KG4RSV: Towards Knowledge Graph-driven Specification and Validation of ML Dataset Quality Requirements: Case Study on e-ICU dataset

Ziad NASRI
Department of Software and Information Systems Technologies
University of Constantine 2, Algeria
ziad.nasri@univ-constantine2.dz

Samir SELLAMI
Department of Mathematics and Computer Science
Higher Normal School of Technological Education Skikda, Algeria
samir.sellami@univ-constantine2.dz

Chabane DJEDDI
Department of Intelligent Systems Engineering
National School of Artificial Intelligence Sidi-Abdellah, Algeria
chabane.djeddi@ensia.edu.dz

Taoufiq DKAKI
IRIT Laboratory
University of Toulouse 2 - Jean Jaurès, France
dkaki@irit.fr

Nacereddine ZAROUR
Department of Software and Information Systems Technologies
University of Constantine 2, Algeria
nasro.zarour@univ-constantine2.dz

Pierre-Jean Charrel
IRIT Laboratory
University of Toulouse 2 - Jean Jaurès
France
pierre-jean.charrel@univ-tlse2.fr

Abstract: *Artificial Intelligence (AI) and Machine Learning (ML) are increasingly embedded in critical domains, raising the challenges for ensuring the quality and reliability of AI/ML solutions. While traditional Requirements Engineering (RE) plays a central role in eliciting and specifying the stockholders' expectations, current practices often lack the systematic specification and verification of dataset requirements, even though poor data quality issues remain a primary cause of ML project failures. This paper introduces a Knowledge Graph-enabled framework for ML Dataset Requirements Specification (KG4RSV) and Validation. KG4RSV addresses three critical challenges in dataset engineering: managing data heterogeneity, ensuring data quality, and supporting systematic specification and validation of dataset requirements. Leveraging the semantic capabilities of Knowledge Graphs (KGs), the proposed method enables a structured representation of both data requirements and datasets using Resource Description Framework (RDF) and SHACL standards. We evaluate KG4RSV on a real-world clinical dataset electronic Intensive Care Unit (eICU) for binary classification of cardiovascular events. Results demonstrate that the framework effectively detects and handles quality issues related to completeness, consistency, and semantic correctness. The validation process led to notable improvements in ML performance-up to a 7% increase in recall for an Artificial Neural Network (ANN) model and an average 5% improvement across multiple ML models, such as Random Forest (RL) and Logistic Regression (LR). By bringing dataset validation into the requirements process, KG4RSV improves data quality and ensures better alignment between data and system goals from the early phases*.

Keywords: *Artificial intelligence, AI/ML-dataset, dataset quality requirements, requirements specification and validation, knowledge graphs, requirements engineering*.

Received August 08, 2025; accepted February 09, 2026
https://doi.org/10.34028/iajit/23/5/9

1. Introduction

Artificial Intelligence (AI) and Machine Learning (ML) technologies are rapidly advancing and being widely adopted across a wide range of domains, including healthcare, finance, transportation, manufacturing, and education [3, 23, 24, 49]. The increasing integration of AI/ML into critical domains/sectors elevates the importance of building AI systems that are trustworthy, transparent, and effective in real-world contexts [20].

Requirements Engineering (RE), a fundamental phase in the software engineering lifecycle, provides structured techniques for eliciting, analyzing, documenting, validating, and evolving the stakeholders' needs and constraints. While RE has traditionally focused on functional and non-functional software requirements, researchers have recently extended its principles and techniques to support Big Data and intelligent systems, including AI and ML-based applications [20, 46]. However, these extensions still face challenges in fully addressing all aspects and components of AI/ML systems, particularly the specification and validation of dataset requirements [4].

Within the context of RE for AI/ML (RE4AI/ML), researchers have proposed a considerable number of contributions to address the challenges across the AI/ML development life cycle, including the problem, data, and model layers. For better problem understanding and modelling, Djeddi *et al*, [17] and Barrera *et al*, [10] introduce two specific extensions of KAOS and i* to elicit Big Data and AI/ML system requirements following a Goal-Oriented RE (GORE) approach [26]. The specification and modelling of data requirements

have also received significant attention due to their crucial role in determining the reliability and quality of AI/ML systems [28, 29, 42].

Finally, the model-related RE activities focus on identifying and specifying:

1) Functional requirements for implementing AI/ML models (e.g., XAI as a recent trend) [57].
2) Non-functional requirements for evaluation [25], which determine the quality attributes of AI/ML systems.

Despite these advancements, the engineering of AI/ML datasets still poses significant challenges. Numerous studies and industrial reports have identified poor dataset quality, insufficient preprocessing, and weak data governance as major reasons for AI/ML project failures [8, 13]. In the literature, Priestley *et al*. [39] provide a survey of data-quality dimensions in ML pipelines, highlighting intrinsic, contextual, representational, and accessibility aspects. They emphasize that an ML model's performance and reliability depend directly on the quality of its training dataset. From a practical perspective, several methods and tools have been developed to support data quality assurance, including XML Schema, JSON Schema, and Great Expectations (a widely-used Python library for dataset validation) [37]. This indicates that data validation is common in ML practice, but usually not formalized as part of RE.

RE for datasets is a key to producing successful AI/ML projects. Defining data quality requirements-such as completeness, consistency, accuracy, credibility, and currentness ensures that the dataset aligns with the system's objectives and the model's learning expectations [13]. Moreover, these requirements must not only be elicited and specified, but also verified and validated against actual datasets before model training. However, dataset preprocessing pipelines rarely incorporate such formal requirement checks, increasing the risk of using poor-quality or biased data.

In this work, we leverage Knowledge Graphs (KGs) technology [50] to support the specification, verification, and validation of ML dataset requirements. KGs provide a rich semantic structure to represent data entities, their relationships, and associated constraints. By using KGs, we can formally represent both the dataset and its requirements, enabling automated reasoning, traceability, and validation through technologies like a standardized query language and protocol used to retrieve and manipulate data stored in Resource Description Framework (RDF) (Resource Description Framework) graph format (SPARQL) [16] and Shape Constraints Language (SHACL) [31, 45]. This semantic foundation bridges the gap between abstract requirements and concrete data instances, allowing for robust and reusable dataset engineering methods.

This paper focuses on addressing the key dataset engineering challenges in ML projects, including:

1) Handling heterogeneity of data types and sources;
2) Ensuring dataset quality requirements such as completeness, consistency, etc.
3) supporting dataset requirements specification, verification, and validation activities. Our proposed contribution is a Knowledge Graph-enabled method for ML-dataset' Requirements Specification and Validation (KG4RSV).

Which provides a structured pipeline that:

1. Specifies and models data requirements in a Requirements Knowledge Graph (Requirements-KG) using RDF standards [11].
2. Automatically generates SHACL shapes from the Requirements-KG.
3. Transforms the dataset into a Dataset Knowledge Graph (Dataset-KG).
4. Verifies and validates the Dataset-KG against the specified requirements, formalized as SHACL constraint shapes.
5. Reconstructs a validated dataset for model training.

To assess the impact of specifying and validating dataset requirements with the KG4RSV framework, we evaluate the performance of downstream ML task. In particular, we train binary classification models to predict cardiovascular events (positive/negative) using the real-world eICU dataset [40], a real clinical dataset. The evaluation demonstrates our approach's ability to detect data-constraint violations early. For instance, constraints on completeness, accuracy, consistency, and complex interrelationships among features are expressed as SPARQL queries and validated to ensure data quality aligns with domain-specific expectations.

As a result, the KG4RSV framework improved the overall quality of the dataset, yielding up to a 7 % increase in the Recall metric for the Artificial Neural Network (ANN) model and an average 5 % improvement across other models-RF, Logistic Regression, and K-Nearest Neighbors (KNN)-along with modest gains in accuracy, precision, and F1-score.

The remainder of this paper is organized as follows: Section 2 presents the background and motivation for our work. Section 3 details the KG4RSV framework, including its components, transformation workflows, and validation process. Section 4 describes the case study on the eICU dataset and experimental results. Section 5 discusses the findings and implications. Section 6 reviews related works. Finally, section 7 concludes the paper and outlines directions for future work.

2. Background and Motivation

This section provides an overview of the interdisciplinary domains and techniques used in this contribution. It begins by introducing ML datasets, then presents the RE activities dedicated to ML-dataset specification and their importance, and finally discusses

Knowledge Graphs (KGs) as a modern approach for representing and managing requirements.

2.1. ML Dataset

The dataset serves as a cornerstones component of any AI/ML-based systems; it often determines the context, boundaries, and potential limitations of the AI/ML system [4]. ML datasets come in various structures and types, including structured tabular data (common in clinical, financial, and administrative domains), unstructured data (e.g., text, audio, images), semi-structured data (e.g., JSON, XML), and time-series data (e.g., sensor readings, ECG signals) [51, 61]. Each type presents unique challenges in terms of data collection, cleaning, integration, and preprocessing. In tabular datasets, which are the focus of this study, a dataset is typically composed of rows (samples or instances) and columns (features or attributes), where one or more columns may represent the target variable(s) for prediction [6, 44].

The inherent heterogeneity in data sources and types presents a significant challenge in AI/ML multi-models dataset engineering. For instance, in healthcare domain such clinical datasets may contains several features across different data formats, including numerical, categorical, and time-Series like ECG Signal, etc.(e.g., Blood Pressure (BP) may appear in different features like systolic and diastolic BP, lab-tests, or signal records depending on the data sources) [24, 36]. This heterogeneity can lead to inconsistency, ambiguity, and misinterpretation unless handled properly during data preprocessing and AI/ML system design [30, 53].

Moreover, the volume and scalability of data lead to difficulties in managing and processing extremely large datasets for training high-performing models. As the amount of data increases, systems must efficiently store, preprocess, and train on this data [34, 59]. For example, a medical machine learning model may require thousands to millions of high-quality MRI images [62]. Poor-quality data tends to increase the overall cost of storage, maintenance, and processing, unless filtered/cleaned early [22].

Motivated by these challenges, the proposed KG4RSV framework leverages KGs foundations to model, integrate, and reason over such heterogeneous dataset formats and data types, in a semantic and machine-processable way.

2.2. Requirements Engineering for ML Datasets (RE4ML-Dataset)

While traditional RE focuses on defining and validating system functionalities and qualities, recent research has highlighted the need to extend RE practices to the data dimension-especially in the context of AI/ML systems [7]. In machine learning pipelines, data preprocessing is the foundation upon which all subsequent learning stages depend, and its quality directly influences the reliability and performance of the final system.

Dataset quality requirements encompass a broad set of properties, including [13]:

- Completeness: ensuring that necessary features or values are present.
- Consistency: verifying that values conform to expected formats, ranges, or inter-feature relationships.
- Conformance: matching the dataset to domain constraints or external ontologies.
- Accuracy/Plausibility: identifying outliers or illogical values (e.g., negative age).
- Credibility: The degree to which a source or information is considered reliable and believable, based on evidence or reputation.
- Confidence: The quantified level of certainty in a result, decision, or prediction, often expressed as a probability or score.
- Trustworthiness: The overall expectation that a system or source will act reliably, securely, and ethically over time.
- Interpretability: labeling features clearly for human and machine understanding.

These requirements and others like fairness, transparency, and explainability should ideally guide the data pre-processing phase before training begins. Unfortunately, existing ML development workflows often lack formal mechanisms to express, trace, and validate such requirements [56]. Instead, ad-hoc scripts and heuristics dominate, making it difficult to ensure or audit dataset quality. This gap motivates the development of a formalized and automatable Requirements Specification and Validation (RSV) process that can be integrated into ML pipelines.

In this context, we argue for a requirements-driven dataset engineering approach where dataset requirements are explicitly modeled, transformed into formal constraints, and systematically validated against the dataset before training. This vision aligns well with the rigor and traceability principles advocated by RE.

2.3. Knowledge Graphs for Requirements Engineering

Knowledge Graphs (KGs) are powerful technology for representing information in a structured, semantically rich, and machine-processable manner. In RDF-based KGs, knowledge is encoded as triples: subject, predicate, and object, connecting nodes entities (subject and object) through labeled edges (properties or predicates), as illustrated in Figure 1. Built upon Semantic Web (SW) standards [12], KGs enable advanced querying via languages like SPARQL [35] and constraint validation through SHACL [31], while supporting automated reasoning to infer implicit knowledge and enforce logical consistency.

Recently, the use of KGs has witnessed a growing

demand across various fields, from healthcare to industry and space research [9, 33]. In healthcare particularly, the integration of KGs into information systems has become increasingly prevalent, driven by the need to manage diverse data modalities (e.g., clinical notes, laboratory results, imaging, sensor devices) and derive actionable insights from interconnected data [14]. For instance, Huang *et al*. [27] demonstrate how symbolic reasoning over KGs can support causal inference in patient profiles. Similarly, towards an explainable AI (XAI), KGs have been employed to enhance transparency in tasks such as detecting healthcare misinformation, predicting adverse drug reactions, and elucidating drug-drug interactions [41].

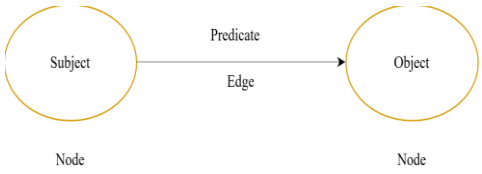

Figure 1. RDF-based knowledge graph components.

Therefore, KGs can offer concrete operational advantages in the context of AI/ML dataset specification and validation. First, they have the capability to enable semantic alignment of heterogeneous data sources and types by linking them to a unified database built on ontology and predefined domain-specific vocabularies. Second, they provide a formal representation of dataset requirements (e.g., value constraints, temporal rules, class balance conditions) that can be programmatically validated using SHACL. This goes beyond syntactic transformations where KGs facilitate semantic validation, allowing detection of violations such as out-of-range values, missing dependencies, or logical inconsistencies. Moreover, RDF schemas and reusable vocabularies promote extensibility and traceability: from the specification of data constraints to their verification and correction. This makes KGs particularly well-suited for structuring and validating datasets in ML workflows where transparency, correctness, and reusability are essential.

This work leverages these capabilities of KGs to establish a structured and semantically driven pipeline, from the formal specification of dataset quality requirements to their automated validation in AI/ML contexts.

3. KG4RSV: Knowledge Graph-enabled Requirements Specification and Validation Framework for AI/ML Datasets

The proposed KG4RSV framework provides a KG-driven pipeline to support the specification and validation of ML dataset requirements using semantic web technologies. By combining the expressiveness of RDF, the querying power of SPARQL, and the constraint-checking capabilities of SHACL, KG4RSV enables a structured, interpretable, and executable process that ensures dataset quality before feeding into ML models training.

Figure 2 illustrates the KG4RSV pipeline, which consists of five main phases:

1) Requirement Specifying and Modeling.
2) Transformation of Requirements into SHACL shapes.
3) Dataset-to-KG Transformation.
4) Dataset-KG Validation.
5) Reverse Transformation (from Valid-Dataset KG to original Dataset format).

The following subsections will detail the KG4RSV process.

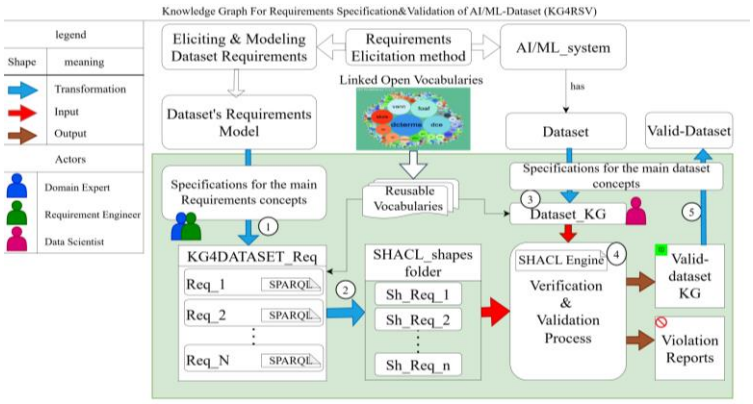

Figure 2. The KG4RSV overview.

3.1. Requirements Eliciting, Analyzing, and Documenting (Out of KG4RSV)

In contrast to the requirements validation activity, this work does not detail the initial three phases of the RE process-commonly referred to as eliciting, analyzing, and documenting stages. Instead, these earlier phases are collectively designated as KG4RSV-preprocess which serves as the essential input to the KG4RSV framework.

This step involves capturing the requirements that define what constitutes a "valid" data(set) value or structure for a specific ML task (e.g., systemic systolic should be strictly greater than systemic-diastolic}), and preparing an ontology mapping file that contains a

reusable domain-specific vocabulary, collected from online specialized published domain ontologies available in vocabulary registries and discovery platforms for the Semantic Web, such as Linked Open Vocabularies (LOV) [54] or the Wikidata Property/PID Directory [21] (see figure 2). These fundamental phases should be performed by a requirements engineer in collaboration with a domain expert.

We have deliberately opted not to provide an in-depth discussion of the exact procedures and outcomes related to preprocess phase in this paper in order to avoid excessively lengthening the manuscript, thereby maintaining its clarity and concentration. This approach ensures that the primary focus is on the validation phase, which is the fundamental contribution of this paper, while appropriately acknowledging the essential preparation function performed by the initial three phases.

3.2. Step 1: Specifying and Modeling Requirements in Knowledge Graphs (RDF)

Unlike the standard specification and modeling activity (which is part of the documentation phase), in this step, each requirement elicited in the previous phase (section 3.1) by the requirement engineer and domain expert is then formalized manually by the same actors using the Requirements RDF Schema (see figure 3). This schema defines key concepts such as requirement identifiers, target classes, category, constraint expression, and their properties (domain and range).

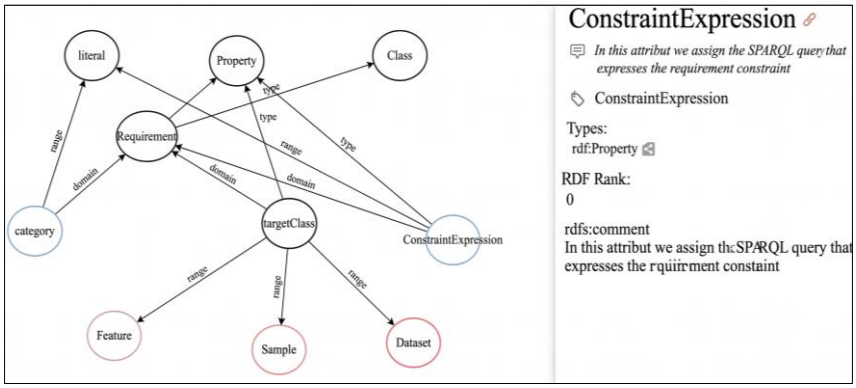

Figure 3. Requirements RDF schema.

The requirements constraints are expressed using SPARQL as a constraint language due to its declarative syntax, semantic precision, and ability to express complex logic. For instance, the requirement mentioned above (section 3.1) is encoded as a SPARQL query to select violations from a dataset graph as illustrates Figure 4).

```
# RQ-CONS-02: systemicsystolic should be strictly
greater than systemicdiastolic
req:RQ-CONS-02 a req:Requirement ;
req:targetClass mlds:Sample ; req:category
"Consistency" ; req:constraintExpression """\"
SELECT ?this WHERE {
?this snomed:72313002 ?sbp ; snomed:271650006
?dbp .
FILTER (?sbp !="NaN" && ?dbp!="NaN" && ?sbp <=
?dbp) }"""\" .
```

Figure 4. Constraint for systolic and diastolic blood pressure.

3.3. Step 2: Transforming Requirements-KG into SHACL Shapes

While SPARQL allows the expression of complex logical conditions, SHACL provides a standardized mechanism for RDF graph validation. SHACL shapes define conformance criteria, and when enriched with SPARQL-based rules, they serve as constraint violation detectors.

This step involves mapping each Requirement modeled in the Requirements_RDF into a SHACL NodeShape, using a Python script that implements the Algorithm 1. Concerning SPARQL constraints, the sh:select construct is used to embed their expression. Additional metadata, such as tsh:message and sh:severity, is included to support traceability and interpretability.

Algorithm 1:GenerateSHACLShapesFromRequirements

Input:
Requirements KG: RDF/Turtle file encoding the Requirements Knowledge Graph
Output_path: Path to store the generated SHACL shapes
Output:
A folder requirements shapes containing SHACL shape files (.ttl), one per requirement

1: Begin Algorithm
2: Parse Requirements KG into RDF graph
3: for each req of type REQ:Requirement do
4: Extract target class
5: Extract constraint expression (SPARQL query)
6: Store (req id, target class, sparql expression) in Constraints list
7: end for
8: Initialize Output Directory:
9: Create folder requirements shapes under output path (if not exists)
10: for each item in Constraints list do :
11: Create new RDF graph shape graph
12: Define a NodeShape:
13: URI: ex:Shape reqID
14: Type: sh:NodeShape
15: Target Class: sh:targetClass = targetClass
16: Link SPARQL constraint to the NodeShape:
17: Create sh:SPARQLConstraint node
18:Add sh:select = sparql expression

19: Set sh:message = "Violation of reqID"
20: Set sh:severity = sh:Violation
21: Serialize shape graph as reqID.ttl in requirements shapes
22: end for
23: End Algorithm

3.4. Step 3: Transforming Dataset into Dataset-KG

After reviewing the approaches of transforming structured and unstructured datasets into knowledge graphs, five mature contributions have been identified. The investigation uncovers a significant lack of attention to AI/ML-datasets specifics and their core concepts (e.g., features, samples, labels). Table 1 summarizes the main steps of these transformation processes, along with commonly used techniques and tools.

Table 1. Summary of dataset-to-knowledge graph transformation approaches.

Contribution (name)	Dataset transformed (input format)	Transformation process (key steps)	Transformation type	Transformation tech / tools	KG (output format)	Ontology (abstraction level)	Validation
Rel2Graph [61]	Relational databases (Spider and Kaggle DBQA benchmarks)	Metadata analysis for PK, FK constraints; incremental conversion to Neo4j property graph; query mapping from SQL to Cypher	Automatic	Neo4j, ETL process, SQL-to-cypher mapping	Neo4j property graph	Schema-based, entity-relationship level	Execution Accuracy metric (EA) to check semantic query results
Transforming tabular datasets [1]	Tabular datasets (Comma-Separated Values, a file format for storing tabular data (CSV), Excel)	Concept prediction (column classification); relation extraction among columns; KG construction; optional enrichment	Semi-automatic	Machine learning (NER, classifiers, relation extraction)	RDF Format (KG Schema + Instances)	Schema-level abstraction	Precision, Recall, and F-Score metrics for KG accuracy
Meta2KG [2]	Metadata files (XML, key-value structure)	Preprocessing metadata; ontology development (BMO); embedding-based ontology matching; automatic population	Semi-automatic	Word embeddings (FastText), RDFLib, SPARQL	RDF Knowledge Graph (BMKG)	Ontology-driven abstraction (BMO)	Ontology matching and triple validation for correctness
DSKG [19]	Dataset metadata in scientific publications (RDF format)	Analyze dataset metadata; parse abstracts and citation contexts (146M publications); author name disambiguation; Enrich with links (ORCID, Wikidata, MAKG)	Semi-automatic	RDF, RDF2Vec, SPARQL endpoint, LOD, author disambiguation	RDF Knowledge Graph	Schema-level abstraction	High accuracy via linking to other data sources, FAIR principles compliance
Industry 4.0 KG for production lines [60]	Football production line dataset (realistic data, RGOM model)	Collect real data (football manufacturing line); map data to RGOM ontology; generate KG; validate with competency questions (SPARQL)	Automated	RGOM ontology, SPARQL, dublin core, SSN, time, OM, OWL, RDFS	RDF Knowledge Graph	Reference Generalized Ontological Model (RGOM)	FAIR evaluation tool (FAIR score: 79%), SPARQL query-based competency testing

Despite existing transformation methods, none explicitly address the main concepts and specifics of AI/ML-datasets components. However, they provide inspiration regarding the systematic steps of the transformation process, from ontology definition to instantiation and verification. In contrast, the KG4RSV framework prioritizes these AI/ML-datasets concepts, while providing a unified, semi-automated pipeline to transform tabular datasets into semantically rich and interoperable KG. Moreover, SHACL-based validation can only be applied after the dataset has been converted to RDF.

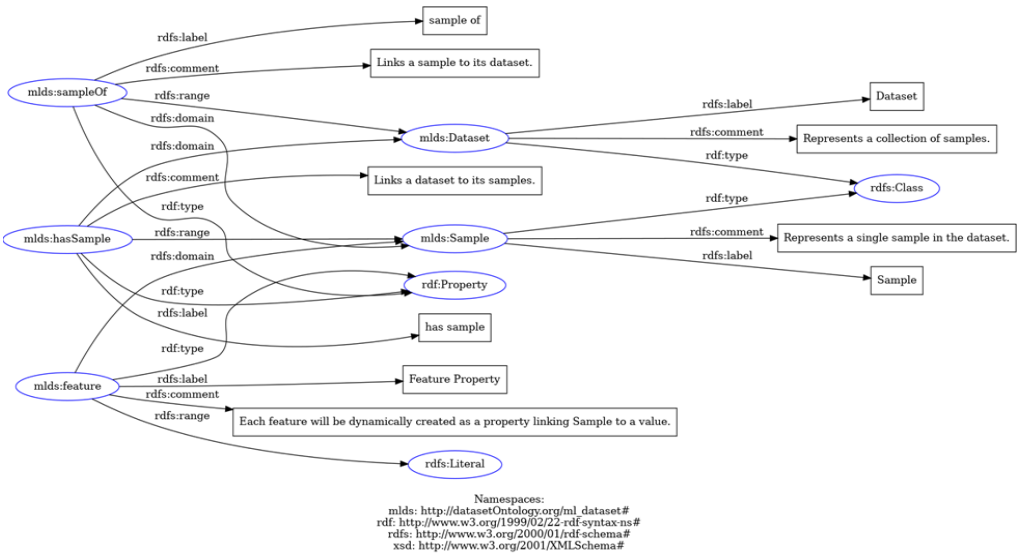

Figure 5. Dataset RDF schema.

Figure 5 represents the dataset RDF schema. It defines entities such as Dataset itself and "Sample" as classes, and "feature" as property, allowing each cell in a tabular dataset (for the first version of KG4RSV) to be represented as RDF triples.

Should be noted that, the dataset RDF schema defines the standard structures of AI/ML datasets, and is independent from any application domain ontology, allowing KG4RSV to generalize to other ML datasets. To ensure data integrity during this transformation, we assert that all information contained in the original dataset is explicitly represented in the resulting RDF graph. (see Equation 1) Formally, let D_{CSV} denotes the original tabular dataset and D_{RDF} the generated RDF graph. Then, for every record $r \in D_{\text{CSV}}$, there exists a corresponding set of RDF triples $T_r \subseteq D_{\text{RDF}}$ such that:

$$\forall r \in DCSV, \exists Tr \subseteq DRDF \text{ such that } sem(Tr) \vDash r \quad (1)$$

Where $sem(Tr) \vDash r$ denotes that the RDF triples T_r semantically encode all values and relationships of record *r*. This invariant ensures complete traceability and preservation of tabular content during transformation and is a foundational requirement for downstream validation and reasoning using SHACL.

The Algorithm 2, which is also implemented using Python script, presents the transformation of structured tabular data into an RDF knowledge graph by integrating schema information and semantic mappings. It begins by loading the dataset's RDFS schema to extract key classes such as *Dataset*, *Sample*, and *Feature*, along with their associated properties like *hasSample* and *sampleOf*.

Algorithm 2. CreateDatasetGraph

Input:
dataset RDFS, dataset CSV, datasetName, datasetPrefix, ontology map file, id feature, target feature (optional)
Output:
RDF Graph in Turtle format
1: Begin Algorithm
2: Load dataset RDFS into memory
3: Extract Classes: Dataset, Sample, Feature
4: Extract Properties: feature, hasSample, sampleOf
5: Read dataset CSV
6: Identify id feature column
7: Identify target feature column (if provided)
8: Identify all feature columns
9: Read ontology map file
10: Load vocabulary and ontology mapping data
11: Create RDF Graph:
12: Instantiate a Dataset instance:
13: datasetPrefix:datasetName a mlds:Dataset
14: for each row in dataset CSV do
15: Create a unique Sample instance (mlds:Sample ID)
16: Link Sample to Dataset using mlds:hasSample, mlds:sampleOf
17: for each feature column do
18: if feature exists in ontology map file then
19: Use mapped property from ontology
20: else
21: Create new property for the feature
22: end if
23: Assign feature value to the Sample instance
24: end for
25: end for
26: Serialize the RDF Graph:
27: Output the graph in Turtle format
28: (Optional) Save to an RDF database
29: End Algorithm

Next, it processes the dataset in Comma-Separated Values, a file format for storing tabular data (CSV) format, identifying the ID feature to uniquely reference each sample, the target feature if provided, and the remaining feature columns as data properties. The algorithm then reads an ontology mapping file to align dataset features with predefined vocabulary terms, ensuring semantic consistency. During graph construction, the algorithm creates an RDF Dataset instance and, for each CSV row, generates a *Sample* instance linked to the dataset. Each sample is enriched with properties corresponding to the feature values; if a feature matches a term in the ontology mapping, it uses the mapped property; otherwise, a new property is generated.

Finally, the RDF graph is serialized in Turtle format, optionally storing it in an RDF database for further use. This current transformation focuses on tabular datasets as a first step, while future work will extend the framework to handle additional data types such as time series, text, and images, enabling richer and more domain-specific KG representations.

3.5. Step 4: Validation of Dataset-KG against Requirements

The Dataset-KG Validation is an automated process for assessing the quality of RDF-based datasets using SHACL constraints. It begins by accepting two key inputs: a dataset-KG in RDF/Turtle format and the directory contains SHACL shapes files that encode the dataset quality requirements. The objective is to validate the dataset against these rules and generate a series of comprehensive reports.

The validation process unfolds in five major steps. First, it initializes the empty report files. Then, it iterates over each SHACL shape file, loading it and performing validation against the dataset using an RDF validator that supports both RDFS inference and advanced SHACL features like SPARQL constraints. The results of each validation run are saved in both RDF and human-readable formats. Next, the RDF validation report is parsed to extract detailed information about each violation, including the violating data (value), the node it was attached to (focus node), the SHACL rule it violated (source shape), and an explanatory message. These are compiled into a structured list and exported to a CSV file.

Finally, the Algorithm 3 supports dataset cleaning by filtering out invalid triples from the original RDF graph. This involves reloading the dataset and omitting any triple associated with a violation, thereby generating a new validated RDF graph. This cleaned graph can be serialized and stored for further use. The entire process has been automated using a SHACL engine (e.g.,

pySHACL or rdflib SHACL validator). It also supports different post-validation strategies, including: eliminating violating samples (applied), imputing them with default values, or retaining them while only reporting the violations, allows the Data Engineer to choose the most appropriate corrective action.

Algorithm 3. ValidateDatasetGraph

Input:
Dataset KG: RDF graph in Turtle format
SHACL Shapes Dir: Directory of SHACL shape files (Turtle format)
Output:
Validation Report RDF: RDF validation report
Validation Report TXT: Human-readable summary report
Violations CSV: CSV file of SHACL violations
Validated Dataset KG: RDF graph excluding invalid triples
1: Begin Algorithm
2: Initialize empty files for Validation Report RDF and Validation Report TXT
3: for each SHACL shape file in SHACL Shapes Dir do
4: Load the SHACL shape file
5: Validate Dataset KG against the SHACL shape using an RDF validator with:
6: RDFS inference
7: Advanced SHACL features (e.g., SPARQL constraints)
8: Store validation results:
9: Save RDF report to Validation Report RDF
10: Save summary to Validation Report TXT
11: end for
12: Parse the final RDF validation report:
13: for each sh:ValidationResult do
14: Extract sh:focusNode, sh:sourceShape, sh:value, sh:resultMessage
15: Add to list of violation entries
16: end for
17: Export Violations CSV with columns: focus node, path, value, message
18: Filter valid dataset triples:
19: Load Dataset KG
20: for each triple (s, p, o) in the graph do
21: if s or o matches a violation value then
22: Skip the triple
23: else
24: Add triple to new RDF graph
25: end if
26: end for
27: Serialize valid RDF graph as Validated Dataset KG
28: End Algorithm

3.6. Step 5: Reverse Transformation from Dataset-KG to Validated Dataset

After validation, the final clean Dataset-KG is transformed back into tabular format using the same Dataset RDF schema that governed the initial transformation. This guarantees structural and semantic consistency between the original dataset and its RDF representation.

To ensure semantic integrity in the reverse transformation, a subset equivalence condition is asserted (see Equations 2 and 3). Specifically, the reconstructed dataset D'_{csv} obtained from the validated RDF graph D_{rdf}, must satisfy:

$$f^{-1}(f(Dcsv)) \subseteq Dcsv \quad (2)$$

With the guarantee that for each row $r' \in Dcsv'$, there exists a corresponding row $r \in Dcsv$ such that:

$$equivalent(r', r) = True \quad (3)$$

Where:

- f denotes the forward transformation function from CSV to RDF.
- $F\text{-}1$ denotes the reverse transformation function from RDF to CSV.
- $Dcsv$ is the original dataset, and
- $D'_{CSV} = f^{-1}(f(D_{CSV}))$ the regenerated dataset.

This condition accounts for transformation scenarios where elimination strategies are applied-such as filtering out incomplete, irrelevant, or non-representable data-resulting in a partial but semantically faithful reconstruction. In cases where no elimination is performed and the process is fully lossless, the strict equality condition $D'csv = Dcsv$ remains valid.

This invariant confirms that the transformation process preserves the integrity of the information, which is essential for enabling accurate downstream analysis, reproducibility, and regulatory traceability.

3.7. Supporting Toolchain

To operationalize the KG4RSV framework, a modular Python-based toolchain was developed. This toolkit facilitates the creation of Requirements-KG, the transformation, validation, and round-trip conversion of structured tabular datasets into RDF-based KG and back.

Table 2 summarizes the key contribution steps, input/output formats, processing types, and supporting technologies used across the KG4RSV pipeline.

Table 2. Supporting toolchain for KG4RSV framework.

Contribution step	Input (format)	Processing type	Used technologies (libraries/tools)	Output (format)
Creation of: -Dataset RDFs -Requirements RDFs -Dataset requirements RDF -Reusable vocabularies	Natural language requirements and domain ontologies (CSV/TTL)	Manual	RDF editors, ontology selection, manual annotation	Dataset RDFs, Requirements RDFs, dataset requirements, reusable vocabularies (CSV/TTL)
Dataset2KG (dSet2kg.py)	Dataset files (CSV)+ Reusable vocabularies (TTL)	Semiautomatic	pandas, rdflib, csv, custom mapping rules	Dataset-KG RDF (Turtle/TTL)
Req2SHACL shapes (SHACL generator.py)	Requirements-KG (RDF)	Automatic	rdflib, SHACL generation scripts	SHACL Shapes (TTL)
Validator (shacl validator.py)	Dataset-KG +SHACL Shapes	Automatic	pySHACL, rdflib, custom reporting scripts	Validation Reports (TTL /TXT / JSON)
Dataset-KG2Dataset (Graph2DS.py)	Validated KG (RDF)	Semiautomatic	rdflib, pandas, rule-based converters	Tabular dataset (CSV)

This toolchain ensures reproducibility, supports extensibility, and enables seamless integration into real-world data preprocessing and requirements validation workflows. The code is publicly available at: https://github.com/ziad97nasri/KG4RSV.git for reuse and further enhancement.

4. Case Study

This section presents a case study to evaluate the proposed framework. It includes a dataset overview for vascular events diagnosis, the application of KG4RSV to the eICU dataset, the experimental results, and a statistical hypothesis test demonstrating the significant improvement achieved by applying KG4RSV.

4.1. ML for Vascular Events Diagnosis: Dataset Overview

Machine learning (ML) is increasingly employed to support the diagnosis and risk prediction of vascular events. Several public and private datasets have been developed for this purpose, including key clinical variables such as age, blood pressure, BMI, lifestyle factors (e.g., smoking, alcohol use), and biochemical markers (e.g., LDL, HDL, glucose) [3].

For this study, we selected a curated subset of the eICU Collaborative Research Database [25], a publicly available dataset. It comprises over 200,000 Intensive Care Unit (ICU) admissions from multiple hospitals in the United States. This dataset provides detailed Electronic Health Records (EHRs), including vital signs, laboratory results, medications, diagnoses, and clinical notes-making it well-suited for AI/ML research in healthcare.

To evaluate the KG4RSV framework, we constructed a tabular dataset containing 2,520 patient records and 203 features encompassing demographics, medical history, laboratory tests, vital signs, medications, and ventilation parameters. Following dataset construction, a feature selection step was applied to retain only the most clinically relevant indicators associated with cardiovascular events, reducing the dimensionality from 203 to 76 features.

The binary target variable *diagnoses* indicates the presence (1,258 cases) or absence (1,262 cases) of a cardiovascular event. The final dataset includes the following key feature categories:

- **Demographics:** age, gender, ethnicity, admission weight.
- **Vital Signs:** heart rate, SpO2, systolic pressure, temperature.
- **Cardiac Biomarkers:** troponin-I, BNP, CRP
- **Hematologic/Inflammatory Markers:** WBC, platelets, PT-INR.
- **Metabolic/Electrolytes:** glucose, creatinine, sodium, lactate, pH.
- **Lipid Profile:** HDL, LDL, triglycerides.
- **Ventilation Data:** FiO2, ventilator rate, tidal volume.
- **Hemodynamics (when available)**: cardiac output, Systemic Vascular Resistance (SVR), Pulmonary Artery Occlusion Pressure (PAOP).
- **Medical History and Diagnoses:** including vascular event indicators (used as target label).

This dataset was chosen due to its high dimensionality, clinical relevance, and inherent data quality challenges-such as missing values, outliers, and semantic inconsistencies. These characteristics make it an ideal candidate to demonstrate KG4RSV's capabilities in requirements-driven dataset validation and quality enhancement.

4.2. Applying KG4RSV to the eICU Dataset

This subsection presents the application of KG4RSV to the eICU dataset, illustrating how the proposed solution can be used for clinical data analysis and prediction.

4.2.1. Eliciting Dataset Quality Requirements

To emulate the roles of a requirements, engineer and domain expert, and to reduce the evaluation overhead of the KG4RSV framework on the eICU dataset, we leverage GPT-4o [32], an advanced language model. Using a few-shot prompting technique with a two-round elicitation template [58], we systematically define the dataset's quality requirements. The goal is to extract one meaningful rule per key data quality dimension including: Completeness, Consistency, Confidence (Credibility), Accuracy, and Inter-feature Relationships. In Round 1, ChatGPT-4o was prompted to provide a preliminary candidate rule per dimension. In Round 2, the results of round 1 is refined for clinical validity, logical coherence, and enforceability on structured data.

Table 3 presents the two-round results for each requirement, the involved attributes, and their meaning. The resulting requirements are annotated with an RQ-DDD-NN code (RQ: requirements, DDD: dimension, NN: number of requirements) and are as follows:

- **RQ-COMP-01 (Completeness):** "age", "systemicsystolic", and "systemicdiastolic" values must not be missing (These attributes are critical for clinical analysis and should always be recorded for reliable patient profiling).
- **RQ-CONS-02 (Consistency):** "systemicsystolic" must be strictly greater than "systemicdiastolic" (Ensures physiological plausibility based on normal blood pressure ranges).
- **RQ-CONF-03 (Confidence/Credibility):** "gender" must be either 'Male' or 'Female' (Restricts values to known valid categories to avoid mislabeling or coding errors).
- **RQ-ACCU-04 (Accuracy):** "glucose" levels must lie

between 50 and 500 mg/dL (Enforces clinically valid ranges and filters out measurement or input anomalies).

- **RQ-INTER-05 (Inter-feature Relationship):** If "glucose" strictly greater than 200, then "systemicmean" blood pressure should not be below 60 (Captures a domain-informed rule that links hyperglycemia with expected hemodynamic responses).

This LLM-based elicitation requirements method serves as a starting point for validation or further refinement by domain experts and illustrates the potential of integrating LLMs into the requirements elicitation process, particularly in some complicated fields such as medicine.

Table 3. Requirements elicitation process using ChatGPT 4o.

Draft requirement: "Act as a data quality analyst and suggest one draft requirement per dimension (completeness, consistency, confidence, accuracy, inter-feature relations) for the eICU dataset based on the following dataset features."			
Refined requirement: "Refine the initial requirements to ensure clinical validity, logical consistency, and verifiability."			
Quality dimension	**Round 1: draft requirement**	**Round 2 refined requirement:**	**Involved attributes**
Complete-ness	age and systemicsystolic should not be null.	age, systemic-systolic, and systemicdiastolic values must not be missing.	age, systemic-systolic, systemicdiastolic
Consistency	Systolic BP must be higher than diastolic BP.	Systemicsystolic must be strictly greater than systemicdiastolic.	systemicsystolic, systemicdiastolic
Confidence (credibility)	Gende should be either male or female.	gender must be either 'Male' or 'Female'.	gender
Accuracy	Glucose values should be within a normal medical range.	glucose levels must lie between 50 and 500 mg/dL.	glucose
Inter-feature relation-ship	High glucose should correspond to a stable hemodynamic response.	If glucose >200, then systemicmean blood pressure should not be below 60.	glucose,sys-temicmean

4.2.2. Step 1: Constructing the Requirements Knowledge Graph (Requirements-KG)

In this step, the quality requirements elicited previously are formally represented as RDF instances for our Requirements_RDF schema (see Figure 3). Each requirement is encoded as an instance of req:Requirement, and includes the following components:

- *req:targetClass*: identifies the class of entities the requirement applies to, typically for example *mlds:Sample.*
- *req:category* : specifies the quality dimension the requirement addresses, such as Accuracy, Completeness, or Consistency.
- *req:constraintExpression* : contains a SPARQL query that identifies samples violating the requirement.

For example, the accuracy requirement: "RQ-ACCU-04: *glucose level should be between 50 and 500 mg/dl*". This constraint ensures that glucose measurements are within medically plausible ranges and not outliers or incorrectly recorded values. The requirement is expressed in RDF/TTL as illustrated in Figure 6.

```
@prefix loinc: <http://loinc.org/> .
@prefix m3lite: <http://purl.org/iot/vocab/m3-
lite#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-
syntax-ns#> . @prefix req:
<http://example.org/req#> .
@prefix snomed:
<http://purl.bioontology.org/ontology/ SNOMEDCT/> .
# RQ-ACCU-04: glucose level should be between 50
and 500 mg/
dL
req:RQ-ACCU-04 a req:Requirement ; req:targetClass
mlds:Sample ; req:constraintExpression """
SELECT ?this WHERE {
?this loinc:2345-7 ?g .
FILTER (?g != "NaN" && (?g < 50 || ?g > 500))
}
""" ; req:category "Accuracy" ;
```

Figure 6. Accuracy requirement as RDF instance.

4.2.3. Step 2: Generating SHACL Shapes

To support automated validation of dataset quality, we implemented a transformation step that converts each SPARQL-based requirement into a corresponding SHACL shape. This was accomplished automatically using our custom script Req2SHACL_shapes.py, which processes the Requirements-KG and generates SHACL NodeShape instances that preserve the constraint logic and metadata (e.g., message, severity, and target class).

For example, the accuracy requirement: "RQ-ACCU-04: Glucose level should be between 50 and 500 mg/dl" is automatically converted into SHACL representation shown in Figure 7.

```
@prefix loinc: <http://loinc.org/> .
@prefix m3lite: <http://purl.org/iot/vocab/m3-
lite#> .
@prefix mlds:
<http://datasetOntology.org/ml_dataset#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-
syntax-ns#> .
@prefix req: <http://example.org/req#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix snomed:
<http://purl.bioontology.org/ontology/SNOMEDCT/> .
req:Shape_RQ-ACCU-04 a sh:NodeShape ; sh:sparql [ a
sh:SPARQLConstraint ; sh:message "Violation of RQ-
ACCU-04" ; sh:select """\r
SELECT ?this WHERE {\r
?this loinc:2345-7 ?g .\r
FILTER (?g != "NaN" && (?g < 50 || ?g > 500))\r
}\r
""" ; sh:severity sh:Violation ] ;
sh:targetClass mlds:Sample .
```

Figure 7. Automatically generated SHACL shape.

4.2.4. Step 3: Constructing the eICU Dataset-KG

In this step, the original tabular dataset is transformed into an RDF knowledge graph using the Dataset2KG.py script. The transformation based on the Dataset RDF schema (Figure 5), which defines core concepts and relationships to semantically represent the dataset concepts:

- *mlds:Dataset*: Encodes the entire collection of samples.
- *mlds:Sample: Represents an individual patient record.*
- *mlds:hasSample: Associates a dataset with its*

samples.

- *mlds:feature: A generic property extended into specific clinical or demographic features using rdfs:subPropertyOf.*

Each row in the tabular data is mapped to a unique mlds:Sample, and each column (i.e., feature) is instantiated as a subproperty of mlds:feature. Before creating these instances, each feature is mapped to a Linked Open Vocabulary when available, using a predefined ontology mapping file created in KG4RSV-preprocess phase. This approach ensures semantic enrichment and interoperability with existing biomedical ontologies. In this study, three prominent ontologies used:

1) m3-lite [7], a lightweight ontology for IoT and healthcare sensor measurements, covering features like m3lite:Temperature and m3lite:Calcium;
2) SNOMED CT [43], a comprehensive clinical terminology for representing clinical concepts, used for features like snomed:364075005 (Heart rate) and snomed:271650006 (Diastolic blood pressure);
3) LOINC [47], an international standard for laboratory and clinical observations, covering laboratory results such as loinc:2345-7 (Glucose) and loinc:19996-8 (FiO2).

This semantic alignment facilitates data sharing, reuse, and reasoning, as illustrated in Figure 8.

```
loinc:2345-7    rdfs:subPropertyOf    mlds:feature    .
m3lite:Calcium  rdfs:subPropertyOf    mlds:feature    .
eicu:ALT__SGPT rdfs:subPropertyOf mlds:feature  .
```

Figure 8. Example of using linked open vocabulary.

```
@prefix eicu:
<http://datasetOntology.org/ml_dataset/e-ICU#> .
@prefix loinc: <http://loinc.org/> .
@prefix m3lite: <http://purl.org/iot/vocab/m3-lite#>
.
@prefix mlds:
<http://datasetOntology.org/ml_dataset#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-
syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
.
@prefix snomed:
<http://purl.bioontology.org/ontology/SNOMEDCT /> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
eicu:eICU_dataset a mlds:Dataset ; mlds:hasSample
eicu:1008970 .
eicu:1008970 a mlds:Sample ; eicu:vitalaperiodicid
"162167707.5"^^xsd:float ; eicu:vitalperiodicid
"525643235.3"^^xsd:float ; eicu:wardid 434 ;
loinc:19996-8 "NaN"^^xsd:string ; loinc:2345-7
"155.8"^^xsd:float ; snomed:271650006
"NaN"^^xsd:string ; snomed:364075005
"88.40998487"^^xsd:float ; snomed:442476006
"98.3853211"^^xsd:float ; snomed:72313002
"NaN"^^xsd:string ; m3lite:Calcium "8.8"^^xsd:float ;
m3lite:Potassium "4.22"^^xsd:float ;
m3lite:Temperature "NaN"^^xsd:string ; mlds:sampleOf
eicu:eICU_dataset .
```

Figure 9. Sample instance excerpt from the eICU-RDF dataset.

Figure 9 represents an instantiation example of the eICU dataset and one sample values. This modeling strategy results in a structured and semantically rich DatasetKG containing over 200000 RDF triples. For instance, Figure 10 represents a few parts of one sample graph, which preserves the full feature set while enabling semantic interconnection and the possibility of SHACL constraint validation and reasoning in subsequent steps of the pipeline.

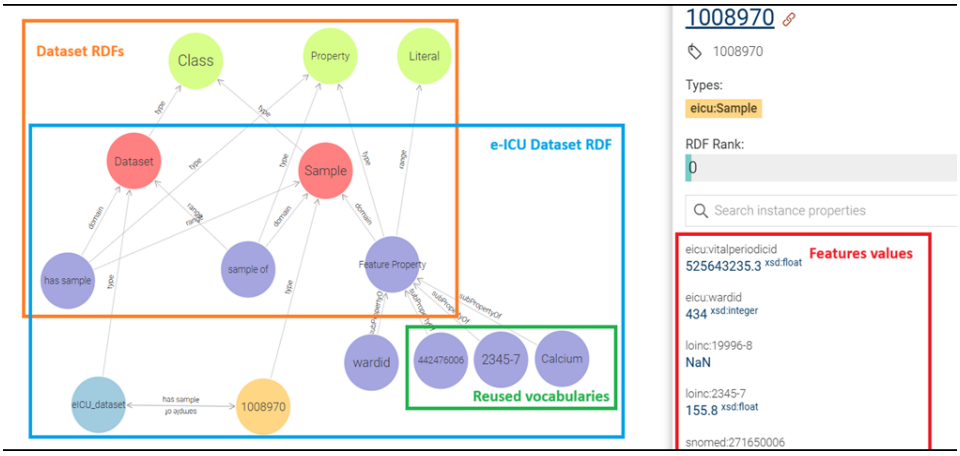

Figure 10. Sample instance excerpt from the eICU-RDF graph.

4.2.5. Step 4: Validation of eICU-Dataset-KG

The validation process is performed using the Validator.py script, which automatically checks the compliance of the Dataset Knowledge Graph (Dataset-KG.ttl) against the set of SHACL constraint files located in the SHACL Shapes/ directory.

Inputs:

- eICU Dataset-KG.ttl: the RDF representation of the original dataset.
- SHACL Shapes: a directory containing SHACL shape files automatically generated from the Requirements-KG;

Outputs:

- Validation Report RDF.ttl: an RDF file containing the complete SHACL validation report;
- Validation Report TXT.txt: a human-readable summary of all validation violations. Table 4 illustrates a part of the validation report resulting from validation of eICU Dataset-KG.
- Violations.csv: a structured CSV file listing all violations with details such as the focus node, path, value, and result message;

- Validated eICU Dataset-KG.ttl: a cleaned RDF graph that excludes triples involving invalid or incomplete data entries.

The script iteratively applies all SHACL shapes to the eICU Dataset-KG, compiles violations, and filters out any invalid triples based on constraint violations as illustrated in Figure 11. The resulting Validated Dataset-KG.ttl is used as base for creating the violation CSV file, which also serves as a reliable foundation for reconstructing a consistent and quality-assured eICU valid Dataset-KG.

```
Validating against SHACL file RQ-CONS-02.ttl
Validation Report
Conforms: True
Validating against SHACL file RQ-ACCU-04.ttl
Validation Report3. Conforms: False
Results (11):
Constraint Violation in SPARQLConstraintComponent:
Severity: sh:Violation
Source Shape: req:Shape_RQ-ACCU-04
Focus Node: eicu:1116712
Value Node: eicu:1116712
Source Constraint: [ rdf:type sh:SPARQLConstraint
;sh:message Literal("Violation of RQ-ACCU-04") ;
sh:select Literal("
SELECT ?this WHERE {
?this loinc:2345-7 ?g .
FILTER (?g != "NaN" && (?g < 50 || ?g > 500))
}
") ; sh:severity sh:Violation ]
Message: Violation of RQ-ACCU-04
Constraint Violation in SPARQLConstraintComponent:
Severity: sh:Violation
Source Shape: req:Shape_RQ-ACCU-04
Focus Node: eicu:3136218
Value Node: eicu:3136218
Source Constraint: [ rdf:type sh:SPARQLConstraint ;
sh:message Literal("Violation of RQ-ACCU-04") ;
sh:select Literal("
SELECT ?this WHERE {
?this loinc:2345-7 ?g .
FILTER (?g != "NaN" && (?g < 50 || ?g > 500)) }") ;
sh:severity sh:Violation ]
28. Message: Violation of RQ-ACCU-04
```

```
Validating against SHACL file RQ-INTER-05.ttl
Validation Report3. Conforms: False
Results (4):
Constraint Violation in SPARQLConstraintComponent:
Severity: sh:Violation
Source Shape: req:Shape_RQ-INTER-05
Focus Node: eicu:2558121
Value Node: eicu:2558121
Source Constraint: [ rdf:type sh:SPARQLConstraint
;sh:message Literal("Violation of RQ-INTER-05") ;
sh:select Literal("
SELECT ?this WHERE {
?this loinc:2345-7 ?g ; eicu:systemicmean ?bpmean
.
FILTER (?g > 200 && ?bpmean < 60)
}
") ; sh:severity sh:Violation ]
Message: Violation of RQ-INTER-05
Constraint Violation in SPARQLConstraintComponent:
Severity: sh:Violation
Source Shape: req:Shape_RQ-INTER-05
Focus Node: eicu:3122044
Value Node: eicu:3122044
Source Constraint: [ rdf:type sh:SPARQLConstraint
sh:message Literal("Violation of RQ-INTER-05") ;
sh:select Literal("
SELECT ?this WHERE {
?this loinc:2345-7 ?g ; eicu:systemicmean ?bpmean
.
FILTER (?g > 200 && ?bpmean < 60)
}
") ; sh:severity sh:Violation ]
28. Message: Violation of RQ-INTER-05
```

Figure 11. Part of validation report of eICU dataset-KG.

4.2.6. Step 5: Reconstruction of the Validated eICU Dataset

Following the validation process, the verified Dataset Knowledge Graph (eICU_valid_Dataset-KG.ttl) was transformed back into a structured tabular format (CSV) using the Dataset-KG2dataset.py script. This transformation preserved all semantic integrity and completeness constraints defined in the Dataset_RDF schema, which originally guided the mapping from tabular data to the graph representation. As a result, no data was lost during the round-trip transformation.

The resulting file, *validated_dataset.csv*, is the final dataset aligned with the defined quality requirements and served as the input for training the machine learning models. In the next section, we demonstrate the effectiveness of the KG4RSV method by performing a comparative analysis between the original dataset and the validated dataset.

4.3. Experimental Results

To demonstrate the impact of the KG4RSV method on data quality and model performance, we evaluated three aspects: dataset statistics, machine learning metrics, and confusion matrices.

The experiments were conducted using a hybrid hardware configuration. Basic ML models (e.g. SVM, KNN, Logistic Regression, Random Forest, and XGBoost) were executed on a CPU environment (Intel Core i5, 13th Gen, 16GB RAM), while deep learning models were trained on a dedicated GPU (NVIDIA RTX 3050 with 6GB VRAM, expandable to 12GB shared memory).

Prior to model training, data preprocessing steps included missing value removal (threshold > 90%), label encoding for categorical variables, mean imputation for numerical features, and feature scaling via standard normalization. The cleaned dataset was split into training and testing subsets using a stratified 80/20 ratio.

Hyperparameters were configured as follows: SVM used an RBF kernel; KNN with k=5; Logistic Regression with a maximum of 1000 iterations; Random Forest with 100 estimators; and XGBoost with use_label_encoder = False and eval_metric = ‘logloss’.

For the deep learning model, a fully connected neural network was employed with three hidden layers (32, 16, 8 units respectively), ReLU activation, L2 regularization, and dropout layers to mitigate overfitting. The model was compiled using the Nadam optimizer with a learning rate of 1*10-4, trained over 200 epochs with a batch size of 32, and monitored via validation loss.

Evaluation metrics included accuracy, precision, recall, F1-score, and a normalized confusion matrix. An ensemble learning strategy was also applied via majority voting across all basic ML models to simulate a federated decision system.

It is important to note that, due to the random strategy of data splitting, repeated runs of the same model on the same dataset may yield slightly different results. The performance metrics reported here correspond to the most accurate and optimal outcomes observed during the dataset evaluation process.

4.3.1. Dataset Statistics Comparison

The Original dataset (Raw) initially contained 2521 samples, including 6 missing values and 21 constraint violations according to the specified requirements in step 1. After applying the KG4RSV validation pipeline, the resulting validated dataset had 2500 samples, with zero missing values and no violations (Figure 12).

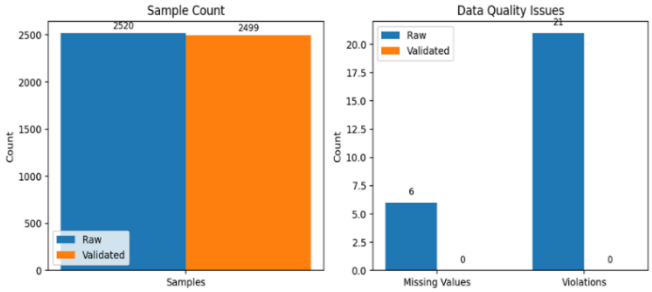

Figure 12. Datasets statistics before and after data validation.

This reduction confirms the method's effectiveness in eliminating semantically inconsistent or incomplete records while retaining the majority of the dataset-an essential balance between data quality and volume must be considered when defining the requirements.

4.3.2. Machine Learning Performance Metrics

To ensure a comprehensive and unbiased evaluation of the proposed approach, six representative ML models were selected: K-Nearest Neighbors (KNN), Random Forest (RF), Logistic Regression (LR), XGBoost, Support Vector Machine (SVM), and Artificial Neural Network (ANN). These models were chosen because they cover a wide range of learning paradigms, complexities, and decision-boundary behaviors, enabling a robust assessment of the method's generalizability [38].

In addition, the first five models (i.e., KNN, Random Forest, Logistic Regression, XGBoost, and SVM) were combined into a federated ensemble using majority voting to leverage their algorithmic diversity and reduce model-specific bias.

Figure 13 presents the evaluation results of these models using key AI/ML metrics, including accuracy, precision, recall, and F1-score. For example, the federated model improved its F1-score from 0.74 (raw dataset) to 0.77 (validated dataset), while recall increased from 0.82 to 0.86. Similarly, the SVM model's F1-score increased from 0.7517 to 0.7539, and its recall improved from 0.8532 to 0.864, indicating better sensitivity in detecting true cases.

These gains show how validated datasets improve not just consistency of data, but also the overall model performance. XGBoost is the only model that showed a decrease in performance compared to the rest of the models, but it remains small compared to the significant improvements witnessed by the remaining models.

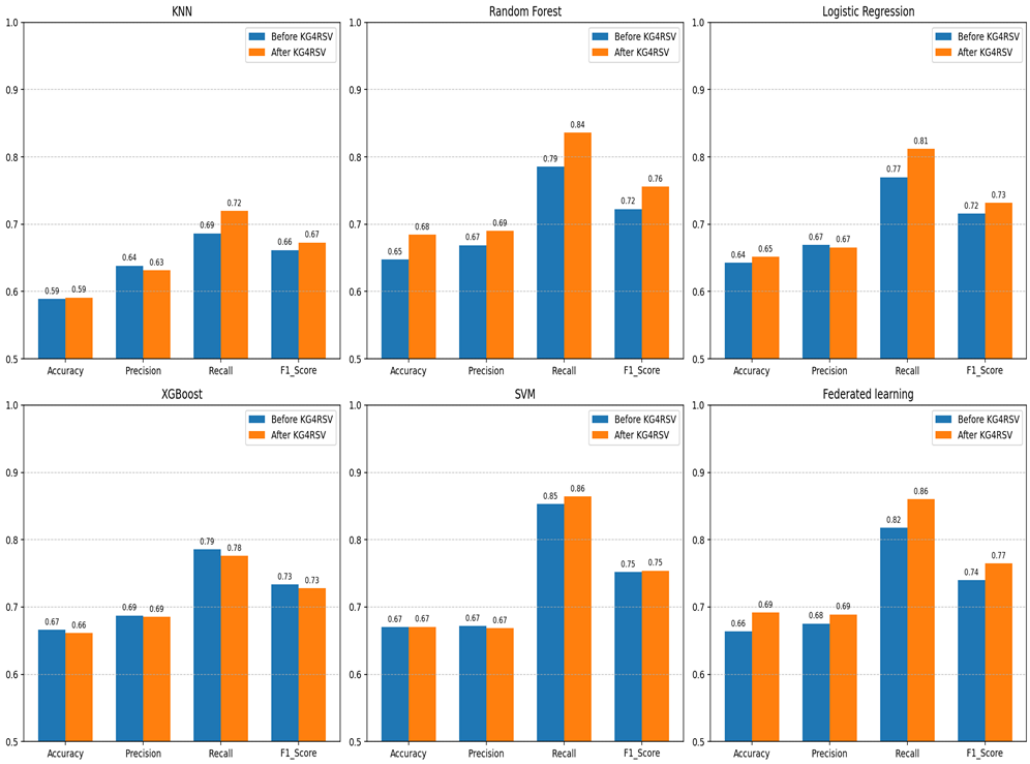

Figure 13. ML-Models' performance before and after data validation.

Figure 14 compares confusion matrices (in %) before and after applying KG4RSV on the eICU dataset. It reveals that optimization consistently improves true positive rates (recall) and slightly reduces false negatives, especially in SVM and Federated Learning. These gains explain the improved performance seen in the earlier recall, precision, F1, and accuracy bar chart analysis.

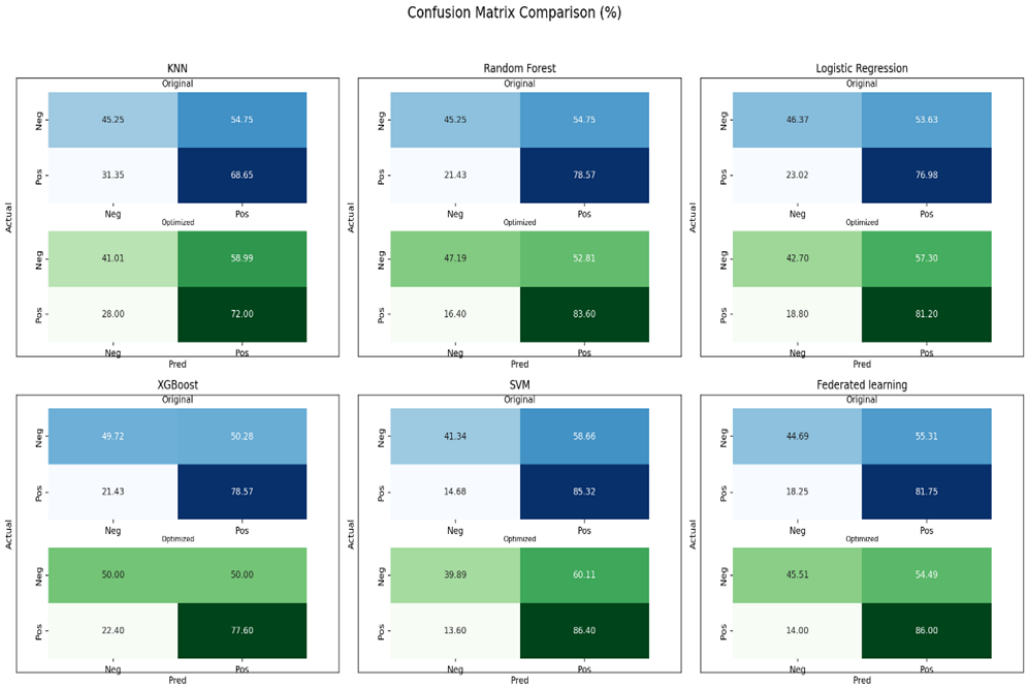

Figure 14. Comparison of confusion matrices.

4.3.3. Performance and Confusion Matrix Analyses of ANN Model (Deep Learning)

The confusion matrices in Figure 15 highlight a notable improvement in classification performance for the ANN model after applying the KG4RSV validation process.

In the Raw Dataset, 43.02% of the actual negative cases (class 0) were correctly classified as negative (True Negatives), while 56.98% were misclassified as positive (False Positives). For the positive class (class 1), 82.14% were correctly detected (True Positives), while 17.86% were missed (False Negatives).

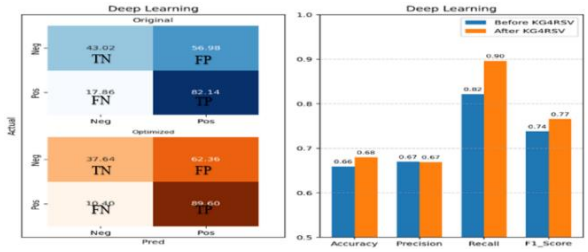

Figure 15. Performance and Confusion Matrices for ANN Model.

In the Raw Dataset, 43.02% of the actual negative cases (class 0) were correctly classified as negative (True Negatives), while 56.98% were misclassified as positive (False Positives). For the positive class (class 1), 82.14% were correctly detected (True Positives), while 17.86% were missed (False Negatives).

After applying KG4RSV (Validated Dataset in right panel), the True Negative rate decreased to 37.64%, but this was counterbalanced by a reduction in False Negatives. The True Positive rate improved significantly to 89.60%, with False Negatives dropping to 10.40%.

This indicates a relative gain of over 7% in True Positive rate and a reduction of over 7% in False Negatives, which is highly beneficial in clinical contexts, where minimizing undetected critical cases (FN) is more important than slightly increasing False Positives.

The trade-off-fewer correct negative predictions in exchange for significantly improved detection of true positives-aligns with the objectives of critical healthcare applications where sensitivity is paramount.

Concerning the model generalization, Figure 16 shows that training on the optimized dataset leads to higher validation accuracy and lower validation loss over epochs compared to the raw dataset. The orange curve demonstrates better generalization and learning efficiency. This supports the earlier findings, confirming that data optimization enhances model performance and convergence stability.

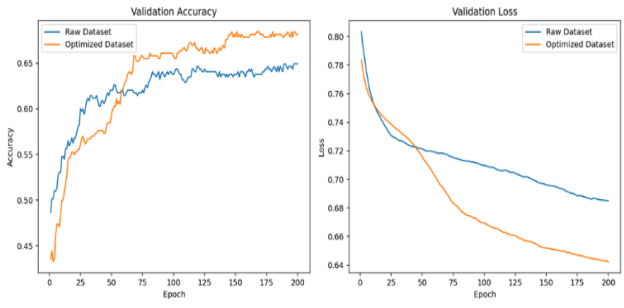

Figure 16. Learning curves of the ANN model.

The validated dataset shows clear improvements in classification performance, consistency, and interpretability of training results. The evaluation outcomes illustrate that KG4RSV not only enhances data integrity (zero violations, no missing values) but also strengthens ML model performance in terms of

both recall and balanced accuracy. The SHACL validation also revealed hidden quality issues that were not obvious during data exploration. This validates its suitability for use in safe-critical AI/ML systems development pipelines.

4.4. Statistical Hypothesis Test

In this section, we apply statistical hypothesis testing [18] to:

1) demonstrate that the KG4RSV pipeline produces a dataset with significantly higher quality,
2) validate that the KG4RSV-cleaned dataset yields statistically significant improvements in AI/ML model performance.

The evaluation method performed on two stages: a dataset-level proportion test and a model-level paired t-test.

4.4.1. Dataset-Level Analysis: Data Quality

Due to the binary nature of the violation checks, the exact binomial test is the most appropriate statistical method [15]. Moreover, unlike large-sample approximations (such as the z-test), the exact binomial test does not rely on normality assumptions and remains valid even when the sample size is small or when the observed proportions are close to 0.

The validated dataset (D2) is obtained directly from the raw dataset (D1) by deterministically removing all rows that violate the KG4RSV requirement constraints. Therefore, the two datasets are not independent; D2 is a filtered version of D1. A summary of both datasets is presented in Table 4.

Table 4. Violation statistics before and after KG4RSV validation.

Dataset	Total Samples (N)	Violations (V)	Violation Rate (P = V/N)
Raw (D1)	2521	21	$P_1 = 21/2521 \approx 0.00833$
Validated (D2)	2500	0	$P_2 = 0/2500 = 0$

- **Exact Binomial Test on the Raw Dataset**

To evaluate whether the raw dataset could be considered fully compliant (i.e., true violation rate p=0), we applied a one-sample exact binomial test (see Equations 4 and 5).

In the raw dataset (D1), we observed 21 violations out of 2521 samples, giving Equation (4):

$$p1 = \frac{21}{2521} \approx 0.00833(0.833\%) \tag{4}$$

Under the null hypothesis H_0:p=0, the probability of observing any violation is (See Equation (5)):

$$P(X \geq 1 \mid n = 2521, p = 0) = 0 \tag{5}$$

Thus, the one-sided exact p-value is effectively zero, which strongly rejects the hypothesis that the raw dataset satisfies the specified constraints.

- **Post-Cleaning Violation Rate Analysis**

After removing the violating rows, the validated dataset (D2) contains 0 violations out of 2500 samples. When zero events are observed, the Clopper-Pearson exact confidence interval reduces to a one-sided upper bound (see Equations 6, and 7):

$$CI(P2) = [0,\ 1 - \alpha^{1/N_2}] \tag{6}$$

Using a 95% confidence level (α=0.05):

$$Upper\ bound = 1 - 0.05^{1/2500} \approx 0.001198. \tag{7}$$

Thus, the 95% exact CI for the violation rate after cleaning is (see Equations 8):

$$[0, |0.001198] or [0\%, |0.12\%] \tag{8}$$

This means that even under the most conservative assumptions, the true violation rate in D2 is below 0.12%.

- **Interpretation**

Together, the exact binomial test on D1 and the exact confidence-interval analysis of D2 provide strong statistical evidence that KG4RSV greatly improves dataset quality.

The violation rate decreases from a statistically unacceptable level in the raw dataset to a fully compliant level in the validated dataset, with a very narrow uncertainty bound.

4.4.2. Model-Level Analysis: Performance Improvement

In order to determine whether the data-quality improvements lead to statistically significant gains in ML model performance, we employ a paired-sample t-test, since we compare the performance of the same models under both datasets.

It is important to emphasize that each dataset (raw and validated) is split only once. In the first round, all models use the same training split of the raw dataset, and in the second round, they use the same training split of the validated dataset. In addition, all models are trained under identical conditions (i.e., same model configurations and training environment). This ensures a fair and consistent paired comparison between both datasets.

Since cross-validation resampling was not repeated, the seven ML models are treated as paired experimental conditions to evaluate the effect of KG4RSV on model-level F1-score. The results should therefore be interpreted as model-level evidence rather than resampling-based evidence.

F1-Score Analysis (Across seven ML-Models)

We selected the F1-Score as the key performance indicator (KPI) as it provides a robust metric for classification tasks, balancing Precision and Recall. Table 5 summarizes the experimental results.

Table 5. Comparative F1-scores (raw vs. validated).

Model	F1-Score (Validated)	F1-Score (Raw)	Difference (d)
SVM	0.7527	0.7517	0.0010
KNN	0.6718	0.6616	0.0102
Logistic Regression	0.7360	0.7259	0.0101
Random Forest	0.7607	0.7226	0.0381
XGBoost	0.7320	0.7333	-0.0013
Federated	0.7787	0.7497	0.0290
Deep Learning	0.7730	0.7413	0.0317

- **Hypothesis Test**

Null Hypothesis H_0 (Equation 9)**:** There is no significant difference in the F1-Score before and after applying KG4RSV.

$$\mu_{validated} - \mu_{original} = 0 \quad (9)$$

Alternative Hypothesis H_a (Equation 10)**:** The mean F1-Score of the validated dataset is significantly greater than that of the raw dataset.

$$\mu_{validated} - \mu_{original} > 0 \quad (10)$$

- **Calculation *T*-Statistic**

$$\bar{d} = \frac{\sum d_i}{n} = \frac{0.1188}{7} \approx 0.017; \quad (11)$$

$$S_d = \sqrt{\frac{\sum (d_i - \bar{d})^2}{n-1}} = 0.0158; \quad (12)$$

$$SE_{\bar{d}} = \frac{s_d}{\sqrt{n}} = \frac{0.0158}{\sqrt{7}} \approx 0.00597; \quad (13)$$

$$t = \frac{\bar{d}}{SE_{\bar{d}}} = \frac{0.017}{0.00597} \approx 2.8476 \quad (14)$$

- $\bar{d}$ is the mean of the differences.
- s_d is the standard deviation of the differences.
- n is the number of pairs (models).

Using a paired *t*-test with degree of freedom: $df = n - 1 = 6$

- *T*-Statistic: 2.8476.
- One-Tailed P-value: 0.015

As a conclusion, since *P*-value < 0.05, we reject H_0. There is strong statistical evidence that the KG-based data validation method significantly improves the general F1-Score performance across the tested ML-models.

ANN F1_score Analysis (Across 10 Runs)

In this case, we assess whether the dataset validation significantly improves the performance of the ANN model using the F1 score as evaluation metric. A paired-sample t-test is applied to evaluate statistical significance. The model training repeated 10 times on the same dataset split for the raw dataset in the first round, and the same procedure is applied in the second round using the validated dataset. The paired t-test is appropriate because each run of the model on the validated dataset corresponds directly to the same run on the raw dataset, forming paired observations (see Table 6).

Table 6. The best validation F1 scores for 10 runs.

Run	Raw dataset	Validated dataset	Difference ($d_i = y_i - x_i$)
1	0.7289	0.7520	0.0231
2	0.7236	0.7590	0.0354
3	0.7275	0.7478	0.0203
4	0.7389	0.7360	-0.0029
5	0.7280	0.7398	0.0118
6	0.7280	0.7431	0.0151
7	0.7285	0.7244	-0.0041
8	0.7092	0.7582	0.0490
9	0.7160	0.7292	0.0132
10	0.7178	0.7340	0.0162

- The mean difference: $\bar{d} = \frac{1}{n}\sum_{i=1}^{n} d_i \approx 0.0177$
- The standard deviation of differences:

$$s_d = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(d_i - \bar{d})^2} \approx 0.0170 \quad (15)$$

- The *t*-statistic:

$$t = \frac{\bar{d}}{S_d/\sqrt{n}} \approx 3.29 \quad (16)$$

- Degrees of freedom is: $df = n - 1 = 9$.
- The p-value is obtained from the *t*-distribution, so Two-tailed *p*-value: $p \approx 0.009$

- **Interpretation**

- The mean difference in F1 score between optimized and original datasets ($\bar{d} \approx 0.0177$), indicating a moderate improvement.
- Since $p < 0.05$, the improvement is statistically significant.

As a conclusion, the optimization method significantly improved the model's validation F1_score compared to the original dataset.

5. Discussion

This section examines the broader implications of our KG4RSV approach in the context of RE4AI/ML dataset. We organize the discussion around four key axes: evaluation findings, alignment with research goals, practical aspects (reusability and adaptability), and relevance to RE4AI/ML.

5.1. Findings

The KG4RSV evaluation on eICU Dataset demonstrated its effectiveness in improving key data quality requirements, particularly those related to completeness, consistency, and conformance. The SHACL validation shapes, derived automatically from the Requirements KG, successfully flagged violations (missing values, out-of-range values, and inter-feature inconsistencies). These results provide experimental evidence of the practical feasibility of formalizing and automatically validating dataset requirements using a KG-based approach.

5.2. Interpretation of Results in the Context of Research Objectives

In light of our research objectives:

1) The proposed KG4RSV method successfully manages the dataset quality requirements despite heterogeneous data types (strings, numbers, categorical) and sources by representing data and requirements in separate KGs. This abstraction enables consistent validation regardless of the feature nature (numerical, categorical, or textual), providing flexibility and scalability across diverse datasets.
2) Enforcing dataset quality requirements, such as completeness and consistency, are formalized and validated using SPARQL queries and SHACL shapes. After applying corrections, the number of data issues dropped (e.g., nineteen violations were resolved), leading to improved model performance. Notably, the validated dataset achieved higher recall and F1-scores, and confusion matrix results showed fewer misclassifications, proving the impact of quality-driven validation.
3) Supporting specification, verification, and validation of dataset requirements. The formal and modular specification of requirements-through identifiers like RQ-COMP-04-enabled clear documentation, machine-verifiable checks, and traceable validation reports, thereby supporting rigorous specification, verification, and validation activities.

These results confirm that KG4RSV provides a structured, transparent, and reusable foundation for integrating Requirements Engineering principles into ML dataset management. The potential gains are expected to be even more pronounced with large-scale datasets, where data governance and semantic consistency are critical.

5.3. Reusability and Adaptability

A key advantage of our approach is its modular design. The Requirements RDF schema is independent of any specific dataset structure domain, so this makes KG4RSV highly adaptable to other domains, such as biotechnology, finance, or agriculture, etc. Early experiments suggest that both the Requirements KG and validation process can be adapted to new domains with limited manual effort.

5.4. Implications for the RE4AI/ML Community

The contribution aligns with the emerging shift in RE for AI/ML, in which quality requirements are formalized and encoded in a manner that supports interpretability and automated validation, going beyond informal documentation. Our findings promote a paradigm shift where dataset preparation includes not only statistical preprocessing but also ensuring alignment with domain-specific requirements. This opens new avenues for collaboration between RE specialists, data scientists, and domain experts in regulated fields such as healthcare.

5.5. Threats to Validity and Limitations

In this section, we reflect on the potential threats to the validity of the KG4RSV framework and its evaluation. These limitations cover both internal and external dimensions and are critical to consider when interpreting the reported results and assessing the generalizability of the proposed approach.

(Internal validity) Several internal factors may have influenced the outcomes of the evaluation. First, a slight decrease in XGBoost's classification accuracy was observed following the semantic data validation process. This reduction could be attributed to shifts in class distribution or feature variance resulting from the filtering of invalid records. Such effects suggest that the framework's impact may be model sensitive and that certain models are more sensitive to changes in data composition.

Second, performance variability was observed across different training runs due to the stochastic nature of ML workflows (e.g., random initialization, data shuffling, and stratified splitting), which poses a threat to the reproducibility and stability of the results. Moreover, the quality of the semantic validation process closely depends on the expressiveness and completeness of the domain-specific requirements provided by experts. Incomplete or ambiguously defined constraints at the early specification stage may limit the effectiveness of the data validation and lead to inconsistent or suboptimal outcomes.

(External validity) The generalizability of the KG4RSV framework beyond the current experimental setting remains a key limitation. The relatively modest sample size (2,520 instances) limits the ability to assess model robustness under broader or more heterogeneous data conditions, particularly for deep learning models that typically benefit from larger volumes of data. Furthermore, the framework's validation was conducted within a healthcare context that benefits from well-established ontologies and rich domain knowledge. Its applicability to domains with unstructured, noisy, or data without a predefined structure (schema), or where expert-defined semantic constraints are less formalized, is yet to be explored. Finally, while the underlying hardware setup (a standard mid-range CPU/GPU setup) was sufficient for the experiments conducted, it may not reflect the performance characteristics of the system in large-scale production environments, particularly for Dataset-KG validation.

Taken together, these limitations outline directions for future research. In particular, extending the framework to more diverse datasets and domains, automating parts of the requirement specification process, and evaluating its performance under varying

computational constraints would help to strengthen the validity and applicability of the approach.

6. Related Work

To position our contribution within existing research, we summarize recent work on RE for AI and data-driven systems, with a focus on dataset quality, validation, and KG-based approaches.

Table 7 and 8 illustrate different approaches to dataset specification and validation across multiple domains. Haar *et al*. [55] utilized knowledge graphs to validate textual security requirements, achieving near-perfect accuracy in detecting inconsistencies. Jahi´c *et al*. [28] focused on improving handwritten digit classification by augmented handwritten digit images, while Jiang and Wang [29] enhanced traffic scene datasets through semantic and relational annotations, leading to measurable gains in object detection performance. Ries *et al*. [42] proposed a formal Alloy-based dataset modeling and validation, but did not assess ML performance. Soukour *et al*. [52] employed a domain specific KG to support the refinement of goal model in flood management, focusing on semantic usefulness and tool responsiveness.

Table 7. Comparison of related works.

study	KG-Based	Ontology	Dataset Focus	RE Phase	Validation Support	Domain Reasoning
[55]	✓	✓	×	Specification	×	✓ (Security)
[28]	×	×	✓(NNs)	Specification	×	×
[29]	×	✓	✓	Specification	✓(Limited)	×
[42]	×	×	✓	Specification & Validation	✓ (Structural)	×
[52]	✓	×	×	Elicitation	×	✓
[37]	×	×	✓	Validation	✓ (Rule-based)	×
[5]	×	×	✓	-	✓ (Computational quality metrics)	×
[48]	×	×	✓	Validation	✓	✓
KG4RSV	✓	✓	✓ (AI/ML)	Specification and Validation	✓ (Semantic)	✓

Table 8. Comparison of datasets produced by related works.

Study	Dataset Type	Dataset Source /Size	Validation method	Performance
[55]	Text: Security Requirements Specification (SRS) documents	62 SRSdocuments written by students	Terms extracted using KG	Correctly flagged 61/62 documents; 1 FP, 1 FN; high rule-based accuracy
[28]	Image (hand-written digits)	Custom image dataset with augmented handwritten “7” digits (The MNIST Database of Handwritten Digits)	Neural Network (NN)	Improve the NN’s recognition accuracy of digit ”7” to 0.98%
[29]	Image (traffic scenes)	CCTSDB2021; 2,574 images with semantic and spatial annotations	YOLO object detection (convolutional neural network CNN)	mAP@0.5 improved from 0.7748 to 0.8018 (semantic) and 0.9015 (semantic + relation coverage)
[42]	Five-Segments Digits (FSD)	Synthetic; +400 generated samples	No AI/ML evaluation (Formal modeling and validating method (DRCM/FDRCM) + Alloy)	N/A
[52]	Domain-specific Knowledge Graph	Manually constructs of flood warning KG (49 nodes, 54 edges)	Semantic similarity + KG traversal + G2T (Goal-to-Text) generation	73.7% of retrieved triples useful for goal refinement; response time: 7.68s (retrieval), 2.49s (text generation)
[5]	The Alzheimer’s Disease Neuroimaging Initiative (ADNI)	Number of samples: 2376. Number of features: 45	Quality metrics: - Pearson Correlation (PC) - Spearman Correlation (ρ) - Missing values - Outliers - Class Overlap - Classification task: - Random Forest -Stochastic Gradient Descent (SGD) - Clustering task: - -k-means clustering algorithms.	Data quality score= 0.4656 Classification accuracy = 0.5134 Clustering accuracy = 0.6012
	The Framingham Heart Study (FHS)	Number of samples: 5209. Number of features: 81.		Data quality score= 0.6481 Classification accuracy = 0.8594 Clustering accuracy = 0.4332
	Wisconsin Diagnosis Breast Cancer (WDBC)	Number of samples: 569. Number of features: 30.		Data quality score= 0.6335 Classification accuracy = 0.4804 Clustering accuracy = 0.5792
[48]	Pediatric Sepsis	10,492	Checks defined by clinicians and data scientists, which based on Patient Centered Outcomes Research Institute (PCORI) to harmonize a data quality framework	Data excluded: 9 Data transformed: 11
	Lung Transplant	719		Data excluded: 22 Data transformed: 9
	Jefferson Sepsis	212,531		Data excluded: 9 Data transformed: 13
	Immune Related Adverse Events	3,962		Data excluded: 12 Data transformed: 18
	Maternal Early Warning Scor	19,832		Data excluded: 4 Data transformed: 10

Moreover, Great Expectations (GE) [37] is a widely adopted open-source framework that validates tabular data using rule-based expectations and is integrated into Extract, Transform, Load (ETL) pipelines. However, it does not incorporate domain semantics, inter-feature reasoning, and ontology integration.

Similarly, Meysam *et al*., developed Dreamers framework [5], that evaluates dataset readiness for ML

projects. The method works by partitioning the dataset into small random subsets, then performing an assessment based on five computational quality metrics: Pearson Correlation (PC), Spearman Correlation (ρ), missing values, outliers, and class overlap. These metrics are aggregated to compute an overall quality score for each subset. In addition, the framework tests each subset using two ML models (supervised and unsupervised) to assess its learning performance. Subsets that fall below predefined quality thresholds are automatically discarded. the results show improvements in the computed quality indicators, although the approach does not incorporate domain knowledge considerations.

In another closely related study, ML-DQA proposes a human-driven data quality assurance framework for ML projects in the healthcare domain. The framework consists of three phases:

1) Entity resolution and data collection.
2) ML-DQA quality checks conducted by a multidisciplinary team-including a data scientist, clinical expert, project manager, and clinical trainee.
3) an adjudication phase where each sample is accepted, transformed, or discarded.

While this approach effectively leverages domain expertise, it is strongly dependent on expert input, which introduces subjectivity and limits its potential for automation.

In contrast, the KG4RSV framework combines formalized dataset requirement specification and KG-based validation applied to a structured tabular dataset. In addition, it demonstrates its effectiveness by improving the performance of the binary classification model using the eICU dataset (a real-world clinical dataset). This joint focus on structured validation and its measurable impact on downstream ML performance sets KG4RSV ahead of previous related works.

7. Conclusion and Future Work

This paper presents KG4RSV, a knowledge graph–driven framework to specify and validate AI/ML datasets using domain-specific requirements. KG4RSV encodes domain knowledge as formal semantic constraints to ensure data quality requirements. After representing data and quality requirements uniformly in RDF with adaptation of specific domain ontologies, KG4RSV enables an automatic validation through the SHACL shapes automatically generated from the Requirements RDF file, and the result is a violation report of all quality requirements. In addition, KG4RSV applies as a first step the elimination strategy for samples whose records violate constraints and generates a new dataset RDF graph that validates all the quality requirements.

An experimental case study using the clinical eICU dataset demonstrates the reliability of this framework. The results show a significant improvement in the performance of AI/ML models after applying KG4RSV, demonstrates its practical impact.

Future work will focus on apply KG4RSV to large-scale datasets to evaluate its scalability and effectiveness, incorporating additional post-validation strategies such as correction and standardization. This will further enhance KG4RSV's adaptability and utility across diverse AI/ML applications. We also plan to support more heterogeneous data types such as images, text, and time series, which are commonly used in AI/ML tasks. Furthermore, we aim to explore training AI/ML models directly on the knowledge graphs produced through the validation process, a promising direction in the field. In parallel, explainable AI (XAI) will be addressed by leveraging the semantic structure and interconnected nature of the Dataset-KG provided by our framework.

References

[1] Abdelmageed N., "Towards Transforming Tabular Datasets into Knowledge Graphs," *in Proceedings of the Semantic Web: ESWC 2020 Satellite Events*, Crete, pp 217-228, 2020. https://doi.org/10.1007/978-3-030-62327-2_37

[2] Abdelmageed N. and König-Ries B., "Meta2KG: An Embeddings-based Approach for Transforming Metadata to Knowledge Graphs," *in Proceedings of the 4th International Workshop on Knowledge Graph Construction@ESWC*, Crete, pp. 1-15, 2023. https://openreview.net/pdf?id=sF2ZN7M6iT

[3] Abduljabbar R., Dia H., Liyanage S., and Bagloee S., "Applications of Artificial Intelligence in Transport: An Overview," *Sustainability*, vol. 11, no. 1, pp. 1-24, 2019. https://doi.org/10.3390/su11010189

[4] Ahmad K., Abdelrazek M., Arora C., Bano M., and Grundy J., "Requirements Engineering for Artificial Intelligence Systems: A Systematic Mapping Study," *Information and Software Technology*, vol. 158, pp. 107176, 2023. https://doi.org/10.1016/j.infsof.2023.107176

[5] Ahangaran M., Zhu H., Li R., Yin L., and et al., "DREAMER: A Computational Framework to Evaluate Readiness of Datasets for Machine Learning," BMC Medical Informatics and Decision Making, vol. 24, pp. 1-14, 2024. Doi:10.1186/s12911-024-02544-w

[6] Alaketu M., Oguntimilehin A., Olatunji K., Abiola O., and et al., "Comparative Analysis of Intrusion Detection Models Using Big Data Analytics and Machine Learning Techniques," *The International Arab Journal of Information Technology*, vol. 21, no. 2, pp. 326-337, 2024. https://doi.org/10.34028/iajit/21/2/14

[7] Agarwal R., Fernandez D., Elsaleh T., Gyrard A.,

and et al., "Unified IoT Ontology to Enable Interoperability and Federation of Testbeds," *in Proceedings of the IEEE 3rd World Forum on Internet of Things*, Reston, pp. 70-75, 2016. DOI:10.1109/WF-IoT.2016.7845470

[8] Arpteg A., Brinne B., Crnkovic-Friis L., and Bosch J., "Software Engineering Challenges of Deep Learning," *in Proceedings of the 44th Euromicro Conference on Software Engineering and Advanced Applications*, Prague, pp. 50-59, 2018. DOI:10.1109/SEAA.2018.00018

[9] Bader S., Grangel-Gonzalez I., Nanjappa P., Vidal M., and Maleshkova M., "A Knowledge Graph for Industry 4.0," *in Proceedings of the European Semantic Web Conference*, Crete, pp. 465-480, 2020. https://doi.org/10.1007/978-3-030-49461-2_27

[10] Barrera J., Reina-Reina A., Lavalle A., Maté A., and Trujillo J., "An Extension of Istar for Machine Learning Requirements by Following the PRISE Methodology," *Computer Standards and Interfaces*, vol. 88, pp. 1-15, 2024. doi:10.1016/j.csi.2023.103806

[11] Beckett D. and McBride B., "RDF/XML syntax specification (revised)," *W3C recommendation*, vol. 10, no. 2.3, pp. 1-56, 2004.

[12] Berners-Lee T., Hendler J., and Lassila O., *Linking the world's information: essays on Tim Berners-Lee's invention of the World Wide Web*, ACM Books, 2023. https://doi.org/10.1145/3591366

[13] Challa H., Niu N., and Johnson R., "Faulty Requirements Made Valuable: On the Role of Data Quality in Deep Learning," *in Proceedings of the IEEE 7th International Workshop on Artificial Intelligence for Requirements Engineering*, Zurich, pp. 61-69, 2020. DOI:10.1109/AIRE51212.2020.00016

[14] Cui H., Lu J., Wang S., Xu R., and et al, "A Survey On Knowledge Graphs for Healthcare: Resources, Application Progress, And Promise," *in Proceedings of the ICML 3rd Workshop on Interpretable Machine Learning in Healthcare*, Hawaii, pp. 1-19, 2023. https://openreview.net/pdf?id=CZCktJoBRh

[15] Du R., Tsougenis E., Ho J., Chan J., and et al, "Machine Learning Application for The Prediction of SARS-Cov-2 Infection Using Blood Tests and Chest Radiograph," *Scientific Reports*, vol. 11, no. 1, pp. 1-13, 2021. DOI:10.1038/s41598-021-93719-2

[16] DuCharme B., *Learning SPARQL*, O'Reilly Media, Inc., 2013. https://www.oreilly.com/library/view/learning-sparql-2nd/9781449371449/

[17] Djeddi C., Zarour N., and Charrel P., "A Requirement Elicitation Method for Big Data Projects," *in Proceedings of the International Conference on Managing Business Through Web Analytics*, Khemis Miliana, pp. 231-242, 2022. https://doi.org/10.1007/978-3-031-06971-0_17

[18] Emmert-Streib F. and Dehmer M., "Understanding Statistical Hypothesis Testing: The Logic of Statistical Inference," *Machine and Knowledge Extraction*, vol. 1, no. 3, pp. 945-961, 2019, DOI:10.3390/make1030054

[19] Färber M. and Lamprecht D., "The Data Set Knowledge Graph: Creating A Linked Open Data Source for Data Sets," *Quantitative Science Studies*, vol. 2, no. 4, pp. 1324-1355, 2021. https://doi.org/10.1162/qss_a_00161

[20] Fernandez-Llorca D. and Gómez E., "Trustworthy Artificial Intelligence Requirements in the Autonomous Driving Domain," *Computer*, vol. 56, no. 2, pp. 29-39, 2023. DOI:10.1109/MC.2022.3212091

[21] Ferranti N., Polleres A., de Souza J., and Ahmetaj S., "Formalizing Property Constraints in Wikidata," *Wikidata@ ISWC*, vol. 3262, pp. 1-13, 2022. https://ceur-ws.org/Vol-3262/paper1.pdf

[22] Foidl H., Golendukhina V., Ramler R., and Felderer M., "Data Pipeline Quality: Influencing Factors, Root Causes of Data-related Issues, and Processing Problem Areas for Developers," *arXiv Preprint*, vol. arXiv:2309.07067, pp. 1-34, 2023. https://doi.org/10.48550/arXiv.2309.07067

[23] Goel P., Wasson V., Singh I., Kaur A., and et al, "Machine Learning Based Water Requirement Prediction for Agriculture Before a Rainfall," *in Proceedings of the International Conference on Emerging Smart Computing and Informatics*, Pune, pp. 1-7, 2024. DOI:10.1109/ESCI59607.2024.10497332

[24] Gudigar A., Kadri N., Raghavendra U., Samanth J., and et al., "Automatic Identification of Hypertension and Assessment of Its Secondary Effects Using Artificial Intelligence: A Systematic Review (2013-2023)," *Computers in Biology and Medicine*, vol. 172, pp. 1-26, 2024. DOI:10.1016/j.compbiomed.2024.108207

[25] Habibullah K., Gay G., and Horkoff J., "Non-Functional Requirements for Machine Learning: Understanding Current Use and Challenges Among Practitioners," *Requirements Engineering*, vol. 28, no. 2, pp. 283-316, 2023. DOI:10.1007/s00766-022-00395-3

[26] Horkoff J., Aydemir F., Cardoso E., Li T., and et al, "Goal-Oriented Requirements Engineering: An Extended Systematic Mapping Study," *Requirements Engineering*, vol. 24, no. 2, pp. 133-160, 2019. doi:10.1007/s00766-017-0280-z

[27] Huang H., Niazmand E., and Vidal M., "Hybrid AI Approach for Counterfactual Prediction over Knowledge Graphs for Personal Healthcare," *in Proceedings of the Artificial Intelligence and Data Science for Healthcare: Bridging Data-Centric AI and People-Centric Healthcare*, Barcelona, pp.1-

6, 2024. DOI:10.13140/RG.2.2.15682.18889

[28] Jahić B., Guelfi N., and Ries B., "SEMKIS-DSL: A Domain-Specific Language to Support Requirements Engineering of Datasets and Neural Network Recognition," *Information*, vol. 14, no. 4, pp. 1-25, 2023. DOI:10.3390/info14040213

[29] Jiang L. and Wang X., "Dataset Constrution through Ontology-Based Data Requirements Analysis," *Applied Sciences*, vol. 14, no. 6, pp. 1-22, 2024. https://doi.org/10.3390/app14062237

[30] Kamm S., Veekati S., Mueller T., Jazdi N., and Weyrich M., "A Survey On Machine Learning Based Analysis of Heterogeneous Data in Industrial Automation," *Computers in Industry*, vol. 149, pp. 1-14, 2023. https://doi.org/10.1016/j.compind.2023.103930

[31] Knublauch H. and Kontokostas D., "Shapes Constraint Language (SHACL)," *W3C Candidate Recommendation*, vol. 11, no. 8, pp. 1, 2017.

[32] Massey P., Montgomery C., and Zhang A., "Comparison of ChatGPT-3.5, ChatGPT-4, and Orthopaedic Resident Performance on Orthopaedic Assessment Examinations," *Journal of the American Academy of Orthopaedic Surgeons*, vol. 31, no. 23, pp. 1173-1179, 2023. DOI: 10.5435/JAAOS-D-23-00396

[33] Mehrabian A., Gerasimov I., Khayat M., Pagán B., and et al, "Unveiling Knowledge: Enhancing Data Discovery Through Integrative Knowledge Graphs at NASA's GES-DISC," *in Proceedings of the 23rd Meeting of the American Geophysical Union*, California, pp. 1-28, 2023. https://ntrs.nasa.gov/api/citations/20230017499/downloads/AGU23-GESDISC-Graph-removed-graphics.pdf

[34] Mohamed A. and Jebapillai C., "A Machine Learning Attempt for Anatomizing Software Risks in Small and Medium Agile Enterprises," *The International Arab Journal of Information Technology*, vol. 22, no. 3, pp. 547-559, 2025. https://doi.org/10.34028/iajit/22/3/10

[35] Mohamed A., Abuoda G., Ghanem A., Kaoudi Z., and Aboulnaga A., "RDFFrames: Knowledge Graph Access for Machine Learning Tools," *The VLDB Journal*, vol. 31, no. 2, pp. 321-346, 2022. https://doi.org/10.1007/s00778-021-00690-5

[36] Montoya-Murillo D., Mora M., Galvan-Cruz S., Munoz-Zavala A., and Alvarez-Rodriguez F., "A Review of SDLCs for Big Data Analytics Systems in the Context of Very Small Entities Using the ISO/IEC 29110 Standard-Basic Profile," *The International Arab Journal of Information Technology*, vol. 22, no. 1, pp. 194-215, 2025. https://doi.org/10.34028/iajit/22/1/15

[37] Narayanan P., *Data Engineering for Machine Learning Pipelines: From Python Libraries to ML Pipelines and Cloud Platforms*, Springer, 2024. https://doi.org/10.1007/979-8-8688-0602-5_6

[38] Pandey D., Niwaria K., and Chourasia B., "Machine Learning Algorithms: A Review," *International Research Journal of Engineering and Technology*, vol. 6, no. 2, pp. 916-922, 2019.

[39] Priestley M., O'donnell F., and Simperl E., "A Survey of Data Quality Requirements That Matter in ML Development Pipelines," *ACM Journal of Data and Information Quality*, vol. 15, no. 2, pp. 1-39, 2023. https://doi.org/10.1145/3592616

[40] Pollard T., Johnson A., Raffa J., and Celi L., "The eICU Collaborative Research Database, A Freely Available Multi-Center Database for Critical Care Research," *Scientific data*, vol. 5, no. 1, pp. 1-13, 2018. https://doi.org/10.1038/sdata.2018.178

[41] Rajabi E. and Kafaie S., "Knowledge Graphs and Explainable AI in Healthcare," *Information*, vol. 13, no. 10, pp. 1-10, 2022. https://doi.org/10.3390/info13100459

[42] Ries B., Guelfi N., and Jahić B., "An MDE Method for Improving Deep Learning Dataset Requirements Engineering using Alloy and UML," *in Proceedings of the 9th International Conference on Model-Driven Engineering and Software Development, Science and Technology Publications*, virtual, pp. 41-52, 2021. DOI:10.5220/0010216600410052

[43] Rouse M., SNOMED CT (Systematized Nomenclature of Medicine-Clinical Terms), SearchHealthIT, 2010. Last Visited, 2025. https://www.snomed.org/

[44] Salazar-Salazar G., Mora M., Duran-Limon H., Alvarez-Rodriguez F., and Munoz-Zavala A., "Review of Agile SDLC for Big Data Analytics Systems in the Context of Small Organizations Using Scrum-XP," *The International Arab Journal of Information Technology*, vol. 21, no. 6, pp. 1089-1110, 2024. https://doi.org/10.34028/iajit/21/6/12

[45] Schaffenrath R., Proksch D., Kopp M., Albasini I., and et al, "Benchmark for Performance Evaluation of SHACL Implementations in Graph Databases," *in Proceedings of the International Joint Conference on Rules and Reasoning*, Oslo, pp. 82-96, 2020. https://doi.org/10.1007/978-3-030-57977-7_6

[46] Semmak F., Gnaho C., Brunet J., and Laleau R., "How to Adapt the KAOS Method to the Requirements Engineering of Cycab Vehicle.," *in Proceedings of the 4th International Conference on Evaluation of Novel Approaches to Software Engineering*, Milan, pp. 87-94, 2009.

[47] Semler S., "LOINC: Origin, Development of and Perspectives for Medical Research and Biobanking-20 Years on the Way to Implementation in Germany," *Journal of Laboratory Medicine*, vol. 43, no. 6, pp. 359-382, 2019. https://doi.org/10.1515/labmed-2019-0193

[48] Sendak M., Sirdeshmukh G., Ochoa T., Premo H.,

and et al, "Development and Validation of ML-DQA-a Machine Learning Data Quality Assurance Framework for Healthcare," *in Proceedings of the 7th Machine Learning for Healthcare Conference,* North Carolina, pp. 741-759, 2022. DOI:10.48550/arXiv.2208.02670

[49] Sheh R., "Explainable Artificial Intelligence Requirements for Safe, Intelligent Robots," *in Proceedings of the IEEE International Conference on Intelligence and Safety for Robotics*, Tokoname, pp. 382-387. 2021. DOI:10.1109/ISR50024.2021.9419498

[50] Sellami S. and Zarour N., "Keyword-Based Faceted Search Interface for Knowledge Graph Construction and Exploration," *International Journal of Web Information Systems*, vol. 18, no. 56, pp. 453-486, 2022. https://doi.org/10.1108/IJWIS-02-2022-0037

[51] Sellami S., Dkaki T., Zarour N., and Charrel P., "KGMap: Leveraging Enterprise Knowledge Graphs by Bridging Between Relational, Social and Linked Web Data," *in Proceedings of the 3rd International Conference on Advances in Artificial Intelligence*, Istanbul, pp. 90-96, 2019. https://doi.org/10.1145/3369114.3369146

[52] Soukour S., Aboucaya W., and Georgantas N., "Leveraging Knowledge Graphs for Goal Model Generation," *in Proceedings of the* 7th *Workshop on Natural Language Processing for Requirements Engineering in conjuction with the 30th International Woking Conference on Requirements Engineering: Foundation for Software Quality*, Winterthur, pp. 1-12, 2024. https://inria.hal.science/hal-04486653

[53] Soureya Y., Amougou N., Ngossaha J., Tsakou S., and M. Ndjodo, "Adaptive Software Development: A Comprehensive Framework Integrating Artificial Intelligence for Sustainable Evolution," *The International Arab Journal of Information Technology*, vol. 22, no. 2, pp. 248-262, 2025. https://doi.org/10.34028/iajit/22/2/4

[54] Vandenbussche P., Atemezing G., Poveda-Villalón M., and Vatant B., "Linked Open Vocabularies (LOV): A Gateway to Reusable Semantic Vocabularies on the Web," *Semantic Web*, vol. 8, no. 3, pp. 437-452, 2016. DOI:10.3233/SW-160213

[55] Vonder Haar L., Reynolds S., Procko T., and Ochoa O., "Validating Security Requirement Specifications through the use of a Knowledge Graph," *in Proceedings of the IEEE 17th International Conference on Semantic Computing*, Laguna Hills, pp. 244-251, 2023. DOI:10.1109/ICSC56153.2023.00048

[56] Vogelsang A. and Borg M., "Requirements Engineering for Machine Learning: Perspectives from Data Scientists," *in Proceedings of the IEEE 27th International Requirements Engineering Conference Workshops*, Jeju, pp. 245-251. 2019. DOI:10.1109/REW.2019.00050

[57] Weihrauch D., Schindler P., and Sihn W., "A Conceptual Model for Developing a Smart Process Control System," *Procedia CIRP*, vol. 67, pp. 386-391, 2018. DOI:10.1016/j.procir.2017.12.230

[58] White J., Hays S., Fu Q., Spencer-Smith J., and Schmidt D., *Generative AI for Effective Software Development*, Springer, 2024. https://doi.org/10.1007/978-3-031-55642-5_4

[59] Yahya M., Ali A., Mehmood Q., Yang L., and et al, "A Benchmark Dataset with Knowledge Graph Generation for Industry 4.0 Production Lines," *Semantic Web*, vol. 15, no. 2, pp. 461-479, 2024. https://doi.org/10.3233/SW-233431

[60] Yogi K., Gowda D., Mouna K., Sujithra L., and et al, "Scalability and Performance Evaluation of Machine Learning Techniques in High-Volume Social Media Data Analysis," *in Proceedings of the 11th International Conference on Reliability, Infocom Technologies and Optimization*, Noida, pp. 1-6, 2024. DOI:10.1109/ICRITO61523.2024.10522361

[61] Zhao Z., Liu W., French T., and Stewart M., "Rel2Graph: Automated Mapping from Relational Databases to a Unified Property Knowledge Graph," *arXiv Preprint*, vol. arXiv:2310.01080, pp. 1-18, 2023. doi:10.48550/arXiv.2310.01080

[62] Zhou S., Greenspan H., Davatzikos C., Duncan J., and et al., "A Review of Deep Learning in Medical Imaging: Imaging Traits, Technology Trends, Case Studies with Progress Highlights, And Future Promises," *IEEE*, vol. 109, no. 5, pp. 820-838, 2021, Doi:10.1109/JPROC.2021.3054390.

Ziad Nasri is a Ph.D. student in Advanced Software Engineering at University of Constantine 2 Abdelhamid Mehri, Algeria. He holds a Bachelor's (2018) and Master's (2020) degree in Software Engineering from the same university. He has experience as a software developer and IT responsible at EPIC-EHC production unit. His research interests include Software Engineering Methodologies, Application Development, and Requirements Engineering (RE) for AI/ML-based system.

Samir Sellami is an Associate Teacher in the Department of Mathematics and Computer Science at the Higher School of Technological Teaching ENSET-Skikda, Algeria. He is also a researcher at the LIRE Laboratory-SIBC Team at the University of Constantine 2-Abdelhamid Mehri, Algeria. His research interests cover Virtual Data Integration, Advanced Information Systems, Semantic Web, Linked Data, Knowledge Graphs and Neuro-Symbolic AI. He holds an Engineering degree INg (2008) in computer science as well as Magister degree Mag (2015) in advanced techniques for parallel and distributed systems, both from the University of 20 August Skikda-1955. He successfully defended his Ph.D. thesis in 2020 at University of Constantine 2 - Abdelhamid Mehri.

Dr. Chabane DJEDDI is an assistant professor of computer science at the National Higher School of Artificial Intelligence (ENSIA) in Algiers, Algeria. He holds a Ph.D. in Computer Science with a specialization in Advanced Information Systems from Constantine 2 University. His research interests and publications include Requirements Engineering for Big Data and AI-Based Systems, Formal Methods for Requirements Verification, and Machine Learning Applications in the Security of UAV AD-HOC Networks.

Taoufiq Dkaki is an Associate Professor (Maître de Conférences HDR) at the University of Toulouse Jean Jaurès, affiliated with the IRIT research laboratory. His research focuses on Information Retrieval, Natural Language Processing, and Legal AI, with particular interest in document representation, neural models, and explainability. He has coordinated and contributed to several interdisciplinary research projects funded by ANR, the Occitanie Region, and international institutions. He is also the pedagogical head of the Master's program in Digital Information Engineering and has supervised numerous doctoral and postdoctoral research projects.

Nacer Eddine Zarour has been a professor since 2009 at the Department of Software Technologies and Information Systems at Constantine 2 University, Constantine, Algeria. He received his PhD in Cooperative Information Systems from Mentouri University, Constantine, in2004. He chaired the Scientific Council of the Faculty of New Information and Communication Technologies (NICT) and was then responsible for the Mathematics-Computer Science field. His current research activities are carried out at the LIRE Laboratory. His research areas include Advanced Information Systems, Requirements Engineering, Cloud Computing, and Data Science. He has been and is responsible for several international (PHC CMEP Tassili) and national (CNEPRU/PRFU) projects.

Pierre-Jean Charrel has been a Professor of Computer Science at the University of Toulouse - Jean Jaurès since 2003, where he shares his passion for software development and research with his students. A graduate of ENSEEIHT, he has been an Assistant and then a Lecturer at the University of Toulouse 1 Capitole since 1982, where he participates in innovative research projects in the field of computer science. He has skills in Requirements Engineering, Databases, Artificial Intelligence, and Information Systems. His mission is to contribute to the Advancement of Knowledge and the Training of Future Computer Scientists, in collaboration with my colleagues and partners. He is motivated by Scientific and Pedagogical Challenges, and seeks to bring his Vision and Experience to his team and organization.