system that integrates real-time anomaly diagnosis with operator feedback through a dual-IoT architecture, enabling users to visualise diagnostic information within the production space. Mao et al. [5] introduced the DILEAF framework, which uses AR overlays to support knowledge capture and procedural guidance in maintenance. Through iterative field validation, they showed that immersive visual cues improve situational awareness and knowledge transfer between technicians and digital systems. Zhang et al. [57] combined AR and human-computer interaction (HCI) to support adaptive assembly, allowing workers to manipulate equipment through hands-free control and real-time visual alignment. In parallel, Cimino et al. [58] proposed an ontology-based VR environment acting as a digital twin for complex systems, allowing operators to interpret live data and intervene in virtual space. Systematic analyses confirm that immersive technologies are evolving from passive visual aids to active components in maintenance, training, and lifecycle management [59].

Yet despite these advances, most implementations remain decoupled from the underlying reasoning systems. AR and VR interfaces typically visualise analytical outputs without influencing how reasoning or learning unfolds. Human validation, while often present, is rarely captured as structured input that can refine models or update ontological knowledge. Consequently, human expertise is recognised but not operationalised; it remains external to the cognitive loop. In this work, human knowledge is operationalised as structured semantic input. Expert validation is captured as updates to ontology entities, relationships and decision constraints, enabling human knowledge to directly influence system reasoning. This includes, for example, refining fault classifications, adjusting severity thresholds and contextual annotation of anomalies. As a result, human interaction contributes not only to validation but to the evolution of the system's knowledge base.

2.4. Research gaps

Although recent research has expanded the theoretical scope of CDTs, their operational realisation in railway maintenance remains insufficiently addressed. Existing studies often reproduce the structural components of cognition, such as data collection, analysis, prediction and guidance, without establishing how these components can be integrated into a unified cognitive process. As a result, several research gaps remain unresolved, limiting the development of maintenance systems as continuous cognitive ecosystems.

i. Lack of a unified cognitive logic across maintenance stages.

Current research typically investigates HSA, PdM and FDD as independent analytical functions. While each stream demonstrates local intelligence, there is limited explanation of how information and knowledge are transferred across stages. In particular, mechanisms that allow outputs from one stage to inform reasoning in another are rarely specified. This results in stage-specific intelligence rather than a continuous reasoning process across maintenance workflows.

ii. Limited semantic coherence in distributed cognitive systems.

Multi-agent architectures have been widely studied for improving scalability and autonomy in maintenance systems. However, existing work predominantly focuses on procedural coordination and data exchange, with insufficient attention to semantic-level interaction. The lack of shared ontologies or causal representations means that agents rarely exchange interpreted knowledge or develop a common understanding of context, leaving the problem of semantic alignment in distributed cognition largely unresolved.

iii. Insufficient integration of human cognition into computational reasoning.

Although immersive technologies such as AR and VR have been explored to enhance human-machine interaction, most studies position humans as validators or information consumers rather than contributors to system cognition. Research has yet to adequately address how human expertise, such as tacit, contextual and situational knowledge, can be formalised and incorporated into reasoning models, symbolic constraints or learning processes, limiting genuine human-machine cognitive co-evolution.

iv Absence of adaptive and evolutionary knowledge mechanisms.

Neurosymbolic approaches have advanced the integration of learning and reasoning, but their knowledge representations are often treated as static. Ontologies are typically modelled as fixed structures, neural representations as frozen embeddings, and temporal dynamics as sequential data rather than evolving context. There remains a lack of research on mechanisms that allow knowledge to be accumulated, updated and recontextualised over time, constraining the development of memory and foresight within CDT-based maintenance systems.

These gaps suggest that while CDT has emerged as a promising conceptual paradigm, its translation into an operational, continuously learning maintenance framework remains incomplete. Addressing these research gaps requires architectures that explicitly integrate semantic reasoning, distributed cognition, human knowledge contribution and adaptive knowledge evolution within a single operational framework.

The MA-CDT proposed in this study addresses this integration problem at the architectural level. The empirical evaluation focuses on the diagnostic module as a proof-of-concept for semantic integration and hybrid reasoning. While the implementation is limited to this module, the architecture is designed to support extension across maintenance stages through shared semantic representations and distributed coordination.

3. Architecture design and methodology

3.1. Design science research process

This study adopts a Design Science Research (DSR) approach to structure the development and evaluation of the proposed MA-CDT architecture. The research follows the problem–design–evaluation cycle described in established DSR frameworks [60]. Within this framework,

the MA-CDT constitutes the primary artefact, defined as an architectural model and partially instantiated through an implemented diagnostic module.

The design problem addressed in this work concerns the fragmentation of railway maintenance intelligence. In prevailing practice, HSA, PdM, FDD are implemented as separate analytical functions. These functions operate on shared data sources but do not maintain structured semantic representations that enable systematic knowledge reuse across stages. Consequently, experience accumulated in one stage is not formally incorporated into subsequent reasoning processes. The architectural objective of the MA-CDT is therefore to organise sensing, reasoning, knowledge representation and human input within a unified structural architecture that supports traceable information exchange across distributed agents.

From this problem definition, three concrete design requirements were derived. First, engineering intervention thresholds must remain explicit decision boundaries within the system, rather than external validation checks. Second, data-driven detection mechanisms must be incorporated to enhance sensitivity to gradual or sub-threshold deviations. Third, semantic representations must allow structured information propagation between agents without introducing uncontrolled feedback into statistical models.

The MA-CDT architecture was designed to satisfy these requirements. During artefact instantiation, alternative configurations were examined at the diagnostic layer. A neural-only configuration was insufficient because it lacked explicit rule-based constraints. A threshold-only configuration could not detect gradual deviations below formal limits. These limitations motivated the integration of neural inference, engineering thresholds and ontology-guided semantic validation within a single diagnostic structure. This diagnostic module represents one implemented component of the broader MA-CDT architecture and provides a concrete basis for empirical examination.

Evaluation was conducted through controlled ablation experiments using operational track-geometry data. Multiple diagnostic configurations were compared to isolate the contributions of neural perception, symbolic constraints and semantic validation. Performance metrics were selected to correspond directly to the architectural requirements defined above. Iterative refinement experiments further examined whether semantic updates could be introduced without modifying neural parameters. This alignment between design requirements and empirical evaluation completes the DSR cycle and grounds the architectural proposal in operational evidence.

3.2. Overview of the multi-agent cognitive digital twin architecture

The proposed MA-CDT architecture provides a system-level organisational structure for integrating sensing, diagnostic reasoning, knowledge representation and human interaction within railway maintenance workflows. The architecture distributes cognitive responsibilities across specialised agents and layered infrastructure components, enabling structured information flow between the physical railway environment and its digital counterpart. Its purpose is to support the transformation of asset-condition data into interpretable maintenance knowledge through distributed agents, shared semantic representations and human feedback.

Cognition in this study is grounded in prior definitions of CDT, where cognition is described as a set of functional capabilities that enable a system to sense, interpret and act upon its physical counterpart. Following existing CDT literature [61,11], six cognitive capabilities are adopted: perception, attention, memory, reasoning, problem-solving and learning.

These capabilities are defined here in operational terms. Perception refers to the transformation of raw asset-condition signals into structured representations. Attention concerns the prioritisation of anomalies based on operational relevance. Memory captures the storage and retrieval of historical condition data and domain knowledge. Reasoning

enables the interpretation of anomalies under engineering and semantic constraints. Problem-solving supports the translation of interpreted states into maintenance actions. Learning reflects the refinement of system knowledge through expert feedback and operational outcomes.

These six capabilities are selected because they correspond to the core information-processing requirements of railway maintenance. Maintenance workflows require continuous sensing of asset condition, selective identification of critical events, interpretation of fault states, decision support under constraints, and iterative updating of knowledge based on inspection and intervention outcomes. The selected capabilities therefore provide a structured basis for operationalising cognition within the MA-CDT architecture.

The MA-CDT comprises six interrelated layers: railway infrastructure, data acquisition and storage, multi-agent intelligence, orchestration, communication protocols and AR/VR-enabled human interaction. These layers do not represent cognition independently. Cognitive behaviour emerges through the interaction between layers and agents. More specifically, cognition emerges through the closed-loop interaction between perception, reasoning, action, and feedback. Raw data is first transformed into structured signals (perception), which are evaluated and prioritised (attention). These signals are then interpreted through neurosymbolic reasoning combining learned patterns and engineering constraints (reasoning). The resulting interpretations are stored and linked to prior knowledge (memory), enabling consistency over time. Decisions derived from this reasoning (problem-solving) are validated or corrected by human operators, whose feedback is incorporated into the semantic layer (learning). This iterative loop transforms the architecture from a static data-processing pipeline into a dynamic cognitive system capable of adaptation and knowledge accumulation.

To clarify the system boundary, the MA-CDT is defined to include the infrastructure, data, multi-agent intelligence, orchestration and communication layers. The human operator is not treated as an internal component of the CDT. Instead, it is modelled as an external decision actor within the broader socio-technical system, interacting with the CDT through the AR/VR presentation layer and communication interfaces. Human involvement is therefore realised through structured validation, contextual input and feedback, rather than as an embedded computational element of the CDT itself.

Within this architecture, maintenance processes are not implemented as isolated modules but as interconnected analytical levels. HSA, FDD, and PdM are linked through shared data, semantic representations and coordination mechanisms. Outputs from HSA provide structured inputs for diagnostic reasoning in FDD, while interpreted fault states are reused by PdM for prediction and planning. HSA produces structured anomaly indicators derived from sensor data, which are passed to FDD for interpretation and fault classification. The outputs of FDD, including fault type and severity, are then used by PdM to estimate degradation trajectories and maintenance timing. These outputs are coordinated through the orchestration layer, which ensures consistency and supports decision-making across stages. Human actors interact at each level, validating anomalies (HSA), confirming diagnoses (FDD) and approving maintenance actions (PdM). The coordination layer integrates these outputs to support consistent decision-making. A detailed description of this interaction is provided in [Section 3.3](#).

Raw rail signals are first captured and structured through the sensing and data layers, corresponding to perception. The diagnostic agent transforms these signals into anomaly candidates. The orchestration layer filters and prioritises anomalies according to severity and operational relevance, corresponding to attention. The semantic layer links anomalies to fault concepts, severity levels, engineering thresholds and maintenance actions, supporting memory and reasoning. Reasoning is realised through the combination of data-driven detection, formal constraints and semantic consistency checks. Problem-solving is implemented through decision synthesis based on interpreted fault states and operational constraints. Learning is supported through expert feedback, where validated or corrected outputs are used to update semantic

representations and decision rules.

The mapping in Table 1 identifies the primary operational locus of each cognitive capability rather than assigning each capability exclusively to one maintenance phase. This mapping does not imply exclusivity but functional emphasis. Each capability across multiple maintenance stages. However, the mapping reflects where each capability plays its dominant role within the workflow. This design choice improves interpretability by linking abstract cognitive functions to concrete operational processes, while still preserving the integrated nature of maintenance cognition. Perception is primarily associated with HSA because HSA begins with sensing and structuring asset-condition data. Reasoning is primarily associated with FDD because fault detection and diagnosis require the interpretation of anomalies under engineering and semantic constraints. Memory is associated with PdM because predictive maintenance depends on historical condition records and degradation patterns. However, these capabilities operate

Table 1
Operationalisation of cognitive capabilities in the MA-CDT architecture.

Cognitive design objective	Intended function in railway maintenance	Architectural instantiation	Primary locus (agent/layer)	Maintenance Phase
Perception	Sense and structure asset-condition signals for downstream interpretation	Data ingestion, pre-processing, and data-driven anomaly detection on track-geometry time series	Diagnostic agent, sensing/data layer	HSA
Attention	Prioritise anomalies for timely human and system action	Event filtering and prioritisation via severity criteria and subscription/trigger logic in the MCP	Coordination/orchestration layer	HSA / PdM
Memory	Preserve operational context and enable reuse of prior knowledge	Ontology-grounded semantic store linking fault taxonomy, standards, and historical inspection context	Knowledge base layer, diagnostic agent interface	PdM
Reasoning	Produce interpretable hypotheses under engineering-consistency constraints	Neurosymbolic fusion combining neural detection, formal thresholds, and ontology-guided consistency constraints	Diagnostic agent	FDD
Problem-solving	Translate hypotheses into actionable maintenance decisions	Decision synthesis and task structuring based on diagnostic outputs and operational constraints	Coordination/orchestration layer	FDD
Learning	Improve models and rules through operational feedback	Expert-in-the-loop review and iterative semantic refinement of constraints/knowledge representations	Human feedback loop, semantic layer	Across all phases

across phases. Perception provides inputs for both diagnosis and prediction. Memory supports both diagnostic interpretation and predictive reasoning. Reasoning contributes to fault classification as well as maintenance planning. The phase mapping should therefore be understood as a primary functional emphasis within an integrated maintenance cognition cycle. This mapping establishes a traceable connection between the system-level architecture (Fig. 1), the maintenance workflow integration (Fig. 4), and the instantiated neurosymbolic pipeline evaluated in Section 5. The mapping between MA-CDT components and ArchiMate concepts used in Figs. 1–3 is provided in Appendix A.

3.2.1. Railway infrastructure layer

The railway infrastructure layer captures physical and operational data that provides both asset-centric condition information and infrastructure-derived context for maintenance decision-making. It integrates sensing and monitoring systems deployed on rolling stock, track infrastructure, and signalling assets to support a situational understanding of railway operations.

Train-mounted sensors and IoT devices continuously collect asset-level parameters such as vibration, temperature, acoustic emissions, and electrical signals, forming the primary inputs for FDD analysis. Track monitoring systems employ techniques including ultrasonic inspection, visual imaging, and strain measurement to assess rail condition and structural integrity. Track condition data are used to characterise infrastructure-induced constraints and stressors that influence asset degradation and maintenance prioritisation.

Signalling systems monitor the operational state of switches, signals, and interlocking components. Within the MA-CDT architecture, signalling information provides operational context, such as routing conditions and abnormal events, that can affect asset loading patterns and failure propagation, thereby informing asset-focused maintenance decisions.

3.2.2. Data storage layer

The data storage layer provides persistent data management and historical context to support temporal analysis, diagnostic reasoning, and maintenance decision-making within the MA-CDT architecture. It integrates heterogeneous data sources generated across sensing, operations, and maintenance activities, ensuring that both real-time observations and accumulated operational knowledge are accessible to downstream intelligent agents.

A time-series database stores high-frequency sensor measurements alongside relevant operational data, including event logs and system states, with explicit temporal annotations. This enables the analysis of long-term trends, anomaly evolution, and condition patterns across assets and operating contexts. Complementing this, the maintenance history repository consolidates records of past failures, inspection results, repair actions, spare parts usage, component specifications, and maintenance outcomes. Together, these repositories provide structured historical evidence that supports predictive maintenance models, diagnostic inference, and knowledge reuse across maintenance cycles.

3.2.3. Multi-agent intelligence layer

The multi-agent intelligence layer realises distributed cognitive functionality by assigning distinct analytical responsibilities to specialised agents. The coordinated interaction of agents operates over shared data resources and semantic representations, addressing different temporal and functional aspects of maintenance decision-making.

The diagnostic agent performs real-time fault detection and diagnosis using data-driven anomaly detection and classification techniques applied to live sensor data. It identifies deviations from normal operating behaviour and generates diagnostic hypotheses with associated confidence estimates and supporting evidence, enabling interpretable fault assessment informed by contextual knowledge.

The predictive maintenance agent focuses on temporal reasoning

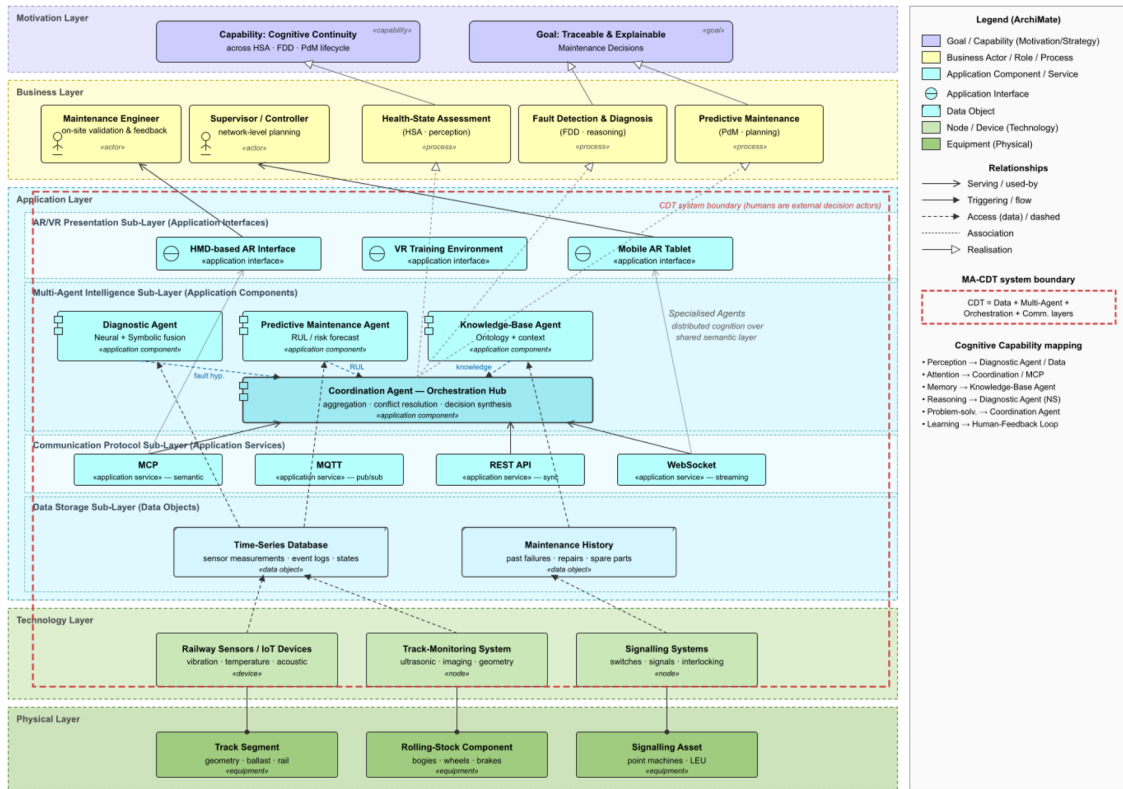


Fig. 1. The multi-agent cognitive digital twin architecture for railway maintenance with AR/VR integration.

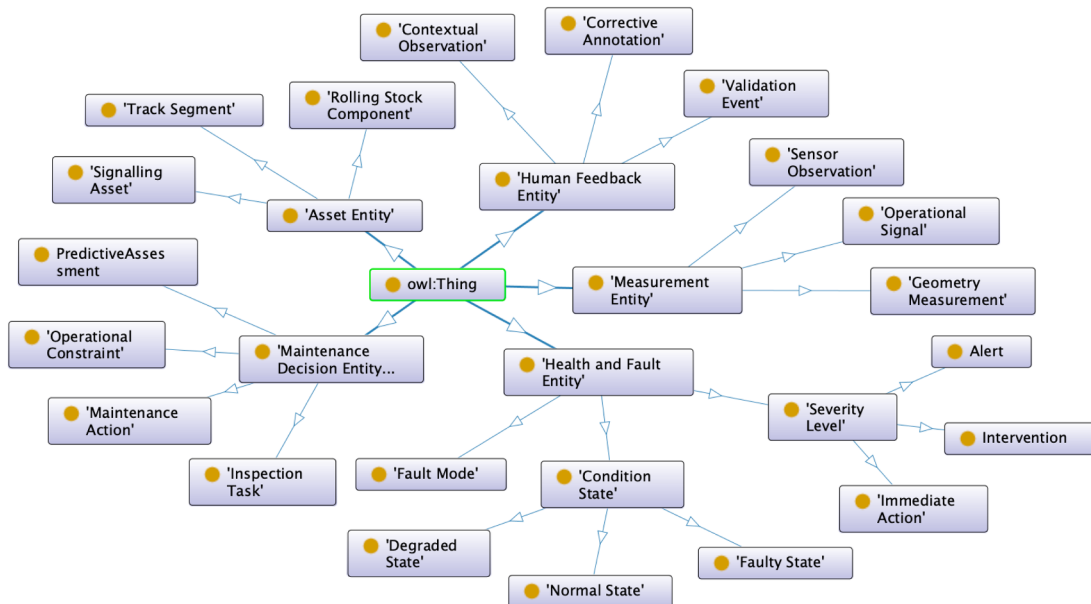


Fig. 2. OntoGraf visualisation of the MA-CDT ontology class hierarchy.

over historical degradation trajectories and maintenance records. It estimates remaining useful life, anticipates potential failure modes, and produces probabilistic risk assessments and maintenance timing recommendations that can be updated as new evidence becomes available.

The coordination agent acts as the orchestration and decision synthesis hub, aggregating diagnostic and predictive outputs and resolving conflicting recommendations through contextual and rule-based prioritisation. By managing communication between agents and external interfaces, it ensures consistency, traceability, and auditability of

maintenance decisions.

The knowledge base agent supports cross-agent reasoning by maintaining a semantic knowledge graph that encodes domain knowledge, maintenance procedures, regulatory constraints, and historical cases. It enables context-aware knowledge retrieval and semantic alignment, grounding data-driven inference within operational and regulatory contexts.

Within the MA-CDT architecture, this semantic knowledge is structured through an ontology that serves as the common representation

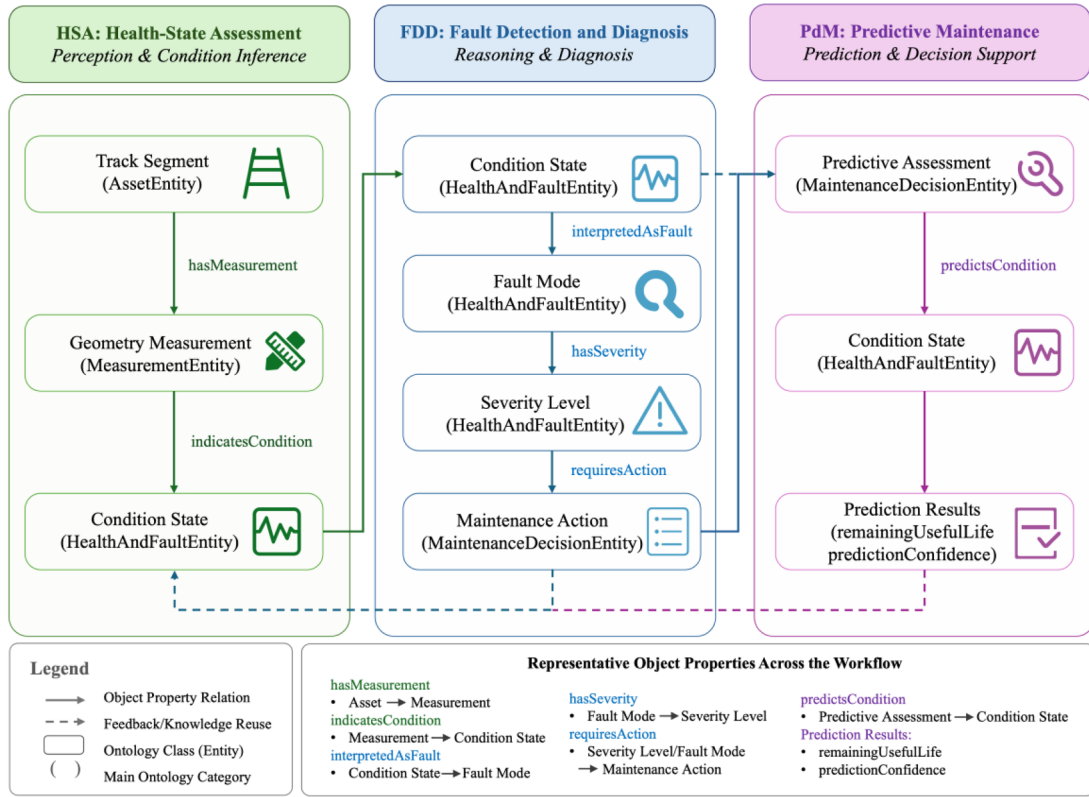


Fig. 3. Semantic reuse of ontology entities across the HSA-FDD-PdM reasoning workflow.

layer for maintenance reasoning. Rather than supporting diagnosis alone, the ontology is designed to provide a shared semantic model across HSA, FDD, and PdM, enabling the same asset, condition and maintenance entities to be progressively enriched as information moves from sensing to diagnosis, prediction, and maintenance planning. The ontology constitutes the core semantic layer of the MA-CDT and is designed to support multiple decision-making levels. It defines a consistent set of entities, relations and constraints through which diagnostic, predictive and coordination agents exchange interpretable information.

The ontology is developed following a knowledge reuse approach [62,63]. Existing studies in maintenance and asset management emphasise the reuse of established ontologies and formal vocabularies to support semantic alignment and interoperability [64,65]. In this work, the ontology is structured as an application-level railway maintenance ontology that can be aligned with foundational ontologies such as BFO

[66]. This alignment provides a consistent conceptual basis for representing assets, observations and maintenance processes across heterogeneous maintenance data sources. The ontology design is further informed by existing semantic models in IoT and railway maintenance domains, including SAREF [67,68] and ROMAIN [69], to support reusable semantic structures for sensor observations, asset descriptions and maintenance knowledge representation.

The ontology is organised around five core groups of entities that collectively represent assets, observations, fault semantics, maintenance decisions, and human feedback within the MA-CDT architecture. These entity groups are intentionally shared across HSA, FDD, and PdM rather than belonging to individual maintenance functions, allowing each stage to contribute additional semantic information to the same evolving knowledge graph. Formally, the ontology is expressed in OWL and structured as a class hierarchy with explicit object and data properties. Fig. 2 presents the resulting OWL class hierarchy generated using

Table 2

Core object properties defined in the MA-CDT ontology.

Object property	Domain	Range	Description
hasMeasurement	AssetEntity	MeasurementEntity	Associates a railway asset with its corresponding measurement observations.
hasConditionState	AssetEntity	ConditionState	Associates an asset with its inferred health condition.
describesAsset	MeasurementEntity	AssetEntity	Specifies the asset represented by a measurement record.
generatedBySensor	MeasurementEntity	SensorObservation	Indicates the sensing observation from which the measurement is derived.
indicatesCondition	MeasurementEntity	ConditionState	Maps observed measurements to interpreted condition states (e.g., Normal, Degraded or Faulty).
interpretedAsFault	ConditionState	FaultMode	Represents the interpretation of abnormal condition states as fault modes.
affectsAsset	FaultMode	AssetEntity	Identifies the asset affected by the detected fault mode.
hasSeverity	FaultMode	SeverityLevel	Associates a fault mode with its engineering severity level (Alert, Intervention or ImmediateAction).
requiresAction	FaultMode	MaintenanceAction	Specifies the maintenance action required to address the identified fault mode.
constrainedBy	MaintenanceAction	OperationalConstraint	Indicates operational constraints governing maintenance execution.
generatesOutcome	MaintenanceAction	ConditionState	Represents the updated asset condition following maintenance execution.
validates	ValidationEvent	FaultMode	Records expert validation of automatically inferred fault hypotheses.
updatesConditionState	ValidationEvent	ConditionState	Updates the condition state in the knowledge graph following expert validation.
corrects	CorrectiveAnnotation	FaultMode	Records expert corrections to fault interpretation when necessary.
predictsCondition	PredictiveAssessment	ConditionState	Represents predicted future asset condition.
derivedFromHistory	PredictiveAssessment	ConditionState	Connects prediction to historical health states.

Protégé OntoGraf, while Tables 2–6 summarise the object properties, data properties, representative axioms, reasoning rules, and SPARQL validation queries that operationalise the ontology. Asset entities represent railway infrastructure elements such as track segments, rolling stock components and signalling systems. The *TrackSegment* (subclass of *Asset*) is linked via the relation *hasMeasurement* to instances of *GeometryMeasurement*, which include attributes such as twist, gauge, and alignment. Measurement entities represent sensor observations and condition indicators, including track geometry measurements and operational signals. Measurements are associated with *ConditionState* through the relation *indicatesCondition*, which can take values such as *Normal*, *Degraded*, or *Faulty*. Health and fault entities represent condition states, anomalies, fault modes and severity levels. Fault modes are represented as instances of the class *FaultMode* and are linked to assets via *affectsAsset*. Each fault is associated with a *SeverityLevel* (e.g., *Alert*, *Intervention*, *ImmediateAction*) through the relation *hasSeverity*, which is constrained by engineering thresholds derived from maintenance standards. Maintenance decision entities represent inspection tasks, intervention actions, work orders and operational constraints. Maintenance actions are represented as instances of *MaintenanceAction* and linked to fault states via *requiresAction*. Human feedback entities represent validation records, contextual observations and corrective annotations. Human feedback is captured through the class *ValidationEvent* together with contextual observations and corrective annotations, enabling expert validation and semantic refinement. For predictive maintenance, the ontology further introduces the auxiliary class *PredictiveAssessment* under the *MaintenanceDecisionEntity* group. This class reuses validated historical condition states to infer future condition states, while representing prediction outputs through data properties such as *remainingUsefulLife* and *predictionConfidence*, thereby extending maintenance decision support from diagnosis to prediction.

The ontology further defines data properties to capture quantitative maintenance information, including asset identifiers, measurement values, timestamps, severity codes, validation status and prediction outputs. Prediction outputs are represented through data properties such as *remainingUsefulLife* and *predictionConfidence*, enabling predictive results to be stored within the same semantic representation as diagnostic knowledge.

The ontology includes axioms and constraints expressed as logic rules. For instance, a constraint enforces that a *FaultMode* with severity *ImmediateAction* must be associated with a maintenance action. Similarly, measurement thresholds are encoded as rule conditions that map numerical values to severity categories. In addition to diagnostic reasoning, the ontology also supports predictive maintenance through representative OWL axioms that associate *PredictiveAssessment* with validated historical condition states and predicted future condition states, thereby enabling semantic continuity from diagnosis to prediction. These rules are implemented through semantic reasoning rules and validated using representative SPARQL queries, enabling consistency checking and semantic validation. The ontology defines the semantic schema, while the 4D knowledge graph stores instantiated asset,

Table 4

Representative OWL axioms implemented in the MA-CDT ontology.

Axiom	Formal representation	Purpose
A1	$\text{TrackSegment} \sqsubseteq \text{hasMeasurement some GeometryMeasurement}$	Every monitored track segment is associated with at least one geometry measurement.
A2	$\text{GeometryMeasurement} \sqsubseteq \text{indicatesCondition some ConditionState}$	Geometry measurements provide evidence for condition assessment.
A3	$\text{ConditionState} \sqsubseteq \text{interpretedAsFault some FaultMode}$	Abnormal condition states may be interpreted as fault modes.
A4	$\text{FaultMode} \sqsubseteq \text{hasSeverity some SeverityLevel}$	Every fault mode is assigned an engineering severity level.
A5	$\text{FaultMode} \sqsubseteq \text{requiresAction some MaintenanceAction}$	Every identified fault requires an appropriate maintenance action.
A6	$\text{ValidationEvent} \sqsubseteq \text{updatesConditionState some ConditionState}$	Human validation updates the semantic condition state stored in the knowledge graph.
A7	$\text{PredictiveAssessment} \sqsubseteq \text{predictsCondition some ConditionState}$	Predictive assessment predicts the future condition state of an asset.
A8	$\text{PredictiveAssessment} \sqsubseteq \text{derivedFromHistory some ConditionState}$	Predictive assessment is grounded in validated historical condition states.

Table 5

Representative semantic reasoning rules.

Rule	Semantic rule	Purpose
R1	IF a measurement exceeds engineering thresholds THEN $\text{ConditionState} = \text{FaultyState}$	HSA
R2	IF $\text{ConditionState} = \text{FaultyState}$ THEN interpret as <i>FaultMode</i>	FDD
R3	IF $\text{FaultMode hasSeverity ImmediateAction}$ THEN $\text{requiresAction MaintenanceAction}$	Maintenance decision support.
R4	IF $\text{ValidationEvent confirms a fault hypothesis}$ THEN update ConditionState	Human-in-the-loop semantic refinement.
R5	IF successive validated ConditionStates indicate progressive degradation THEN generate a $\text{PredictiveAssessment}$ with updated $\text{remainingUsefulLife}$ and $\text{predictionConfidence}$	PdM knowledge propagation

condition and temporal relations used during reasoning.

The ontology is aligned at a conceptual level with the basic formal ontology, where assets correspond to material entities, measurements to processes or observations, and condition states to dependent continuants. This alignment ensures interoperability and conceptual grounding, while domain-specific extensions capture railway maintenance semantics.

Together, these object properties define the semantic knowledge flow across the ontology. Asset entities have associated measurements and condition states. Measurements may indicate anomalies or degradation patterns. Anomalies can be interpreted as fault modes, which are assigned severity levels based on engineering constraints. Fault states may require maintenance actions, and actions produce outcomes that

Table 3

Representative data properties in the MA-CDT ontology.

Data property	Domain	Datatype	Description
<i>assetID</i>	<i>AssetEntity</i>	xsd:string	Unique identifier of a railway asset.
<i>measurementValue</i>	<i>MeasurementEntity</i>	xsd:float	Numerical value recorded by a monitoring system.
<i>twistValue</i>	<i>GeometryMeasurement</i>	xsd:float	Three-metre twist value used in track-geometry threshold reasoning.
<i>gaugeValue</i>	<i>GeometryMeasurement</i>	xsd:float	Gauge value used in track-geometry threshold reasoning.
<i>ruleStrengthGlobal</i>	<i>GeometryMeasurement</i>	xsd:integer	Maximum symbolic rule strength across geometry fault mechanisms.
<i>timestamp</i>	<i>MeasurementEntity</i>	xsd:dateTime	Time at which the measurement was acquired.
<i>severityCode</i>	<i>SeverityLevel</i>	xsd:string	Engineering severity category (Alert, Intervention or ImmediateAction).
<i>validationStatus</i>	<i>ValidationEvent</i>	xsd:string	Status of expert validation.
<i>annotationText</i>	<i>CorrectiveAnnotation</i>	xsd:string	Expert comments describing corrective knowledge.
<i>remainingUsefulLife</i>	<i>PredictiveAssessment</i>	xsd:float	Estimated remaining useful life.
<i>predictionConfidence</i>	<i>PredictiveAssessment</i>	xsd:float	Confidence of predictive inference.

Table 6
Representative SPARQL-based validation queries for the track-geometry ontology instance.

Competency question	Validation purpose	Representative SPARQL query	Expected validation result
Which asset does a given observation describe?	Validate that the geometry measurement is linked to the monitored track segment.	PREFIX: < http://www.semanticweb.org/wanting/ontologies/2026/6/untitled-ontology-5# > SELECT ?asset ?measurement WHERE {?measurement:describesAsset ?asset .}	GeometryMeasurement_001 describes TrackSegment_001.
Which measurement abnormality indicates a fault state?	Validate that a track-geometry measurement exceeding symbolic thresholds is associated with a faulty condition state.	PREFIX: < http://www.semanticweb.org/wanting/ontologies/2026/6/untitled-ontology-5# > SELECT ?measurement ?condition ?twist ?ruleStrength WHERE {?measurement:indicatesCondition ?condition::twistValue ?twist::ruleStrengthGlobal ?ruleStrength .}	GeometryMeasurement_001 indicates FaultyState_001, with twistValue = 35.0 and ruleStrengthGlobal = 3.
Which severity level is associated with a detected fault?	Validate that the detected fault is assigned an engineering severity level.	PREFIX: < http://www.semanticweb.org/wanting/ontologies/2026/6/untitled-ontology-5# > SELECT ?fault ?severity WHERE {?fault:hasSeverity ?severity .}	FaultMode_001 has severity ImmediateAction_001.
Which maintenance action is required for a given fault and severity level?	Validate that an immediate-action fault is linked to a maintenance action.	PREFIX: < http://www.semanticweb.org/wanting/ontologies/2026/6/untitled-ontology-5# > SELECT ?fault ?severity ?action WHERE {?fault:hasSeverity ?severity::requiresAction ?action .}	FaultMode_001 with ImmediateAction_001 requires MaintenanceAction_001.
How can validated fault states and intervention outcomes be reused in predictive maintenance?	Validate that expert validation updates the condition state stored in the knowledge graph.	PREFIX: < http://www.semanticweb.org/wanting/ontologies/2026/6/untitled-ontology-5# > SELECT ?validation ?fault ?status ?updatedCondition WHERE {?validation:validates ?fault::validationStatus ?status::updatesConditionState ?updatedCondition .}	ValidationEvent_001 confirms FaultMode_001 and updates the condition state to FaultyState_001.

update the condition state of the asset. Human feedback is linked to fault hypotheses, actions and asset states, enabling validated knowledge to be incorporated into subsequent reasoning.

The ontology encodes constraints that define admissible configurations of fault states and severity levels. These constraints are expressed as rule-based conditions that ensure consistency between measurement evidence, fault interpretation and engineering limits. They serve as formal reasoning rules during semantic validation, enabling the system to detect and correct inconsistent or physically implausible interpretations.

These constraints support reasoning across maintenance stages, as illustrated in Fig. 3. At the HSA level, measurement entities represent observed condition data. Semantic reasoning transforms these observations into explicit condition states while preserving links to the originating measurements and assets. At the FDD level, these observations are interpreted as fault entities with associated severity levels and explanatory context. The validated fault and condition entities are then retained within the knowledge graph as temporal maintenance histories rather than discarded after diagnosis. At the PdM level, validated fault knowledge together with historical condition information is reused through the auxiliary class *PredictiveAssessment* to infer future condition evolution and remaining useful life estimates. Prediction results, including *remainingUsefulLife* and *predictionConfidence*, are represented as ontology data properties, enabling continuous knowledge reuse across maintenance stages. This structure enables knowledge to move from observation to diagnosis and then to prediction within a shared semantic representation.

To guide the scope of the ontology, a set of competency questions is defined. The competency questions were defined before ontology development and directly informed the selection of classes, relationships, constraints and reasoning rules. Together, they ensure that the ontology captures the semantic requirements needed to support health assessment, diagnosis, prediction and maintenance decision-making. These include: which asset a given observation describes; which measurement abnormality indicates a fault state; which severity level is associated with a detected fault; which maintenance action is required for a given fault and severity level; and how validated fault states and intervention outcomes can be reused in predictive maintenance. These questions ensure that the ontology supports the information requirements of maintenance decision-making. Representative SPARQL validation queries corresponding to these competency questions are provided in Table 6.

In the present study, the ontology is instantiated in a restricted form for track-geometry monitoring and is used for semantic validation

within the diagnostic module. The complete ontology nevertheless specifies the semantic structures required for HSA, FDD, PdM, coordination and human feedback within the proposed MA-CDT architecture. While only the diagnostic reasoning pathway is implemented and experimentally evaluated in this work, the ontology itself defines the shared semantic backbone intended to support information exchange and knowledge reuse across all maintenance stages in future implementations.

3.2.4. Communication protocol layer

The communication protocol layer enables real-time coordination and semantic interoperability across the MA-CDT architecture. It supports heterogeneous interaction patterns required by distributed agents, data streams, and immersive interfaces while ensuring consistent interpretation of exchanged information.

At the semantic level, the MCP facilitates high-level communication between agents by allowing the exchange of ontologically grounded beliefs, goals, and intentions. This ensures that agents maintain aligned interpretations of shared concepts during collaborative reasoning and decision-making. For data-centric communication, MQTT provides lightweight publish-subscribe messaging for real-time sensor streams and event notifications, making it suitable for bandwidth-constrained IoT environments. REST APIs enable standardised synchronous interactions with legacy systems and external services, while WebSocket channels support low-latency, bidirectional streaming of diagnostic outputs and AR/VR content updates required for immersive HCI.

3.2.5. AR/VR presentation layer

The AR/VR presentation layer provides the interface through which human operators access CDT outputs and return operational feedback. In the proposed architecture, AR/VR is not treated as an independent reasoning component. It functions as a presentation and feedback channel between human operators and the semantic layer of the MA-CDT.

AR interfaces support on-site maintenance by presenting diagnostic outputs, asset-specific information, maintenance instructions and component annotations in relation to the physical railway asset. Head-mounted displays (HMDs) enable hands-free interaction during inspection and repair tasks, while tablet-based AR provides a more accessible option when headsets are unsuitable. Through these interfaces, engineers can review system-generated fault hypotheses, confirm or reject diagnostic outputs, add observations from the field, and record the action taken.

The knowledge exchanged through this layer includes three main

types. First, validation knowledge records whether a system output is confirmed, corrected or rejected by the engineer. Second, contextual knowledge captures site-specific observations, such as asset condition, environmental factors, access constraints, visible damage or unexpected task conditions. Third, procedural knowledge records maintenance actions and outcomes, including inspections completed, repairs performed and whether the fault condition was resolved.

These inputs are represented in the MA-CDT as structured feedback records linked to asset identifiers, fault concepts, severity levels, maintenance actions and timestamps. Where appropriate, they can update semantic assertions, refine decision rules or provide labelled cases for later model review. In this way, AR supports human-system knowledge exchange by making system outputs inspectable and by capturing operator feedback in a form that can be reused by the semantic and diagnostic components.

VR environments support training and scenario exploration rather than direct operational refinement. They allow engineers and supervisors to rehearse diagnostic and repair procedures, examine rare or hazardous fault scenarios, and review maintenance plans without exposing assets or staff to operational risk. In the current study, VR is included as an architectural interface for future training and planning functions, rather than as part of the empirically validated diagnostic pipeline.

3.2.6. Human operator layer

The human operator layer defines how domain expertise is incorporated into the MA-CDT architecture. Human operators are treated as external decision actors who interact with the CDT through defined interface channels. Their role is to interpret system outputs, provide contextual judgement, validate or correct diagnostic hypotheses, and record maintenance actions and outcomes.

Maintenance engineers interact with the system primarily during on-

site inspection and repair activities. Through AR interfaces and, where appropriate, VR-based simulation support, engineers receive context-aware diagnostic explanations, asset-specific maintenance history, and guided repair instructions aligned with the physical task at hand. Information is presented at an operational level, prioritising immediacy, spatial relevance, and actionability. Engineers retain authority over execution decisions, while their validation, annotations, and corrective actions provide feedback that supports subsequent model refinement and knowledge updates.

Supervisors and controllers interact with the system at a coordination level. They review aggregated diagnostic outputs, maintenance priorities, predictive recommendations and resource implications. Their feedback concerns decision context rather than direct fault validation. For example, they may adjust intervention priorities, record operational constraints, or approve maintenance actions based on network-level risk and resource availability. These inputs support the alignment between diagnostic reasoning and maintenance planning.

The knowledge exchanged between human operators and the MA-CDT includes confirmation, correction, contextual annotation, intervention records and decision constraints. Within the architecture, this knowledge is represented as structured semantic feedback linked to assets, faults, severity levels, actions and outcomes. The feedback can support three forms of refinement: updating the semantic state of an asset, adjusting rule-based constraints, and creating reviewed cases for later model improvement.

Fig. 1 presents the system-level architecture of the MA-CDT, specifying the functional layers, agent roles, and semantic communication mechanisms required to support distributed cognitive maintenance. The human operator is positioned as an external decision actor connected to the CDT through the AR/VR presentation layer and feedback channels. To operationalise the architecture within a realistic engineering context, Fig. 4 maps these architectural components onto the maintenance

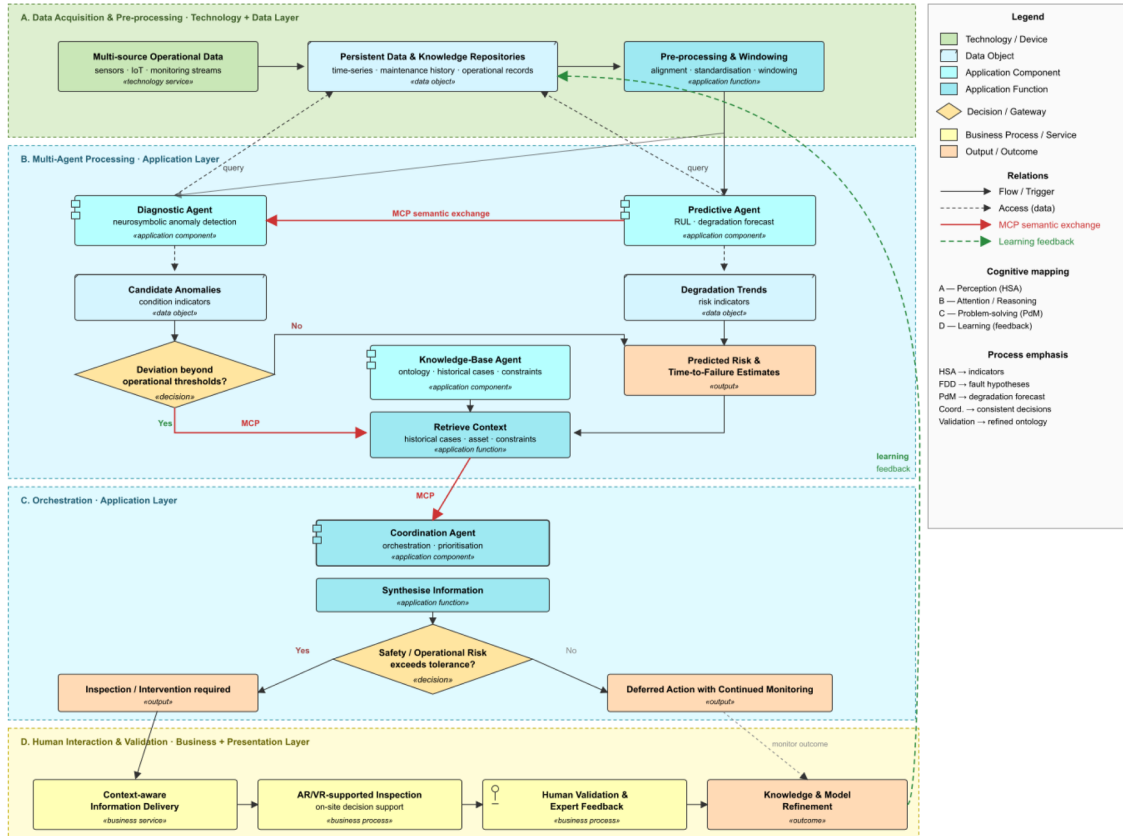


Fig. 4. Operational information flow diagram of the MA-CDT architecture.

lifecycle, thereby clarifying how perception, reasoning, coordination, and learning functions are situated across inspection, diagnosis, and decision-making phases. While the current implementation focuses on validation of semantic refinement, the architectural design supports broader forms of human cognitive contribution. In particular, expert knowledge can influence system behaviour at three levels: (i) operational validation, where engineers confirm or reject system outputs, (ii) semantic refinement, where domain concepts, thresholds and relations are updated, and (iii) strategic input, where supervisors introduce contextual constraints and priorities that influence decision-making policies. In the present study, only the first two levels are empirically instantiated. The third level, involving higher-order reasoning and adaptive policy formation, is defined at the architectural level and remains a direction for future work.

3.3. Functional flow and communication

The workflow mapping in Fig. 4 situates the architectural layers within specific maintenance phases and clarifies where cognitive functions are enacted during operational practice. To further explain this mapping, the interaction between HSA, FDD, and PdM is described in terms of information transformation and decision flow.

The workflow in Fig. 4 reflects the interaction between these three maintenance processes. HSA corresponds to the initial transformation of raw sensor data into condition indicators and anomaly candidates. This is implemented in the data acquisition and diagnostic stages, where time-series data are structured and deviations from normal behaviour are identified. The threshold-based decision point further filters anomalies based on operational criteria, determining whether they require deeper diagnostic interpretation.

FDD is realised through the diagnostic agent and the knowledge base agent. Candidate anomalies are interpreted as fault hypotheses by combining data-driven detection with engineering thresholds and semantic constraints. The knowledge base agent enriches these hypotheses with contextual information, including historical cases, asset characteristics and operational conditions. This allows anomalies to be translated into structured fault states with associated severity and explanatory context.

PdM operates in parallel with FDD through the predictive agent. It analyses degradation trends and historical data to estimate risk levels and time-to-failure. The outputs of PdM are integrated with diagnostic results through the coordination agent. In this step, diagnostic evidence and predictive assessments are synthesised to produce consistent maintenance decisions.

Information flows across these processes in a structured manner. HSA outputs provide inputs to FDD in the form of anomaly candidates. FDD transforms these into fault representations, which are then reused by PdM for prediction and planning. The coordination agent combines outputs from both processes to evaluate risk and determine whether intervention is required.

Human operators participate at different points in this flow. Maintenance engineers are primarily involved after the FDD stage, where they validate diagnostic outputs and provide contextual observations. Supervisors operate at the coordination stage, where predictive outputs and system-level risks are assessed before decisions are finalised.

Validated outcomes and expert feedback are incorporated into the knowledge base, allowing updated fault interpretations, contextual knowledge and maintenance outcomes to be reused in subsequent HSA, FDD and PdM processes.

Within this mapped structure, the diagnostic module occupies a central role in the FDD phase, where raw inspection data are transformed into interpretable hypotheses and actionable assessments. The following section therefore instantiates the diagnostic component as an operational realisation of the perception and reasoning layers defined in Fig. 1 and positioned within the lifecycle framework illustrated in Fig. 4.

3.4. Diagnostic agent: neurosymbolic reasoning design

This section details the instantiation of the diagnostic module as a representative component of the proposed MA-CDT architecture. The module operationalises the perception and reasoning objectives outlined in Table 1 and implements them through a structured neurosymbolic pipeline. By grounding the architectural abstractions in a concrete implementation, the diagnostic agent provides the primary vehicle for empirical validation in this study.

In this work, neurosymbolic reasoning is realised through a decision-level integration between neural inference and symbolic reasoning components. This implementation corresponds to a decision-level hybrid neurosymbolic architecture, where symbolic reasoning operates as a post-inference validation and interpretation layer rather than being embedded within neural model training. While this approach differs from tightly integrated neurosymbolic systems that incorporate symbolic constraints directly into representation learning, it is intentionally adopted to preserve modularity, interpretability and engineering compatibility within industrial maintenance contexts. The architecture therefore prioritises explainability and controllability over deep integration, while still enabling interaction between learned patterns and symbolic knowledge. Neural models are used to detect anomalies from sensor data, while symbolic rules and ontology-based constraints are applied to interpret and validate these outputs. The symbolic component operates after neural inference and does not influence model training or internal representations. As shown in Fig. 5, the diagnostic agent consists of two parallel processing branches followed by a decision integration stage.

The neural branch performs multi-modal feature extraction and temporal pattern analysis. Sensor data streams, including vibration, temperature, and positional information, are first pre-processed through a feature extraction stage. The Convolutional Neural Network (CNN) feature encoder captures spatial correlations and localised signal features, while the LSTM temporal analyser models sequential dependencies and degradation trends across time. These features are aggregated in the anomaly detection network, which outputs probabilistic anomaly scores indicating deviations from normal operational patterns. This branch ensures high sensitivity to complex fault signatures that may not be linearly separable or explicitly rule-defined.

Parallel to the neural processing, a symbolic reasoning pipeline encodes domain knowledge and structural relationships derived from the railway ontology. The railway ontology loader accesses the structured knowledge base, including asset hierarchies, failure modes, and maintenance rules. The SPARQL Query Engine retrieves relevant patterns and historical fault contexts from the 4D knowledge graph, which captures the spatial configuration of railway assets together with their temporal evolution and condition states over the maintenance lifecycle. The rule-based classifier then applies logic-based inference over retrieved entities to generate candidate fault hypotheses consistent with domain constraints.

Outputs from the neural and symbolic branches are integrated through the neurosymbolic fusion engine, which acts as the cognitive decision layer. The fusion process combines the neural anomaly embeddings and symbolic rule matches to produce a fault hypothesis, allowing the agent to achieve both accuracy and interpretability. This module includes three sub-functions:

- Confidence scoring, which quantifies diagnostic certainty by weighting neural probabilities with symbolic rule confidence;
- Explanation generator, which produces human-interpretable justifications linking detected patterns to ontology concepts and maintenance rules;
- Symbolic validator, which verifies the consistency of neural outputs against logical constraints, thereby reducing false alarms and hallucinated predictions.

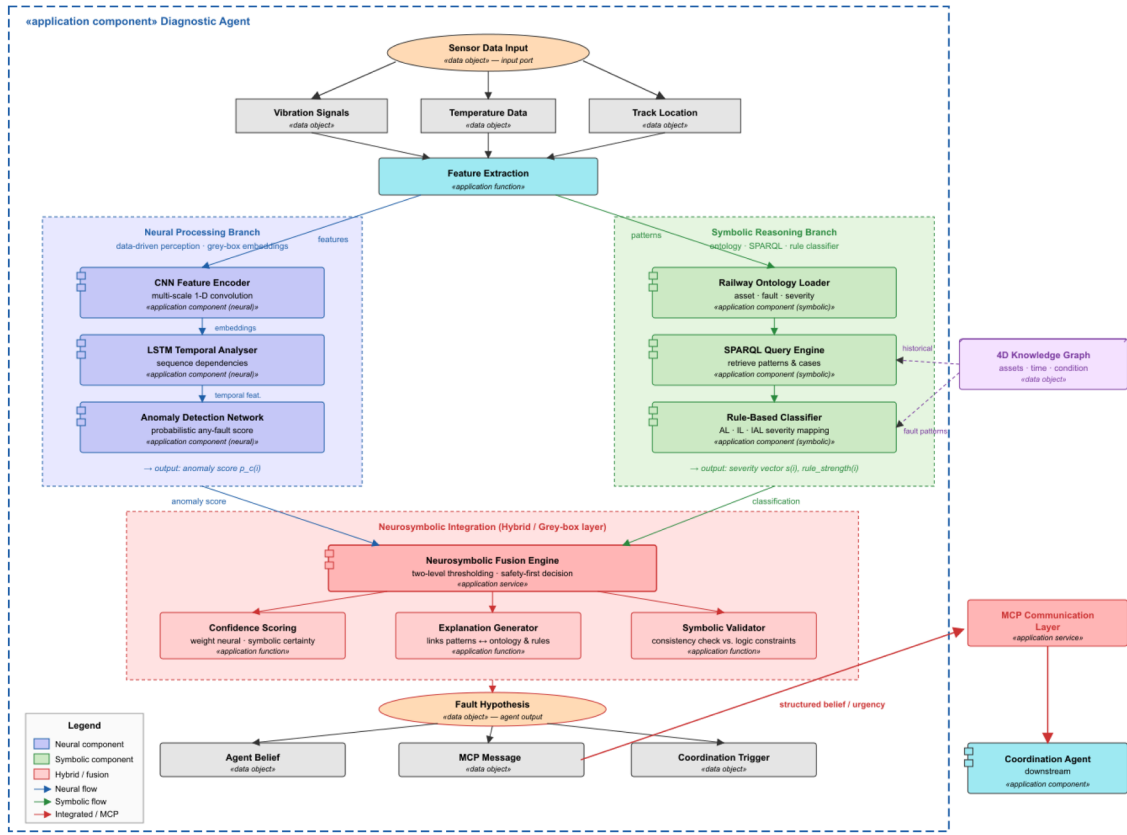


Fig. 5. Detailed architecture of the diagnostic agent showing the neurosymbolic integration approach.

Validated fault hypotheses are then formatted into structured messages and transmitted to the coordination agent through the MCP layer. The diagnostic agent communicates its belief state, reasoning explanation, and urgency indicators, enabling downstream agents to synthesise and act upon this information. The coordination stage ensures that anomaly assessments are propagated consistently across the multi-agent system and linked to AR/VR interfaces for operator validation.

4. Case study: diagnosis of railway track geometry

4.1. Case overview

In many countries, rail transport plays a critical role in economic cohesion and growth. Given this, it is of the utmost importance to maintain the railway network, and in particular the railway tracks. An example can be seen in England, Scotland and Wales, where the network operator, Network Rail spent £921 million in 2023–24 on track maintenance alone [13]. Given such large sums, improving efficiency and reducing costs in track maintenance is essential for network operators. This study investigates a diagnostic methodology using real-world data from Network Rail.

Railway track geometry provides a controlled yet challenging environment in which to examine the proposed neurosymbolic diagnostic architecture. Geometry cars measure track at metre-level resolution along fixed routes, producing long one-dimensional profiles rather than high-frequency vibration signals. Each position is described by four engineering quantities: three-metre twist, gauge, and left/right alignment, whose semantics are standardised in maintenance manuals and directly linked to operational limits. Fault conditions are defined normatively by the Network Rail standard, which specifies alert, intervention and immediate-action limits for each parameter. Accordingly, ground-truth labels used in the evaluation are derived directly from these normative engineering thresholds. Each geometry record is

therefore associated with a deterministic fault label determined by the corresponding threshold exceedance conditions.

This combination of clear semantics and real-world variability creates a characteristic tension. On the symbolic side, low dimensionality and codified thresholds make rule-based reasoning precise, reproducible and easy to audit. On the data side, geometry logs contain measurement noise, localised artefacts, alignment drift and residual effects of historic interventions. Many patterns of interest manifest as small yet coherent excursions around the formal limits, or as clusters of sub-threshold exceedances that are too weak to trigger an alarm but are routinely noticed by experienced inspectors. As a result, purely rule-driven diagnostics tend to miss emerging defects, while unconstrained neural models risk reacting to benign irregularities and inflating false alarms.

The case study uses a corpus of approximately 200,000 geometry samples collected from routine inspection runs. Each sample records the four geometry features and the corresponding rule-based fault indicators, aggregated into three fault types: gauge, twist and alignment. The dataset is highly imbalanced: around two-thirds of the samples correspond to healthy track; each individual fault type represents only a single-digit percentage of the data, with twist being the scarcest. The any-fault class, defined as the union of the three fault types at any severity, accounts for roughly one-third of all samples.

Diagnostic requirements follow operational practice. From a traffic management perspective, the primary question is binary: whether a given location should be treated as geometrically safe or as requiring follow-up, while the exact subtype and severity of the deviation can be resolved later using engineering rules and human inspection. The evaluation therefore reports both: i) binary performance for any-fault detection, which determines workload and safety; and ii) per-fault coverage, quantifying how often gauge, twist, and alignment events are captured by a positive decision.

A further complication arises from the sequential nature of the data. Fault labels are attached to individual metre positions, yet physical

defects may influence several neighbouring measurements and detections are useful even when slightly displaced along the track. The case study therefore adopts an offset-tolerant matching scheme, which credits detections that fall within a small spatial neighbourhood of the labelled position, reflecting how inspectors interpret geometry reports in practice.

Within this setting, the case study instantiates and evaluates the diagnostic core of the MA-CDT architecture. The objective is to examine whether combining neural inference with rule-based engineering constraints can produce stable and interpretable diagnostic decisions under noisy and imbalanced geometry measurements. The detailed evaluation protocol and hypotheses are presented in Section 5.

The empirical evaluation is therefore limited to the diagnostic component of the proposed MA-CDT architecture. Other components, including predictive maintenance functions, coordination mechanisms, and AR/VR-based human interaction, are defined at the architectural level but are not empirically validated in this study. The results should be interpreted as evidence of the feasibility of the diagnostic reasoning approach, rather than as a full-system validation of the entire architecture.

4.2. Implementation of the diagnostic agent

The Diagnostic Agent is implemented as a modular pipeline that mirrors the conceptual structure of the neurosymbolic core (shown in Fig. 6). The input to the agent is a normalised, windowed representation of track geometry; the output is a fault/no-fault decision, accompanied by per-fault indicators and a reasoning tag that records which component dominated the decision.

Pre-processing and windowing. Raw geometry records are first aligned to a common spatial index and filtered to remove obvious acquisition artefacts. For each route, four features are retained: three-metre twist, gauge, left alignment and right alignment. To stabilise neural training and prevent data leakage, feature-wise means and standard deviations are computed on the training split only and then used to standardise all splits. The normalised sequences are segmented into overlapping windows of length $L = 21$ samples, using a stride of one metre. Each window is labelled by the fault status of its centre sample, so that the model learns to use neighbouring context while remaining evaluated on precise positions.

Neural branch. The neural component is a compact CNN-LSTM-Attention network implemented in PyTorch. Three parallel one-dimensional convolutional branches with kernel sizes 3, 5 and 7 extract local patterns at different spatial scales from each window; their outputs are concatenated and passed to a two-layer bidirectional LSTM

that captures longer-range structure along the track. A multi-head self-attention layer re-weights the LSTM outputs to focus on the most informative positions within the window. A per-time-step classification head then produces logits for four classes: no fault, gauge fault, twist fault and alignment fault.

Let $\mathbf{x}_i \in \mathbb{R}^4$ denote the geometry vector at location i , and

$$p_c(i) = \Pr \left(y_i = c \mid \mathbf{x}_{i-\frac{L}{2}+1}^{\frac{L}{2}} \right), \quad c \in \{\text{none, gauge, twist, alignment}\}, \quad (1)$$

denote the class probabilities output by the network for the centre position of each window. Training uses a weighted cross-entropy loss with class weights derived from a clipped inverse square-root of class frequencies, so that rare twist and alignment events retain sufficient influence despite their scarcity. Early stopping and learning-rate scheduling on a held-out validation set prevent over-fitting.

Although the network outputs a full four-way distribution, its predictions are subsequently collapsed into binary any-fault decisions where needed, and per-fault scores are retained for detailed analysis and for conditioning the symbolic validator. This design keeps the neural branch flexible: it can express preferences between fault types, while aligning the final decision semantics with maintenance practice.

Symbolic branch. In parallel, the symbolic component evaluates each centre sample using a two-layer semantic pipeline. The first layer implements the normative rule set defined in Network Rail documentation. For each geometry mechanism

$$f \in \{\text{twist, gauge, alignment}\}, \quad (2)$$

the corresponding measurement $x_f(i)$ is mapped to a discrete severity level using the alert (AL), intervention (IL) and immediate-action limits (IAL):

$$\text{severity}_f(i) = \begin{cases} 3, & |x_f(i)| \geq \theta_f^{\text{IAL}}, \\ 2, & \theta_f^{\text{IL}} \leq |x_f(i)| < \theta_f^{\text{IAL}}, \\ 1, & \theta_f^{\text{AL}} \leq |x_f(i)| < \theta_f^{\text{IL}}, \\ 0, & |x_f(i)| < \theta_f^{\text{AL}}. \end{cases} \quad (3)$$

The threshold parameters θ_f^{IAL} , θ_f^{IL} and θ_f^{AL} are derived from the alert, intervention and immediate-action limits defined in Network Rail maintenance standards. All geometry measurements and threshold values θ are expressed in millimetres (mm) in accordance with the Network Rail track-geometry standard. These thresholds represent

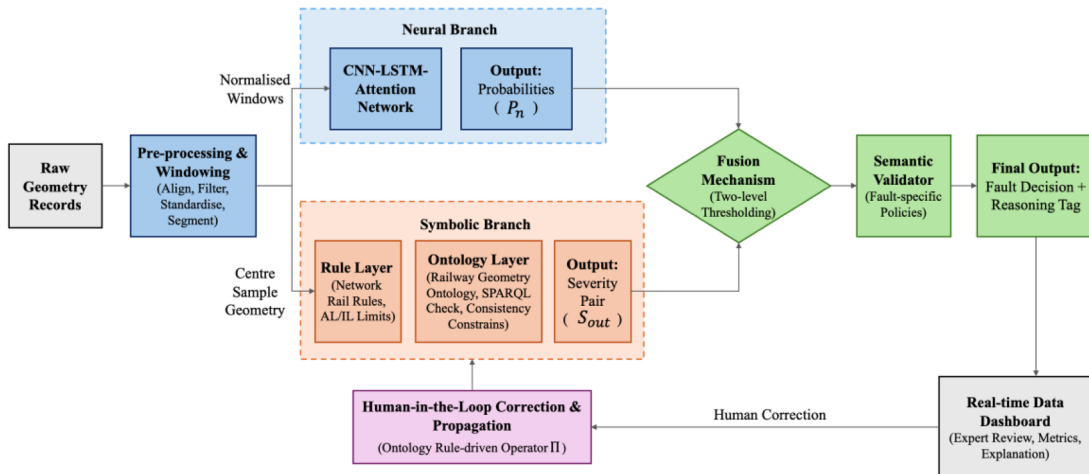


Fig. 6. Neurosymbolic diagnostic agent implementation.

normative engineering constraints used in operational inspection systems. A global rule-strength score:

$$\text{rule_strength}(i) = \max_f \text{severity}_f(i), \quad (4)$$

summarises the most critical exceedance and provides deterministic, auditable symbolic evidence.

The symbolic branch implements a track-geometry subset of the ontology structure described in Section 3.2.3. This subset focuses on the representation of track segments, geometry measurements, fault states and severity levels required for diagnostic reasoning.

To support semantic validation, a lightweight railway geometry ontology is constructed to capture relationships between track segments, measurement entities and admissible fault configurations. Geometry mechanisms, including twist, gauge and alignment, are represented as measurement entities linked to specific track-segment instances. These measurements are associated with fault states and severity levels derived from engineering thresholds.

The ontology encodes admissible combinations of severity levels across multiple geometry mechanisms, reflecting engineering knowledge about multi-mechanism compatibility and physical constraints. This allows rule-derived severity assignments to be checked against a set of semantically valid configurations.

The second layer enriches these rule-derived severities using a lightweight railway geometry ontology. The ontology loader retrieves the admissible semantic configurations $\mathcal{C}$, and a SPARQL query engine checks whether the severity vector:

$$\mathbf{s}(i) = (\text{severity}_{\text{twist}}(i), \text{severity}_{\text{gauge}}(i), \text{severity}_{\text{alignment}}(i)), \quad (5)$$

is consistent with track-geometry dependencies and multi-fault compatibility constraints.

If $\mathbf{s}(i)$ violates any ontological rule, it is projected onto the nearest admissible configuration

$$\mathbf{s}^*(i) = \arg \min_{\mathbf{z} \in \mathcal{C}} d(\mathbf{z}, \mathbf{s}(i)), \quad (6)$$

ensuring that symbolic outputs remain physically meaningful and semantically coherent. The symbolic branch therefore outputs the pair $(\mathbf{s}^*(i), \text{rule_strength}(i))$, combining rule-based interpretation with ontology-guided semantic validation.

Fusion mechanism. Neural and symbolic evidence are combined using a two-level thresholding strategy. For each location i , the neural any-fault confidence is taken as

$$p_{\text{nf}}(i) = 1 - p_{\text{none}}(i) = p_{\text{gauge}}(i) + p_{\text{twist}}(i) + p_{\text{alignment}}(i), \quad (7)$$

and the fusion module applies a safety-first decision rule

$$\hat{\mathbf{y}}_{\text{fusion}}(i) = \begin{cases} 1, & \text{rule_strength}(i) \geq 2, \\ 1(p_{\text{nf}}(i) \geq \tau_{\text{low}}), & \text{rule_strength}(i) = 1, \\ 1(p_{\text{nf}}(i) \geq \tau_{\text{high}}), & \text{rule_strength}(i) = 0, \end{cases} \quad (8)$$

where $\hat{\mathbf{y}}_{\text{fusion}}(i) \in \{0, 1\}$ denotes the any-fault decision, τ_{low} and τ_{high} are calibration thresholds with $\tau_{\text{low}} < \tau_{\text{high}}$, and $1(\cdot)$ is the indicator function. High-severity symbolic exceedances (intervention or immediate-action, $\text{rule_strength} \geq 2$) are always preserved and cannot be overridden by the neural branch. For low-severity rule triggers ($\text{rule_strength} = 1$), neural confidence acts as a filter: weak neural support leads to suppression of isolated marginal exceedances, whereas strong neural support confirms them as actionable. In the absence of any rule trigger ($\text{rule_strength} = 0$), the neural branch may still raise a fault if its confidence exceeds the stricter threshold τ_{high} , enabling detection of patterns that fall just below formal limits but are consistently associated with historical faults. Thresholds are calibrated on the validation set to maximise F_1 under a target upper bound on the false alarm rate.

Semantic validator and reasoning tags. The final stage is a per-fault semantic validator that encodes fault-specific decision policies. Let $\hat{c}(i) = \arg \max_{c \neq \text{none}} p_c(i)$ denote the dominant fault type predicted by the neural branch at location i . The validator applies type-dependent constraints, for example

$$\hat{\mathbf{y}}_{\text{twist}}(i) = \begin{cases} 1, & \text{severity}_{\text{twist}}(i) \geq 1 \text{ or } p_{\text{twist}}(i) \geq \theta_{\text{twist}}, \\ 0, & \text{otherwise,} \end{cases} \quad (9)$$

and

$$\hat{\mathbf{y}}_{\text{gauge}}(i) = \begin{cases} 1, & p_{\text{gauge}}(i) \geq \theta_{\text{gauge}}, \\ 0, & \text{otherwise,} \end{cases} \quad (10)$$

with analogous rules for alignment. In practice, twist decisions remain conservative and privilege symbolic evidence because of their safety-critical nature and severe class imbalance, whereas gauge and alignment decisions rely more heavily on neural confidence, reflecting their higher signal-to-noise ratio in the data. For each sample, the validator reconciles the fusion outcome with these policies and assigns a reasoning tag such as symbolic-dominated, neural-dominated or hybrid-agreement. These tags are logged alongside predictions and are later used to interpret system behaviour and structure the ablation studies reported in the subsequent section.

Human-in-the-loop correction and propagation. After the agent produces a fault hypothesis, experts may validate or amend the decision via the MCP communication interface. A correction at location i is represented as

$$\delta(i) = \{\text{updated per} - \text{fault label and severity}\}, \quad (11)$$

and is not applied as a local override. Instead, the correction is propagated through the ontology using a rule-driven operator Π :

$$\Delta = \Pi(\delta(i), \mathcal{C}), \quad (12)$$

where $\mathcal{C}$ defines the admissible semantic relationships between geometry mechanisms and their compatible fault configurations. The propagation operator expands a single expert correction into a set of semantically consistent updates by traversing ontology constraints and dependency rules. In the present implementation, Π is realised through SPARQL rule queries over the railway geometry ontology.

Algorithmically, the propagation proceeds as follows. First, the corrected fault label and severity are mapped to the corresponding ontology entities associated with the affected track segment. Second, related mechanisms and dependency rules defined in $\mathcal{C}$ are retrieved to identify admissible co-occurring fault states. Third, rule constraints are applied to generate a set of semantically consistent candidate updates. Finally, these propagated adjustments form the update set Δ , which refines the agent's internal belief state. The propagated updates influence subsequent refinement cycles without altering the parameters of the neural branch, allowing the diagnostic logic to adapt to route-specific conditions while preserving semantic consistency across mechanisms.

Real-time data dashboard and expert-in-the-loop. To demonstrate that the Diagnostic Agent can operate under streaming conditions, the neurosymbolic pipeline was deployed within a web-based real-time dashboard, shown in Fig. 7. The interface ingests live track-geometry measurements, applies the full diagnostic workflow, and visualises twist, gauge and alignment traces alongside their corresponding alert and intervention limits. A performance panel presents the live confusion matrix and key indicators, including F1-score, precision, recall and false-alarm rate. Detected alerts are listed in an interactive review panel that enables experts to confirm or reject each alarm and, where relevant, record short explanatory notes. These judgements are logged as structured feedback and provide an initial mechanism for expert-in-the-loop monitoring, forming the operational foundation for future integration of human corrections into the diagnostic refinement process.

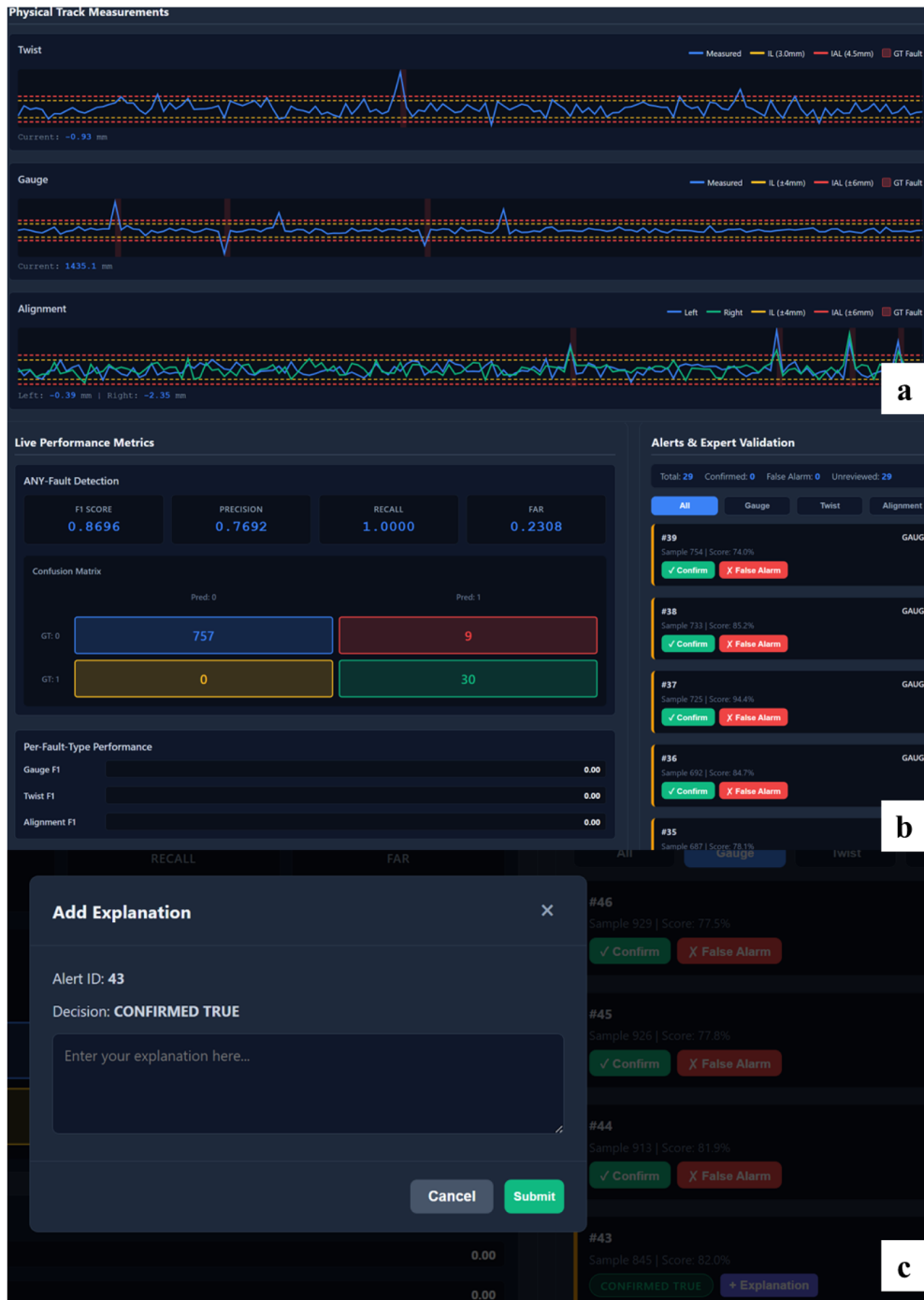


Fig. 7. Real-time diagnostic dashboard for the MA-CDT diagnostic agent.

(a) Streaming track-geometry signals with engineering thresholds (AL/IL/IAL) and detected anomalies. (b) Live binary performance metrics, including confusion matrix and precision-recall statistics. (c) Expert validation panel and explanation interface for reviewing and confirming or rejecting model-generated alerts.

5. Empirical evaluation of the diagnostic agent

5.1. Evaluation objectives and hypotheses

The empirical evaluation aims to assess whether the instantiated diagnostic module achieves the design objectives associated with perception and reasoning as defined in Table 1. In particular, the evaluation examines whether the proposed neurosymbolic fusion mechanism:

- H1: reduces false positive rates relative to standalone neural anomaly detection;
- H2: preserves detection coverage for severe deviations under rule-strength constraints;
- H3: maintains semantic consistency across iterative updates without destabilising detection performance.

These hypotheses reflect the architectural intent of integrating data-driven perception with ontology-grounded constraint validation within a structured diagnostic pipeline.

5.2. Diagnostic variants

To evaluate the contribution of each cognitive mechanism within the Diagnostic Agent, four variants were implemented. All variants are grounded in the same underlying geometry measurements acquired from the Track Recording Vehicle (TRV) and processed through an identical preprocessing pipeline, including normalisation, windowing, and feature preparation. However, the effective inputs differ across variants depending on which architectural stages are activated.

Specifically, the neural-only variant operates directly on pre-processed geometry features and produces probabilistic fault scores. Variants involving symbolic reasoning additionally consume intermediate representations derived from geometry-based severity indicators, which are computed using engineering thresholds and domain rules. The full diagnostic agent further incorporates structured semantic inputs, including fault-type constraints and validation logic, which act on the fused neural-symbolic outputs rather than on raw sensor data. In this way, all variants share a common physical data source while progressively activating higher-level interpretive stages of the architecture.

Because the TRV onboard alarm flag originates from an external hardware subsystem, independent of the proposed architecture, it is treated strictly as an operational reference rather than an internal mechanism. Accordingly, it is renamed as an Operational-Alarm Baseline. The four variants are:

(1) Operational-Alarm Baseline (external reference).

This variant reproduces the behaviour of the onboard TRV alarm stream. Decisions correspond exactly to the alarm flag recorded during inspection runs. No geometry thresholds, neural inference or fusion logic are used. This baseline reflects how alerts are generated in current practice and provides an external reference for understanding how the proposed system differs from existing operational behaviour.

(2) Neural-Only Variant (perception without engineering semantics).

The neural CNN-LSTM-Attention model operates without symbolic information or validation. The probability of any-fault,

$$p_{nf}(i) = 1 - p_{none}(i), \quad (13)$$

is thresholded to obtain a binary decision using a calibration value determined on the validation set. This configuration captures the behaviour of a purely data-driven detector: highly sensitive to weak patterns but unaware of severity definitions, rule structure, or safety semantics, and therefore prone to noise-induced false alarms.

(3) Fusion-Only Variant (hybrid integration without consistency enforcement).

This variant preserves both neural and symbolic branches and applies the two-level fusion rule,

$$\hat{y}_{fusion}(i) = \begin{cases} 1, & \text{rule_strength}(i) \geq 2, \\ 1(p_{nf}(i) \geq \tau_{low}), & \text{rule_strength}(i) = 1, \\ 1(p_{nf}(i) \geq \tau_{high}), & \text{rule_strength}(i) = 0, \end{cases} \quad (14)$$

but omits the semantic validator. As a result, contradictions between neural and symbolic cues remain unresolved. Weak neural anomalies cannot be suppressed when they lack symbolic support, and fault-specific safety constraints, particularly the conservative treatment of twist, are not enforced.

(4) Full Diagnostic Agent.

This variant implements the complete neurosymbolic pipeline:

neural inference, symbolic severity evaluation, two-level fusion and structured semantic validation. The validator suppresses noise-driven anomalies, resolves conflicts and applies type-specific semantics, yielding stable and interpretable diagnostic behaviour. This configuration represents the intended operation of the proposed system.

5.3. Evaluation metrics

Evaluation is formulated as a binary any-fault detection task, consistent with operational practice. For each variant, predictions are compared against ground-truth fault locations using the offset-tolerant matching scheme commonly employed in track-geometry diagnostics. Four domain-specific quantities define the confusion matrix:

- True Positive (TP): a reported fault lies within the tolerance window of a ground-truth fault.
- False Positive (FP): a reported fault occurs where the ground truth is healthy, generating unnecessary workload.
- False Negative (FN): a true deviation is not detected, directly affecting safety.
- True Negative (TN): a healthy segment is correctly identified.

Derived metrics (Table 7) assess detection accuracy, false-alarm behaviour and operational cost. FAR is particularly emphasised in relation to H1, while recall directly informs H2.

5.4. Evaluation protocol

All experiments were conducted on the university high-performance computing (HPC) cluster using PyTorch. The neural diagnostic model contains approximately 293k trainable parameters and was trained with early stopping based on validation loss. The dataset was divided into training (70%), validation (15%), and test (15%) subsets, corresponding to approximately 140k, 30k, and 30k samples respectively.

All variants follow the same inference pipeline structure. Each metre-indexed geometry record is transformed into a 21-sample window, standardised using statistics from the training set, and passed to both neural and symbolic components. The neural branch outputs class probabilities over the four geometry fault categories; the symbolic branch evaluates the Network Rail standard thresholds. Fusion and semantic validation are executed only for variants in which they are enabled.

The dataset is partitioned using stratified sampling to preserve the relative frequency of fault categories across the training, validation and

Table 7

Evaluation metrics used for comparative analysis.

Metric	Definition	Purpose in this study
False Alarm Rate (FAR)	$FAR = \frac{\#False\ Alarms}{\#Total\ Alarms} \quad (15)$	Measures the trustworthiness of the alarm stream by quantifying spurious alerts.
Precision	$Precision = \frac{TP}{TP + FP} \quad (16)$	Indicates the proportion of alarms that correspond to true faults, reflecting operational usefulness.
Recall	$Recall = \frac{TP}{TP + FN} \quad (17)$	Ensures safety by evaluating the proportion of true anomalies that are successfully detected.
F1-score	$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (18)$	Provides a balanced measure of detection accuracy when precision and recall must be traded off.
Expert Workload	$Workload = \frac{\#Alerts\ requiring\ validation}{\#Total\ Alarms} \quad (19)$	Represents the operational cost arising from alerts that the system cannot resolve and thus require expert intervention.

test subsets. Only the test set is used for reported performance. Ground truth plays no role during inference.

For each variant, predicted fault locations are matched to the ground truth under the offset tolerance. The metrics defined in Section 5.3 are computed directly from the resulting confusion matrix. To account for stochastic variation in the neural component, all reported metrics are averaged across five independent runs, and confidence intervals are estimated using bootstrap resampling. Because all variants operate on identical inputs and share identical preprocessing stages, differences in performance reflect the cognitive mechanisms removed or retained in each configuration.

A further evaluation setting is applied only to the Full Agent to reflect its intended human-in-the-loop operating mode. After each inference pass, the agent receives a correction signal derived from the ground-truth labels; this signal is used solely to update the agent's semantic belief state through ontology-guided propagation, while all neural parameters remain fixed. A new inference pass is then executed using the updated semantic state, yielding the next refinement cycle. Ten such cycles are performed, mirroring how expert feedback would gradually constrain inconsistent hypotheses in operational use. No other variant undergoes iterative refinement because they lack the symbolic structures required for ontology-based updates.

This controlled comparison framework enables a precise assessment of how: i) neural perception behaves without semantic grounding, ii) symbolic thresholds constrain or stabilise predictions, iii) fusion without validation propagates inconsistencies, and iv) the full agent reconciles heterogeneous cues into stable, safety-aligned decisions. This controlled setup ensures that performance differences can be attributed directly to the cognitive mechanisms activated in each configuration.

5.5. Validation of hypotheses

5.5.1. Validation of H1: reduction of false positives

H1 posits that the integration of symbolic severity evaluation and semantic validation reduces false positive rates relative to standalone neural anomaly detection. Table 8 demonstrates a monotonic decrease in false positives across the four variants. The Neural-Only configuration substantially reduces false alarms compared to the Operational-Alarm Baseline, indicating that learned geometric representations already improve selectivity relative to hardware-triggered alarms. However, residual false positives remain considerable (FP=525), reflecting the sensitivity of purely data-driven perception to noise-induced fluctuations.

The introduction of symbolic reasoning in the Fusion-Only configuration further reduces false positives (FP=249), confirming that severity thresholds introduce corrective structure into neural predictions. The Full Diagnostic Agent achieves the lowest false positive count (FP=144) and the lowest false alarm rate (FAR=0.0150), indicating that semantic validation effectively suppresses unsupported neural triggers and resolves contradictions between heterogeneous cues.

Per-fault analysis (Tables 9a; 9b; 9c) confirms this trend across twist, gauge, and alignment categories. Although absolute FAR values remain high for twist and alignment due to intrinsic oscillatory signal behaviour, incremental reductions are observed as cognitive mechanisms are progressively activated. These findings support H1 by demonstrating that neurosymbolic fusion and semantic validation systematically reduce spurious alarms relative to neural-only perception.

5.5.2. Validation of H2: maintenance of detection coverage

H2 proposes that detection coverage for high-severity deviations is preserved when symbolic constraints are introduced, ensuring that severe cases are not suppressed by semantic filtering. Across all hybrid configurations, recall remains at 1.0 in the binary evaluation (Table 8), indicating that no ground-truth faults are omitted. This behaviour arises from the architectural design of the symbolic layer rather than purely statistical optimisation: rule-based severity thresholds act as deterministic anchors that override neural predictions when high-severity deviations are detected. As a result, severe exceedances cannot be suppressed by semantic filtering.

Per-fault results similarly show recall values of 1.0 for twist, gauge, and alignment faults across the Full Agent configuration. The absence of false negatives demonstrates that symbolic validation operates conservatively and does not eliminate actionable anomalies. These results support H2, indicating that the integration of engineering thresholds and ontology-grounded constraints preserves detection coverage while reducing false alarms.

5.5.3. Validation of H3: iterative stability and semantic consistency

H3 states that semantic refinement through iterative belief updating maintains detection stability without destabilising performance. Table 10 reports performance across ten refinement rounds. Initial refinement cycles reduce false positives from 144 to 121, corresponding to an improvement in F1-score from 0.9925 to 0.9936. This behaviour indicates effective suppression of borderline neural activations as semantic constraints propagate.

Beyond the second iteration, moderate oscillations appear, with temporary increases in false positives (e.g., iteration 7 and iteration 10). Importantly, no false negatives are introduced throughout all iterations, and recall remains perfect. The strongest performance is observed at iteration 9 (F1=0.9945), after which marginal degradation suggests diminishing returns.

Per-fault iteration behaviour (Tables 11a; 11b; 11c) reveals that twist and alignment faults exhibit greater fluctuation due to intrinsic geometric noise, whereas gauge faults remain comparatively stable. These dynamics reflect the interaction between noise-sensitive neural perception and conservative symbolic thresholds.

The multiclass confusion matrix (Fig. 8) further demonstrates global decision coherence. Misclassifications predominantly occur between fault and normal classes when severity thresholds fall marginally below actionable limits. Cross-fault confusions remain minimal, indicating that semantic validation preserves category boundaries.

Collectively, these findings partially support H3. Iterative refinement stabilises predictions during early cycles and preserves safety, although later iterations may amplify noise-driven inconsistencies when semantic propagation exceeds the information content of the signal.

5.6. Empirical scope and limitations

The empirical evaluation presented in this study is confined to the diagnostic module of the proposed MA-CDT architecture. While the broader architecture defines predictive and coordination agents to support distributed cognitive maintenance, only the diagnostic component is instantiated and validated here. Consequently, the findings relate specifically to perception and reasoning mechanisms and should not be interpreted as a full-system validation of the entire multi-agent

Table 8
Overall binary performance of the four diagnostic variants.

Model	TP	FP	FN	TN	FAR	Precision	Recall	F1	Workload
Operational-Alarm Baseline	9423	8077	41	12459	0.4615	0.5385	0.9957	0.6989	0.5833
Neural-Only	9464	525	0	20011	0.0526	0.9474	1	0.9730	0.3330
Fusion-Only	9464	249	0	20287	0.0256	0.9744	1	0.9870	0.3238
Full Agent	9464	144	0	20392	0.0150	0.9850	1	0.9925	0.3203

Table 9a

Twist fault detection performance across diagnostic variants.

Model	TP	FP	FN	TN	FAR	Precision	Recall	F1
Operational-Alarm Baseline	2929	14146	11	12914	0.8285	0.1715	0.9963	0.2927
Neural-Only	2940	7049	0	20011	0.7057	0.2943	1	0.4548
Fusion-Only	2940	6773	0	20287	0.6973	0.3027	1	0.4647
Full Agent	2940	6668	0	20392	0.6940	0.3060	1	0.4686

Table 9b

Gauge fault detection performance across diagnostic variants.

Model	TP	FP	FN	TN	FAR	Precision	Recall	F1
Operational-Alarm Baseline	3728	13366	22	12884	0.7819	0.2181	0.9941	0.3577
Neural-Only	3750	6239	0	20011	0.6246	0.3754	1	0.5459
Fusion-Only	3750	5963	0	20287	0.6139	0.3861	1	0.5571
Full Agent	3750	5858	0	20392	0.6097	0.3903	1	0.5615

Table 9c

Alignment fault detection performance across diagnostic variants.

Model	TP	FP	FN	TN	FAR	Precision	Recall	F1
Operational-Alarm Baseline	2766	14291	8	12935	0.8378	0.1622	0.997116	0.2790
Neural-Only	2774	7215	0	20011	0.7223	0.2777	1	0.4347
Fusion-Only	2774	6939	0	20287	0.7144	0.2856	1	0.4443
Full Agent	2774	6834	0	20392	0.7113	0.2887	1	0.4481

Table 10

Overall iterative performance of the Full Diagnostic Agent.

Iteration	TP	FP	FN	TN	FAR	Precision	Recall	F1	Workload
baseline	9464	144	0	20392	0.0150	0.9850	1	0.9925	0.3203
1	9464	133	0	20403	0.0139	0.9861	1	0.9930	0.3199
2	9464	121	0	20415	0.0126	0.9874	1	0.9936	0.3195
3	9464	180	0	20356	0.0187	0.9813	1	0.9906	0.3215
4	9464	126	0	20410	0.0131	0.9869	1	0.9934	0.3197
5	9464	156	0	20380	0.0162	0.9838	1	0.9918	0.3207
6	9464	131	0	20405	0.0137	0.9863	1	0.9931	0.3198
7	9464	373	0	20163	0.0379	0.9621	1	0.9807	0.3279
8	9464	170	0	20366	0.0176	0.9824	1	0.9911	0.3211
9	9464	105	0	20431	0.011	0.989	1	0.9945	0.319
10	9464	314	0	20222	0.0321	0.9679	1	0.9837	0.3259

Table 11a

Iterative twist-fault detection performance of the Full Diagnostic Agent.

Iteration	TP	FP	FN	TN	FAR	Precision	Recall	F1
baseline	2940	6668	0	20392	0.6940	0.3060	1	0.4686
1	2940	6657	0	20403	0.6937	0.3063	1	0.4690
2	2940	6645	0	20415	0.6933	0.3067	1	0.4695
3	2940	6704	0	20356	0.6951	0.3049	1	0.4673
4	2940	6694	0	20366	0.6948	0.3052	1	0.4676
5	2940	6680	0	20380	0.6944	0.3056	1	0.4682
6	2940	6655	0	20405	0.6936	0.3064	1	0.4691
7	2940	6897	0	20163	0.7011	0.2989	1	0.4602
8	2940	6694	0	20366	0.6948	0.3052	1	0.4676
9	2940	6629	0	20431	0.6928	0.3072	1	0.4701
10	2940	6838	0	20222	0.6993	0.3007	1	0.4623

architecture. The neurosymbolic integration demonstrated in this study is limited to post hoc symbolic validation and decision-level fusion. Tighter forms of integration, where symbolic knowledge is embedded within neural learning processes, are not explored in the current implementation.

The empirical results can be interpreted in relation to the cognitive capabilities defined in [Section 3.2](#). The neural detection component operationalises perception by transforming raw geometry signals into anomaly representations. The prioritisation of detected anomalies

through threshold filtering reflects attention. The ontology-based semantic layer provides memory by linking current observations with historical fault knowledge and engineering standards. The neurosymbolic fusion process operationalises reasoning by combining data-driven evidence with symbolic constraints to produce interpretable fault hypotheses. The generation of maintenance-relevant outputs corresponds to problem-solving, while the incorporation of expert validation into semantic updates reflects learning. Although the empirical evaluation focuses on the diagnostic module, these results demonstrate

Table 11b

Iterative gauge-fault detection performance of the Full Diagnostic Agent.

Iteration	TP	FP	FN	TN	FAR	Precision	Recall	F1
baseline	3750	5858	0	20392	0.6097	0.3903	1	0.5615
1	3750	5847	0	20403	0.6093	0.3907	1	0.5619
2	3750	5835	0	20415	0.6088	0.3912	1	0.5624
3	3750	5894	0	20356	0.6112	0.3888	1	0.5560
4	3750	5840	0	20410	0.609	0.391	1	0.5622
5	3750	5870	0	20380	0.6102	0.3898	1	0.561
6	3750	5845	0	20405	0.6092	0.3908	1	0.562
7	3750	6087	0	20163	0.6188	0.3812	1	0.552
8	3750	5884	0	20366	0.6108	0.3892	1	0.5604
9	3750	5819	0	20431	0.6081	0.3919	1	0.5631
10	3750	6028	0	20222	0.6165	0.3835	1	0.5544

Table 11c

Iterative alignment-fault detection performance of the Full Diagnostic Agent.

Iteration	TP	FP	FN	TN	FAR	Precision	Recall	F1
baseline	2774	6834	0	20392	0.7113	0.2888	1	0.4481
1	2774	6823	0	20403	0.7110	0.2890	1	0.4485
2	2774	6811	0	20415	0.7106	0.2894	1	0.4489
3	2774	6870	0	20356	0.7124	0.2876	1	0.4468
4	2774	6816	0	20410	0.7107	0.2893	1	0.4487
5	2774	6846	0	20380	0.7116	0.2884	1	0.4476
6	2774	6821	0	20405	0.7109	0.2891	1	0.4485
7	2774	7063	0	20163	0.718	0.282	1	0.4399
8	2774	6860	0	20366	0.7121	0.2879	1	0.4471
9	2774	6795	0	20431	0.7101	0.2899	1	0.4495
10	2774	7004	0	20222	0.0321	0.2837	1	0.442

Test Set					
Actual \ Prediction	Normal	Twist	Gauge	Alignment	SUM
Normal	20222 67.41%	0 0.00%	1 0.00%	0 0.00%	20223 100.00% 0.00%
Twist	0 0.00%	2940 9.80%	0 0.00%	0 0.00%	2940 100.00% 0.00%
Gauge	192 0.64%	0 0.00%	3749 12.50%	0 0.00%	3941 95.13% 4.87%
Alignment	122 0.41%	0 0.00%	0 0.00%	2774 9.25%	2896 95.79% 4.21%
SUM	20536 98.47% 1.53%	2940 100.00% 0.00%	3750 99.97% 0.03%	2774 100.00% 0.00%	29685 / 30000 98.95% 1.05%

Fig. 8. Multiclass confusion matrix for the full agent on the test set.

how core cognitive functions are instantiated within the MA-CDT architecture and provides a basis for extending cognition across additional maintenance stages.

Learning within the evaluated system is restricted to semantic belief refinement. Neural model parameters remain fixed during iterative cycles, and updates occur solely within the ontology-guided constraint layer. The knowledge evolution process therefore reflects controlled semantic propagation rather than autonomous rule generation, online retraining, or cross-asset transfer learning. The scope of learning

demonstrated in this study is intentionally bounded to ensure stability and interpretability within operationally conservative maintenance settings.

Performance is partially constrained by the characteristics of operational track-geometry data. Severe class imbalance and the oscillatory nature of twist and alignment signals introduce inherent ambiguity near severity thresholds. As a result, absolute false alarm rates for certain fault types remain high despite relative improvements across configurations. These behaviours reflect properties of the underlying signal

distribution rather than limitations of the fusion mechanism alone.

Evaluation is conducted on a single operational dataset derived from one inspection context. Cross-network validation, longitudinal deployment studies, and adaptive retraining scenarios fall outside the scope of this work. Future extensions would be required to assess generalisability across different infrastructure environments and inspection regimes.

In addition, several architectural components remain at the conceptual level in the present study. In particular, AR/VR interfaces are defined as mechanisms for human interaction, contextual information delivery and feedback, but their impact on usability, interpretability and decision-making is not empirically evaluated. Similarly, predictive maintenance and coordination functions are specified within the architecture but not instantiated in the current case study. The reported results therefore reflect the behaviour of the diagnostic component rather than the full MA-CDT system. Further work is required to evaluate these components and to assess the scalability of the proposed architecture across different assets, maintenance scenarios and operational contexts.

The role of human operators in the current implementation is also limited to supervisory and corrective interaction. Human input is incorporated through validation of diagnostic outputs, contextual annotation and semantic refinement of constraints. This supports consistency and interpretability, but does not constitute full human-machine co-cognition. Human expertise is not directly integrated into higher-level reasoning processes such as adaptive decision-making, policy formation or restructuring of diagnostic logic. The broader architectural design allows for deeper integration of human knowledge into reasoning and coordination processes, but these capabilities are not instantiated in the present study.

6. Discussion

The empirical findings indicate that the improvements observed across diagnostic variants arise not merely from the addition of model components, but from the structured interaction between numerical predictions and engineering constraints. The progression from neural-only detection to fusion and semantic validation reflects a layered interpretive process: data-driven perception identifies candidate deviations, symbolic thresholds anchor high-severity conditions, and semantic validation resolves inconsistencies between heterogeneous cues. This interaction restructures the effective decision boundary by enforcing compatibility with established engineering rules rather than relying solely on probabilistic confidence.

The empirical results can also be interpreted in relation to the cognitive capabilities defined in Section 3.2. The stability of anomaly detection under noisy and imbalanced conditions reflects perception, where raw measurements are transformed into structured representations that remain robust to local variation. The reduction in unsupported alerts reflects attention, as the interaction between neural predictions and rule-based constraints prioritises operationally relevant deviations.

Reasoning is supported through the combination of neural inference, engineering thresholds and ontology-based validation, which ensures that diagnostic outputs remain consistent with domain constraints. Problem-solving is reflected in the generation of actionable diagnostic outcomes, where fault states and severity levels are translated into decisions aligned with maintenance practice. Learning is observed in a bounded form through semantic refinement, where expert-informed adjustments to constraints improve consistency across iterations.

These results provide a partial empirical grounding of the cognitive framework. The evaluation focuses on perception and reasoning within the diagnostic module, while attention and learning are demonstrated in a limited form. Broader cognitive functions, particularly those involving cross-stage coordination and predictive reasoning, remain to be validated.

The behaviour observed during iterative refinement further clarifies the operational logic of the neurosymbolic core. Early refinement cycles

stabilise predictions by suppressing unsupported activations, indicating that ontology-guided propagation acts as a constraint-propagation mechanism. Subsequent fluctuations are modest and non-monotonic, reflecting statistical properties of the dataset rather than architectural instability. Similar non-monotonic behaviour has been documented in imbalanced and noise-sensitive classification settings, where minority classes are sparsely distributed and boundary decisions are highly sensitive to local perturbations [70,71]. The refinement mechanism therefore operates within representational limits imposed by the available sensing modality.

These representational limits are central to interpreting the per-fault results. Each metre-level record is described by a small set of scalar geometry measurements within a fixed temporal window, yielding a low-dimensional and locally correlated feature space. Under such conditions, anomaly discrimination relies on subtle amplitude variations near predefined engineering thresholds. Threshold-based categorisation introduces structural discontinuities in the decision space: minor signal fluctuations around severity boundaries may alter symbolic rule-strength classification even when neural confidence remains similar. This interaction constrains achievable per-class precision, particularly for oscillatory fault types such as twist and alignment, whose geometric signatures are inherently noisy. The observed performance ceiling therefore reflects not only modelling choices but also the informational constraints of the underlying data representation.

Class imbalance further shapes model behaviour. With each fault category constituting a relatively small fraction of the overall dataset, decision surfaces are skewed towards the majority class, increasing variance in minority precision estimates. Aggregation into a binary intervention task mitigates this effect by enlarging the effective positive sample pool, which explains why binary detection remains comparatively stable while per-fault discrimination exhibits higher false-alarm rates. This distinction is consistent with practical railway maintenance workflows, where automated systems are primarily deployed to flag segments requiring attention, while fine-grained fault categorisation is subsequently refined through deterministic engineering standards and expert judgement [72,73].

From a deployment perspective, the reduction of unsupported alerts has implications beyond statistical performance. False positives increase inspection workload, reduce confidence in automated diagnostics, and complicate resource allocation decisions in maintenance planning [74]. By constraining neural activations through severity-aware symbolic validation, the architecture shifts towards conservative and interpretable decision behaviour that remains compatible with established engineering logic. The structured outputs produced by the Diagnostic Agent: fault presence, severity and rule-based justification, can be contextualised through AR field interfaces and supervisory dashboards, enabling inspection, validation and action within human-centred maintenance processes.

Several limitations should be acknowledged. First, the empirical evaluation focuses on the diagnostic component instantiated within the MA-CDT architecture, while the remaining agents are defined at the architectural level. The current implementation therefore serves as a partial operationalisation rather than a full system deployment. Second, the evaluation is conducted on a single operational track-geometry inspection dataset. Although the dataset reflects realistic inspection conditions, validation across additional track segments, inspection periods, or external datasets would be necessary to further examine the robustness of the approach.

Third, the diagnostic inputs are limited to standard track-geometry measurements, which provide only a coarse representation of track condition. Richer sensing modalities, such as vibration spectra, high-resolution imaging, or uncertainty-aware measurements, could potentially provide additional diagnostic cues and may influence model behaviour. Finally, the empirical comparison focuses on operational alarm streams and neural-only detectors in order to isolate the effect of integrating neural inference with engineering constraints and semantic

validation. Classical anomaly detection baselines (e.g., Isolation Forest or One-Class SVM) are therefore not included, as the objective of this study is to examine the architectural integration of learning and rule-based reasoning rather than standalone anomaly detection accuracy. Within these boundaries, the observed behaviour illustrates how engineering constraints and semantic validation can stabilise diagnostic decisions under noisy sensing conditions.

7. Conclusion

This study has presented a MA-CDT architecture for railway maintenance, designed to address the fragmentation of maintenance intelligence across HSA, PdM and FDD. The proposed architecture is designed to organise sensing, reasoning, knowledge representation and human interaction through distributed agents that communicate through shared semantic structures.

To empirically ground the proposed architecture, the study instantiated and evaluated its diagnostic component using operational track-geometry inspection data from Network Rail. The results show that integrating neural inference, engineering thresholds and ontology-guided semantic validation within a single diagnostic structure improves the stability of intervention-triggering detections under noisy and imbalanced inspection data. Compared with both operational alarm streams and neural-only configurations, the integrated design reduces unsupported alerts while preserving threshold-defined decision coverage.

Beyond the diagnostic evaluation itself, the study specifies how the proposed architectural design can be translated into implementable maintenance mechanisms. The proposed MA-CDT architecture defines how distributed agents, semantic knowledge representations and human-in-the-loop validation are intended to be coordinated within a railway maintenance environment. The implemented diagnostic module therefore provides an operational implementation of the perception and reasoning functions defined within the proposed architecture, but does not constitute a full implementation or validation of the complete multi-agent coordination.

Within the scope of the present study, the findings indicate that

combining statistical inference with domain-constrained reasoning can improve decision robustness in threshold-driven maintenance systems. Future work will extend the architecture by implementing and evaluating additional cognitive components, including predictive coordination and cross-stage knowledge integration, and by experimentally validating the information flow and coordination between multiple interaction agents defined in the proposed architecture. Richer sensing modalities and operational datasets will also be incorporated to support broader deployment of CDT systems in railway infrastructure management.

CRedit authorship contribution statement

Wanting Mao: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Sara Scheffer:** Writing – review & editing, Supervision. **John Ahmet Erkoyuncu:** Writing – review & editing, Supervision, Resources, Project administration. **Xi Vincent Wang:** Writing – review & editing. **Lihui Wang:** Writing – review & editing. **Arnab Majumdar:** Writing – review & editing, Supervision. **Bernadin Namoano:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Investigation, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The authors would like to thank Network Rail for providing access to the railway track monitoring data used in this research. This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Appendix

Appendix A. Mapping between MA-CDT components and ArchiMate concepts

Original concept	ArchiMate element	Layer
Maintenance Engineer / Supervisor	Business Actor	Business
HSA · FDD · PdM	Business Process	Business
HMD AR / Mobile AR / VR	Application Interface	Application
Diagnostic / Predictive / Knowledge-Base / Coordination Agents	Application Component	Application
MCP · MQTT · REST · WebSocket	Application Service	Application
Time-Series DB · Maintenance History · 4D KG · Fault Hypothesis	Data Object	Application
Sensors · Track-Monitoring · Signalling Systems	Node / Device	Technology
Track Segment · Rolling Stock · Signalling Asset	Equipment	Physical
Cognitive Continuity / Traceable Decisions	Capability / Goal	Motivation
CNN / LSTM / Anomaly Detection	Application Component (neural sub-type)	Application
Ontology Loader · SPARQL · Rule Classifier	Application Component (symbolic sub-type)	Application
Fusion Engine · Confidence · Explanation · Symbolic Validator	Application Service / Function (hybrid)	Application

Data availability

The data that has been used is confidential.

References

- [1] A. Saleh, R. Remenyte-Priscott, D. Prescott, M. Chlachfó, Intelligent and adaptive asset management model for railway sections using the iPN method, *Reliab. Eng. Syst. Saf.* 241 (2024) 109687, <https://doi.org/10.1016/j.res.2023.109687>.
- [2] M. Shimizu, S. Perinpanayagam, B. Namoano, A real-time fault detection framework based on unsupervised deep learning for prognostics and health management of railway assets, *IEEE Access* 10 (2022) 96442–96458, <https://doi.org/10.1109/ACCESS.2022.3205352>.

- [3] V.J. Hodge, S.O. Keefe, M. Weeks, A. Moulds, Wireless sensor networks for condition monitoring in the railway industry: a survey, *IEEE Trans. Intell. Transp. Syst.* 16 (3) (2015) 1088–1106, <https://doi.org/10.1109/TITS.2014.2366512>.
- [4] R.D. D'Amico, J.A. Erkoynucu, S. Addepalli, S. Penver, Cognitive digital twin: an approach to improve the maintenance management, *CIRP J. Manuf. Sci. Technol.* 38 (2022) 613–630, <https://doi.org/10.1016/j.cirpj.2022.06.004>.
- [5] W. Mao, S. Scheffer, A. Majumdar, Augmented reality-enabled knowledge management in industrial maintenance: the DILEAF framework, *Comput. Ind. Eng.* 208 (2025) 111363, <https://doi.org/10.1016/j.cie.2025.111363>.
- [6] R. Dirnfeld, L. De Donato, A. Somma, M.S. Azari, S. Marrone, F. Flammini, V. Vittorini, Integrating AI and DTs: challenges and opportunities in railway maintenance application and beyond, *Simulation* 100 (9) (2024) 903–917, <https://doi.org/10.1177/00375497241229756>.
- [7] Q. Zhou, D. Zhou, C. Dai, J. Chen, Z. Guo, Knowledge-driven innovation in industrial maintenance: a neural-enhanced model-based definition framework for lifecycle maintenance process information propagation, *J. Manuf. Syst.* 82 (2025) 976–999, <https://doi.org/10.1016/j.jmsy.2025.08.001>.
- [8] Dirnfeld, R., De Donato, L., Flammini, F., Azari, M.S., & Vittorini, V. (2022). Railway digital twins and artificial intelligence: challenges and design guidelines. *Dependable Computing – EDC 2022 Workshops*, Cham.
- [9] S. Zappa, C. Franciosi, A. Polenghi, A. Voisin, Cognitive digital twin for industrial maintenance: operational framework for fault detection and diagnosis, *J. Ind. Inf. Integr.* 48 (2025) 100974, <https://doi.org/10.1016/j.jii.2025.100974>.
- [10] M.A.A. Faruque, D. Muthirayan, S.Y. Yu, P.P. Khargonekar, Cognitive digital twin for manufacturing systems, in: 2021 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2021.
- [11] X. Zheng, J. Lu, D. Kiritis, The emergence of cognitive digital twin: vision, challenges and opportunities, *Int. J. Prod. Res.* 60 (24) (2022) 7610–7632, <https://doi.org/10.1080/00207543.2021.2014591>.
- [12] Y. Liu, T. Ji, X. Guo, X. Xu, J. Polzer, Cognitive digital twin frameworks in manufacturing—a critical survey, evaluation criteria, and future directions, *J. Manuf. Syst.* 83 (2025) 597–611, <https://doi.org/10.1016/j.jmsy.2025.10.004>.
- [13] L. Network Rail, *Network Rail Limited's Annual Report and Accounts 2025* (CP 1364), N. Rail, 2025. www.gov.uk/official-documents.
- [14] S. García-Méndez, F. de Arriba-Pérez, F. Leal, B. Veloso, B. Malheiro, J. C. Burguillo-Rial, An explainable machine learning framework for railway predictive maintenance using data streams from the metro operator of Portugal, *Sci. Rep.* 15 (1) (2025) 27495, <https://doi.org/10.1038/s41598-025-08084-1>.
- [15] S. Sharma, Y. Cui, Q. He, R. Mohammadi, Z. Li, Data-driven optimization of railway maintenance for track geometry, *Transp. Res. C* 90 (2018) 34–58, <https://doi.org/10.1016/j.trc.2018.02.019>.
- [16] T.D. Bruin, K. Verbert, R. Babuška, Railway track circuit fault diagnosis using recurrent neural networks, *IEEE Trans. Neural Netw. Learn. Syst.* 28 (3) (2017) 523–533, <https://doi.org/10.1109/TNNLS.2016.2551940>.
- [17] J. Yin, W. Zhao, Fault diagnosis network design for vehicle on-board equipments of high-speed railway: a deep learning approach, *Eng. Appl. Artif. Intell.* 56 (2016) 250–259, <https://doi.org/10.1016/j.engappai.2016.10.002>.
- [18] J. Wang, F. Yang, T. Chen, S.L. Shah, An overview of industrial alarm systems: main causes for alarm overloading, research status, and open problems, *IEEE Trans. Autom. Sci. Eng.* 13 (2) (2016) 1045–1061, <https://doi.org/10.1109/TASE.2015.2464234>.
- [19] A.M. Peco Chacón, F.P. García Márquez, False Alarms management by data science, in: F.P. García Márquez, B. Lev (Eds.), *Data Science and Digital Business*, Springer International Publishing, 2019, pp. 301–316, https://doi.org/10.1007/978-3-319-95651-0_15.
- [20] P. Goel, A. Datta, M.S. Mannan, Industrial alarm systems: challenges and opportunities, *J. Loss Prev. Process Ind.* 50 (2017) 23–36, <https://doi.org/10.1016/j.jlp.2017.09.001>.
- [21] J.U. Ko, K. Na, J.-S. Oh, J. Kim, B.D. Youn, A new auto-encoder-based dynamic threshold to reduce false alarm rate for anomaly detection of steam turbines, *Expert Syst. Appl.* 189 (2022) 116094, <https://doi.org/10.1016/j.eswa.2021.116094>.
- [22] D. López, I. Aguilera-Martos, M. García-Barzana, F. Herrera, D. García-Gil, J. Luengo, Fusing anomaly detection with false positive mitigation methodology for predictive maintenance under multivariate time series, *Inf. Fusion* 100 (2023) 101957, <https://doi.org/10.1016/j.inffus.2023.101957>.
- [23] P. Gyasi, J. Wang, Design of serial alarm systems based on deadbands and delay timers for removing false alarms, *Process Saf. Environ. Prot.* 162 (2022) 1033–1041, <https://doi.org/10.1016/j.psep.2022.05.006>.
- [24] H. Hashim, E. Clifford, P.C. Ryan, False alarm moderation for performance monitoring in industrial water distribution systems, *Adv. Eng. Inform.* 52 (2022) 101592, <https://doi.org/10.1016/j.aei.2022.101592>.
- [25] H.H. Lin, C.H. Mao, H.M. Lee, False alarm reduction by weighted score-based rule adaptation through expert feedback, in: 2009 2nd International Conference on Computer Science and its Applications, 2009.
- [26] T. Jia, Y. Li, Y. Yang, G. Huang, Z. Wu, Augmenting log-based anomaly detection models to reduce false anomalies with human feedback, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022.
- [27] C. Chai, L. Cao, G. Li, J. Li, Y. Luo, S. Madden, Human-in-the-loop outlier detection, in: Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, Portland, OR, USA, 2020, <https://doi.org/10.1145/3318464.3389772>.
- [28] M. Liu, L. Jiang, J. Liu, X. Wang, J. Zhu, S. Liu, Improving learning-from-crowds through expert validation, *IJCAI* (2017).
- [29] G. Nigmatulin, K. Pavlov, Anomaly detection-aided active learning, in: *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*, 2023.
- [30] Deng, Z., Xuan, X., Ma, K.-L., & Kong, Z. (2024). A reliable framework for human-in-the-loop anomaly detection in time series. *arXiv preprint arXiv:2405.03234*.
- [31] Islam, M.R., Das, S., Doppa, J.R., & Natarajan, S. (2018). Glad: globalized anomaly detection via human-in-the-loop learning. *arXiv preprint arXiv:1810.01403*.
- [32] I. Durazo-Cardenas, B. Namoano, A. Starr, R.D. Sala, J. Lai, False alarm reduction in railway track quality inspections using machine learning, in: *PHM Society European Conference*, 2024.
- [33] P. Klein, L. Malburg, R. Bergmann, Combining informed data-driven anomaly detection with knowledge graphs for root cause analysis in predictive maintenance, *Eng. Appl. Artif. Intell.* 145 (2025) 110152, <https://doi.org/10.1016/j.engappai.2025.110152>.
- [34] Smits, G., Lesot, M.-J., Yepmo Tchaghe, V., & Pivert, O. (2022). PANDA: human-in-the-loop anomaly detection and explanation. *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Cham.
- [35] W. Xu, H. Yang, Z. Ji, M. Ba, Cognitive digital twin-enabled multi-robot collaborative manufacturing: framework and approaches, *Comput. Ind. Eng.* 194 (2024) 110418.
- [36] H.A. Simon, Information-processing models of cognition, *J. Am. Soc. Inf. Sci.* 32 (5) (1981) 364–377, <https://doi.org/10.1002/asi.4630320517>.
- [37] J.M. Rožanec, J. Lu, J. Rupnik, M. Škrjanc, D. Mladenčić, B. Fortuna, X. Zheng, D. Kiritis, Actionable cognitive twins for decision making in manufacturing, *Int. J. Prod. Res.* 60 (2) (2022) 452–478, <https://doi.org/10.1080/00207543.2021.2002967>.
- [38] L. Jinzhi, Y. Zhaorui, Z. Xiaochen, W. Jian, K. Dimitris, Exploring the concept of cognitive digital twin from model-based systems engineering perspective, *Int. J. Adv. Manuf. Technol.* 121 (9) (2022) 5835–5854, <https://doi.org/10.1007/s00170-022-09610-5>.
- [39] J. Du, Q. Zhu, Y. Shi, Q. Wang, Y. Lin, D. Zhao, Cognition digital twins for personalized information systems of smart cities: proof of concept, *J. Manag. Eng.* 36 (2) (2020) 04019052.
- [40] V. Belle, M. Fisher, A. Russo, E. Komendantskaya, A. Nottle, Neuro-symbolic AI + agent systems: a first reflection on trends, opportunities and challenges. *Autonomous Agents and Multiagent Systems, Best and Visionary Papers*, Cham, 2024.
- [41] F. Chen, W. Ren, On the control of multi-agent systems: a survey, *Found. Trends® Syst. Control* 6 (4) (2019) 339–499, <https://doi.org/10.1561/26000000019>.
- [42] S. Wang, J. Wan, D. Zhang, D. Li, C. Zhang, Towards smart factory for industry 4.0: a self-organized multi-agent system with big data based feedback and coordination, *Comput. Netw.* 101 (2016) 158–168, <https://doi.org/10.1016/j.comnet.2015.12.017>.
- [43] M. Wooldridge, *An Introduction to Multiagent Systems*, John Wiley & sons, 2009.
- [44] K.M. Hanga, Y. Kovalchuk, Machine learning and multi-agent systems in oil and gas industry applications: a survey, *Comput. Sci. Rev.* 34 (2019) 100191, <https://doi.org/10.1016/j.cosrev.2019.08.002>.
- [45] D. Hedjazi, F. Layachi, D.E. Boubiche, A multi-agent system for distributed maintenance scheduling, *Comput. Electr. Eng.* 77 (2019) 1–11, <https://doi.org/10.1016/j.compeleceng.2019.04.016>.
- [46] A. Salvador Palau, M.H. Dhada, K. Bakliwal, A.K. Parlikad, An industrial multi agent system for real-time distributed collaborative prognostics, *Eng. Appl. Artif. Intell.* 85 (2019) 590–606, <https://doi.org/10.1016/j.engappai.2019.07.013>.
- [47] M.G. Juarez, V.J. Botti, A.S. Giret, Digital twins: review and challenges, *J. Comput. Inf. Sci. Eng.* 21 (3) (2021), <https://doi.org/10.1115/1.4050244>.
- [48] L. De Simone, E. Caputo, M. Cinque, A. Galli, V. Moscatò, S. Russo, G. Cesaro, V. Criscuolo, G. Giannini, LSTM-based failure prediction for railway rolling stock equipment, *Expert Syst. Appl.* 222 (2023) 119767, <https://doi.org/10.1016/j.eswa.2023.119767>.
- [49] Tran, S., Mota, E., & Garcez, A.D.A. (2025). Reasoning in neurosymbolic AI. *arXiv preprint arXiv:2505.20313*.
- [50] B.P. Bhuyan, A. Ramdane-Cherif, R. Tomar, T.P. Singh, Neuro-symbolic artificial intelligence: a survey, *Neural Comput. Appl.* 36 (21) (2024) 12809–12844, <https://doi.org/10.1007/s00521-024-09960-z>.
- [51] P. Ganguly, I. Mukherjee, Bridging the gap: the rise of neurosymbolic artificial intelligence in advanced computing, *IT Prof.* 27 (2) (2025) 48–53, <https://doi.org/10.1109/MITP.2025.3532388>.
- [52] E.E. Kosasih, E. Papadakis, G. Baryannis, A. Brintrup, A review of explainable artificial intelligence in supply chain management using neurosymbolic approaches, *Int. J. Prod. Res.* 62 (4) (2024) 1510–1540, <https://doi.org/10.1080/00207543.2023.2281663>.
- [53] Shindo, H., Dhami, D.S., & Kersting, K. (2021). Neuro-symbolic forward reasoning. *arXiv preprint arXiv:2110.09383*.
- [54] L.N. DeLong, R.F. Mir, J.D. Fleuriot, Neurosymbolic AI for reasoning over knowledge graphs: a survey, *IEEE Trans. Neural Netw. Learn. Syst.* 36 (5) (2025) 7822–7842, <https://doi.org/10.1109/TNNLS.2024.3420218>.
- [55] S. Scheffer, A. Martinetti, R. Damgrave, S. Thiede, L. van Dongen, How to make augmented reality a tool for railway maintenance operations: operator 4.0 perspective, *Appl. Sci.* 11 (6) (2021) 2656, <https://www.mdpi.com/2076-3417/11/6/2656>.
- [56] Y.-C. Liang, S.-C. Shiu, C.-W. Liu, Enhancing yield and process efficiency through dual internet of things and augmented reality for AI-driven human-machine interaction in centering mass production, *J. Manuf. Syst.* 82 (2025) 26–41, <https://doi.org/10.1016/j.jmsy.2025.06.003>.
- [57] X. Zhang, S. Meng, B. Wang, L. Zheng, R. Zhang, X. Li, Integrated assembly, measurement, and adjustment method of reconfigurable flexible fixture for aircraft panels based on augmented reality and human-computer interaction, *J. Manuf. Syst.* 79 (2025) 117–133, <https://doi.org/10.1016/j.jmsy.2025.01.003>.

- [58] A. Cimino, F. Longo, G. Mirabelli, V. Solina, S. Verteramo, An ontology-based, general-purpose and Industry 4.0-ready architecture for supporting the smart operator (part II – virtual reality case), *J. Manuf. Syst.* 73 (2024) 52–64, <https://doi.org/10.1016/j.jmsy.2024.01.001>.
- [59] Z. Guo, D. Zhou, Q. Zhou, X. Zhang, J. Geng, S. Zeng, C. Lv, A. Hao, Applications of virtual reality in maintenance during the industrial product lifecycle: a systematic review, *J. Manuf. Syst.* 56 (2020) 525–538, <https://doi.org/10.1016/j.jmsy.2020.07.007>.
- [60] K. Peffers, T. Tuunanen, M.A. Rothenberger, S. Chatterjee, A design science research methodology for information systems research, *J. Manag. Inf. Syst.* 24 (3) (2007) 45–77.
- [61] M.A. Al Faruque, D. Muthirayan, S.-Y. Yu, P.P. Khargonekar, Cognitive digital twin for manufacturing systems, in: 2021 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2021.
- [62] B.C. Grau, I. Horrocks, Y. Kazakov, U. Sattler, Modular reuse of ontologies: theory and practice, *J. Artif. Intell. Res.* 31 (2008) 273–318.
- [63] A. Polenghi, I. Roda, M. Macchi, A. Pozzetti, H. Panetto, Knowledge reuse for ontology modelling in maintenance and industrial asset management, *J. Ind. Inf. Integr.* 27 (2022) 100298, <https://doi.org/10.1016/j.jii.2021.100298>.
- [64] J.J. Montero Jiménez, R. Vingerhoeds, B. Grabot, S. Schwartz, An ontology model for maintenance strategy selection and assessment, *J. Intell. Manuf.* 34 (3) (2023) 1369–1387, <https://doi.org/10.1007/s10845-021-01855-3>.
- [65] D. Rajpathak, R. Chougule, A generic ontology development framework for data integration and decision support in a distributed environment, *Int. J. Comput. Integr. Manuf.* 24 (2) (2011) 154–170.
- [66] J.N. Otte, J. Beverley, A. Ruttenberg, BFO: basic formal ontology, *Appl. Ontol.* 17 (1) (2022) 17–43.
- [67] L. Daniele, F. den Hartog, J. Roes, Created in close interaction with the industry: the smart appliances reference (SAREF) ontology, *Int. Workshop Form. Ontol. Meet Ind.* (2015).
- [68] R. García-Castro, M. Lefrançois, M. Poveda-Villalón, L. Daniele, The ETSI SAREF ontology for smart applications: a long path of development and evolution, *Energy Smart Appl.: Appl. Methodol. Chall.* (2023) 183–215.
- [69] M.H. Karray, F. Ameri, M. Hodkiewicz, T. Louge, ROMAIN: towards a BFO compliant reference ontology for industrial maintenance, *Appl. Ontol.* 14 (2) (2019) 155–177.
- [70] K. Ghosh, C. Bellinger, R. Corizzo, P. Branco, B. Krawczyk, N. Japkowicz, The class imbalance problem in deep learning, *Mach. Learn.* 113 (7) (2024) 4845–4901, <https://doi.org/10.1007/s10994-022-06268-8>.
- [71] C. Huang, C.C. Loy, X. Tang, Discriminative sparse neighbor approximation for imbalanced learning, *IEEE Trans. Neural Netw. Learn. Syst.* 29 (5) (2017) 1503–1513.
- [72] M. Noruzoliaee, B. Zou, Y. Liu, Roads in transition: integrated modeling of a manufacturer-traveler-infrastructure system in a mixed autonomous/human driving environment, *Transp. Res. C* 90 (2018) 307–333, <https://doi.org/10.1016/j.trc.2018.03.014>.
- [73] J. Vyas, Bhumika, D. Das, S. Chaudhury, Federated learning based driver recommendation for next generation transportation system, *Expert Syst. Appl.* 225 (2023) 119951, <https://doi.org/10.1016/j.eswa.2023.119951>.
- [74] G. Pang, C. Shen, L. Cao, A.V.D. Hengel, Deep learning for anomaly detection: a review, *ACM Comput. Surv.* 54 (2) (2021) 38, <https://doi.org/10.1145/3439950>.



DSPACE

<https://dspace.org/>

A multi-agent cognitive digital twin architecture with neurosymbolic diagnostic integration for railway maintenance

Mao, Wanting; Scheffer, Sara; Erkoyuncu, John Ahmet; Wang, Xi Vincent; Wang, Lihui ...

2026-09

Elsevier

<https://doi.org/10.1016/j.jii.2026.101165>

Mao W, Scheffer S, Erkoyuncu JA, et al., (2026) A multi-agent cognitive digital twin architecture with neurosymbolic diagnostic integration for railway maintenance. Journal of Industrial Information Integration, Volume 53, September 2026, Article number 101165