ML System Engineering Supported by a Body of Knowledge

Juliette Mattioli[1], Dominique Tachet[2], Fabien Tschirhart[2], Henri Sohier[2], Loic Cantat[2] and Boris Robert[2,3]

[1]*Thales, France*

[2]*IRT SystemmX, France*

[3]*IRT Saint Exupéry, France*

Keywords: Body-of-Knowledge, Knowledge Graph, Knowledge Extraction, Knowledge Fusion, ML Engineering.

Abstract: A body of knowledge (BoK) can be defined as the comprehensive set of concepts, terminology, standards, and activities that facilitate the dissemination of knowledge about a specific field, providing guidance for practice or work. This paper presents a methodology for the construction of a body of knowledge (BoK) based on knowledge-based artificial intelligence. The process begins with the identification of relevant documents and data, which are then used to capture concepts, standards, best practices, and state-of-the-art. These knowledge items are then fused into a knowledge graph, and finally, query capacities are provided. The overall process of knowledge collection, storage, and retrieval is implemented with the objective of supporting a trustworthy machine learning (ML) end-to-end engineering methodology, through the ML Engineering BoK.

1 RATIONALE

Systems and products are developed in competitive, volatile, uncertain, complex and ambiguous contexts, influenced by external factors such as regulations and societal expectations. Artificial Intelligence (AI) technologies, in particular Machine Learning (ML) approaches, improve product quality and production efficiency (Li et al., 2017). Reliability is crucial for critical systems to remain reliable throughout their lifecycle and to evolve cost-effectively. However, the rapid adoption of AI technologies is leading to a specialization of engineers and a dispersion of the required skills. Current demographics are making highly skilled and experienced engineers scarce, leading to a lack of project support and mentorship. In addition, the complexity of AI-based solutions and past trends in each component require the management of AI engineering knowledge and general engineering practices. This highlights the need for effective management of AI engineering knowledge and practices. Knowledge management and knowledge engineering (KE) are often used interchangeably, with ”manage” referring to executive leadership and ”engineer” to planning, construction, or design activities. The main difference is that the knowledge manager sets the process direction, while the knowledge engineer develops the means to achieve it. The Confiance.ai[1] program's end-to-end methodology serves as a foundational framework for knowledge management in trustworthy AI engineering (Awadid et al., 2024). It addresses non-functional requirements for successful implementation of ML-based components in critical systems (Adedjouma et al., 2022). The methodology covers various process levels and aligns with industrial best practices. It is essential to define the scope and position of the methodology in relation to other engineering disciplines. KE is a sub-field of AI that focuses on understanding, designing, and implementing methods for representing information effectively (Shapiro, 2006). It facilitates the management of all types of knowledge based on labelled graphs and provides guidance for resolving ML engineering-related issues. In order to address these issues, Confiance.ai program put forth the argument that there is a need to consolidate the AI engineering field's largely fragmented body of knowledge (Mattioli et al., 2024), which encompasses data engineering, algorithm engineering, software and system engineering, safety and cyber-security, similar to the SWEBOK definition of software engineering (Robert et al., 2002).

[1]www.confiance.ai/en

Mattioli, J., Tachet, D., Tschirhart, F., Sohier, H., Cantat, L. and Robert, B.
ML System Engineering Supported by a Body of Knowledge.
DOI: 10.5220/0013048500003838

In *Proceedings of the 16th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2024) - Volume 3: KMIS*, pages 331-338
ISBN: 978-989-758-716-0; ISSN: 2184-3228

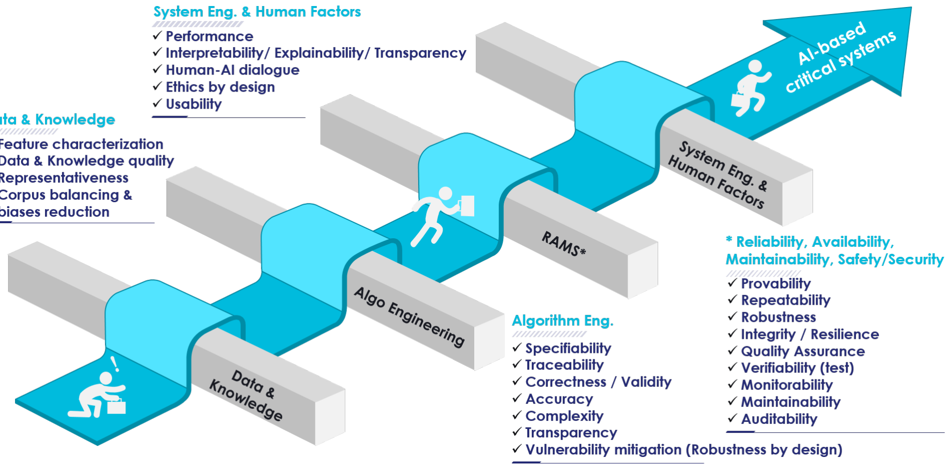

Figure 1: AI/ML deployment induces some (engineering) challenges.

2 KNOWLEDGE ENGINEERING

From this perspective, Knowledge Engineering (KE) represents a sub-field of AI that is concerned with the understanding, design and implementation of methods for representing information in a manner that enables computers to utilize it effectively. In other words, the objective of KE is the understanding and subsequent representation of human knowledge in the form of data structures, semantic models (conceptual graph of the data in relation to the real world) and heuristics (or rules). The induced process comprises three principal elements: a) knowledge acquisition, representation, and validation; b) inference; and c) explanation and justification. It is evident that approaches to knowledge representation, such as ontologies, taxonomies, thesauri, and vocabularies, can be employed to support a diverse array of activities. Techniques that could be used for such purpose include:

- Using a taxonomy that is well-known and accepted by the organization in work-products;
- Using patterns to help a computer “understand” the content of work-products;
- Encoding the breakdown structures that represent well-established knowledge within the organization;
- Using inference rules to reason about the quality of the content of different work-products.

An organization must manage knowledge about its systems and engineering practices, known as a Body of Knowledge (BoK). The BoK provides a comprehensive description of ML system engineering contents and practices, establishing a foundation for curriculum development and creating a coherent curriculum for qualification towards certification.

3 ML ENGINEERING BoK

A **body of knowledge** (BoK) is”*structured knowledge that is employed by members of a discipline to inform their practice or work*” (Ören, 2005). It is used to define concepts and activities, such as accuracy in data engineering, machine learning models and system-level applications. The core components of a BoK include concepts, knowledge, skills, standards, terminology, guidelines, practices and activities. It serves as the ”ground truth” for ML engineering activities, covering various domains such as data engineering, algorithm engineering, software engineering, systems engineering, cyber-security, safety, and cognitive engineering.

3.1 The BoK Design

The BoK design is an iterative process involving knowledge acquisition, fusion, storage, and retrieval. Knowledge is acquired from structured, semi-structured, and unstructured data, with extraction focusing on entities, attributes, and relations. Knowledge fusion requires ongoing ontology construction and quality evaluation. Currently, knowledge is typically stored in KG databases. Confiance.ai BoK (fig 2) is a comprehensive guide for engineers on the lifecycle of AI-based critical systems. It offers guidance and support throughout the development, maintenance, and evolution of these systems. The guide defines trustworthy ML engineering concepts and provides an outline of essential knowledge, skills,

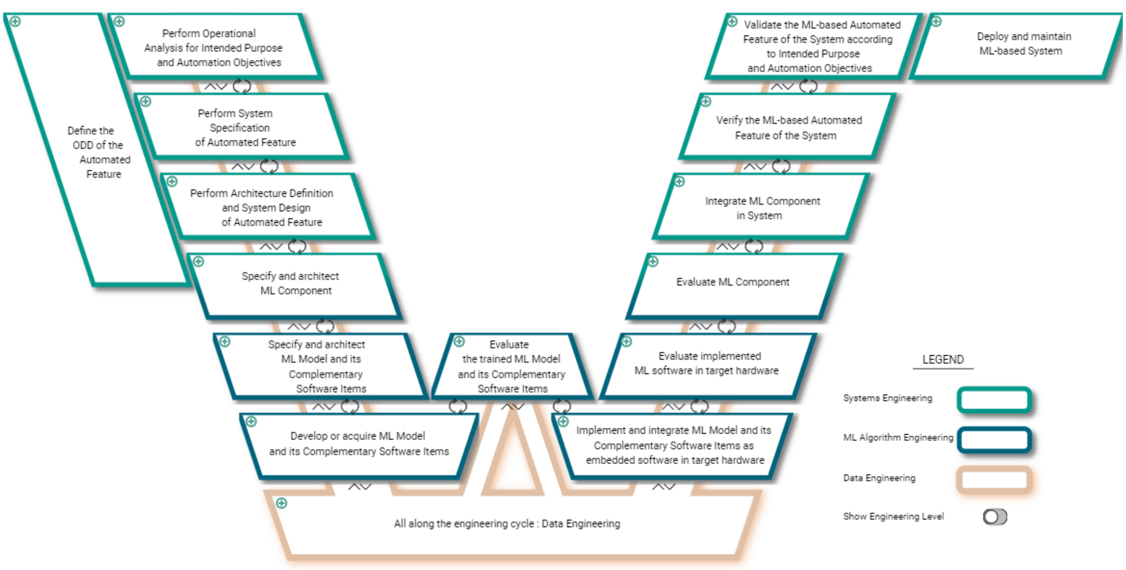

Figure 2: Trustworthy ML engineering Body-of-Knowledge - https://bok.confiance.ai/.

and practices, covering all fundamental competencies required by professionals in the field. The initial step of the ML Engineering BoK design is to identify the domain of application and compile a list of relevant knowledge sources. Secondly, a conceptual model will be devised with the objective of gathering together the entities of interest, their inter-relationships and the categories. A valuable resource for conceptual modeling is Capela©, which is a model-based engineering solution that has proven effective in numerous industrial contexts. Thirdly, the logical and physical models will provide a logical representation and assertions for the entities and relationships that have been collected. Fourthly, the technical development and implementation must take into account the coding language to be employed (for example, RDF and OWL), as well as the serialization formats (such as RDF/XML, Turtle and JSON-LD). The final stage is the deployment of the BoK as a service, thereby facilitating reuse and enabling the engineering community to provide feedback. In essence, this process entails the transformation of knowledge held by engineering experts and end-users into a machine-readable format. Designing a ML Engineering BoK induces some general issues:

- **Raw Data Acquisition.** How to select the relevant data and information to be fed into the BoK?
- **Knowledge Extraction, Representation and Validation.** How do we represent human knowledge as it currently exists in state of the art reports, scientific articles, standards and norms, Engineering best practices, and the minds of the experts in terms of data structures that can be processed by a computer? How to determine the best representation for any given engineering problem?
- **Knowledge Integration and Fusion.** How do we use these knowledge item to generate useful information in the context of a ML engineering?
- **KG Design.** How to manipulate the knowledge to provide explanations to the engineer/user?
- **Knowledge Query.** How do we use these abstract knowledge structures to generate useful information in the context of a specific case?

3.2 Step 1: Raw Data Acquisition

BoK developers create knowledge bases from scratch, dealing with diversity and heterogeneity of knowledge representation formalisms and mismatch of different knowledge items. Knowledge engineers focus on modeling structural use cases and expert knowledge concepts. The first step is to define a taxonomy of the ML engineering domain aligned with ISO/IEC DIS 5338 standard for AI systems and safety and reliability standards. This taxonomy is the classification of concepts induced by Trustworthy ML Engineering activities. The operational definition of trustworthy AI includes a taxonomy and keywords that define core domains such as AI Engineering, Data Engineering, and Safety Engineering, covering the entire life cycle of AI-based critical systems. This framework aids in harmonizing design and support activities, including monitoring and maintenance, and serves as a foundation for the Confiance.ai methodology, which outlines requirements and recommendations. The initial stage is devoted to the identification of data sources, as this has a significant impact on the entire knowledge graph (KG) development process, as well as on the selection of knowledge extraction techniques. Definitions have been collated from various sources, including European and worldwide standardization bodies (ISO/IEC 5338, Aerospace Standard 6983, IEEE 7000...), National and European projects (Confiance.ai, DEEL project[2], JRC Flagship on AI[3]), scientific publications and working groups (e.g. the HLEG or the AI Safety Landscape initiative[4]), as well as other relevant sources. The working group responsible for the state of the art sourced definitions from external literature in most cases. Sometimes the existing literature did not match the scope of ML Engineering. We created a new definition. This phase was also based on data about making AI reliable. This includes making AI reliable through design, data engineering for trusted AI, IVVQ Strategy (Integration, Verification, Validation and Qualification), and targeted embedded AI.

3.3 Step 2: Knowledge Extraction, Representation and Validation

The extraction of knowledge from semi-structured sources is easier than from unstructured sources, which hold more information. The second phase focuses on extracting knowledge from unstructured data to create and enhance knowledge graphs, identifying entities and relationships. This process involves natural language processing (NLP) and knowledge representation technologies to automatically extract structured information from various data types, facilitating the effective use of external data. An entity represents the most fundamental unit of a knowledge graph. It represents a concept. Furthermore, the quality of knowledge graph construction is contingent upon the accuracy and integrity of its extraction. Subsequently, relationship extraction entails the identification of associations between entities, thereby establishing semantic relations and forming a knowledge network. These types of graphs embed a structured representation of facts, consisting of entities, relationships, and semantic descriptions, which are modeled with an RDF (Resource Description Framework) structure. An RDF model is a flexible data representation model comprising three-element tuples, with no fixed schema requirements. It is a graph-based model for the description of entities and their relationships on the Web. Many researchers prefer to conceptualize RDF as a set of triples, although it is commonly described as a directed and labelled graph, each consisting of a subject, predicate and object in the form of $< subject, predicate, object >$. In this context, the predicate represents the relationship between the subject and the object. For example: $< Data\_Engineering, is\_an\_activity, MLOps >$. The triples are stored in a triple store and can be queried using the SPARQL query language. In comparison to both inverted indices and plain text files, triple stores and the SPARQL query language facilitate the formulation of sophisticated queries, enabling users to satisfy complex information needs. Although a model is required for representing data in triples (similar to relational databases), RDF enables the expression of rich semantics and supports knowledge inference (Hertz et al., 2019). Like any model, such a BoK is only an approximation of reality. New observations based on ML engineering use-cases can guide the further acquisition of knowledge. Therefore, an evaluation of the represented knowledge with respect to reality is indispensable for the creation of an adequate model. These limitations relate to the so-called symbol grounding problem (Harnad, 1990), and concern the extent to which representational elements are hand-crafted rather than learned from data. The most common methods employed include pattern matching, machine learning, and semantic rule extraction. Furthermore, generative models such as large language models (LLMs) can play a pivotal role in the construction of knowledge graphs by extracting entities, relationships, and attributes from unstructured text data (Meyer et al., 2023). They can be pre-trained through the structurally consistent linearization of text, which facilitates the transition from traditional understanding to structured understanding and increases knowledge sharing (Wang et al., 2022). In contexts with Named Entity Recognition (NER), as demonstrated by (Straková et al., 2019), the proposed generative method implicitly models the structure between named entities. This approach effectively avoids the complexity inherent to multi-label mapping. Similarly, extracting overlapping triples in relation extraction is also challenging to address for traditional discriminating models (Zeng et al., 2018), introducing a new perspective for addressing this issue through a general generative framework. Moreover, several features must be taken into account when developing a BoK:

- Redundancy: Are there identical or equivalent knowledge items that is a special case of another (subsumed)?
- Consistency: Are there ambiguous or conflicting knowledge, is there indeterminacy in its ap-

[2] https://www.deel.ai/

[3] https://joint-research-centre.ec.europa.eu/jrc-mission-statement-work-programme/facts4eufuture/artificial-intelligence-european-perspective/future-ai_en

[4] https://www.aisafetyw.org/ai-safety-landscape

plication? Is it intended? Are several outcomes possible, for example, depending on the strategy (the order in which the knowledge models are ordered)?

- Minimality: Can the knowledge set be reduced and simplified? Is the reduced form logically equivalent to the first one?
- Completeness: Are all possible entries covered by the knowledge of the set?

Thus, a good BoK must have properties such as:

- Representational Accuracy: It should represent all kinds of required knowledge.
- Inferential Adequacy: It should be able to manipulate the representational structures to produce new knowledge corresponding to the existing structure.
- Inferential Efficiency: The ability to direct the inferential knowledge mechanism into the most productive directions by storing appropriate guides.
- Acquisitional Efficiency: The ability to complete with new knowledge easily using automatic methods.

Peer reviews with various stakeholders (data scientists, software and system engineers, safety and cybersecurity engineers...) were carried out to assess the appropriateness and quality of the acquired knowledge in relation to the ML engineering end-to-end methodology (Adedjouma et al., 2022).

3.4 Step 3: Knowledge Integration and Fusion

For (Sowa, 2000), "*Knowledge Representation is the application of logic and ontology to the task of constructing software models for some domain*". Therefore, the way a knowledge representation is conceived reflects a particular insight or understanding of how people reason. The selection of any of the currently available representation technologies (such as logic, knowledge bases, ontology, semantic networks...) commits one to fundamental views on the nature of intelligent reasoning and consequently very different goals and definitions of success. As we manipulate concepts with words, all ontologies use human language to "represent" the world. Thus, ontology is expressed as a formal representation of knowledge by a set of concepts within a domain and the relationships between these concepts. Nevertheless, the "fidelity" of the representation depends on what the knowledge-based system captures from the real thing and what it omits. If such system has an imperfect model of its universe, knowledge exchange or sharing may increase or compound errors during the ML Engineering process. As such, a fundamental step is to establish effective knowledge representation (symbolic representation) that can be used for query. The sheer complexity, variety and volume of data available today presents a significant challenge to achieving efficient and accurate knowledge graph fusion. The process of integrating disparate sources of knowledge, also known as knowledge fusion, entails the elimination of redundancies, inconsistencies, and ambiguities from the integrated corpus. The field of engineering is one in which knowledge is typically subject to updates. In the majority of cases, users will have the capacity to supplement existing external knowledge graphs with external knowledge. Thus, the objective of knowledge fusion is to merge semantically equivalent elements, for example, the concepts of "accuracy" and "machine learning accuracy", with the intention of integrating novel forms of knowledge within existing conceptual frameworks or factual assertions. The sub-tasks of knowledge fusion include the alignment of attributes, the matching of entities with small-scale incoming triples, and the alignment of entities with a complete knowledge graph. This stage is beneficial for both the generation and completion of knowledge graphs. We employ the knowledge graph representation for knowledge fusion, as proposed by (Laudy et al., 2007), which is based on the conceptual graph model. This representation is used to store and combine knowledge. The approach is to examine observations with domain knowledge and graph operators. This removes any bias from translating data from one format to another with different models. We suggest using it for a high-level information fusion approach based on the Maximal Join operator, which is an aggregation operator on conceptual graphs (Laudy, 2011).

3.5 Step 4: Knowledge Graph

At worst, the effort involved in specifying the relevant knowledge forces us to think more deeply about the relevant ways of characterizing the ML engineering models that we as researchers implicitly construct anyway. The use of Knowledge Graphs (KG) as a means of representing knowledge is becoming increasingly prevalent. Their versatility in terms of representation allows for the integration of diverse data sources, both within and across engineering boundaries. Therefore, our primary strategy to support this step in practice was the creation of a knowledge graph, which collects information about each ML development activity, the artifacts and processes used in the entire ML-based system lifecycle, the end-to-end methodology, and the motivation behind the key design decisions. Several replications have been carried out in this way, contributing to a growing body of knowledge about ML engineering techniques. By employing the graph architecture, KGs are capable of modeling a range of relationship types (edges) and entities (nodes) (Chen et al., 2020). KGs comprise an additional embedded layer, designated a reasoner (or inference engine), which enables them to extract implicit information from existing explicit concepts, in contrast to plain graph or non-relational databases. The most well-known examples of knowledge graphs (KGs) – DBpedia, Freebase, Wikidata, YAGO, and so forth – encompass a diverse array of domains and are either derived from Wikipedia or created by volunteer communities (Heist et al., 2020). The Google Knowledge Graph is one of the largest and most comprehensive KGs in existence, aiming to model and link all structured information found on the internet, including persons, organizations, skills, events, products, and more. This is one of the reasons why the Google search engine is so effective. A graph-based knowledge representation and reasoning formalism derived from conceptual graphs has been formalized as finite bipartite graphs, as outlined in (Mugnier and Chein, 1992). In this formalism, the set of nodes is divided into concept and conceptual relation nodes. In such a graph, concept nodes represent classes of individuals, and conceptual relation nodes illustrate the relationships between the aforementioned concept nodes. This is in accordance with the findings of (Sowa, 1976). As outlined in (Ehrlinger and Wöß, 2016), a KG acquires information and integrates it into an ontology, subsequently applying a reasoner to derive new knowledge. Furthermore, in accordance with the definition provided by (Ji et al., 2021), KGs are "structured representations of a fact, consisting of entities, relations, and semantics." Entities may be either real-world objects or abstract concepts. Relationships represent the relationship between entities, and semantic descriptions of entities and their relationships contain types and properties with defined semantics. Property graphs, in which nodes and relations possess properties or attributes, or attribute graphs, are extensively employed. All of these facets rely on a knowledge inference over knowledge graphs, which represents one of the core technologies in the design of our ML engineering BoK. The Semantic Web community has reached a consensus on the use of RDF to represent a knowledge graph. Then, RDF model also allows for a more expressive semantics of the modeled data that can be used for knowledge inference. As a result, a KG is a set of interconnected information on a specific set of facts that includes characteristics of many data management paradigms:

- Database: Structured queries can be used to explore data in a database.
- Graph: KGs can be analyzed in the same way that any other network data structure can be.
- Knowledge Base: Formal semantics are encoded in KGs, which can be used to understand data and infer new facts.

3.6 Step 5: Knowledge Query

In this context, a body of knowledge (BoK) is conceptualized as a graph of knowledge, as proposed by (Mattioli et al., 2022). Ultimately, the utility of the ingested, transformed, integrated and stored knowledge is contingent upon the efficiency with which answers can be retrieved by users in an intuitive manner. At the present time, keyword queries and specialized query languages (e.g. SQL and SPARQL) represent the prevailing approaches to information retrieval. However, in order to facilitate the search for a specific ML engineering knowledge by querying the KG and selecting the set of relevant engineering views to perform specific ML engineering activities, it is necessary to enable the identification of similarities between Confiance.ai documents by searching for isomorphisms between the graphs representing the knowledge extracted from the text. A number of algorithms have been defined which implement subgraph isomorphism; however, the subgraph isomorphic problem is an NP-complete problem. The initial component is a generic sub-graph matching mechanism that functions in conjunction with fusion schemes. This component is responsible for ensuring the structural consistency of the merged information with respect to the structures of the initial documents throughout the fusion process. The fusion approach is constituted by the similarity and compatibility functions applied to the members of the graphs to be fused. The generic fusion algorithm can be adapted to suit the context in which it is used by adopting these strategies. The knowledge graph fusion method offers two additional operations, contingent upon the fusion strategies employed. Information synthesis is the collection and organization of data on a subject. Information is then put together into a network through information synthesis, where any repetitions are removed. Fusing techniques are used to combine information about the same thing, even though it is in different forms. When different sources of information are used to create a representation of something, inconsistencies may appear. This function finds all the information in a network that follows a specific pattern. The structure of the query graph must match that of the data graph. To find the information query function, look for a one-to-one mapping between the query graph and the data graph.

4 ILLUSTRATION ON ML ROBUSTNESS EVALUATION

The utilization of keyword-based queries has become a prevalent methodology for enabling non-technical users to access expansive RDF data sets. At the present time, the user is able to select an engineering activity within the graph that utilizes the end-to-end methodology, and the underlying knowledge will then be presented to them. The ML Engineering BoK is a trustworthy ML end-to-end engineering guideline that engineers should follow throughout their activities. It is based on a comprehensive set of descriptions, engineering knowledge, metrics, and key performance indicators, which are capitalized in the BoK. These elements enable engineers to assess both functional and non-functional properties. For example, an engineer is seeking information on the assessment of ML model robustness in the context of the activity of evaluating an ML model in order to analyze and characterize the system's sensitivity to changes in the input, with a view to determining its overall resilience. For this engineering activity, the BoK suggests a strategy made of two successive phases: 1) Robustness test by sampling and perturbation (empirical evaluation) and 2) Formal verification of robustness (formal evaluation). Each step is described by a Capela model and a textual content. It consists in selecting the most appropriate tool for this robustness test. To make this selection, the key criteria are: the ML Model Algorithm, the type of data (images, time series, and language), the type of perturbation and its intensity (examples of perturbation include image luminance, image blurring, geometric transformation of plane position), and the target level of robustness. There may be different tools for data perturbation and for execution of the test., or it can be the same tool. With the selected tool, the specified perturbation is applied on the Test Dataset and the ML Model is executed on this perturbed dataset and the resulting behavior of the ML Model is captured.

5 CONCLUSIONS

The objective of building a Trustworthy ML Engineering Knowledge Graph is to facilitate more effective specification, design, comprehension, monitoring, and maintenance of ML-based systems for system design engineers and ML-based system operations personnel. Ultimately, this should enhance safety, cyber security, reliability, and performance, while also improving availability. The construction of a reliable ML engineering framework entails the utilization of an array of data sources and artificial intelligence methodologies, encompassing knowledge representation, knowledge graphs, semantic networks, high-level information fusion, graph theory, and numerous other techniques. Confiance.ai's methodological contributions span the entire development process of an ML-based system, from initial specification and design through to the commissioning and subsequent supervision of operational deployment, and even to the embedding of the latter in other systems. These contributions are manifold and include:

- A taxonomy used in trustworthy AI;
- A complete documentation of the process, including modeling of activities and roles, with elements enabling corporate engineering departments to implement it;
- A first development of a Trustworthy AI ontology, linking the main concepts of the process and the taxonomy;
- And a "Body-of-Knowledge" which brings together all these elements and makes them accessible on the website of the same name.

Furthermore, the ML-engineering BoK provides support to stakeholders across the ML value chain, offering invaluable assistance in the elicitation, validation, and verification of safety and cyber-security relevant quality attributes. Based on the Confiance.ai end-to-end methodology, it is able to guarantee that the heterogeneous requirements of stakeholders are met, thereby further consolidating its status as a fundamental element in the field of safety within the context of Machine Learning. While the deployment of these methodological instruments does not inherently guarantee the compliance of ML-based systems with regulatory requirements, it may serve as a basis for justifying such compliance, particularly in light of the prevailing standards set by the notified bodies responsible for verification. Furthermore, this ML Engineering BoK serves as a valuable resource in addressing the following key challenges:

- How to design AI models, so that, by construction, they satisfy trustworthy properties (accuracy, robustness, etc.)?
- How to characterize these AI models, for example, to understand and explain their behavior and their adequacy to the operational domain?
- How to implement and embed those AI models on hardware, by making them fit for the target with-

out losing their trustworthy properties.

- What are the data engineering method to apply in order to manage important volumes of data, account for the evolution of the operational domain, etc.?
- What are the appropriate verification, validation, and certification processes to consider for AI-based systems?

ACKNOWLEDGEMENTS

This work has been supported by the French government under the ”France 2030” program, as part of the SystemX Technological Research Institute within the *Confiance.ai* Program (www.confiance.ai).

REFERENCES

Adedjouma, M., Alix, C., et al. (2022). Engineering dependable ai systems. In *2022 17th Annual System of Systems Engineering Conference (SOSE)*, pages 458–463. IEEE.

Awadid, A., Le Roux, X., et al. (2024). Ensuring the Reliability of AI Systems through Methodological Processes. In *The 24th IEEE International Conference on Software Quality, Reliability, and Security*.

Chen, Z., Wang, Y., et al. (2020). Knowledge graph completion: A review. *IEEE Access*, 8:192435–192456.

Ehrlinger, L. and Wöß, W. (2016). Towards a definition of knowledge graphs. *SEMANTiCS (Posters, Demos, SuCCESS)*, 48(1-4):2.

Harnad, S. (1990). The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3):335–346.

Heist, N., Hertling, S., et al. (2020). Knowledge graphs on the web–an overview. *Knowledge Graphs for eXplainable Artificial Intelligence: Foundations, Applications and Challenges*, pages 3–22.

Hertz, S., Olof-Ors, M., et al. (2019). Machine learning-based relationship association and related discovery and search engines. US Patent 10,303,999.

Ji, S., Pan, S., et al. (2021). A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems*, 33(2):494–514.

Laudy, C. (2011). Semantic knowledge representations for soft data fusion. *Efficient Decision Support Systems-Practice and Challenges from Current to Future*.

Laudy, C., Ganascia, J.-G., and Sedogbo, C. (2007). High-level fusion based on conceptual graphs. In *2007 10th International Conference on Information Fusion*, pages 1–8. IEEE.

Li, B. et al. (2017). Applications of artificial intelligence in intelligent manufacturing: a review. *Frontiers of Information Technology & Electronic Engineering*, 18(1):86–96.

Mattioli, J., Laudy, C., et al. (2022). Body-of-knowledge development by using artificial intelligence. In *17th IEEE System of Systems Engineering Conference (SOSE)*, pages 14–19.

Mattioli, J., Tachet, D., et al. (2024). Leveraging Knowledge Graph to design the Machine-Learning Engineering Body-of-Knowledge. In *IEEE International Conference on AI x Science, Technology, and Technology (AIxSET)*, Laguna hills, United States.

Meyer, L., Stadler, C., et al. (2023). LLM-assisted knowledge graph engineering: Experiments with ChatGPT. In *Working conference on Artificial Intelligence Development for a Resilient and Sustainable Tomorrow*, pages 103–115. Springer Fachmedien Wiesbaden Wiesbaden.

Mugnier, M.-L. and Chein, M. (1992). Conceptual graphs: Fundamental notions. *Revue d'intelligence artificielle*, 6(4):365–406.

Ören, T. (2005). Toward the body of knowledge of modeling and simulation. In *Interservice/industry training, simulation, and education conference (I/ITSEC)*, volume 2005.

Robert, F., Abran, A., and Bourque, P. (2002). A technical review of the software construction knowledge area in the swebok guide. In *10th International Workshop on Software Technology and Engineering Practice*, pages 36–42. IEEE.

Shapiro, S. C. (2006). Knowledge representation. *Encyclopedia of cognitive science*.

Sowa, J. F. (1976). Conceptual graphs for a data base interface. *IBM Journal of Research and Development*, 20(4):336–357.

Sowa, J. F. (2000). Guided tour of ontology. *Retrieved from*.

Straková, J., Straka, M., and Hajič, J. (2019). Neural architectures for nested ner through linearization. *arXiv preprint arXiv:1908.06926*.

Wang, C., Liu, X., et al. (2022). Deepstruct: Pretraining of language models for structure prediction. *arXiv preprint arXiv:2205.10475*.

Zeng, X., Zeng, D., et al. (2018). Extracting relational facts by an end-to-end neural model with copy mechanism. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 506–514.