DOI: https://doi.org/10.15276/aait.09.2026.22
UDC 004.82:004.6

The taxonomy of knowledge graph validation approaches

Ivan V. Osiichuk[1)]
ORCID: https://orcid.org/0009-0003-6844-2374; ivan_osiichuk0108@tntu.edu.ua. Scopus Author ID: 59296923000
Nataliya V. Zagorodna[1)]
ORCID: https://orcid.org/0000-0002-1808-835X; zagorodna.n@gmail.com. Scopus Author ID: 57189380553
Yuriy L. Skorenkyy[1)]
ORCID: https://orcid.org/0000-0002-4809-9025; skorenkyy@tntu.edu.ua. Scopus Author ID: 6507755672
Mikolaj P. Karpinski[2)]
ORCID: https://orcid.org/0000-0002-8846-332X; mikolaj.karpinski@uken.krakow.pl. Scopus Author ID: 57226717849
Aizhan K. Tokkuliyeva[3)]
ORCID: https://orcid.org/0000-0002-5019-2413; tokkuliyeva_ak_1@enu.kz. Scopus Author ID: 57445077400
[1)] Ternopil Ivan Puluj National Technical University, 56, Ruska Str. Ternopil, 46025, Ukraine
[2)] University of the National Education Commission, 2, Podchorazych Str. Krakow, 30-084, Poland
[3)] L. N. Gumilyov Eurasian National University. Astana, 010008, Kazakhstan

ABSTRACT

Relevance. Knowledge graphs encoded in Resource Description Framework have become a standard foundation for data integration, semantic search, and automated reasoning across domains ranging from biomedicine to enterprise knowledge management. As these graphs grow in scale and are maintained by distributed teams, it becomes increasingly important to ensure that their data follows the defined structural and semantic rules. **Aim.** This survey presents a systematic review of knowledge graph validation approaches published between 2011 and 2026, covering three principal families of methods and proposes a taxonomy that classifies existing papers within following three families: (i) ontology-based validation via Web Ontology Language reasoning, (ii) constraint-based validation using Shapes Constraint Language, Shape Expressions Language, and SPARQL Inferencing Notation (SPIN) language, and (iii) statistical and machine-learning-based error detection using knowledge graph embeddings and contrastive learning. The following **objectives** were conducted: to review the literature sources, to carry out comparative analysis across listed above methods, to suggest taxonomy of graph validation approaches, to discover gaps in research field. **Scientific novelty:** the survey introduces a single three-family taxonomy that characterizes each family by its underlying assumptions (Open World Assumption vs. Closed World Assumption), expressiveness, and computational cost, and it formally defines constraint drift, the progressive misalignment of Shapes Constraint Language shapes with their underlying ontology. **Practical significance.** The survey provides architectural design choices that resolve the tension between the Open World Assumption (OWA) of Web Ontology Language reasoning and the Closed World Assumption (CWA) of Shapes Constraint Language shapes across all three families, serving as a reference for practitioners and a roadmap for researchers. **Results:** a taxonomy of validation approaches is proposed and a comparative analysis of the three families is conducted. Constraint-based validation and Shapes Constraint Language in particular, is identified as the practical standard for Resource Description Framework data quality enforcement, with ontology-based reasoning complementing it for logical consistency and machine-learning methods extending it to distributional anomaly detection. Four open research gaps are identified: provenance-aware validation, incremental validation, validation under ontology evolution (including constraint drift, formally defined in subsection Gap 3), and hybrid constraint-plus-machine-learning approaches. **Conclusions**: no single family satisfies all validation requirements; ontology-based, constraint-based, and statistical/ML-based methods are complementary, and the OWA/CWA inconsistency remains a central design consideration across all three. Closing the identified gaps, in particular constraint drift and hybrid approaches, defines a roadmap for future work requiring coordinated effort across the standardisation, knowledge-graph-engineering, and machine-learning-on-graphs communities.

Keywords: Knowledge graph; data validation; ontology reasoning; constraint languages; machine-learning-based error detection; Resource Description Framework database

***For citation*: Osiichuk I., Zagorodna N., Skorenkyy Yu., Karpinski M., Tokkuliyeva A. "The taxonomy of knowledge graph validation approaches". *Applied Aspects of Information Technology*. 2026; Vol.9 No.3: 326–339. DOI: https://doi.org/10.15276/aait.09.2026.22**

INTRODUCTION

Knowledge graphs represent knowledge as directed labelled graphs, typically serialised as Resource Description Framework (RDF) triples of the form (subject, predicate, and object).

Large-scale knowledge graphs such as Wikidata, DBpedia, and enterprise-internal graphs now contain billions of triples and are consumed by downstream applications including question answering and recommendation systems. The utility of these applications depends directly on the quality of the underlying graph: inaccurate triples propagate errors into query results, incomplete data produces

© Osiichuk I., Zagorodna N., Skorenkyy Yu., Karpinski M., Tokkuliyeva A., 2026

This is an open access article under the CC BY license (https://creativecommons.org/licenses/by/4.0/deed.uk)

misleading silence, and structural violations break automated pipelines [1].

Validation, as defined in this survey, is the task of verifying that a knowledge graph conforms to a declared set of structural and semantic constraints.

This definition is deliberately narrow. We distinguish validation from knowledge graph completion, which adds missing facts, and from knowledge graph enrichment, which augments a graph with external information [2], [3]. Validation checks whether the graph satisfy its specified rules.

A central problem in RDF validation arises from the Open World Assumption (OWA) which is typical for OWL reasoning.

Under OWA, the absence of a triple does not imply its negation; it may simply be unknown. Constraint languages such as Shapes Con Close straint Language Shapes Constraint Language (SHACL), operate under the d World Assumption (CWA), where absent data is treated as a violation if the constraint demands it. This OWA-CWA contradiction explains why reasoning alone is insufficient for practical validation and why dedicated constraint languages were developed [4].

The paper is structured as follows: initially provides background on RDF and the formal definition of knowledge graph validation. Then paper presents the taxonomy of validation approaches, covering ontology-based, constraint-based, and statistical/ML-based methods. Next section discusses open research gaps. The last section concludes the paper.

RELATED WORKS

This review covers peer-reviewed publications and standardisation documents on RDF knowledge graph validation published between 2011 and 2026. The start date coincides with the introduction of SPIN, the earliest widely adopted mechanism for attaching SPARQL-based constraints to RDF classes. The end date captures recent work, including SHACL validation under graph updates [5], ML-based error detection at AAAI and WSDM 2023-2024, and emerging neuro-symbolic approaches (2024-2026) [6], [7].

We searched ACM Digital Library, IEEE Xplore, Semantic Scholar, and arXiv using queries combining the terms “SHACL”, “knowledge graph validation”, “RDF data quality”, “ontology consistency was checking”, and “knowledge graph error detection”. We included papers that propose, evaluate, or survey validation methods for RDF graphs. We excluded papers on property graph validation (e.g., Neo4j Cypher constraints), relational database integrity constraints, and general data quality frameworks not specific to RDF. Some literature sources, such as W3C specifications, product documentation, were deliberately included to this study because there is a lack of comprehensive academic research covering these topics.

More than 200 scientific papers were processed, but after critical analysis of their abstracts, and relevance screening, 29 were selected and cited throughout the survey. Table 1 lists the inclusion and exclusion criteria applied during screening.

Table 1. **Inclusion and exclusion criteria for source selection**

Inclusion criteria	Exclusion criteria
Peer-reviewed papers and standardisation documents on RDF knowledge graph validation. Published between 2011 and 2026. Works that propose, evaluate, or survey validation methods for RDF graphs. W3C specifications and product documentation where academic coverage is absent.	Property graph validation, for example Neo4j Cypher constraints. Relational database integrity constraints. Generic data quality frameworks not specific to RDF. Works that propose no evaluable validation method. Sources superseded by later standard specifications.

Source: **compiled by the authors**

The selection followed a four-stage screening process. Database and index searches across the sources produced an initial set of candidate records. Duplicate records were removed, and the remaining items were screened by title and abstract against the scope of the review. Records concerned with property graph validation, relational database integrity constraints, or generic data quality frameworks not specific to RDF were excluded at this stage. The surviving records underwent full-text assessment, during which works that proposed no evaluable validation method, were not specific to RDF, or had been superseded by later standard specifications, were removed.

CONTRIBUTIONS

This survey makes three contributions. First, we propose a taxonomy that organizes knowledge graph validation approaches into three families: ontology-based, constraint-based, and statistical/ML-based, and characterises each by its assumptions, expressiveness, and scalability. While prior surveys such as Paulheim [2] and Xue and Zou [1] address individual aspects of knowledge graph quality in depth, neither organises existing methods around a unified three-family taxonomy that characterises

each family by its underlying assumptions (OWA vs. CWA), expressiveness, and computational cost.

Second, we identify four open research gaps that current methods do not fully address: provenance-aware validation, incremental validation, validation under ontology evolution, and hybrid constraint-plus-ML approaches.

Third, we introduce and formally define constraint drift as part of Gap 3: the progressive misalignment of SHACL shapes with their underlying ontology. As ontologies evolve, validation shapes may become outdated. Although several studies have addressed related sub-problems [5], [8], [9], this phenomenon has not previously been formally defined, or treated as a unified operational problem.

BACKGROUND

Resource Description Framework and the Triplestore Data Model

The Resource Description Framework (RDF) represents knowledge as a set of triples, each consisting of a subject, predicate, and object [4]. Subjects are Internationalized Resource Identifiers (IRIs) or blank nodes; predicates are IRIs, objects may be IRIs, blank nodes, or typed literals. An RDF dataset may additionally organise triples into named graphs, producing quads of the form (subject, predicate, object, graph) [10]. SPARQL is the standard query language for RDF datasets, supporting pattern matching, optional joins, aggregation, and federated queries.

Triplestores are database systems optimised for storing and querying RDF data. Prominent open-source implementations include Apache Jena (with its Fuseki SPARQL endpoint) and Eclipse RDF4J. Commercial systems include Stardog, Ontotext GraphDB, and Franz AllegroGraph. These systems differ in their resource demands, reasoning support, and, critically for this survey, their support for constraint validation languages [11].

Validation Definition

We define the knowledge graph validation problem as follows: given an RDF graph G and a constraint specification C, determine whether G satisfies C (written $G \vDash C$). The constraint specification C may be expressed in any of several formalisms: OWL axioms, SHACL shapes, ShEx schemas, or learned statistical models (e.g., embedding-based anomaly detectors, as discussed in following sections) [1], [4]. The output is either a boolean conformance verdict or a structured violation report identifying specific triples or nodes that fail to satisfy their constraints.

This definition deliberately excludes knowledge graph completion (predicting missing triples via link prediction), knowledge graph alignment (mapping entities across graphs), and knowledge graph enrichment (adding new facts from external sources). While these tasks may improve graph quality indirectly, they do not verify compliance with the graph's declared constraints. Prior surveys have addressed these adjacent tasks comprehensively [1], [2], [3].

The quality dimensions of knowledge graphs most directly served by validation are accuracy (the correctness of stated facts), consistency (the absence of contradictory assertions in the graph), and conformance (the alignment of the graph structure with the declared schema). Other quality dimensions such as completeness, timeliness, and provenance are relevant but require different techniques [1], [3].

TAXONOMY OF VALIDATION APPROACHES

We organise knowledge graph validation methods into three families based on their underlying mechanism: ontology-based approaches that use Web Ontology Language (OWL) reasoning, constraint-based approaches that use dedicated validation languages (SHACL, ShEx, SPIN), and statistical/ML-based approaches that rely on learned representations from graph structure. The first two families are well-established in the RDF validation literature [1], [4], while the third has emerged more recently from the knowledge graph embedding and error detection communities [2]. Our contribution is to unify all three under a single comparative taxonomy organised along three dimensions: world assumption (OWA vs. CWA), constraint expressiveness, and computational cost. Fig. 1 summarises the taxonomy. This taxonomy serves as the core organizing principle of this survey and the framework for the subsequent comparative analysis.

As shown in Fig. 1, the taxonomy is a hierarchy: the root node denotes the knowledge graph validation task, and the three branches correspond to the ontology-based, constraint-based, and statistical or ML-based families. Each branch lists the representative methods examined in its subsection. The ontology-based branch covers OWL consistency checking, combined OWL and SHACL workflows, and entailment-aware validation in Re-SHACL. The constraint-based branch follows the line from SPIN (2011-2017) to SHACL (a W3C

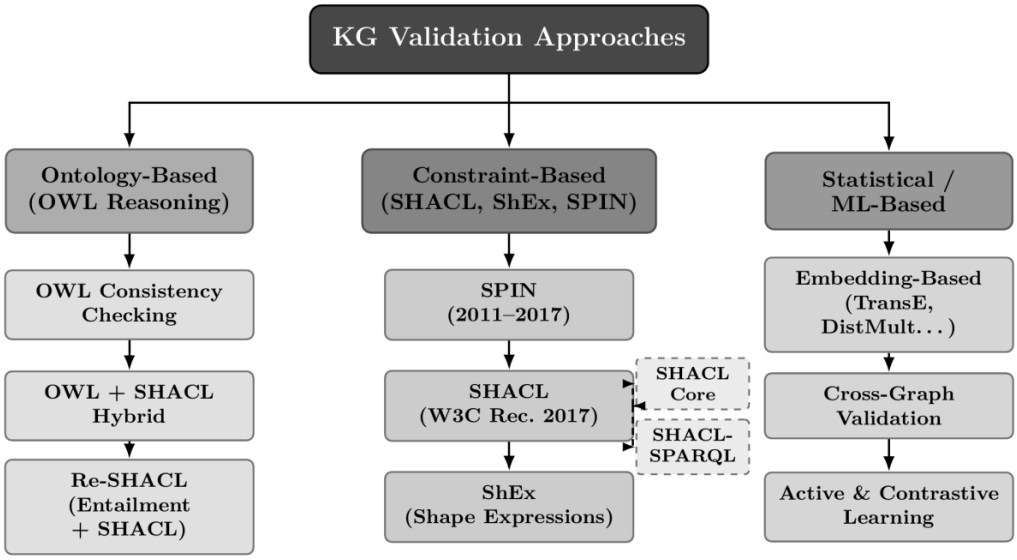

Fig 1. **Taxonomy of knowledge graph validation approaches**
***Source*: compiled by the authors**

Recommendation since 2017, with its Core and SPARQL profiles) and the parallel ShEx language. The statistical and ML-based branch groups embedding-based validation, cross-graph validation, and active and contrastive learning. The figure organises the methods by family; their comparison along world assumption, expressiveness, and computational cost is given in Table 2.

The study provides a detailed analysis of each knowledge graph validation approach, focusing on its expressiveness and the range of constraints that can be formally specified.

Ontology-Based Validation

The Web Ontology Language (OWL) extends RDF Schema with formal semantics for defining classes, properties, and logical axioms. Ontology-based validation uses OWL axioms to encode constraints implicitly. A reasoner checks whether the combination of an ontology (the TBox) and its instance data (the ABox) is consistent. Inconsistency indicates that the data violates the declared axioms. Constraint types expressible in OWL include disjoint class declarations, property domain and range restrictions, cardinality bounds on properties, and functional property constraints. To address varying practical requirements, OWL 2 defines several profiles that explicitly trade expressiveness for computational guarantees to target different use cases [12]:

• DL (Description Logic) provides the highest expressiveness but is often less scalable for massive datasets (decidable, 2NEXPTIME-complete);

• EL (Existential Language) is tailored for large ontologies requiring efficient, polynomial-time subsumption and classification;

• QL (Query Language) is designed for scalable conjunctive query answering;

• RL (Rule Language) is optimized for rule-based reasoning with tractable, polynomial-time data complexity.

Standard OWL reasoners such as HermiT, Pellet, and ELK perform consistency checking (determining whether an ontology has a model), classification (computing the subsumption hierarchy), and instance retrieval. In practice, the choice of OWL profile heavily dictates the computational performance and scalability of these tasks over large knowledge graphs [12].

The principal limitation of ontology-based validation is the Open World Assumption. Because OWL assumes that any unstated fact may be true elsewhere, the absence of a required property value does not produce a violation. This makes OWL unsuitable for enforcing minimum cardinality constraints or mandatory property patterns in a data quality context. Additionally, full OWL 2 DL reasoning is 2NEXPTIME-complete, making it impractical for knowledge graphs with millions of triples. The OWL 2 RL profile mitigates this by restricting the language to rules amenable to polynomial forward-chaining inference, at the cost of reduced expressiveness [4], [12].

Let consider the example to demonstrate why OWL reasoning does not replace constraint validation. A data graph asserts only that

```
:Alice is a :Person,
```

together with an ontology stating that every person has at least one birth date. An OWL reasoner reports no violation, because under the Open World Assumption, the missing birth date is treated as unknown rather than absent [4].

A SHACL shape expresses the same requirement as a closed-world constraint and reports a violation:

```
:PersonShape a sh:NodeShape ;
    sh:targetClass :Person ;
    sh:property [ sh:path :birthDate ;
                  sh:minCount 1 ] .
```

In the domain of data quality meta-modelling, the BIGOWL4DQ system defines an OWL 2 ontology for data quality dimensions and uses Semantic Web Rule Language (SWRL) rules to evaluate the correctness and completeness of the data quality rules via semantic reasoning [13]. This approach is notable because it uses the ontology not to validate data directly but to validate the quality rules themselves – checking whether a declared quality metric is compatible with the data domain to which it is applied.

Constraint-Based Validation

SPARQL Inferencing Notation

The SPARQL Inferencing Notation (SPIN) was introduced by TopQuadrant as a mechanism for attaching SPARQL-based constraints and rules to OWL classes. In SPIN, constraints are expressed as SPARQL ASK queries (where a result `true` indicates a violation) or CONSTRUCT queries (which produce structured violation triples). Within these queries, the reserved SPARQL variable "`?this`" (the leading question mark is SPARQL's notation for a variable) is pre-bound to the focus node, that is, the instance currently being validated, so one constraint definition applies to every instance of the class. This notation follows an object-oriented design pattern in which a method operates on the current object [4].

SPIN served as a primary constraint mechanism for RDF data from 2011 until the standardisation of SHACL in 2017. It is implemented in Eclipse RDF4J (via the SPIN Sail module) and Apache Jena (via TopQuadrant's SPIN API) and was used in TopBraid products. The W3C Data Shapes Working Group subsequently developed SHACL as an official standard, heavily influenced by SPIN's design. Following SHACL's release, the official SPIN documentation was updated to recommend migration from SPIN to SHACL for new projects.

Shapes Constraint Language

The Shapes Constraint Language (SHACL) is a W3C Recommendation from 2017 for describing and validating RDF graphs against a set of conditions called shapes. A SHACL validation process takes two inputs: a data graph (the RDF data to be validated) and a shapes graph (the set of SHACL shape declarations). The output is a validation report expressed as an RDF graph, containing one `sh:ValidationResult` resource for each detected violation.

SHACL defines two levels of expressiveness. SHACL Core provides built-in constraint components for datatype checking, cardinality bounds, value ranges, string patterns, class membership, and property path constraints. SHACL-SPARQL extends this with the ability to define arbitrary constraints as SPARQL queries, enabling validation logic that cannot be expressed in the core language. SHACL-SPARQL is strictly more expressive than SHACL Core but introduces non-monotone evaluation and potential performance costs.

SHACL supports class-based targets (all instances of a given class), node targets (specific IRIs), and implicit targeting via property shapes. Targets determine which nodes in the data graph are validated against which shapes. екгShapes may reference other shapes, creating recursive dependencies. Because the SHACL specification leaves the semantics of recursive shapes partially underspecified, formal treatments have proposed alternative approaches, such as well-founded and stable model semantics, to handle these dependencies [5], [14].

In practice, OWL and SHACL address validation complementarily: OWL provides ontological entailments and domain-range reasoning, while SHACL enforces data validity against concrete shapes under a closed-world assumption for data instances [15], [16]. These papers have explored combining OWL modelling with SHACL validation in industrial knowledge graph projects. Ruckhaus et al. (2025) [15] report practical experience using OWL for domain modelling and SHACL for validation constraints, identifying design patterns that avoid the most common pitfalls of mixed-formalism development. Their proposed approach uses OWL and SHACL together in a joint development workflow, where classes and properties can be represented in both languages and constraints may appear in one or both, depending on whether

ontology reasoning or data correctness requirements take precedence. Key practical findings include the absence of standard SHACL engine support for reusable shape components and the continued manual effort required despite automated OWL-to-SHACL translation tools.

The Re-SHACL system [16] bridges ontology-based and constraint-based validation by applying ontological entailment before SHACL shape evaluation. Instead of inefficient full-graph materialisation, Re-SHACL improves the data graph by materialising only shape-relevant inferences from the ontology and merging semantically equivalent entities, then validates the enriched graph against dynamically rewritten SHACL shapes.This approach captures violations that would be missed if entailment were ignored – for example, a shape requiring all instances of a class to have a specific property can be evaluated against instances that are only implicitly members of that class via subsumption reasoning.

Shape Expressions

Shape Expressions (ShEx) is an alternative RDF validation language developed by the W3C ShEx Community Group, which formed after splitting from the W3C Data Shapes Working Group responsible for SHACL [4]. Unlike SHACL, which is itself an RDF vocabulary, ShEx is designed as a compact, human-readable grammar inspired by schema languages such as RelaxNG and Turtle [4], although it also supports RDF syntax (ShExR) and a JSON-LD syntax (ShExJ). In ShEx, shapes are defined using a notation based on regular expressions over nodes, and shape maps are used to associate shapes with specific nodes or node patterns in the data graph [4].

ShEx and SHACL share substantial expressiveness for common validation tasks: both support datatype constraints, cardinality bounds, value sets, and recursive shape references. In most cases, constraints expressed in one language can be translated into the other, though the translation is not fully symmetric – ShEx shapes can generally be converted to equivalent SHACL shapes, while the reverse is restricted because SHACL includes features absent in ShEx, such as qualified cardinality constraints, property pair comparisons, and unique language constraints. ShEx requires shape definitions to be decoupled from their application to data nodes (via shape maps), while SHACL embeds target declarations directly in shapes. ShEx provides a more formal grammar-based foundation, while SHACL provides tighter integration with the RDF ecosystem since shapes are themselves RDF triples.

ShEx is used in practice by Wikidata, which uses ShEx and SPARQL for validation of its knowledge graph [4]. SHACL is implemented in a broader set of various enterprise triplestore products (Stardog, GraphDB, Jena, RDF4J) and holds W3C Recommendation status, giving it a stronger position in standards-based procurement contexts [11].

Statistical and ML-Based Approaches

These approaches use the same data representation models as knowledge graph completion and link prediction, which based on machine learning methods, but they address the different tasks. Completion and link prediction add missing triples and treat absence as incompleteness to be filled, whereas validation examines triples that are already present and flags those that appear suspicious [1], [2], [3]. The distinction matters for interpretation: a low plausibility score is a statistical signal, not a formal proof of error, and a rare but correct fact may receive a low score for the same reason a genuine error does. Statistical anomaly is therefore not equivalent to a formal data error. For this reason, the methods discussed below are best positioned as an auxiliary screening layer that surfaces candidate triples for review, complementing rather than replacing SHACL or OWL-based validation, which provide formal and explainable conformance guarantees [1], [6].

Embedding-Based Validation

Knowledge graph embedding models represent entities and relations as vectors in a continuous space, learning representations that capture the statistical regularities of the graph. The TransE model introduced the translation principle: for a valid triple (h, r, t), the embedding of h plus the embedding of r should approximate the embedding of t. Subsequent models, including DistMult, ComplEx, TransR, HolE, refine this principle with bilinear scoring, complex-valued embeddings, and relation-specific projection spaces. RDF2Vec takes a fundamentally different approach: rather than optimising a geometric scoring function over triples, it generates random walks over the RDF graph and applies word-embedding training (Word2Vec) to the resulting sequences, producing entity representations that capture graph neighbourhood structure.

These embedding models can be applied to validation through triple classification: given a triple, compute its plausibility score and compare it

to a learned threshold. Triples scoring below the threshold are flagged as potentially erroneous.

However, embedding-based methods carry significant limitations as a primary validation mechanism. They do not produce formal violation reports interpretable by downstream systems, making it hard to explain to domain experts why a specific triple was flagged [1], [6]. Furthermore, the quality of learned representations depends on the training graph: since the model learns from what is there rather than what should be there, systematic errors in the graph can be reinforced rather than detected [1]. Finally, setting appropriate anomaly-detection thresholds requires labelled examples of errors, which are expensive to acquire and may not exist for specialised domain graphs [17]. These limitations explain the complementary, rather than replacement, role of embedding methods relative to constraint-based approaches.

Cross-Graph Validation

The CrossVal framework [18] takes a different approach by using an external curated knowledge graph to validate facts in a target graph.CrossVal learns cross-graph representations that align entity and relation embeddings between the target and reference graphs. Rather than simply penalising triples that lack direct corroboration, the framework generates informative cross-KG negative samples based on mutually conflicting relations between the graphs, which are used alongside a confidence estimation component to assign validation scores. The framework achieves approximately 8% improvement in Recall of Ranking and over 20% improvement on Filtered Mean Rank compared to single-graph embedding methods. Cross-graph validation occupies a middle position within this taxonomy: it draws on explicit structural knowledge from external reference graphs, as constraint-based methods do, but uses that knowledge through statistical alignment rather than formal logical inference.

A fundamental limitation of simple matching approaches of cross-graph validation is its coverage: the approach applies only to the fraction of triples whose entities and relations are present in both the target and reference graphs and provides no signal for triples unique to the target graph. However, CrossVal was specifically designed with this limitation in mind and uses cross-KG negative sampling to borrow information from the external reference graph, allowing it to validate the majority of triples even when they are unique to the target graph [18].

Error Detection via Active and Contrastive Learning

Recent work addresses the annotation bottleneck in supervised error detection. Dong et al. (2023) [17] propose an active ensemble learning framework (KAEL) that ensembles outputs of multiple off-the-shelf base error detectors, including embedding-based models (TransE, ComplEx) and path-based models (CKRL and KGTtm), with an active learning strategy that selects the most informative triples for human annotation using a multi-armed bandit (MAB) formulation, reducing labelling effort while maintaining detection accuracy.

Liu et al. (2024) [19] tackle a specific weakness of prior embedding methods: the difficulty of distinguishing erroneous triples from semantically similar correct ones (the near-miss problem or semantically-similar noise as the authors refer to it). Their Contrastive Confidence Adaption (CCA) model integrates both textual descriptions and graph structural information using a contrastive learning objective. CCA explicitly trains the model to separate valid triples from plausible but incorrect alternatives. On the FB15K-237 and WN18RR benchmarks, CCA outperforms prior embedding-based detection methods across precision and recall measures.

Both methods address a shared constraint on ML-based validation: the cost of obtaining labeled error annotations at scale, which limits the applicability of fully supervised approaches to knowledge graphs where systematic manual curation is not feasible.

Strength and Limitations Relative to Constraint-Based Methods

ML-based methods detect statistical anomalies that constraint languages cannot express, for example, that a particular entity-relation-entity combination is implausible given the overall distribution of the graph, even though no formal constraint prohibits it.

However, these methods have significant limitations for validation. They cannot guarantee completeness: a low anomaly score does not prove correctness. They do not produce formal violation reports interpretable by downstream systems, so their decisions are difficult to explain to domain experts who need to understand why a triple was flagged [1], [6]. They require training data that may not exist for specialised domain graphs [17].

The complementary strengths of constraint-based and ML-based methods suggest that hybrid approaches can combine formal constraint enforcement with statistical anomaly detection [6]. Such systems use SHACL shapes for structural rules and embedding models for distributional plausibility. While several recent systems have begun to implement this integration [7], [20], producing standardised, unified reports that formally distinguish between logical structural violations and statistical anomalies remains an open problem [21], and is identified as an open gap in the next section.

Summarized Comparison of Families

Table 2 consolidates the three families along the comparative dimensions used throughout this section, giving a single reference for selecting an approach in practice.

OPEN RESEARCH GAPS

Gap 1: Provenance-Aware Validation

Current SHACL validation produces a conformance report listing violations but does not record the provenance of the validation process itself: which shapes were applied, by which engine, at what time, against which version of the data graph. For regulated domains (pharmaceutical, financial, governmental), audit trail requirements demand precisely this information. The PROV-O ontology provides a standard vocabulary for describing provenance. Sikos and Philp (2020) [10] and other recent papers on RDF-star-enabled provenance tracking show how triple-level change tracking can be implemented efficiently [22]. While custom pipelines have been built to persist SHACL validation metadata, no native SHACL engine automatically emits its execution trail as a standardised PROV-O-compliant provenance model.

Recent work has begun to close parts of this gap from two directions. On the theoretical side, Delva et al. (2023) [23] provide the first formal provenance semantics for SHACL, introducing the concept of a “neighbourhood” – a sufficient subgraph of the RDF data graph that witnesses why a node satisfies or violates a given shape. This directly addresses the absence of an explanation mechanism in the SHACL specification itself. On the applied side, the InterpretME framework (Chudasama et al., 2025) [20] traces metadata through a machine-learning-over-KG pipeline and explicitly captures SHACL validation results inside its own knowledge graph, extending ML Schema with classes such as `SHACLValidation` and properties such as `hasSHACLResult` and `hasSHACLShape` to record what was validated and against which constraints.

Table 2. **Comparison of the three knowledge graph validation families**

Family	World assumption	Constraint types and expressiveness	Strengths and limitations	Example tools	Application complexity and typical use
Ontology-based (OWL reasoning)	Open World Assumption	Disjointness, domain and range, cardinality bounds, functional properties; constraints encoded implicitly as logical axioms; OWL 2 profiles (DL, EL, QL, RL) trade expressiveness for tractability.	Infers violations from logical contradictions and captures implicit class membership through subsumption; cannot flag absent data under OWA, and full OWL 2 DL reasoning is 2NEXPTIME-complete.	HermiT, Pellet, ELK; BIGOWL4DQ	Reasoning cost depends on the chosen profile; suited to schema consistency checking and domain modelling.
Constraint-based (SHACL, ShEx, SPIN)	Closed World Assumption for data instances	Datatype, cardinality, value ranges, string patterns, class membership, property paths (SHACL Core); arbitrary SPARQL constraints (SHACL-SPARQL); grammar-based shapes (ShEx).	W3C standard with broad triplestore support, structured violation reports, and enforcement of mandatory properties; requires complete schema knowledge, and recursive shape semantics are underspecified.	Apache Jena, Eclipse RDF4J, Stardog, GraphDB, TopBraid; Wikidata (ShEx)	Moderate cost, though most engines re-validate the full graph on update; the practical standard for data quality enforcement.
Statistical and ML-based (embeddings, contrastive learning)	Data-driven and distributional; no formal world assumption	Distributional plausibility via embeddings (TransE, DistMult, ComplEx, RDF2Vec), triple classification, cross-graph alignment, active and contrastive learning.	Detects statistical anomalies that no formal constraint can express; produces no formal violation report, gives no completeness guarantee, and depends on labelled training data.	CrossVal, KAEL, CCA; TransE, ComplEx, RDF2Vec implementations	High training cost; an auxiliary screening layer that surfaces candidate errors rather than a replacement for formal validation.

Source: **compiled by the authors**

Tracing SHACL outputs is technically feasible, but often this procedure relies on a bespoke schema rather than a shared standard. The open gap is therefore more precisely defined: no SHACL engine natively produces a provenance record conforming to the W3C PROV-O model, and no interoperable vocabulary for cross-system SHACL audit trails yet exists.

Gap 2: Incremental Validation

Existing SHACL engines typically re-validate the entire data graph when shapes are evaluated. For graphs containing millions of triples that receive frequent updates, full re-validation is prohibitively expensive. Incremental validation, which re-evaluates only the shapes affected by a given update, is a recognised need but remains at an early research stage. To advance this area, Ahmetaj et al. formalise the problem of static validation under graph updates, studying whether a graph that satisfies a SHACL specification will continue to do so after a defined update sequence [5].

While foundational work in this area was theoretical, recent advancements include a prototype implementation that extends the SHACL2FOL tool to perform static validation on medium-sized shape graphs [5]. In practice, triplestores such as GraphDB, building on the open-source Eclipse RDF4J, provide commit-time SHACL validation that runs when data changes are committed. This feature determines when validation runs, but still does not limit it to the shapes affected by an update, which is the goal of incremental validation. Commit-time validation is therefore a practical engineering mechanism, not a general solution to incremental validation.

A further extension of incremental validation is streaming validation, where triples arrive continuously from a data stream rather than in discrete update batches. In this setting, a validation engine must decide whether each incoming triple satisfies the active shapes without re-evaluating the entire graph, a constraint that rules out most existing incremental approaches and represents an open research problem.

Gap 3: Validation Under Ontology Evolution

Knowledge graph ontologies evolve over time: classes are added or removed, property domains change, and subsumption relationships are restructured. When the ontology changes, existing SHACL shapes may become inconsistent with the new schema – a phenomenon we term constraint drift. We define constraint drift as the progressive misalignment between a set of SHACL shapes and the ontology they were designed to enforce, arising from ontology changes that are not propagated into the shape graph, causing false violations or missed violations that do not reflect genuine data quality problems. For example, a shape targeting all instances of a class that is subsequently split into two subclasses may silently stop validating instances of one subclass. A key distinguishing property is its silent nature: a shape affected by constraint drift may stop catching violations without producing any error, unlike a breaking change that generates explicit failures. Ontology evolution has received sustained attention in the knowledge representation community [8], and recent work has begun to address constraint drift directly through automated propagation frameworks. The severity of this problem scales with graph size: in large-scale knowledge graphs comprising millions to trillions of edges, the ontology can no longer be treated as a static schema. It must instead evolve in lockstep with data ingestion, reasoning pipelines, and compliance requirements, making constraint drift an operational bottleneck rather than a routine maintenance task [24].

Three following situations make the problem tangible. First, a property rename: if an ontology revision replaces `:hasAuthor` with `:authoredBy`, a shape whose `sh:path` still references `:hasAuthor` matches no data and silently stops detecting documents that lack an author, a missed violation. Second, a domain or range change: if the range of `:hasAge` is changed from `xsd:integer` to a structured date type, a shape asserting `sh:datatype xsd:integer` reports every conforming new value as a violation, a false violation. Third, a class hierarchy restructuring: if `:Employee` is split into `:FullTimeEmployee` and `:PartTimeEmployee`, a shape with `sh:targetClass :Employee` no longer targets instances typed only under the new subclasses, leaving them unvalidated. In each case the ontology change is valid and intentional, yet the unpropagated shapes produce false or missed violations rather than explicit errors, which is what distinguishes constraint drift from the general study of ontology evolution [9], [25], [26].

Three frameworks are particularly relevant. Astrea [9] automatically generates SHACL shapes from ontology axioms by executing SPARQL-based

mapping rules, translating conceptual restrictions in the ontology into equivalent SHACL constraint patterns. Pürmayr (2025) [25] improves shape extraction for evolving graphs using graph changesets – sets of added or removed triples, updating only the affected SHACL shapes rather than re-extracting the full shape graph. However, Pürmayr states that this tool is ineffective for large updates when it is computationally cheaper to just re-extract the whole graph. OntoRipple (2026) [26] formalises automated propagation at the lifecycle level: it uses the OWL Change Ontology (OCH) to document differences between ontology versions and applies 35 evolution handler functions via declarative SPARQL templates to propagate changes directly in a wide range of structural elements, including node shapes, property shapes, and SPARQL constraints.

However, the declarative evolution handlers in such tools operate on deterministic templates and cannot cover all categories of ontology change. OntoRipple manages two interdependent artifact types: SHACL shapes, which express validation constraints on the RDF graph, and RML mappings, which define how heterogeneous data sources are transformed into RDF triples. For ontology changes with ambiguous effects on source-to-graph mappings, the affected RML mapping rules are exported for manual human review before application, preventing fully automated end-to-end maintenance [26].

Despite this progress, the ecosystem remains fragmented. The constraint drift problem is particularly acute in joint OWL-SHACL development environments, where ontology and shape graph evolve in parallel but are maintained independently [15]. Each framework addresses a specific facet of the constraint drift problem in isolation, but no integrated pipeline currently combines ontology versioning, automated shape propagation, and post-propagation quality assurance into a single auditable workflow.

A more fundamental gap is the lack of standardised semantic diff tools for SHACL shape graphs. While recent prototypes such as the ShapeComparator have been developed to automatically compare two versions of a shape graph and identify which constraints have been added, removed, or structurally modified [25], there remains an absence of widely adopted, interoperable tooling. Without such standardised tooling, knowledge engineers maintaining large-scale or high-velocity graphs cannot systematically audit the extent of constraint drift or verify that automated propagation has been correctly applied.

Gap 4: Hybrid Constraint and ML Validation

Constraint-based methods enforce formal rules but cannot detect statistical anomalies. ML-based methods detect distributional irregularities but cannot guarantee formal compliance. A unified validation framework that combines both would use SHACL shapes for structural and semantic rules and embedding-based models for distributional plausibility, producing a combined report that distinguishes formal violations from statistical anomalies. Such a system could, for example, flag a triple that passes all SHACL constraints but is statistically implausible given the graph's overall structure, or conversely, confirm that a statistically anomalous triple is nonetheless formally valid. Recent literature has characterised these integrated approaches as neuro-symbolic reasoning systems [6], combining the logical rigor of symbolic constraint evaluation with the uncertainty tolerance of neural methods.

Several systems now implement this integration. InterpretME combines SHACL constraint validation with ML predictive models (including LIME and decision trees) and exposes both validation status and prediction probabilities through a single queryable knowledge graph [20].

ETCOD takes a different approach: it uses semantic embeddings combined with pattern mining to identify anomalous triples, which are forwarded to a large language model (LLM) for human-readable explanation and context-aware reasoning to help an analyst determine if the anomaly is an actual error or a rare fact. Unlike Poderoso and InterpretME, ETCOD's secondary layer is probabilistic rather than formally deterministic, providing interpretable context rather than rule-based guarantees [7].

Two open problems remain within this class of systems. First, running full SHACL validation or symbolic reasoning inside a neural training loop is computationally expensive at scale, remaining a persistent bottleneck for billion-triple graphs [6]. Second, no standard schema currently specifies how a Unified Validation Report should formally distinguish a structural violation, which constitutes a

logical breach of a SHACL constraint, from a statistical anomaly, which is a fact that is mathematically improbable but structurally valid. This absence limits the applicability of hybrid validation in regulated and risk-sensitive domains, where audit trails must separately categorise rule breaches and statistical flags [21].

Apart from this, recent work continues to strengthen the machine-learning component that hybrid validation must integrate: error-detection frameworks combine structural embeddings with semantic or textual signals to flag implausible triples [27], contrastive objectives separate correct triples from plausible but erroneous ones [28], and applied systems pair large-language-model agents with graph-based methods to automate rule-compliance checking, as shown for constructability checking in building engineering [29].

CONCLUSIONS

This survey has reviewed knowledge graph validation approaches published between 2011 and 2026. Authors suggest taxonomy organizing validation aproaches into three families: ontology-based validation via OWL reasoning, constraint-based validation using SHACL, ShEx, and SPIN, and statistical/ML-based error detection using knowledge graph embeddings and contrastive learning.

Constraint-based validation, and SHACL in particular, has emerged as the practical standard for RDF data quality enforcement. SHACL provides a W3C-standardised language with broad triplestore support, structured violation reports, and two levels of expressiveness (Core and SPARQL). Ontology-based reasoning complements SHACL by detecting consistency violations that arise from logical axiom interactions, and recent work on combined OWL-SHACL workflows shows how the two can be used together effectively. ML-based methods extend validation to distributional anomaly detection, catching errors that no formal constraint can express, though at the cost of interpretability and completeness guarantees [6].

The inconsistency between Open World Assumption and Closed World Assumption remains a central design consideration across all three families: reasoning approaches can infer violations from logical contradictions but cannot flag absent data, while constraint-based approaches enforce explicit closure conditions but require complete schema knowledge [4].

This study discovers four open research gaps: provenance-aware validation, incremental validation, validation under ontology evolution and hybrid constraint-plus-ML approaches. Provenance-aware validation would enable audit trails for regulated domains. Incremental validation would reduce the cost of validating frequently updated graphs. Validation under ontology evolution would address constraint drift when schemas change. The naming and formal definition of constraint drift is introduced here for the first time, since no prior work treats it as one operational problem.

Hybrid constraint-plus-ML approaches, now implemented in emerging neuro-symbolic pipelines [6], face open challenges in scalability and the standardisation of unified validation reporting. The consolidation of four open problems into a single research agenda is the authors' synthesis. Addressing these gaps will require coordinated effort across the W3C Data Shapes Working Group, the knowledge graph engineering community, and the ML-on-graphs research community.

This survey is limited to RDF-based knowledge graphs and does not cover property graph validation systems (e.g., Neo4j constraints) or the emerging PG-Schema standard.

REFERENCES

1. Xue, B. & Zou, L. “Knowledge graph quality management: a comprehensive survey”. *IEEE Transactions on Knowledge and Data Engineering*. 2022; 35 (5): 4969–4988. DOI: https://doi.org/10.1109/TKDE.2022.3150080.

2. Paulheim, H. “Knowledge graph refinement: a survey of approaches and evaluation methods”. *Semantic Web*. 2017; 8 (3): 489–508. DOI: https://doi.org/10.3233/SW-160218.

3. Issa, S., Adekunle, O., Hamdi, F., Cherfi, S. S. S., Dumontier, M. & Zaveri, A. “Knowledge graph completeness: a systematic literature review”. *IEEE Access*. 2021; 9: 31322–31339. DOI: https://doi.org/10.1109/ACCESS.2021.3056622.

4. Gayo, J. E. L., Prud’hommeaux, E., Boneva, I. & Kontokostas, D. “Validating RDF data”. *Synthesis Lectures on the Semantic Web: Theory and Technology*. 2017; 1 (1): 1–328. San Rafael, CA, USA: Morgan & Claypool. DOI: https://doi.org/10.2200/S00786ED1V01Y201707WBE016.

5. Ahmetaj, S., Konstantinidis, G., Ortiz, M., Pareti, P. & Simkus, M. “SHACL validation under graph updates (extended paper)”. *arXiv*. 2025. DOI: https://doi.org/10.48550/arXiv.2508.00137.

6. DeLong, L. N., Fernández Mir, R. & Fleuriot, J. D. “Neurosymbolic AI for reasoning over knowledge graphs: a survey”. *IEEE Transactions on Neural Networks and Learning Systems*. 2024; 36 (5): 7822–7842. DOI: https://doi.org/10.1109/TNNLS.2024.3420218.

7. Nguyen, T. T. N., Senaratne, A. & Seneviratne, L. “ETCOD: embedding-based anomaly detection and LLM-driven validation framework for knowledge graphs”. *Australasian Conference on Data Science and Machine Learning*. 2026. p. 454–469. Singapore: Springer Nature. DOI: https://doi.org/10.1007/978-981-95-6786-7_31.

8. Flouris, G., Manakanatas, D., Kondylakis, H., Plexousakis, D. & Antoniou, G. “Ontology change: classification and survey”. *The Knowledge Engineering Review*. 2008; 23 (2): 117–152. DOI: https://doi.org/10.1017/S0269888908001367.

9. Cimmino, A., Fernández-Izquierdo, A. & García-Castro, R. “Astrea: automatic generation of SHACL shapes from ontologies”. *European Semantic Web Conference*. 2020. p. 497–513. Cham, Switzerland: Springer. DOI: https://doi.org/10.1007/978-3-030-49461-2_29.

10. Sikos, L. F. & Philp, D. “Provenance-aware knowledge representation: a survey of data models and contextualized knowledge graphs”. *Data Science and Engineering*. 2020; 5 (3): 293–316. DOI: https://doi.org/10.1007/s41019-020-00118-0.

11. Schaffenrath, R., Proksch, D., Kopp, M., Albasini, I., Panasiuk, O. & Fensel, A. “Benchmark for performance evaluation of SHACL implementations in graph databases”. *Rules and Reasoning: 4th International Joint Conference on Rules and Reasoning (RuleML+RR 2020)*, LNCS, 2020; 12173. DOI: https://doi.org/10.1007/978-3-030-57977-7_6.

12. Polleres, A., Hogan, A., Delbru, R. & Umbrich, J. “RDFS and OWL reasoning for linked data”. *Reasoning Web International Summer School*. Berlin, Heidelberg, Germany: Springer. 2013. p. 91–149. DOI: https://doi.org/10.1007/978-3-642-39784-4_2.

13. Barba-González, C., Caballero, I., Varela-Vaca, Á. J., Cruz-Lemus, J. A., Gómez-López, M. T. & Navas-Delgado, I. “BIGOWL4DQ: ontology-driven approach for Big Data quality meta-modelling, selection and reasoning”. *Information and Software Technology*. 2024; 167: 107378.
DOI: https://doi.org/10.1016/j.infsof.2023.107378.

14. Martínez-Sarmiento, E., Ruckhaus, E., Toledo, J., Doña, D. & Corcho, O. “ERA-SHACL-Benchmark: a real-world benchmark to assess the performance and quality of in-memory SHACL engines”. *Semantic Web Journal*. 2025. DOI: https://doi.org/10.5281/zenodo.15423772.

15. Ruckhaus, E., Toledo, J., Martínez Sarmiento, E. A. & Corcho, O. “Lessons learned from the combined development of OWL and SHACL”. *Proceedings of the 13th Knowledge Capture Conference*. 2025. p. 1–4. DOI: https://doi.org/10.1145/3731443.3771340.

16. Ke, J., Zacouris, Z. & Acosta, M. “Efficient validation of SHACL shapes with reasoning”. *Proceedings of the VLDB Endowment*. 2024; 17 (11): 3589–3601. DOI: https://doi.org/10.14778/3681954.3682023.

17. Dong, J., Zhang, Q., Huang, X., Tan, Q., Zha, D. & Zihao, Z. “Active ensemble learning for knowledge graph error detection”. *Proceedings of the 16th ACM International Conference on Web Search and Data Mining*. 2023. p. 877–885. DOI: https://doi.org/10.1145/3539597.3570368.

18. Wang, Y., Ma, F. & Gao, J. “Efficient knowledge graph validation via cross-graph representation learning”. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2020. p. 1595–1604. DOI: https://doi.org/10.1145/3340531.3411993.

19. Liu, X., Liu, Y. & Hu, W. “Knowledge graph error detection with contrastive confidence adaption”. *Proceedings of the AAAI Conference on Artificial Intelligence*. 2024; 38 (8): 8824–8831. DOI: https://doi.org/10.1609/aaai.v38i8.28678.

20. Chudasama, Y., Purohit, D., Rohde, P. D., Gercke, J. & Vidal, M. E. “InterpretME: a tool for interpretations of machine learning models over knowledge graphs”. *Semantic Web*. 2025; 16 (2): SW-233511. DOI: https://doi.org/10.48786/edbt.2023.03 DOI: 10.3233/SW-233511.

21. Kolli, C. K. “Hybrid neuro-symbolic models for ethical AI in risk-sensitive domains”. *arXiv*. 2025. DOI: https://doi.org/10.48550/arXiv.2511.17644.

22. Dibowski, H. “Full traceability and provenance for knowledge graphs”. *Formal Ontology in Information Systems.* Amsterdam, Netherlands: *IOS Press*. 2024. p. 223–237. DOI: https://doi.org/10.3233/FAIA241309.

23. Delva, T., Dimou, A., Jakubowski, M. & Van den Bussche, J. “Data provenance for SHACL”. *OpenProceedings*. 2023. DOI: https://doi.org/10.48786/edbt.2023.23.

24. Zghal, S. & Kachroudi, M. “The co-evolution of ontologies and extensive knowledge graphs on a web scale”. *The Journal of Supercomputing*. 2026; 82 (5): 264. DOI: https://doi.org/10.1007/s11227-026-08380-1.

25. Pürmayr, E. “SHACL shapes extraction for evolving knowledge graphs”. *Doctoral dissertation, Technische Universität Wien*. 2025. DOI: https://doi.org/10.34726/hss.2025.120502.

26. Conde-Herreros, D., Corcho, O. & Chaves-Fraga, D. “OntoRipple: making waves in the knowledge graph lifecycle”. *SoftwareX*. 2026; 33: 102542. DOI: https://doi.org/10.1016/j.softx.2026.102542.

27. Zhao, Y., Liu, Y., Ao, X. & He, Q. “A flexible knowledge graph error detection framework combined with semantic information”. *Big Data (BigData 2024), CCIS*. 2025; 2301: 1–14, https://www.scopus.com/pages/publications/85218465475?origin=resultslist.
DOI: https://doi.org/10.1007/978-981-96-1024-2_1.

28. Wang, X., Ao, X., Zhang, F., Zhang, Z. & He, Q. “Knowledge error detection via textual and structural joint learning.” *Big Data Mining and Analytics*. 2025; 8 (1): 233–240, https://www.scopus.com/pages/publications/86000179469?origin=resultslist.
DOI: https://doi.org/10.26599/BDMA.2024.9020040.

29. Li, A., Wang, B., Gong, X., Hou, F., Tao, X., Ma, J., Cheng, Jack C. P. “Multi-agent integration of LLMs and graph-based method for automated MEP constructability checking”. *Automation in Construction*. 2026; 187: 106886. ISSN 0926-5805, https://www.scopus.com/pages/publications/105034619292?origin=resultslist. DOI: https://doi.org/10.1016/j.autcon.2026.106886.

Conflicts of Interest: The authors declare that they have no conflict of interest regarding this study, including financial, personal, authorship or other, which could influence the research and its results presented in this article
Author Mikolaj P. Karpinski is member of the Editorial Board of this journal. This role had no influence on the peer review process or editorial decision regarding this manuscript

Received 20.04.2026
Received after revision 15.06.2026
Accepted 19.06.2026

DOI: https://doi.org/10.15276/aait.09.2026.22
УДК 004.82:004.6

Таксономія підходів до валідації графів знань

Осійчук Іван Васильович[1)]
ORCID: https://orcid.org/0009-0003-6844-2374; ivan_osiichuk0108@tntu.edu.ua. Scopus Author ID: 59296923000
Загородна Наталія Володимирівна[1)]
ORCID: https://orcid.org/0000-0002-1808-835X; zagorodna.n@gmail.com. Scopus Author ID: 57189380553
Скоренький Юрій Любомирович[1)]
ORCID: https://orcid.org/0000-0002-4809-9025; skorenkyy@tntu.edu.ua. Scopus Author ID: 6507755672
Карпінський Миколай Петрович[2)]
ORCID: https://orcid.org/0000-0002-8846-332X; mikolaj.karpinski@uken.krakow.pl. Scopus Author ID: 57226717849
Токулієва Айжан Конурбаевна[3)]
ORCID: https://orcid.org/0000-0002-5019-2413; tokkuliyeva_ak_1@enu.kz. Scopus Author ID: 57445077400
[1)] Тернопільський національний технічний університет, вул. Руська, 56. Тернопіль, 46025, Україна
[2)] Педагогічний університет імені Комісії національної освіти, Podchorazych St2.Краків, 30-084, Польща
[3)] Євразійський національний університет імені Гумільова. Астана, 010008, Казахстан

АНОТАЦІЯ

Актуальність. Графи знань, закодовані у форматі Resource Description Framework, стали стандартною основою для інтеграції даних, семантичного пошуку та автоматизованого виведення в галузях від біомедицини до корпоративного управління знаннями. У міру того як ці графи зростають за масштабом, дедалі важливішим стає забезпечення відповідності даних визначеним структурним і семантичним правилам розподіленими командами. **Мета.** У цьому огляді представлено

систематичний аналіз підходів до валідації графів знань, опублікованих у період з 2011 по 2026 рік, що охоплює три основні сімейства методів, та запропоновано таксономію, яка класифікує наявні роботи за такими трьома сімействами: (i) онтологічна валідація засобами виведення на основі Web Ontology Language, (ii) валідація на основі обмежень із використанням мови Shapes Constraint Language (SHACL), мови Shape Expressions (ShEx) та мови SPARQL Inferencing Notation (SPIN), (iii) статистичне, засноване на машинному навчанні, виявлення помилок із застосуванням вбудовань графів знань і контрастного навчання. Виконано такі **задачі**: огляд літературних джерел, порівняльний аналіз перелічених методів, запропонована таксономія підходів до валідації графів, виявлені прогалини у досліджуваній галузі. **Наукова новизна:** запропоновано єдину трирівневу таксономію, що характеризує кожне сімейство за його базовими критеріями (припущення відкритого світу OWA або припущення закритого світу CWA), виразністю та обчислювальними витратами. Крім того, формально визначено поняття дрейфу обмежень (constraint drift) – поступового неузгодження правил Shapes Constraint Language із відповідною онтологією. **Практична значущість.** Огляд надає архітектурні проектні рішення, що дозволяють подолати протиріччя між припущенням відкритого світу (OWA) у виведенні на основі Web Ontology Language і припущенням закритого світу (CWA) у формах Shapes Constraint Language в усіх трьох сімействах, що може бути корисним для практиків і дорожньою картою для дослідників. **Результати:** запропоновано таксономію підходів до валідації та проведено порівняльний аналіз трьох сімейств. Валідація на основі обмежень, і зокрема Shapes Constraint Language, визначена як практичний стандарт забезпечення якості Resource Description Framework-даних, тоді як онтологічне міркування доповнює її з точки зору логічної узгодженості, а методи машинного навчання розширюють її до виявлення статистичних аномалій. Виявлено чотири відкриті прогалини в області досліджень: валідація з урахуванням походження даних (provenance-aware validation), інкрементна валідація, валідація в умовах еволюції онтологій (включно з дрейфом обмежень, формально визначеним у підрозділі «Gap 3»), а також гібридні підходи, що поєднують обмеження з машинним навчанням. **Висновки:** жодне окреме сімейство не задовольняє усіх вимог до валідації; онтологічні, обмежувальні та статистичні/ML-методи є взаємодоповнювальними, а несумісність OWA/CWA залишається центральним проектним аспектом у всіх трьох сімействах. Усунення виявлених прогалин – зокрема дрейфу обмежень і гібридних підходів – окреслює дорожню карту для майбутніх досліджень, що потребують скоординованих зусиль спільнот зі стандартизації, інженерії графів знань і машинного навчання на графах.

Ключові слова: граф знань; перевірка даних; онтологічне виведення; мови обмежень; виявлення помилок на основі машинного навчання; Resource Description Framework бази даних

ABOUT THE AUTHORS

Ivan V. Osiichuk - PhD student in Computer Science. Ternopil National Technical University, 56, Ruska Str. Ternopil, 46025, Ukraine
ORCID: https://orcid.org/0009-0003-6844-2374; ivan_osiichuk0108@tntu.edu.ua. Scopus Author ID: 59296923000
***Research field*:** Knowledge Representation; Data Engineering; Data Science

Осійчук Іван Васильович - аспірант спеціальності Комп'ютерні науки. Тернопільський національний технічний університет імені І. Пулюя, вул. Руська 56. Тернопіль, 46001, Україна

Nataliya V. Zagorodna - PhD in Engineering Sciences, Associate Professor, Head of the Cybersecurity Department. Ternopil National Technical University, 56, Ruska Str. Ternopil, 46025, Ukraine
ORCID: https://orcid.org/0000-0002-1808-835X; zagorodna_n@tntu.edu.ua. Scopus Author ID: 57189380553
***Research field*:** Computer Security and Cryptography; Artificial Intelligence; Signal Processing

Загородна Наталія Володимирівна - кандидат технічних наук, доцент, завідувач кафедри Кібербезпеки. Тернопільський національний технічний університет імені І. Пулюя, вул. Руська 56. Тернопіль, 46001, Україна

Yuriy L. Skorenkyy - PhD in Physical-Mathematical Sciences, Associate Professor, Associate Professor of the Physics Department. . Ternopil National Technical University, 56, Ruska Str. Ternopil, 46025, Ukraine
ORCID: https://orcid.org/0000-0002-4809-9025; skorenkyy@tntu.edu.ua. Scopus Author ID: 6507755672
***Research field*:** Cyber-physical systems, Technical Protection of Information, Physics

Скоренький Юрій Любомирович - кандидат фіз.-мат. наук, доцент кафедри Фізики. Тернопільський національний технічний університет імені І.Пулюя, вул. Руська 56. Тернопіль, 46001, Україна

Mikolaj P. Karpinski - Doctor of Engineering Sciences, Head of Department of Software Engineering of Institute of Security and Computer Science. University of the National Education Commission, 2, Podchorazych St. Krakow, 30-084, Poland
ORCID: https://orcid.org/0000-0002-8846-332X; mikolaj.karpinski@uken.krakow.pl. Scopus Author ID: 57226717849
***Research field*:** Information Technologies and Computer Systems; Computer Security and Cryptography; Computing Systems

Карпінський Микола Петрович - доктор технічних наук, завідувач кафедри Програмної інженерії Інституту кібербезпеки і комп'ютерних наук. Університет Комісії національної освіти. Краків, Польща

Aizhan K. Tokkuliyeva - doctoral student in Information Security Systems. L.N. Gumilyov Eurasian National University. Astana, 010008, Republic of Kazakhstan
ORCID: https://orcid.org/0000-0002-5019-2413; tokkuliyeva_ak_1@enu.kz. Scopus Author ID: 57445077400
***Research field*:** Information Security Risk Assessment, Digital Technologies, Digital Forensics, AI in Information Security

Токулісва Айжан Конурбаевна - докторант за спеціальністю Інформаційні кіберсистеми. Євразійський національний університет імені Л. М. Гумільова. Астана, 010008, Казахстан