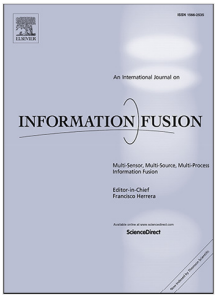
Journal Pre-proof



Explainable knowledge graph embeddings for industrial process monitoring & control

Michael Weyns, Thibault Blyau, Bram Steenwinckel, Filip De Turck, Sofie Van Hoecke, Femke Ongenae

PII: S1566-2535(25)00315-X
DOI: https://doi.org/10.1016/j.inffus.2025.103242
Reference: INFFUS 103242

To appear in: *Information Fusion*

Received date : 29 December 2024
Revised date : 11 March 2025
Accepted date : 17 April 2025

Please cite this article as: M. Weyns, T. Blyau, B. Steenwinckel et al., Explainable knowledge graph embeddings for industrial process monitoring & control, *Information Fusion* (2025), doi: https://doi.org/10.1016/j.inffus.2025.103242.

Highlights

Explainable Knowledge Graph Embeddings for Industrial Process Monitoring & Control

Michael Weyns, Thibault Blyau, Bram Steenwinckel, Filip De Turck, Sofie Van Hoecke, Femke Ongenae

- Hierarchical knowledge graph (KG) to unify process domain knowledge & data insights
- Event-specific embedding algorithm for numerical features & long-range sequences
- Explanations identifying & visualising the most salient embedded features on the KG
- Evaluations performed on real-world use cases in the chemical engineering domain
- Empirically shown improved performance with respect to traditional machine learning

Explainable Knowledge Graph Embeddings for Industrial Process Monitoring & Control

Michael Weyns[a,*], Thibault Blyau[a], Bram Steenwinckel[a], Filip De Turck[a], Sofie Van Hoecke[a], Femke Ongenae[a]

[a] *IDLab, Ghent University - imec, Technologiepark-Zwijnaarde 126, Gent, B-9052, Belgium*

Abstract

AI-driven solutions are being employed in process monitoring and control to learn typical system behaviours under various conditions based on historical data. However, they are unable to take advantage of the rich, tacit domain expertise of experienced process engineers pertaining to these behaviours. Hybrid AI solutions aim at fusing domain knowledge into machine learning models, but have so far been limited to specific industrial subdomains or applications and support only domain expertise in the form of equations. We propose an explainable hybrid AI methodology that can integrate any kind of tacit knowledge in an interpretable manner. First, we introduce a method to consolidate process data and domain-specific expertise in a generic fashion using Knowledge Graphs. Second, we propose a Knowledge Graph transformation technique to better capture the sequential aspects of a process and an accompanying white-box Knowledge Graph embedding technique that allows us to integrate domain knowledge directly into the feature space of a data-driven model. Third, we show how our methodology can be combined with explainability techniques, such as SHAP, to highlight directly in the graph which paths contributed most to the AI-driven decision. Our methodology has been evaluated on two real-world chemical engineering use cases. It outperforms data-driven baselines on all performance metrics, with average

*Corresponding author

Email addresses: michael.weyns@ugent.be (Michael Weyns), thibault.blyau@ugent.be (Thibault Blyau), bram.steenwinckel@ugent.be (Bram Steenwinckel), filip.deturck@ugent.be (Filip De Turck), sofie.vanhoecke@ugent.be (Sofie Van Hoecke), femke.ongenae@ugent.be (Femke Ongenae)

improvements of up to 8.57% and 10.21%.

Keywords:
Knowledge graph, Knowledge graph embedding, Process monitoring, Process control, Industry 4.0, Hybrid AI, XAI

1. Introduction

With the advent of what has been called the fourth industrial revolution (Industry 4.0), traditional forms of industrial process management are poised to undergo a substantial revolution, as they become increasingly exposed to a host of new technologies, ranging from smart sensors and the Internet of Things (IoT) to various forms of Artificial Intelligence (AI) [1]. Prime examples of applications that are set to be revolutionised include predictive maintenance, asset tracking, inventory management, quality control, production process monitoring, energy efficiency management, safety monitoring, and supply chain optimisation, among others [1]. For most of these areas of application, Industry 4.0 demands both the passive and routine disclosure of extra information to process managers (i.e., process monitoring) and the active and semi-automatic exploitation of said information for the sake of improving the underlying processes being monitored and managed (i.e., process control).

For example, when considering inventory management, IoT sensors can be used to accurately and dynamically track inventory characteristics (e.g., stock levels, locations of different pieces of equipment), but the same data gathered through such sensors can also be used to predict or forecast future inventory demands, such that managers can more efficiently and effectively adapt their inventories to real conditions. Alternatively, consider a straightforward predictive maintenance case, where sensors may be used to measure certain chemical properties of a product being manufactured, and predictive models can learn from historical measurements to determine whether manufacturing is proceeding as desired. Many more examples abound, with a whole host of different industrial manufacturing processes—from cutting, drilling and welding processes to additive and multi-stage manufacturing and assembly processes—all providing opportunities to develop advanced process monitoring and control capabilities [2].

Process monitoring & control (PMC) encompasses most of the activities associated with the proliferation of principles and technologies involved in the

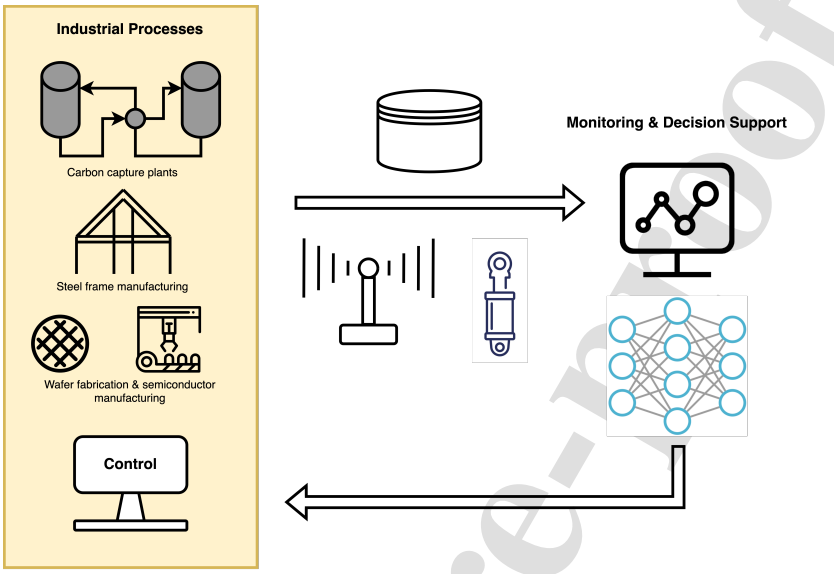


Figure 1: Schematic overview of several examples of ML-driven PMC/PSE applications. In particular, emissions forecasting in carbon capture plants; online inspection for screw-fastening operations in light-gauge steel frame manufacturing; and fault detection and diagnosis in semiconductor manufacturing are all examples of ML being applied to enable PMC/PSE for specific industrial processes [3, 4, 5]. In each case, the same overall framework of data-driven decision support is in effect.

Industry 4.0 paradigm. Notably, PMC overlaps to a significant degree with what is sometimes called Process Systems Engineering (PSE), although the latter is usually positioned more specifically within the domain of chemical engineering, while the former should be understood to apply more broadly [6]. Within the scope of Industry 4.0, both PMC and PSE stand to gain performance-wise from the inclusion of machine learning (ML) algorithms, which are advantaged with respect to traditional modelling techniques by being able to exploit the increasing availability of sensor data in modernised industrial settings [1, 6, 7]. Much effort has already gone towards leveraging ML solutions for the purpose of improving PMC/PSE applications in various subdomains, from emissions or energy demand forecasting to energy consumption reduction [8, 3, 9, 10]. Figure 1 can be consulted for a schematic overview of a small selection of industrial processes where ML-driven solutions for PMC/PSE have already been proposed.

Nonetheless, considerable outstanding challenges remain. Specifically,

one of the key challenges impeding the adoption and integration of ML solutions in the domain of PMC/PSE relates to the implicit trade-off between data-driven approaches and older mechanistic, expert-driven modelling solutions (also referred to as *first principles models*) that better account for operational deviations and system-specific dynamics [6]. First principles models are typically built on mechanistic domain knowledge encoded predominantly in the form of mathematical equations or physical constraints, as in e.g., the work by Azarpour et al., where industrial fixed-bed catalytic reactors are modelled using a set of energy balance and species mass balance equations under various initial and boundary conditions [11, 12, 13]. Sometimes, although more rarely, domain knowledge is also captured in process flowsheets, as in e.g., the work of Vedam et al., where signed digraphs reflecting non-transient relations between the different variables of a chemical process are used to assist in multiple fault diagnosis [14]. While ML solutions are able to make use of the high volumes of data made available by Industry 4.0 trends, they are not straightforwardly capable of harnessing *a priori* domain expertise. To remedy this, we can turn to a kind of approach that is typically referred to as hybrid modelling or hybrid AI [12]. This type of approach tries to leverage the benefits of ML while simultaneously making use of industry insights tacitly maintained by human experts. There already exists a substantial body of research on this type of approach, but existing solutions are currently limited in various ways [12, 15]. Most existing hybrid modelling approaches to PMC/PSE are highly domain-specific and are thus usually tailored towards particular subdomains or even individual applications. Indeed, most existing ways of combining data-driven and expert-driven approaches operate in one direction: they usually try to improve the latter with the former, without considering more generic solutions that involve integrating the former into the latter [12]. This hinders the adaptation of these approaches to alternative use cases and domains, as they are centred around case-specific first principles models. The few approaches that have explored the latter option are usually limited to expert-driven architecture selection, loss function optimisation, and architecture initialisation, which only leverage expert knowledge indirectly insofar as it can be used to influence the learning procedure; for the most part, knowledge is not used directly as additional input or as input augmentation, such that many forms of knowledge will not be harnessed to their full potential [16]. Relatedly, most prior knowledge leveraged by existing approaches is captured in equations, thus excluding knowledge that is of a more qualitative nature, e.g., workflows, taxonomies [12, 17].

Within this broader context, Knowledge Graphs (KGs) have emerged as a promising way of capturing both quantitative and qualitative expert knowledge for various PMC/PSE task domains. KGs have specifically been singled out as as a potential *default data model* for heterogeneous data in general, and as a way of meaningfully integrating and contextualising the complex, fragmented data landscape disclosed by PMC/PSE in particular [18, 19]. Given the prominence and centrality of (industrial, manufacturing) pipelines and procedures in PMC/PSE contexts, using KGs to model process knowledge is especially promising [19]. In fine, whether it be predictive maintenance, digital twins, fault diagnosis, supply chain management, or job scheduling, KGs are already operative within myriad PMC/PSE activities, with knowledge fusion as one of the core unifying factors underlying their adoption [20, 21, 22, 23]. However, at the same time, KGs have made only limited inroads into PMC/PSE-specific hybrid AI, as they are currently used mainly as knowledge repositories [24, 25].

Usually framed within the broader Semantic Web paradigm, KGs are multigraphs interlinking named entities with labelled, directed edges, in order to facilitate human-machine interoperability using formal semantics [26]. Attempts at leveraging these multigraphs for the sake of enhancing ML tasks usually rely on KG embedding techniques to translate graph structures into numerical vector representations, as these are more readily digestible by ML models [27, 26]. As of yet, KG embeddings have, to our knowledge, not yet been used to enable hybrid AI for PMC/PSE, even as KGs themselves are becoming more prominent in this domain, and would be able to address many of the outstanding shortcomings in existing hybrid solutions. As it stands, KG embeddings are only seldom applied to embed process knowledge, which is precisely the kind of knowledge that is of primary relevance to PMC/PSE domains, as most embedding techniques are designed for generic forms of knowledge graph construction and curation [26]. Also, most KG embeddings are latent feature models, meaning that the individual dimensions of the vector representations obtained after translation are opaque in terms of what they refer to. Latency makes it difficult to elicit meaningful interpretations from ML decisions made based on embedded inputs. Since achieving a greater degree of decision transparency is a well-known challenge when applying ML in the context of PMC/PSE, this poses a problem that only few embedding approaches would be able to address, prompting the need for explicitly explainable approaches [6, 7].

Responding to these challenges, this paper presents a novel approach to

explainable hybrid modelling for PMC and PSE, which relies on KGs and KG embeddings to integrate process knowledge directly into the feature space of the historical data used to train a data-driven model for a downstream task. Specifically, the contributions of this paper are as follows:

- We rely on a general-purpose process ontology to formally represent process knowledge potentially originating from diverse PMC/PSE subdomains.
- We design a novel methodology based on the Track Activities by Linking Knowledge (TALK) [28] paradigm to combine process knowledge and historical data in a single KG-based data structure. To our knowledge, fusing data and processual knowledge in this manner has not been attempted before.
- We propose a generic method for hybrid AI in the context of PMC/PSE by relying on the Instance Neighbouring by using Knowledge (INK) [29] KG embedding technique in order to generate vector representations of individual process events inside a TALK graph in function of their contextual characteristics. These representations are then used to train a data-driven model on a downstream PMC/PSE task. Different from most other embedding techniques, INK elicits numerical representations with transparent dimensions, thus facilitating explainable decision-making down the line.
- We enable explainability by efficiently combining SHapley Additive exPlanations (SHAP) [30], a widely used, game-theoretic approach to ML-based explainability, with INK embeddings. As such, we are able to make individual process forecasts, or other kinds of model decisions, more transparent, as we can map salient features onto the TALK graphs from which they were extracted.
- We evaluate our methodology on two separate, real-world use cases situated in the chemical engineering domain. Both of these use cases require us to make forecasts about the further evolution of a given production process, for which both real historical data and various kinds of procedural knowledge have been made available by industrial partners, P&G and Allnex.

The rest of this research paper is structured as follows. In Section 2, we present a brief recapitulation of the existing research on hybrid modelling in PMC/PSE, with a particular emphasis on prior work involving the use of KGs. We also provide a brief overview of the research on KG embeddings, which in recent years has grown into a comprehensive field of its own. In Section 3, we provide background information regarding the techniques and ontologies we will be relying on when defining our proposed methodology. Following this, Section 4 contains everything pertaining to the novel hybrid methodology put forward by this research, outlining the generic, explainable hybrid AI pipeline required to solve individual PMC/PSE problems. Section 5 gives an overview of the two chemical engineering use cases and the evaluation setup, and lists all the results for various test settings. Section 6 draws conclusions from these results. Finally, Section 7 offers an overall reflection on what has been accomplished, by assessing the merits of the research, while Section 8 concludes the paper as a whole and discusses what will be addressed in future work.

2. Related Work

As discussed in the previous section, many hybrid modelling approaches already exist, but these suffer from a number of shortcomings. In particular, very few techniques offer a generic methodology to augment data-driven approaches with domain-specific expertise subsuming various modalities [12]. Most techniques that do leverage domain knowledge to augment data-driven approaches are limited to specific architectural alterations, such that they may not be generally applicable or even straightforwardly transportable to different applications [31, 32, 33]. Because of these limitations, we have singled out KGs to facilitate hybrid AI for PMC/PSE, insofar as their central characteristics, namely, their flexible, distributed architectures combined with a formal overlay to integrate across different domains, are eminently suited to networked, heterogeneous PMC/PSE settings, and can be combined with data-driven ML solutions via KG embeddings [19].

2.1. KGs & KG Generation for PMC/PSE

To enable a unified view of both knowledge and data, KGs make use of a two-tiered representation, namely, a TBox and an ABox [34]. A KG's TBox is also referred to as an ontology, and is comprised of classes of individuals, relationships between classes, and various universally quantified, i.e.,

axiomatic, assertions relating to these classes and relationships. The ABox, on the other hand, comprises a set of assertions and facts associated with a specific instantiation of a given domain, expressed in terms of the concepts and roles laid out in the ontology. Overall, KGs can be used to facilitate data integration by unifying a plurality of heterogeneous data sources under overarching semantic schemata [19].

Various techniques have been developed to generate KGs suited specifically to PMC/PSE use cases. Zhou et al. have developed a methodology to extract KGs from documents describing various aspects of industrial processing using a Bidirectional Long Short-Term Memory (BiLSTM) model [35]. Wen et al. confirms our intuition that process knowledge, while often tacitly available in the minds of engineers and managers, is ideally captured formally, and that KGs are well suited to this purpose [36]. They have developed a methodology to extract a manufacturing process KG from existing manufacturing process texts by making use of text templates, fine-grained topic recognition, and a general-purpose domain ontology for process knowledge. Similarly, Han et al. have set up a complex pipeline involving various textual embedding models, graph neural networks, and reinforcement learning to automatically extract fault diagnosis KGs [37]. Also operating within the domain of fault diagnosis (more specifically, root cause analysis), Steenwinckel et al. have developed knowledge extraction methodologies based on mapping scripts to integrate risk documents and raw sensor data using a dedicated FOLIO domain ontology [38, 39]. By contrast, Liang et al. suggest a more manual approach, using the open-source editor Protégé to construct a papermaking process ontology [40]. Finally, Bachhofner et al. rely on the BPMN-based Ontology (BBO) together with RDF Mapping Language (RML) rules to convert XML-based BPMN diagrams into industrial process KGs. While each of these techniques focus on knowledge extraction and construction for PMC/PSE use cases, they do not make use of the constructed KGs to improve model performance on specific downstream PMC/PSE tasks.

Besides PMC/PSE-specific KG generation techniques, some work also tries to incorporate KGs more directly in PMC/PSE solutions. For example, Banerjee et al. rely on KGs to facilitate a question-answering mechanism for large-scale production data [24]. The KGs in question are used to link production data to a number of predefined concepts and relations. Using a Path Ranking Algorithm, graph queries can be translated to edge sequences between the source and target nodes extracted from each query. The question-answering mechanism developed by Banerjee et al. is a kind of

digital twin, in that it can provide an in-depth query-based analysis of the specific target domain being modelled (in this case, an industrial production line), but cannot be used to solve any downstream tasks by itself apart from the query-based framework. Along similar lines, Stavropoulou et al. have proposed a way to augment digital twins with KGs, specifically by utilising KGs as a flexible repository of process configurations and execution orders which can be accessed dynamically by the corresponding digital twins [25]. While this approach was designed with a more general applicability in mind (i.e., the methodology should be extensible to different use cases with minimal effort), the KG is not directly used to improve task performance. To our knowledge, little to no work has been done to investigate how KGs might be used to enable hybrid AI for PMC/PSE by directly incorporating process knowledge, as we are trying to do in this paper.

2.2. KG Embeddings for PMC/PSE

Having discussed prior work focusing on KG generation and KG-based solutions for PMC/PSE, we now turn towards KG embeddings, which are a central part of the hybrid methodology put forward in this paper. KG embeddings have already been shown to outperform other kinds of graph encoding methods on various machine learning tasks, thus offering a solid mechanism to enable KG-based hybrid AI, and are part of an extensive literature within the broader domain of statistical relational learning [41].

When trying to integrate process knowledge, it is important that we employ the most appropriate embedding technique for that kind of knowledge. In general, KG embeddings are used to "propositionalize" the various substructures of a KG. This is useful because KGs are effectively semantic networks, composed of interconnected representatives of real-world entities that are not directly usable by existing machine learning approaches. As such, KG embedding techniques are often used to translate elements inside a KG (or even entire KGs) into numerical vectors embedded in Euclidean space. It should be noted that this translation is by no means straightforward or even necessarily feasible. By design, embeddings are a approximate effort, as there is no guarantee that a clear, non-reductive, one-to-one mapping between networked entities and their corresponding embeddings even exists [41]. KG embeddings are thus always to some extent heuristic in nature, implying a probable loss of knowledge where the end result is concerned.

Most KG embeddings operate under the assumption that all elements inside the KG are conditionally independent given a limited set of latent

Embedding model	Type	Interpretable	Literals	Long-range sequences	Scoring function $f(x_{ikj}, \Theta)$
TransE	node + edge				$\|v_{e_i} + v_{p_k} - v_{e_j}\|$
PTransE	node + edge			✓	$E(e_i, p_k, e_j) + E(e_i, P, e_j)$
RTransE	node + edge			✓	$\|s_n(e_i, \{p_i\}_{i=1}^n, e_j) - v_{e_j}\|$
RESCAL	node + edge				$v_{e_i}^T W_p v_{e_j}$
DistMult	node + edge				$\langle v_{e_i}, v_{p_k}, v_{e_j} \rangle$
ComplEx	node + edge				$\mathrm{Re}(\langle v_{e_i}, v_{p_k}, \overline{v_{e_j}} \rangle)$
SimplE	node + edge	✓			$\frac{1}{2}(\langle v_{e_i}, v_{p_k}, v_{e_j} \rangle + \langle v_{e_j}, v_{p_k^-1}, v_{e_i} \rangle)$
R-GCN	node + edge			✓	$v_{e_i}^T D_p v_{e_j}$
TransGCN	node + edge			✓	$\|v_{e_i}^{(l)} + v_{p_k}^{(l)} - v_{e_j}^{(l)}\| \,/\, \|v_{e_i}^{(l)} \odot v_{p_k}^{(l)} - v_{e_j}^{(l)}\|$
RDF2Vec	node + edge			✓	N/A
TraceWalk	node + edge			✓	N/A
Graph2Vec	subgraph + graph			✓	N/A
INK+TALK	node + edge	✓	✓	✓	N/A

Table 1: Overview of different embedding approaches in terms of their most relevant characteristics from the standpoint of PMC/PSE.

variables [26, 41]. These latent variables or features are usually associated with the subjects, objects, and relationships already populating the KG, i.e., the set of observable facts. Formally, this means each latent feature model's probability model can be described in terms of $\mathrm{P}(Y \mid T, \Theta)$, where $Y \subseteq \{0,1\}^{N_e \times N_p \times N_e}$, N_e is the total number of named entities, N_p is the total number of labelled edges, T is the set of observable facts, and Θ is the model parameter set. Each latent feature model defines a scoring function $f(x_{ikj}; \Theta)$ to estimate $\mathrm{P}(Y)$, with x_{ikj} a *possible* fact inside the KG.

More intuitively, the latent feature approach implies that observable facts are explained in terms of features not explicitly observed inside the KG. For example, the fact that the novelist Thomas Mann won the Nobel Prize in 1929 may be attributed to the fact that he is a good author. Here "being a good author" is a latent entity feature, while "won the Nobel Prize in 1929" is an observable fact. We should note that latent features, because they are latent, are by definition unobservable, opaque—thus hampering their interpretability. This opaqueness is undesirable when it comes to designing explainable hybrid AI, and thus plays an important role when selecting for the optimal embedding approach.

Apart from transparency or observability, which is paramount when trying to enable explainable decision-making, PMC/PSE process graphs impose a number of additional constraints on prospective embedding techniques: namely, that they should support numerical literals (e.g., pertaining to sensor readings), and that they should be able to capture long-range sequential knowledge. On the whole, there are many different embedding techniques, each of which has its own distinct advantages and disadvantages and each

of which is appropriate to different KG substructures, e.g., nodes, relations, subgraphs, etc. First, distance-based models mostly derive from the original TransE model developed by Bordes et al [42]. Although they continue to attract attention, having been applied more recently to e.g., facilitate cross-domain embeddings in a federated learning setting [43], such models are generally incapable of capturing long-range dependencies (with a few exceptions like e.g., PTransE and RTransE [44, 45]), are not natively equipped to incorporate numerical sensory information, and, like most latent feature models, are not interpretable—all of which are unfortunately characteristics that one would desire when trying to capture process knowledge. Second, multiplicative models, including bilinear models like RESCAL, DistMult, and ComplEx, suffer from many of the same shortcomings as distance-based models [41, 46, 47]. Except for SimplE [48], most of these techniques are not interpretable. They are also incapable of capturing long-range dependencies or encoding (numerical) literals. Next, there are the neighbourhood-based models, including the Relational Graph Convolutional Network (R-GCN), TransGCN, RDF2Vec, TraceWalk, and Graph2Vec [49, 50, 27, 51, 52]. This last category is much better at capturing long-range (neighbourhood) dependencies, but often still lacks native support for literals, and typically remains unintelligible. Finally, it should also be mentioned that certain embedding techniques have made attempts to integrate time-sensitive or temporal information by e.g., adding time constraints to relationships representing topological connections [53]. As industrial processes usually reflect time-sensitive operations or, as mentioned previously, sequences of discrete execution steps, integrating temporality is also desirable.

With regard to the neighbourhood-based category, it should be noted that TraceWalk was specifically designed to embed process knowledge, offering native support in modelling different process execution branches. However, TraceWalk still suffers from the same shortcomings as the other neighbourhood-based models. By contrast, INK [29] together with TALK [28] also belongs to the neighbourhood-based category, but additionally offers native support for literals and is also intrinsically interpretable. By adhering to an event-centric graph organisation paradigm, TALK also allows us to capture the temporal aspects involved in process modelling. In conclusion, INK+TALK most closely approximates the requirements for capturing process knowledge out of the embedding techniques that were surveyed. For an overview of the various embedding approaches discussed so far, refer to Table 1. Here, the final column pertaining to each embedding technique's scoring function

is only relevant to those models that need to attribute scores to individual triples, usually to perform the link prediction task. As such, some of the surveyed embeddings do not come equipped with a specific scoring function, in which case the entry was marked as not applicable.

2.3. KG-based Explainability for PMC/PSE

To round off this section, we should also note the work that has already been done on the topic of KG-supported explainability or interpretability, especially within the scope of PMC/PSE, given that we ourselves aim to facilitate decision-level explainability by relying on KGs. Based on what we were able to find, this field is still in its infancy. Rožanec et al. have proposed using a KG specifically to improve explanations within the domain of demand forecasting [54]. The KG in question is used mainly as a repository to combine process knowledge and decision explanations, i.e., as support for feedback collection, though it is not used to improve the explanations directly. In a separate work, the same authors build on their previous insights and attempt to use the concepts encoded in the KG (as well as some external sources) to enrich model explanations [55]. This latter approach is similar to what we are proposing, in that KGs are used to enrich explanations based on feature rankings obtained via some explainability technique, and make them more intelligible to end users, i.e., engineers and process managers [55]. However, while the authors focus mostly on demand forecasting and separate the semantic enrichment and explanation modules, we offer a more generalisable and integrated approach, where the features being ranked are already semantically enriched, because the domain expertise captured by the KG is also used to improve prediction performance before explanations are even generated.

3. Background

As noted earlier in Section 1, our methodology is tailored towards a broad range of process monitoring and control problems. Similar to the work of Stavropoulou et al., our methodology is extensible to various particular applications of a generic manufacturing process [25]. In what follows, we will take a closer look at the specific KG embedding approaches, namely INK and TALK, that will be used to integrate domain expertise and historical data in order to perform a downstream PMC/PSE task. We will also take a look

at the kind of knowledge we will need to embed in order to formally model manufacturing processes.

3.1. INK Embeddings

As mentioned in the previous section, INK is a neighbourhood-based, white-box KG embedding model [29]. It can be used to extract an interpretable vector encoding for different elements inside a KG. As an example, consider the graph depicted in Figure 2. Here, two different atoms, e.g., Hydrogen and Oxygen, have been linked to the compound they compose, e.g., Water. The atoms and the compound are each accompanied by various pieces of contextual information, e.g., group, density at standard temperature, phase at standard temperature, etc. These are modelled as properties and organised in a graph structure.

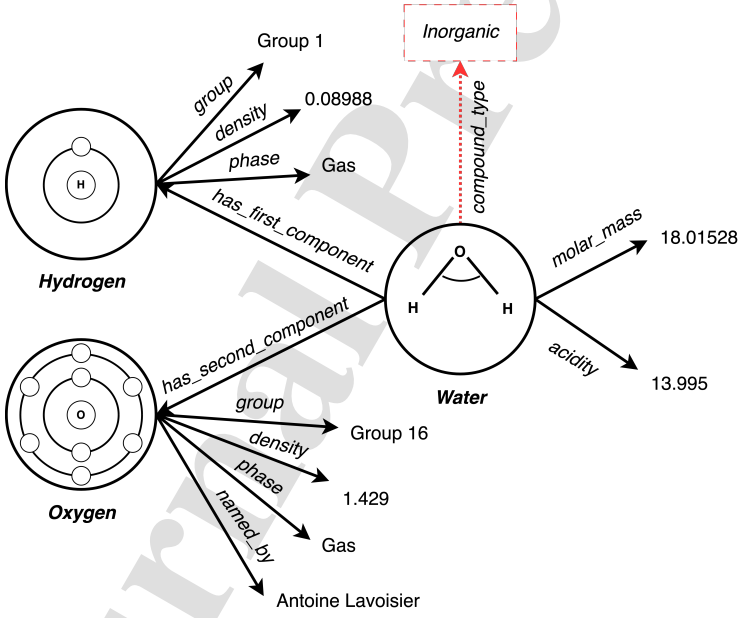


Figure 2: Informal knowledge graph linking atoms with compounds for the sake of classifying compound types, e.g., organic or inorganic. Here, the directed, labelled arrows signify relationships between different entities, e.g., *has_first_component*, or properties of entities, e.g., *density* and *acidity*.

The graph depicted in Figure 2 can be represented as a list of triples or subject-predicate-object tuples, with each triple expressing an atomic fact inside the graph. Table 2 shows what this alternative representation looks

like. For each of the three core entities inside the graph, i.e., the two atoms and the compound, we have a number of different properties, each of which can be associated with an object. In some cases the object is a literal value, e.g., "18.01528"^^xsd:float, usually accompanied by a specific literal type, e.g., the molar mass is encoded as a floating-point number. In other cases, the object is another entity, e.g., <Hydrogen>.

Subject	**Predicate**	**Object**
<Hydrogen>	<group>	<Group 1>
<Hydrogen>	<density>	"0.08988"^^xsd:float
<Hydrogen>	<phase>	"Gas"^^xsd:str
<Oxygen>	<group>	<Group 16 >
<Oxygen>	<density>	"1.429"^^xsd:float
<Oxygen>	<phase>	"Gas"^^xsd:str
<Oxygen>	<named_by>	<Antoine Lavoisier>
<Water>	<has_first_component>	<Hydrogen>
<Water>	<has_second_component>	<Oxygen>
<Water>	<molar_mass>	"18.01528"^^xsd:float
<Water>	<acidity>	"13.995"^^xsd:float

Table 2: Triple-based representation of the knowledge graph depicted in Figure 2. Here, <...> is used to indicate entities and relationships, and literal objects are rendered as a value followed by an XML Schema Definition (XSD) literal type, e.g., *xsd:float* or *xsd:str*.

To embed one of the entities in the graph with INK, we must set up a neighbourhood dictionary, which will be used to create a binary encoding of the entity in question expressed in terms of its neighbourhood information. The first column in Table 3 shows some of the entries in the neighbourhood dictionary for <Water> up until extraction depth 2. (Note that this is the maximum depth for the graph displayed in Figure 2.) Paths are extracted by starting from the entity <Water> and travelling down the various predicates associated with this entity up to the maximum depth. Nested paths encountered along the way are encoded as "prediction_depth_1.prediction_depth_2" (the individual predicates are separated by a full stop, and any nodes encountered along the way are ignored). The value encountered at the end of any given (nested) path is given between brackets. The neighbourhood dictionary also contains entries indicating whether a certain relationship is present at all; in these cases, the value encountered between brackets at the end of a given path is "True". This is done to ensure that different neighbourhoods can be compared on a predicate level, without reference to specific

tail nodes. Note that we cannot construct paths ending in "False" given that we can only extract from what is at hand, and that endless paths are absent from any given graph.

Dictionary entry	**Binary format**
has_first_component → [Hydrogen]	has_first_component§Hydrogen
has_second_component → [Oxygen]	has_second_component§Oxygen
has_first_component.group → [Group 1]	has_first_component.group§Group 1
has_second_component.group → [Group 16]	has_second_component.group§Group 16
has_first_component.density → [0.08988]	has_first_component.density§0.08988
has_second_component.density → [1.429]	has_second_component.density§1.429
acidity → [13.995]	acidity§13.995
has_first_component → [True]	has_first_component§True
has_second_component → [True]	has_second_component§True
has_first_component.group → [True]	has_first_component.group§True
has_second_component.group → [True]	has_second_component.group§True
has_first_component.density → [True]	has_first_component.density§True
has_second_component.density → [True]	has_second_component.density§True
acidity → [True]	acidity§True

Table 3: Neighbourhood dictionary (of depth 2) for the <Water> concepts in the graph of Figure 2 in two formats.

After this neighbourhood information has been extracted, the dictionary needs to be converted into a binary format (i.e., embedded). The second column in Table 3 shows the result of this transformation. Each value in the dictionary is appended to the associated nested path using a §-sign. The INK embedding of any given entity in the graph will simply correspond to the binary encoding of that entity. This binary encoding can be constructed by checking which fields in the binary format of the neighbourhood dictionary are valid for the given entity. For instance, for <Hydrogen> we know that "group§Group 1" is True, while "group§Group 16" is False (see Table 2). The first three rows in Table 4 show the resulting binary encodings for the primary elements in the graph.

The problem, however, with this approach is that it discards the literal information that was present in the original graph. To remedy this, INK distinguishes those relationships ending in a real value (e.g., "density§1.429" and "density§0.08988") from other relationships (e.g., "group§True" and

	density§0.08988	density§1.429	has_first_component.density§0.08988	has_second_component.density§1.429 ...
Hydrogen	1	0	0	0 ...
Oxygen	0	1	0	0 ...
Water	0	0	1	1 ...
	density_real_value		**has_first_component.density_real_value**	**has_second_component.density_real_value** ...
Hydrogen	0.08988		NaN	NaN ...
Oxygen	1.429		NaN	NaN ...
Water	NaN		0.08988	1.429 ...

Table 4: Fragment of a binary representation of two atoms and compound in the knowledge graph, represented in Figure 2, based on the dictionary of Table 3.

"has_second_component§Oxygen"). For the former it inserts the actual value (which will be NaN when the path is not present) instead of the binary encoding, while for the latter it retains the binary encoding. The last three rows in Table 4 demonstrate what this means for the "density" paths.

The final INK embedding of any given entity will contain a mix of binary and real-valued fields, which can all be traced back to different neighbourhood paths in the original graph. Note that these embeddings fulfil the primary requirements imposed by process KGs for PMC/PSE. By extracting neighbourhood information for every focal node, long-range, sequential information can be incorporated into the embedded representation. As shown in Table 4, each embedding also maintains the literal values encountered during neighbourhood extraction, such that sensory information can be accommodated. Finally, all embedding dimensions refer to observable features inside the neighbourhood dictionary, thus avoiding the opaqueness that makes most embedding techniques ill-suited for explainable hybrid AI.

3.2. TALK Graphs

TALK is a paradigm that can be used to structure event-based knowledge in a way that allows us to use INK embeddings to make knowledge-informed outcome predictions for successive events [28]. Figure 3 shows what a TALK graph could look like. In general, the TALK paradigm demands that subsequent events are linked to one another and that each event is described by a number of states, which are themselves, in turn, characterised by a number of observations. Applied to an industrial process, events might represent observations of certain chemical properties, which are contextualised by states representing an ongoing reaction or a set of manufacturing presets. Note that the states contextualising each event may vary over time; the same goes for the observations characterising each event state.

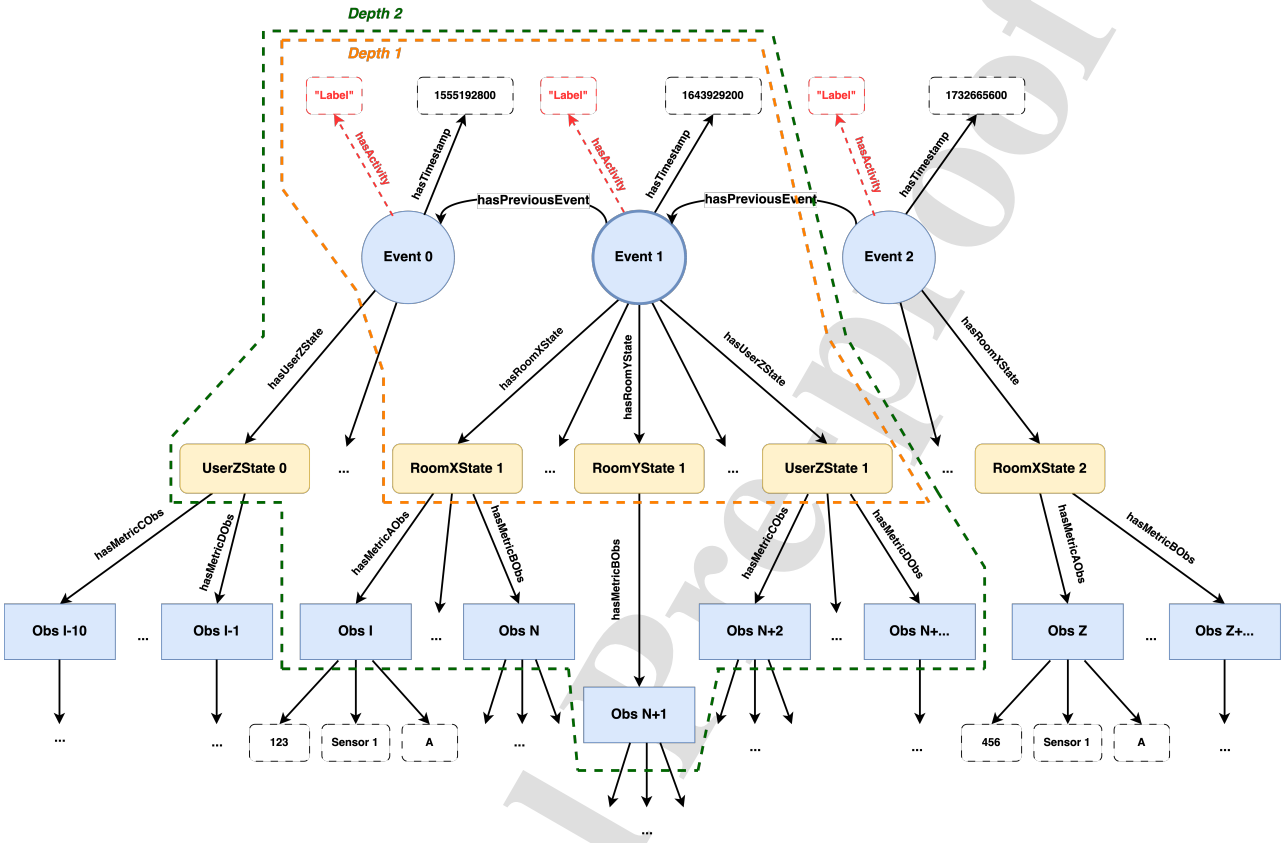


Figure 3: Abstract TALK graph, with neighbourhood depths 1 and 2 indicated for Event 1. As in Figure 2, directed labelled arrows signify relationships between different entities, e.g., *hasRoomXState*, or properties of entities, e.g., *hasTimestamp*. Each event entity is linked to the preceding event via the relationship *hasPreviousEvent*. Additionally, each event is contextualised by a number of states, e.g., *RoomXState 1* and *RoomYState 1*, which are in turn contextualised by observations, e.g., *Obs I* and *Obs N+1*.

Using INK, each event in the TALK graph can be associated with an interpretable vector encoding. Table 5 shows what the INK embeddings would look like for the first three events in the TALK graph of Figure 3. Here, an extraction depth equal to 3 was used to include observation values.

3.3. Modelling Process Knowledge

The TALK paradigm was originally devised to facilitate Human Activity Recognition (HAR) [28]. To apply TALK to the domain of HAR, Steen-

	hasUserZState	hasUserZState§UserZState1	hasUserZState.hasMetricDObs.hasValue	...	hasTimestamp
Event 0	1	0	Value		1555192800
Event 1	1	1	Value		1643929200
Event 2	0	0	NaN		1732665600

Table 5: INK embeddings for the event entities in the TALK graph, represented in Figure 3.

winckel et al. made use of an applied ontology based on the Smart Applications REFerence (SAREF) ontology, which is specifically tailored towards capturing the spatiality of sensory monitoring [28]. When transposed from HAR to PMC/PSE, TALK requires a different ontology to enrich the available data. Within the domain of process monitoring and control, events correspond to certain process behaviours that need to be predicted over time. To model this kind of domain knowledge, and with reuse of existing knowledge in mind, we propose an extension and combination of the OWL-S Process ontology on the one hand and the Semantic Sensor Network (SSN) ontology on the other hand [56, 57].

The OWL-S Process ontology is a sub-ontology of OWL-S and contains a relatively large number of control constructs needed to build complex process flows. Each process in the OWL-S Process ontology can also be associated with a number of variables that describe its pre- and post-conditions, as well as its internal state. As such, this ontology provides many of the core workflow concepts in a structure that is not overly complex. The SSN ontology supplements OWL-S by offering a comprehensive, yet intuitive suite of concepts and relationships for modelling everything related to sensor observations, samplings, and actuations. This includes temporal information about specific sensory actions, the value that was measured or inputted, what produced the action and by what procedure it was produced, as well as what about the process is being captured by the sensor in question.

Note that both OWL-S and SSN are recommendations made by the World Wide Web Consortium (W3C) and are recognised as standards within the Semantic Web community. As will be further illustrated in the next section, combining them offers a compendious toolkit with which to describe the generic architecture of a process workflow involving an interconnected sensor network, and in contrast to other comprehensive process ontologies like OntoCAPE, OPMW, and BBO, which comprise multilayered and complex frameworks, the proposed combination of OWL-S and SSN is relatively lightweight [58, 59, 60].

4. Methodology

As indicated before, we propose a hybrid AI methodology for PMC/PSE which leverages KGs to integrate both historical data and process knowledge in a single data structure. To accomplish this, we need a general-purpose manufactoring process ontology, the main concepts of which have been listed in Table 6, together with a succinct description for each[1]. To illustrate more intuitively what these concepts express and how they logically cohere, Figure 4 shows a workflow diagram visualising some of the primary components of the manufacturing ontology. Three different kinds of direct steps (pmc:ObservationStep, pmc:ActuationStep, and pmc:SamplingStep) have been linked together with the *nextStep* relationship, indicating that these steps are executed in sequence. The actuation step leads into a conditional step, which can either branch towards the observation step (via the *thenStep* property) or back to the actuation step preceding it (via the *elseStep* property). Either way, the process is a closed execution loop, encapsulated within a single module.

It should be noted that the manufacturing ontology presented here is a meta-ontology, containing abstract concepts that will need to be inherited by domain ontologies representing specific processes. As shown in Figure 5, a number of specific ontologies exist on the level just below the meta-ontology, each of which inherits from the concepts shared throughout the meta-ontology. These sub-ontologies can be instantiated with data belonging to the data sources of each respective domain. The main benefit of relying on a multi-level framework pertains to the generic nature of the meta-ontology, which can remain invariant throughout different domain-specific instantiations [61]. Specifically within a PMC/PSE setting, the meta-ontology can model the abstract building blocks needed to describe any kind of process flow, while the individual local ontologies codify the different steps making up a domain-specific process. By further instantiating each local ontology with domain-specific data, concrete KGs are realised reflecting actual execution patterns.

By relying on this multi-level framework, we can set up KGs to represent specific PMC/PSE domains. Once a KG has been set up, it must be translated to a TALK graph reflecting the event structure of the industrial

[1]Refer to `https://github.com/predict-idlab/KGE-for-PMC` for the complete definition of the pmc meta-ontology.

Concept	Description
pmc:Step	An abstract building block in a manufacturing process, representing a single action to be taken in the PMC process' execution order.
pmc:Module	This abstracts a logically coherent sequence of pmc:Step nodes. Importantly, a pmc:Module is itself a kind of pmc:Step, one that requires exactly one beginning and one end, both of which are defined by other pmc:Step nodes, which bracket the scope of the module. It is possible to nest modules inside other modules.
pmc:ConditionalStep	A kind of pmc:Step, which additionally inherits from the OWL-S owls:If-Then-Else class. As such, an instantiated pmc:ConditionalStep requires either one then-branch or one else-branch, depending on which direction the process takes.
pmc:Process	This inherits from both the SSN ssn:Procedure class and the OWL-S owls:Process class, thus forming a conceptual bridge between these two ontologies. It represents the procedure or subroutine that is executed by a single step and should not be confused with any given PMC process as a whole.
pmc:ObservationProcess	A kind of pmc:Process which involves specifically making a pmc:Observation.
pmc:ActuationProcess	A kind of pmc:Process which involves specifically making a pmc:Actuation.
pmc:SamplingProcess	A kind of pmc:Process which involves specifically making a pmc:Sampling.
pmc:DirectStep	A kind of pmc:Step which first executes a pmc:Process before being followed by one or more pmc:Step nodes.
pmc:Condition	A condition that needs to be fulfilled in order for the then-branch of a pmc:ConditionalStep to be taken.
pmc:Observation	This inherits from the SSN ssn:Observation class and has the same semantic scope, namely, an act of observing a value of some feature of interest.
pmc:Actuation	This inherits from the SSN ssn:Actuation class and has the same semantic scope, namely, an act of changing the state of the world using an actuator.
pmc:Sampling	This inherits from the SSN ssn:Sampling class and has the same semantic scope, namely, an act of taking or modifying samples.
pmc:ObservationStep	A particular kind of pmc:DirectStep. As such, its shape follows the basic outline of the pmc:DirectStep shape. However, the pmc:ObservationStep always executes a pmc:ObservationProcess instead of a generic pmc:Process.
pmc:ActuationStep	This is very similar to a pmc:ObservationStep, except that actuations are involved instead of observations.
pmc:SamplingStep	This is very similar to a pmc:ObservationStep, except that samplings are involved instead of observations.

Table 6: Main concepts in the manufacturing process ontology, defined in the pmc namespace.

process being modelled. The TALK graph will, in turn, serve as the basis for a pipeline designed for a specific downstream task. By building upon existing explainability techniques, we can provide a rationale for individual predictions in terms of the most salient features involved in those predictions. Because features reflect neighbourhood information extracted from

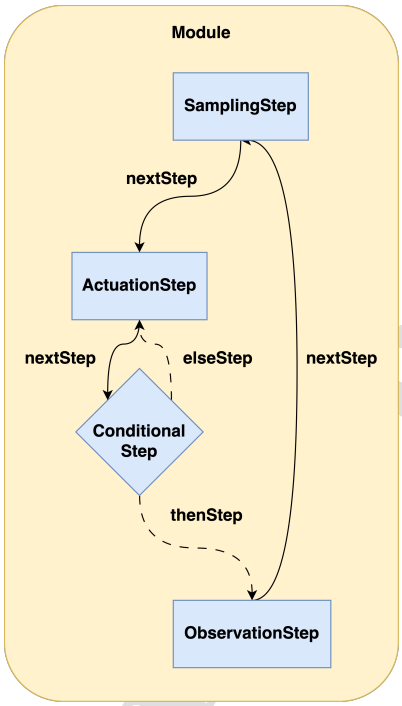


Figure 4: Intuitive representation of concepts in the PMC manufacturing ontology.

the TALK graph, it becomes possible to express decision-level explanations in terms of rooted subgraphs, making them more intelligible. In what follows, we will go over each of the steps involved in the methodology in more detail.

4.1. Embedding Process Knowledge with TALK Graphs and INK Embeddings

To transform process knowledge such that it adheres to the TALK paradigm, the PMC ontology for the domain in question first needs to be integrated with the corresponding historical process data. In the context of what was shown in Figure 5, this means instantiating the bottom arrows between domain-specific ontologies and corresponding databases. Ontology integration means defining a mapping from different data fields to the various steps characterising the process. More specifically, in cases where the order of execution is known for a sequence of steps (which is not always the case), this order must be instantiated as well. Ontology integration might otherwise be seen

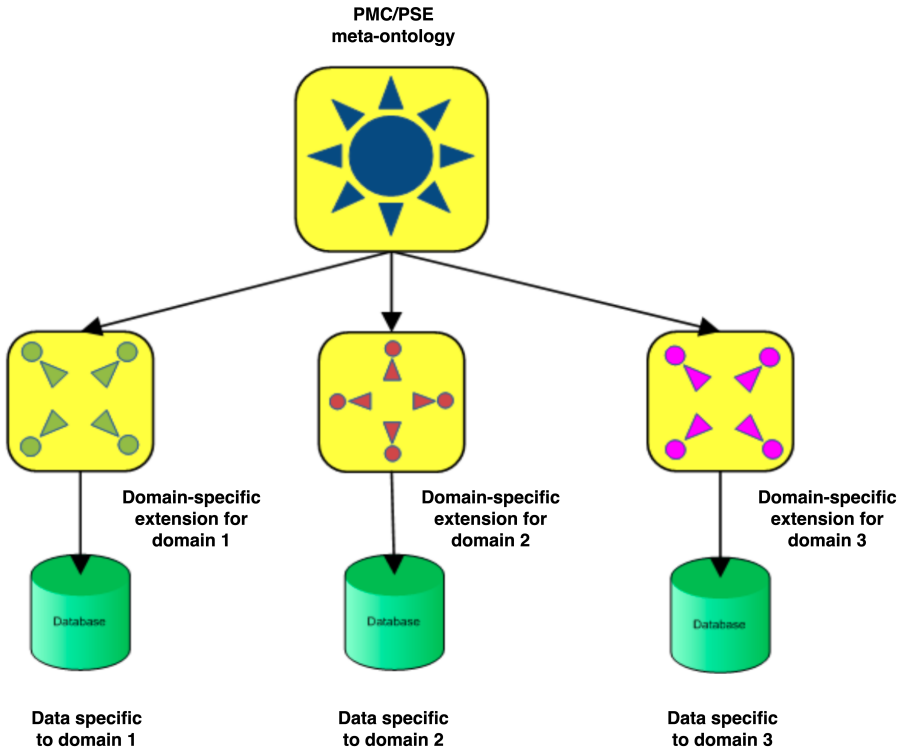


Figure 5: Hierarchical ontology modelling [61]. A single PMC/PSE meta-ontology is used as a skeleton to instantiate different domain-specific KGs.

as *semantic enrichment*, where individual data fields are linked with a whole constellation of concepts, each of which are in turn linked with other data fields [28].

To demonstrate what *sequence instantiation* looks like specifically—as it differs from generic *semantic enrichment* in that it is peculiar to process knowledge—we turn to an example from the first evaluation use case (which will be presented in further detail in Subsection 5.1). Part of the process knowledge involved in this use case pertains to the different reaction phases characterising the chemical state of liquid fabric enhancers. There are five phases in total, three of which are pmc:ObservationSteps and two of which are pmc:ConditionalSteps indicating whether or not a transition is taking

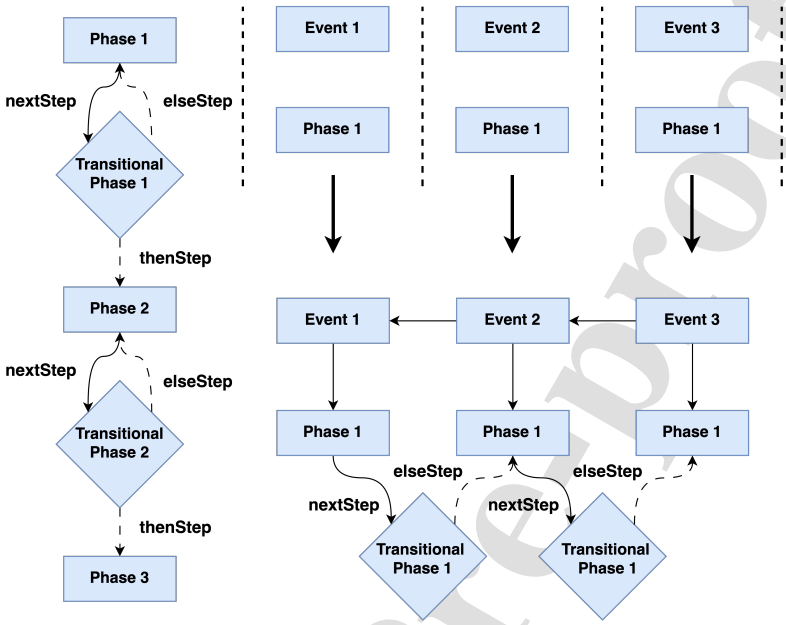


Figure 6: Successive reaction phases as an example of modelling sequential process knowledge.

place between observable chemical states. Figure 6 shows how these reaction phases logically relate to one another. To add phase information to a concrete KG, we need to make connections between actually occurring subsequent phases. For example, say we have an experiment in which the first three events all correspond with the first phase in the reaction. When we model the process knowledge up to Event 3, we loop over pairs of phases in [phase 1 $\rightarrow$ phase 1 $\rightarrow$ phase 1] to make the necessary connections between the phases in each pair. In our example, subsequent phases are identical, such that we get the following list of pairs: [(phase 1, phase 1) $\rightarrow$ (phase 1, phase 1) $\rightarrow$ (phase 1, phase 1)]. Unfortunately, based on what is shown in Figure 6, it is not possible for identical phases to follow one another directly. If phase 1 is repeated twice, this means that for the conditional step representing the transition between phase 1 and phase 2, we need to take the else branch back to phase 1. In the case where no intermediate step is needed, we can simply make the connection to the subsequent phase in the chain. In exceptional cases, where two conditional steps follow one another (i.e., where the transition takes more than one time step), we simply link these steps to-

gether via an else branch. What we have demonstrated in reference to phase transitions in particular, can be extended to step sequences in general, where the predetermined logical ordering of subsequent steps, as well as the different types of these steps (i.e., pmc:DirectSteps or pmc:ConditionalSteps), determine how they should be concretely instantiated.

Once semantic enrichment has been performed, we can make use of the process structure itself to translate the instantiated KG to one that follows the TALK paradigm. Recall that the PMC ontology defines both the pmc:Module and the pmc:Step classes, the prior gathering different steps together in a single logically cohering block, and the latter representing an atomic step (either direct or conditional) inside an industrial process.

As demonstrated in Figure 7, the TALK paradigm requires us to define events, states, and observations. Events can be identified by selecting a specific step in the PMC KG containing all the values on which a prediction by the downstream PMC/PSE task can be based, for example, a specific kind of pmc:ObservationStep. Events can be linked with the timestamps at which they occur, as well as the target values that need to be predicted. For each event, a number of states needs to be defined. These can be extracted from the modules defined in the PMC KG, as modules characterise coherent parts of the overall process representing the states in which that process can find itself. A number of modules may accord with essentially static information (e.g., Static Module 1 in Figure 7) that remains the same throughout the entire process (e.g., relating to process setup), while some other modules may only be accessed intermittently as the process is in a particular state or step (e.g., the phases in Dynamic Module 2 in Figure 7), and again others may be more continuously accessed throughout the process, e.g., continuous observations being made by sensors. Since different events occur at different times and might even be distributed across multiple stages of the industrial process, they will not all be contextualised by the same modules. In general, an event will be characterised by the modules preceding it as well as those that co-occur with it. Modules that remain invariant will correspond with static states, while modules that change over time will be instantiated as distinct states. Finally, observations can be extracted from the steps composing the various modules. These steps encode either semantically enhanced data fields (e.g., ObservationStep 2, SamplingStep 1), or qualitative process knowledge (e.g., ActuationStep 2); and they can exist either in isolation (when the exact step sequence is not known by process managers) or in connection with other

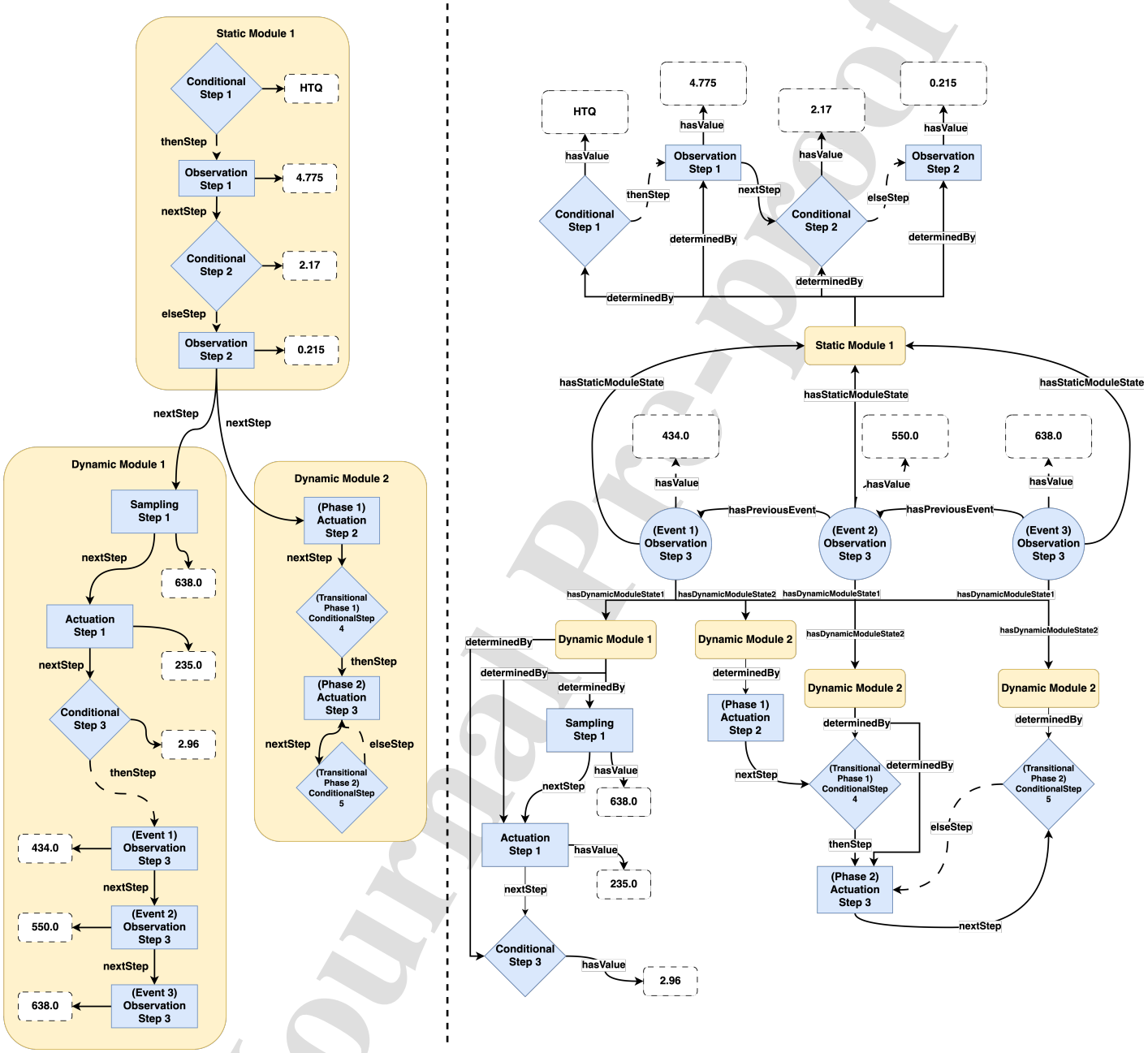


Figure 7: Converting a process flow to a TALK KG.

steps (e.g., reaction phases that occur in a particular sequence depending on the exact evolution of the reaction in a specific experiment). Note that in practice any connections established between steps belonging to the neighbourhoods of different events will have to be inverted, as we want to avoid situations where future knowledge is accessible when it has not yet been disclosed by the process flow.

Once the industrial process has been described in the form of a TALK KG, INK embeddings can be used to create a knowledge-enhanced data-matrix with which a dedicated ML model can be trained to solve a specific downstream PMC/PSE task. As explained in Subsections 3.1 and 3.2, INK embeddings can be used to capture the neighbourhood information of individual TALK events. For a given neighbourhood depth, INK will traverse all paths (following the directions defined by the relationships between separate nodes) starting from a particular event node. Using these paths, it will set up a neighbourhood dictionary with relevant contextual features. Within a TALK graph, paths of depth 3 will include the observations of the event in question. Paths of depth 4 will additionally include the observations of the preceding events, and so on. The extraction depth can be tuned depending on the use case and the process involved in that use case.

In general, the training stage for this methodology supposes that a TALK graph has been set up (in the manner discussed above) for each experiment in the training set. This is the first step in the procedure shown by Figure 8. For each event in each TALK graph, INK is used to extract event-specific neighbourhood information, which is in turn used to create an INK embedding, as discussed previously. The INK embeddings that are created during the training stage will be subject to diverse dimensionalities, with many of the same dimensions across separate embeddings referring to different features. This is a direct consequence of the fact that states and observations tend to vary between subsequent events. In Figure 8, steps 3 and 4 show simplified a version of the extraction step, where INK extractions per event are already aggregated per TALK graph, such that the resulting embedded representations of subsequent events within a single graph already share a common feature set. Note that, at this point, the features sets constructed per graph still differ between different TALK graphs, such that, at the end of the training stage, a common feature set needs to be constructed by taking the union of all features encountered so far. All individual embeddings are then permuted in order to align their separate dimensions with the ordering

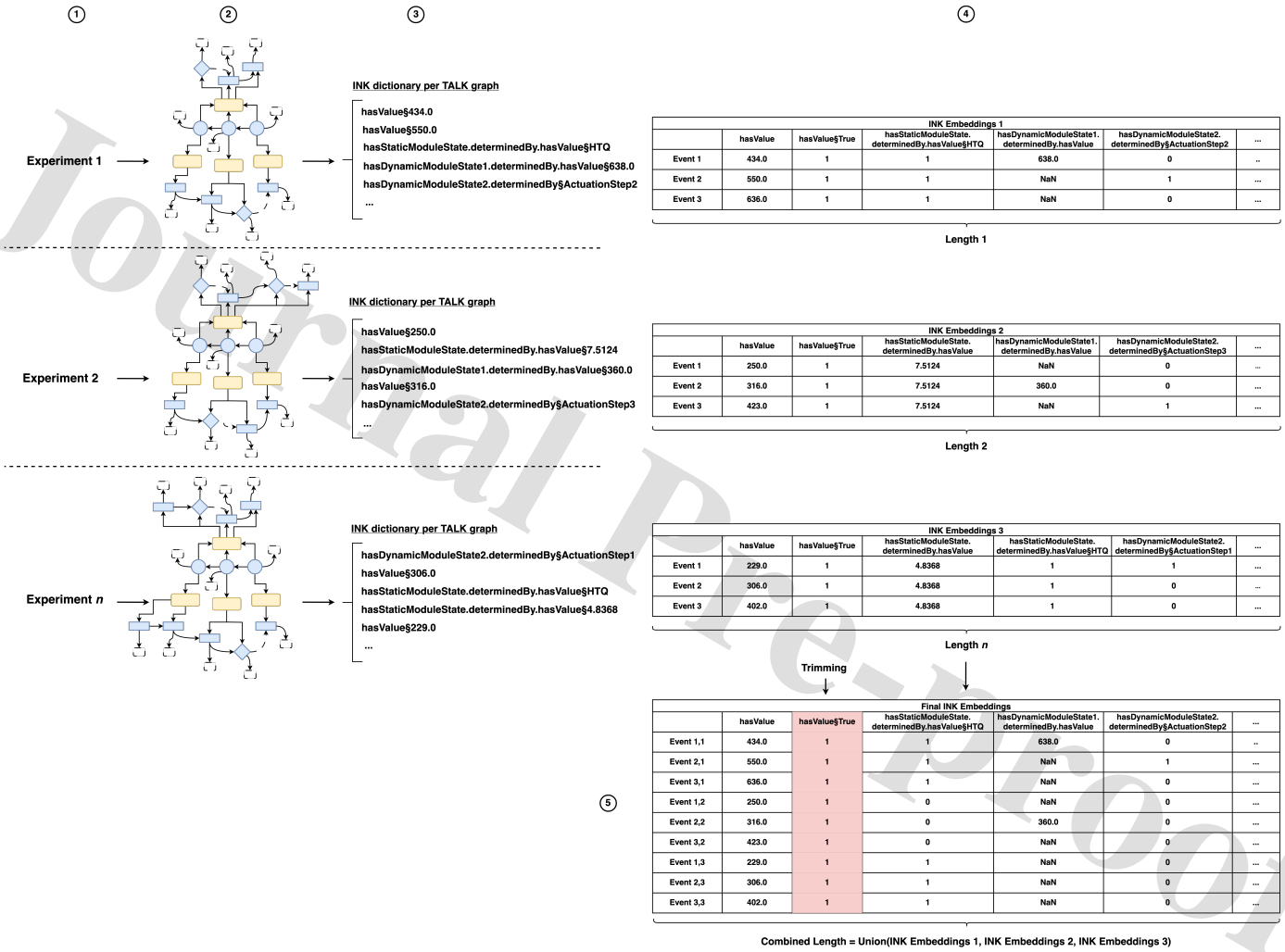


Figure 8: Conversion of TALK graphs for different experiments to create a knowledge enhanced data matrix for a downstream PMC/PSE task.

enforced by the final common feature set. This is shown in step 5 of Figure 8, where all event embeddings, originating from separate graph-specific tables, have now all been given the same ordered feature set. When we encounter new feature dimensions in the validation and test sets, these are discarded. All embeddings extracted from the validation and test experiments are similarly permuted to fit the common feature set constructed during the training stage.

Once all embeddings have been created, we can further optimise them for the downstream task by trimming less informative dimensions. For instance, binary features indicating the presence or absence of a certain neighbourhood path can be safely discarded if accompanied by a real-valued feature indicating the value encountered at the end of the path (e.g., the column for the feature *hasValue§True* in step 5 of Figure 8). Furthermore, binary or categorical features that appear either very seldom or very often are likely to have low informational value and can also be discarded in some cases—although it is difficult to establish a solid criterion with which this can be assessed in a reliable, *a priori* fashion. After these final post-processing steps have been taken, the embeddings can be used to train an ML model or a pipeline of ML models to solve a downstream PMC/PSE task.

4.2. Explaining Model Decisions

As stated in previous sections, one of the main benefits of using INK+TALK is its intrinsic interpretability. With this in mind, Steenwinckel et al. have proposed using INK's task-specific semantic rule mining functionality to learn rule sets with the aim of providing some insight into an underlying dataset and a specific decision process [29]. However, this approach is limited to classification tasks, while PMC/PSE tasks may require specific values to be predicted reflecting a process' operational status. There are a number of methods that are able to provide model-agnostic, decision-level explainability on the condition that input dimensions are themselves explainable. Because INK extracts observable features as opposed to latent ones, such methods can be used in tandem with INK to facilitate knowledge-informed explanations. Additionally, since INK maintains features that can be mapped back onto extracted paths with a formal interpretation, the resulting embedding dimensions will convey relevant information about the contextual process characteristics that inform model decisions. In other words, not only are the dimensions of INK embeddings in themselves transparent, the fact that they reflect process knowledge further adds to their intelligibility.

Some of the most well-known techniques used to enable decision-level explainability include SHapley Additive exPlanations (SHAP), Local Interpretable Model-agnostic Explanations (LIME), Local Agnostic attribute Contribution Explanation (LACE), Local Rule-based Explanation (LoRE), and anchors [62, 30, 63, 64, 65]. Of these, the last three are rule-based and are thus, similar to INK's own rule mining approach, limited to classification tasks in terms of their applicability. Also, in those cases where the downstream task *is* amenable to rule-based explanations, we would prefer INK's innate functionality, given that it is especially geared towards generating KG-based explanations. Besides the rule-based approaches, SHAP and LIME have very similar scopes in terms of their functionality and applicability. Both are model-agnostic, can be applied to all manner of prediction tasks, and offer explanations in terms of feature importances, though LIME is limited to local explanations while SHAP can also provide global explanations. When comparing the two further, LIME is effective when dealing with simpler models, but SHAP has been found to better approximate human explanations and is on the whole a more unified and more robust approach to explainability [30]. Because we want our methodology to apply to a broad range of potentially complex PMC/PSE tasks, SHAP was opted for to explain model decisions.

Within the scope of our hybrid methodology, we can apply SHAP directly by first selecting an appropriate explainer. We then apply the appropriate explainer to specific predictions made by an ML model trained on INK embeddings and use it to estimate the SHAP values for the embedding samples pertaining to the predictions we wish to see explained. We then rank the resulting importance values based on their absolute values and associate them with the corresponding features. As we are providing local (i.e., decision-specific) explanations, we expect to rank those paths highest which have as their root the specific event for which we are making the prediction. However, as embeddings contain features extracted from all events in the training set, this is not guaranteed. Paths which do not show up in the neighbourhood of the event to which the explanation applies, are therefore discarded. Once all features have been cleaned and filtered appropriately, we can provide explanations in the form of a list of paths with normalised importance values. Each path has as its root node the focal event and, by following the relationships involved in the feature names extracted by INK, the intermediate nodes along the path can be extracted as well. (Note that these intermediate nodes were ignored during the original INK extraction to avoid

a dimensionality explosion in the final feature set.) To further increase the intelligibilty of a given decision, this list of paths can be visualised by, for example, highlighting them (using different gradations of a colour scale) in a graphical rendering of the TALK graph representing the process.

5. Evaluation Setup

Having outlined the proposed methodology in detail, we will now go over the evaluation setup. In particular, we will outline the two chemical engineering use cases on which the methodology will be evaluated.

5.1. Use Case 1: Intermittent Viscosity Forecasting

In the first use case, we focus on an industrial process for developing liquid fabric enhancers, i.e., complex mixtures with surfactants, polymers, and perfume. The objective of this use case involves an assessment of the product's stability with respect to its shelf-life, after the product has already been synthesised. As a rule, fabric enhancers are not thermodynamically stable and will ultimately settle [66]. To prevent this during shelf-life time of use, their formulation needs to be optimised by measuring stability and stability processes to optimise design inputs. Monitoring is also performed by putting products in storage at different temperatures to assess their stability for as long as their shelf-life is supposed to hold, under a variety of real-life storage conditions. To assess stability levels, the product's physical and chemical properties are measured at different temporal frequencies (from weekly to monthly) while it remains in storage.

By making use of historical stability data together with prior knowledge about the production process, a hybrid methodology can be used to better predict the long-term stability of specific formulations in advance, based on shorter tests, which will in turn make it feasible to launch new fabric enhancers faster without incurring an increased risk of unsatisfactory results. By also incorporating explainability, it becomes possible to provide insights to engineers with the purpose of further optimising the development of new products.

For the first use case, historical stability data was provided by P&G, consisting of 547 different experiments in total, with each experiment representing a post-production time series measuring viscosity evolution over 8 time steps, with intervals of multiple days separating consecutive time steps.

Specifically, for each experiment we have a data sample for each of the following 8 days: 0, 14, 28, 56, 84, 120, 150, and 180. (Note that the interval between each pair of time steps is identical across different experiments.) The product viscosity at each point in time for a given experiment is measured manually and entered by humans into a database. Various features concerning the product composition, such as, e.g., the amount of silk, perfume, glycerol, structurant, and surfactant, are known in advance for each execution of the production process. Available expert knowledge includes range information which can be associated with subsequent viscosity measurements to indicate whether any given measurement is within the appropriate boundaries. As mentioned briefly in Section 4, for this use case, we also have an indication of the phase of the reaction we are in during post-production. There are a total of five reaction phases characterising the viscosity evolution at different points in time (although the final two phases are fused in practice). These reaction phases are not known in advance, and can only be identified with some degree of certainty in hindsight. As such, definite knowledge pertaining to the reaction is only available for past viscosity measurements, not for ones that still need to be forecast (where it needs to be predicted instead). Reaction phase predictions are performed by a separate bump prediction model, which predicts the highest point of the viscosity curve and the time step at which this high point is going to occur based on the product composition. Using the bump prediction model, it becomes straightforward to identify the transition point in the curve shown in Figure 9, as well as the phases preceding and following this transition point. Note that the final phase (hydrolytic gelling) is difficult to identify based on the transition point alone, but is also less consequential in practice as it occurs only rarely.

When translating the information associated with the first use case to the TALK paradigm, individual events will correspond with discrete points in time at which forecasts should be made. All in all, the forecasting of future viscosity values happens rather straightforwardly: At first, all events in the graph need to be forecast, with only the context of the first event available to the forecasting model. In a second step, the viscosity for the first event will have been sampled and no longer needs to be forecast. Now, all events minus the first need to predicted, using both the original knowledge and the newly measured viscosity of the first event. This continues until the last viscosity measurement has been taken. Following the TALK paradigm, each event is contextualised by a number of states, which include both the information

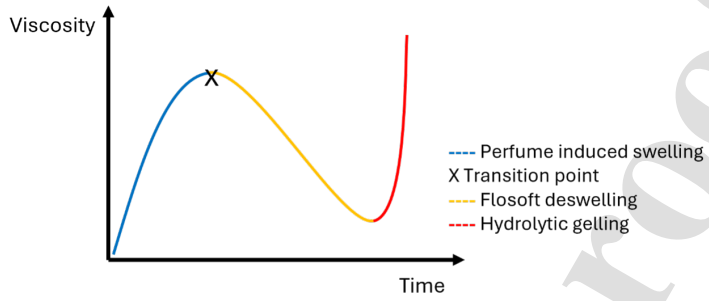


Figure 9: Different reaction phases in terms of viscosity evolution.

about the original product composition and expert knowledge (as mentioned earlier). A fragment of what the TALK graph looks like for the first use case can be found in Figure 10. INK is used to embed the different TALK events inside the graph. INK does not require model training and can be used directly to extract neighbourhood embeddings of a certain depth. Based on prior experimentation and with the aim of being able to capture partial reaction chains, depth 4 was used during the extraction phase for this use case.

5.2. Use Case 2: Continuous Residual Acrylic Acid & Epoxy Forecasting

In the second use case, we consider a resin production process. The goal of this use case is to forecast leftover epoxy and acid values so that operators can determine when to stop feeding acid or whether a correction needs to be made to the process, without having to do the time-consuming work of taking a sample and sending it to a quality control lab where it usually takes a while to perform the analysis. The ultimate goal of this forecasting process is to make sure the production process will result in high-quality resin.

The data for this use case was provided by Allnex and describes experiments in which an epoxy resin is made. These experiments are performed in a small-scale reactor, separate from the larger reactor that would be used during production. The epoxy resin is made by combining acrylic acid and epoxy during a six-to-eight-hour-long exothermic reaction. Currently, the data includes 31 usable experiments, for which the input parameters have been tweaked to mimic the variability or possible malfunctions during a real production run. These experiments may contain as few as 250 consecutive samples or as many as 423, with an average of 353 consecutive data samples

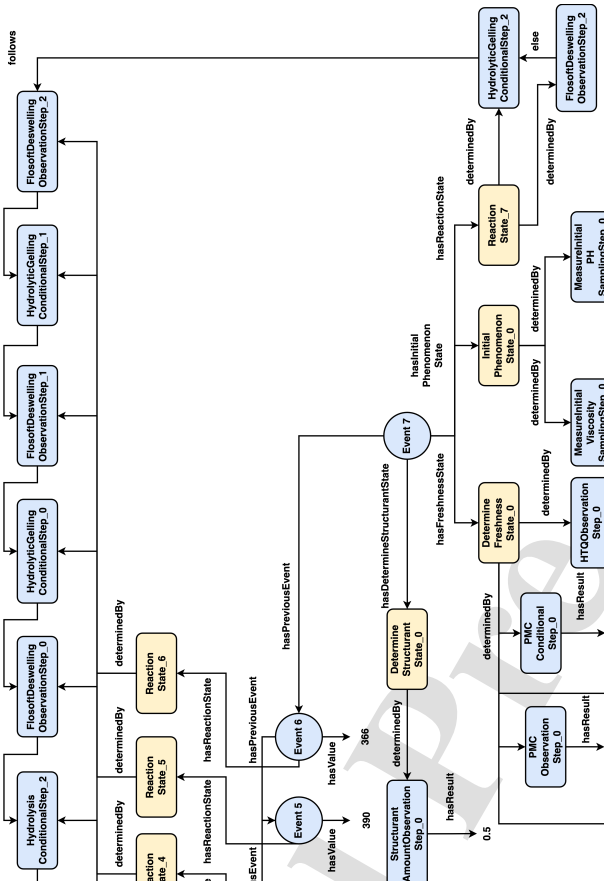


Figure 10: TALK graph fragment for Use Case 1: Intermittent Viscosity Forecasting. All eight successive events are shown together with a selection of different states characterising each event. For the first seven events, only reaction states are shown, with the entire reaction chain also displayed. For the last event, Event 7, a number of static states are visible.

per experiment, where each sample is comprised of 54 different features.

For each experiment, data is collected from three different sources. The first data source produces the process data, which is collected at one sample per minute over the course of the experiment. The most important parameters collected in this data source are:

- Amount of acrylic acid & epoxy loaded into the reactor
- Reactor temperature and setpoint, i.e., the desired or target temperature value
- Cooling bath temperature and setpoint, i.e., the desired or target temperature value

A second data source is based on near-infrared spectroscopy (NIR), where the absorbance in the near-infrared region (833 nm to 2061 nm) of the electromagnetic spectrum is captured every minute. This reflects what is actually happening in the reactor during the process. Some key wavelengths that are monitored include the following:

- 1052 nm: Turbidity
- 1157 nm: Epoxy
- 1210 nm: Ether groups
- 1429 nm: OH groups
- 1575 nm: Acid groups
- 1622 nm: Double bonds
- 1904 nm: Water

The last data source concerns quality control. For each experiment some samples are taken during the process and sent to the quality control lab for inspection of different components. The samples are taken at irregular intervals and are not even taken at the same times for different experiments. The quality control report contains titration information about the leftover acrylic acid and epoxy. These values are reported back to the operators,

which allows them to evaluate the quality of the product and determine possible corrections in the current workflow.

Based on these sources, an integrated dataset was constructed containing both time-variable and static features for each experiment. Besides historical data, available expert knowledge includes range information for various information fields, including acid, epoxy, and viscosity; anomaly detection for the NIR wavelengths, the reactor temperatures, and the feed rates; as well as a mechanistic simulation model to forecast future residual acrylic acid and epoxy values. For the latter, the kinetics and reaction rate equation (albeit simplified) involved in the mechanistic simulation were derived from Merfeld et al. [67].

When translating the information associated with the second use case to the TALK paradigm, individual events will correspond with near-continuous points in time (separated by one-minute intervals) at which forecasts should be made. This is similar to the *modus operandi* of the first use case (albeit with a higher forecasting frequency), in that forecasts will be iteratively improved the closer we get to the event being forecast. All in all, the forecasting of future residual acrylic acid and epoxy values happens rather straightforwardly: At first, instead of predicting the values of all events succeeding the current one (as in the previous use case), we only predict up to one hour in the future, with only the context of the first event available to the forecasting model. In a second step, the residual acrylic acid and epoxy for the first event will have been sampled and no longer needs to be forecast. Now, again, we forecast up to one hour in the future, using both the original knowledge and the newly acquired measurements of the first event. This continues until the last measurements have been taken. Following the TALK paradigm, each event is contextualised by a number of states, which include both the information about the original process specifications, the ongoing feeding process, quality assessment and corrections, as well as additional expert knowledge (as mentioned earlier). A fragment of what the TALK graph looks like for this second use case can be found in Figure 11. INK is used to embed the different TALK events inside the graph. INK does not require model training and can be used directly to extract neighbourhood embeddings of a certain depth. Based on prior experimentation, depth 3 was used during the extraction phase for this use case.

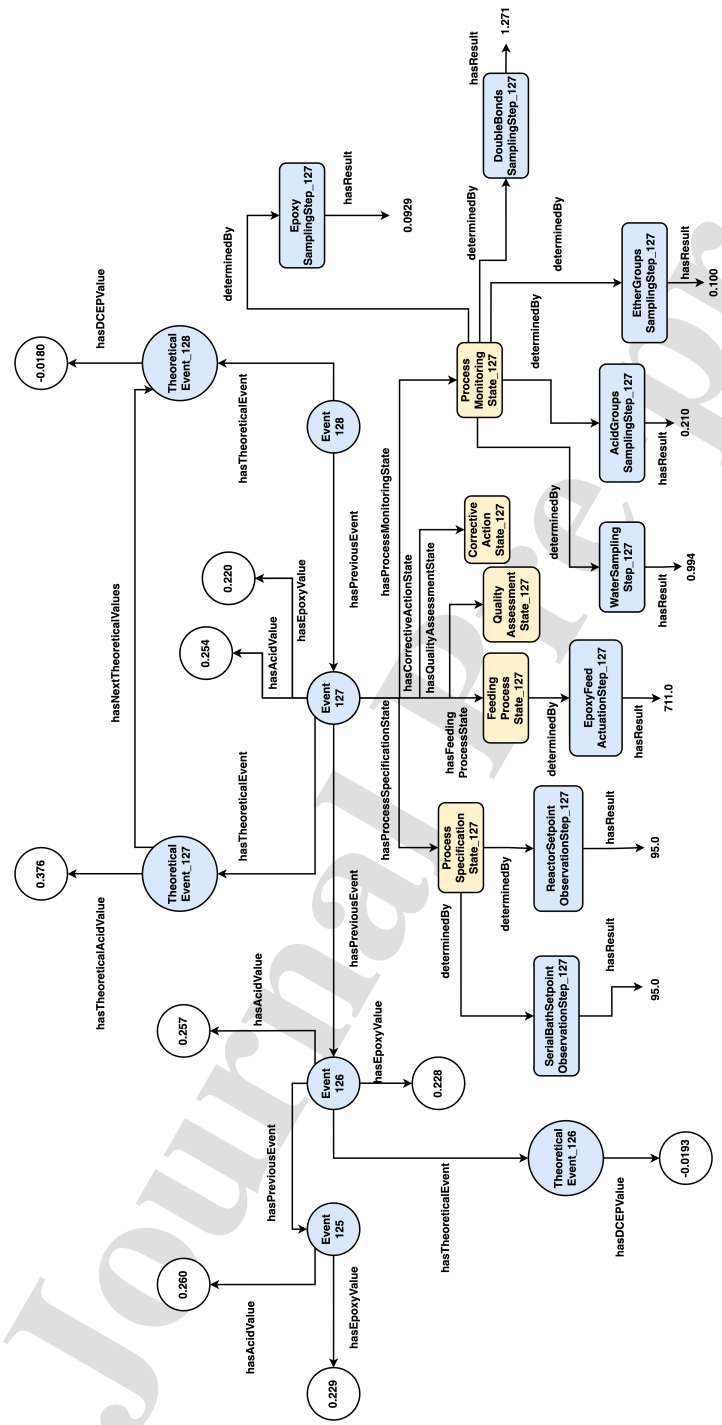


Figure 11: TALK graph fragment for Use Case 2: Continuous Residual Acrylic Acid & Epoxy Forecasting. Three successive events are shown together with a selection of different states characterising the last event. For some of the events, a number of theoretical events are also displayed, which refer to simulated values calculated beforehand by using a mechanistic model. For Event 127, a number of intermittent and dynamic states have been visualised.

5.3. Evaluation Pipelines & Models

Each of the previous two use cases is associated with a separate downstream task. While the proposed hybrid methodology is broadly applicable to different use cases, the individual tasks that need to be performed require specific evaluation pipelines and forecasting models best suited to the respective task domains.

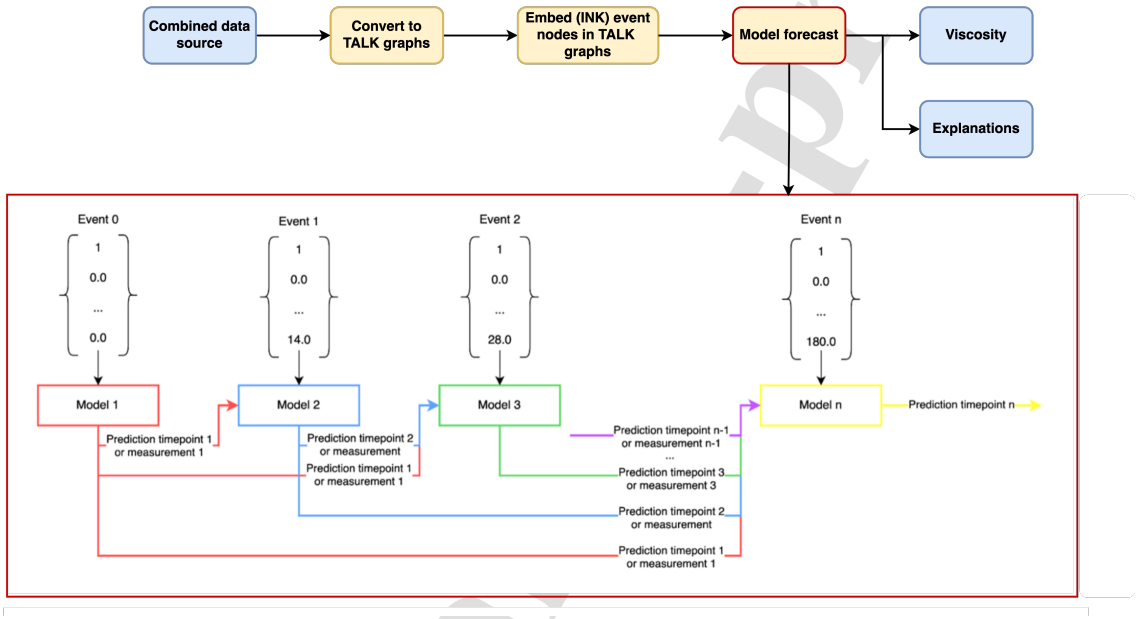


Figure 12: Forecasting pipeline for Use Case 1: Intermittent Viscosity Forecasting.

For both use cases, the downstream PMC/PSE task involves regression. Given that INK embeddings contain a mixture of numerical, binary, and categorical features, we opted for Categorical Boosting (CatBoost) regressors to solve the downstream tasks [68]. Note, however, that the presented methodology is not limited to CatBoost models solely and can be applied to other model types as well.

For Use Case 1: Intermittent Viscosity Forecasting, we set up a forecasting pipeline composed of a sequence of CatBoost models (one for each TALK event), as shown in Figure 12. The easiest way to explain this forecasting pipeline is by considering the dataset as a stream of incoming measurements. When no measurements have been taken yet, we want to make predictions for all time points (from 1 to n). Model 1 is given only the embedding of

Event 0, which does not contain any viscosity measurements. Based on this input, it produces a prediction for time point 1. This prediction gets passed on to all subsequent models (Model 2 to Model n), each of which will need to make a prediction for its respective time point. So next, Model 2 integrates the predictions made by Model 1 into its embedded representation of Event 0. The updated embedding can then be used by Model 2 to generate its own prediction for time point 2, which it can then pass on to all subsequent models (Model 3 to Model n)—and so on. The same procedure holds when we consider the next point in time (at which the first measurement, associated with time point 1, has already been taken), where we start by making a prediction with Model 2 for time point 2 based on the embedded representation of Event 1 (which incorporates the first measurement). This prediction is passed on to all subsequent models, where it is integrated in the respective embeddings of Event 1. This process continues until we have measured the viscosity value associated with time point $n-1$, at which point only Model n still needs to make a forecast (for Event n).

TALK statistics								
Metric	**FFFFFFFF**	**TFFFFFFF**	**TTFFFFFF**	**TTTFFFFF**	**TTTTFFFF**	**TTTTTFFF**	**TTTTTTFF**	**TTTTTTTF**
Avg. Triples	71	95	119	141	161	181	200	219
Avg. Entities	51	57	62	66	71	75	80	85
Avg. Relationships	46	49	50	50	50	52	52	52

Table 7: Overview of TALK statistics for Use Case 1: Intermittent Viscosity Forecasting. Averages per experiment are given for each information level.

Based on prior experimentation, each of the CatBoost models featured in the pipeline of Figure 12 have been trained for 2000 epochs; they have been trained on 349 experiments, validated on 86 experiments, and finally evaluated on 112 experiments. These distributions were obtained by first taking an 80%-20% train-test split and then additionally splitting the resulting 80% training data according to another 80%-20% train-validation split. In the end, this means 64% of the original 547 experiments was used as training data, 16% was used as validation data, and 20% was used as test data. Regarding the TALK graphs from which the embeddings were extracted to set up these datasets, we refer to Table 7 for an overview of the most relevant average statistics. In Table 7, each column represents an information level. Per information level, each experiment corresponds to a single graph. In all, the average number of unique triples, entities, and relationships across all 547 experiments in the datasets has been shown for each information level. Note

that the average number of relationships does not increase at the same steady rate as the average number of triples and entities because reuse is much more frequent for relationships than it is for the other graph elements. Usually, as the process evolves, the primary form of novelty in terms of relationships comes from modelling reaction phases.

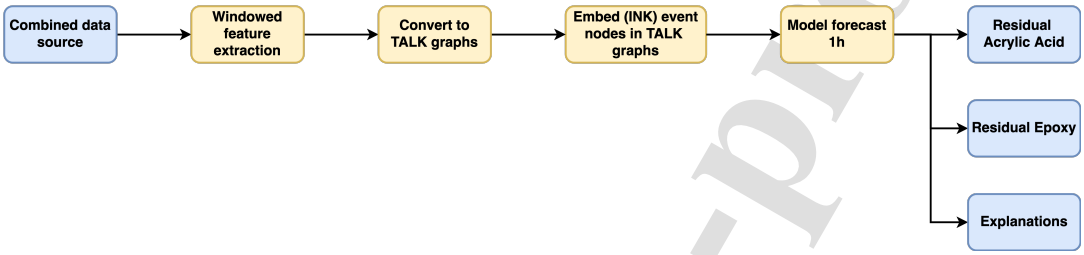


Figure 13: Forecasting pipeline for Use Case 2: Continuous Residual Acrylic Acid & Epoxy Forecasting.

For Use Case 2: Continuous Residual Acrylic Acid & Epoxy Forecasting, the pipeline, as shown in Figure 13, is a bit more complex. Apart from the steps in which the TALK graphs are set up and the INK embeddings are extracted, both of which are integral parts of the proposed methodology, the pipeline for this second use case also includes a step in which *windowed feature extraction* is performed, which itself involves two substeps. First, the measured features are aggregated to reflect windowed information. For these features, we calculate the mean, mean absolute, and standard deviation of the slope of the window, making sure to skip any NaN values encountered during the operation. We also retain the final value of each window. In sum, four new features are created per original feature: *mean_diff_feat*, *mean_abs_diff_feat*, *std_diff_feat*, and *last_feat* with feat the original feature.

Second, after extractions have been made, we also want to make sure that each aggregated sample records some information of the samples preceding it. While the embeddings themselves will contain features representing links to previous events, we can further enhance this trait by adding shifted features to the original samples. For example, if we use $LAGS = [1, 2, 3]$, we will create three new features for each original feature, plus the original feature itself: *feat_SHIFT=0*, *feat_SHIFT=1*, *feat_SHIFT=2*, *feat_SHIFT=3*, with *feat_SHIFT=0* the original feature and *feat_SHIFT=*i the corresponding feature from i time steps back. After applying both of these operations, we have (in our example, where $LAGS = [1, 2, 3]$) 16 features for every original

feature. Note that for our evaluations, we made use of $LAGS = [1, 2, 3, 4, 5]$, which, based on prior experimentation, seemed to offer a good balance between computational complexity (i.e., in terms of feature dimensionality) and information value.

Using this final data setup, we perform forecasts one hour into the future, with separate forecasts for each minute. As such, we use two separate CatBoost regressor models for every minute we want to forecast (so, 120 in total, with each pair of models used to forecast acrylic acid and epoxy separately). Based on prior experimentation, each of these CatBoost models was trained for 2000 epochs; overall, the models are trained on 22 experiments, validated on 3 experiments, and finally evaluated on 6 experiments. (Note that the lower number of experiments in this use case *vis-à-vis* the previous one is offset by the greater number of samples in each experiment. This is also reflected in the TALK graph statistics, which are as follows: 41244 average triples, 21883 average entities, and 57 average relationships per experiment (averaged over all 31 experiments). For the second use case, only a single average is provided for each type of graph element, given that the experiment is not characterised by different information levels.) Finally, because of the large number of extracted features elicited by the INK algorithm, PowerShap is used to perform feature selection on numerical and categorical features jointly [69]. PowerShap was opted for because it has a number of advantages over traditional filter- and wrapper-based feature selection methods. Specifically, PowerShap is able to maintain a relatively low computational complexity, while also being able to take into account complex interactions between features. Different downstream tasks may give preference to other feature selection approaches, but, due to the large number of forecasting models involved in the current use case, managing the computational load was given priority.

5.4. Evaluation Criteria

To evaluate the pipelines and models expounded on above, data-driven baselines were set up for each use case as a point of comparison. In each case, the objective is to measure whether the hybrid approach incorporating process knowledge via KG embeddings can outperform a corresponding data-driven approach that is architecturally similar but does not leverage process knowledge. Additionally, we need to demonstrate how KG-based explanations can be generated and visualised for specific forecasts. These explanations are evaluated on a qualitative basis and the evaluations themselves may be

treated as a proof of concept demonstrating the principle behind the methodology.

For both use cases, the data-driven baseline adheres to the same downstream architecture as the hybrid model, except that manual feature engineering was used instead of an INK+TALK-based approach. The data-driven baseline thus combines the most salient data features identified by engineers with a forecasting pipeline of CatBoost regressor models, as also shown in Figures 12 and 13, where only the steps pertaining to the construction of TALK graphs and INK embedding extraction are omitted. Additionally, in terms of knowledge integration, for Use Case 1: Intermittent Viscosity Forecasting, the data-driven baseline does not incorporate knowledge pertaining to reaction phases or range information indicating whether successive viscosity samplings are within appropriate boundaries. For Use Case 2: Continuous Residual Acrylic Acid & Epoxy Forecasting, the data-driven baseline does not make use of theoretical simulations, range information, or anomaly detection. For a more in-depth and exhaustive description of the data-driven baselines used during evaluation, please refer to Blyau et al. [70].

6. Results & Discussion

Given the evaluation setup detailed in the previous section, a different set of results was obtained for each use case, reflecting various model settings. In what follows, these results are shown and discussed for each use case separately.

6.1. Use Case 1: Intermittent Viscosity Forecasting

For Use Case 1: Intermittent Viscosity Forecasting, we first show results for the hybrid model without using phase predictions; in other words, for these results, the ground truth was used to reflect the reaction phases even though these are not all ready to hand when the reaction is still ongoing. We also show the results for the hybrid model while using a prediction model to predict the most recent reaction phase. Tables 8, 9, and 11 show the raw Root Mean Squared Error (RMSE) results for the data-driven baseline and the hybrid model (in two different settings) as applied to the first use case.

Tables 8, 9, and 11 can be interpreted along the lines of how the forecasting process was explained in Subsection 5.3. The columns express how many measurements have been taken at a given point (i.e., the information level, where FFFFFFFF means that no measurements have yet been taken,

TFFFFFFF means that only one measurement has been taken, and so on), while the rows express the successive future timepoints at which forecasts need to be made for a given information level. In the case of the first column, all the average RMSE values for the forecasts of the eight total time points across the 112 evaluation experiments are reported. In the second column, the same is done for a situation where the first measurement has already been taken, so that the first time point (i.e., the first row) no longer needs to be forecast. In the final column, all measurements have been taken except for the one corresponding to the final time point, such that only one value is reported.

Table 10 shows the differences between the data-driven baseline and the hybrid model, using the ground truth to model reaction phases. These differences were calculated with reference to the hybrid model itself, meaning that every score is the difference in RMSE values between the baseline and the hybrid, divided again by the baseline RMSE, such that we get a percentage improvement of the hybrid *vis-à-vis* the baseline for every time point (row) and information level (column). The average improvement across all rows and columns is also reported. Tables 12, 13, and 14 do the same with predicted reaction phases instead of phases reflecting the ground truth (thus more accurately reflecting a real-world situation) for three different settings of the hybrid model. Here, Table 13 omits what was previously designated as expert knowledge, i.e., knowledge concerning the reaction phase active at any given time step, and the range information indicating whether successive viscosity samplings are within appropriate boundaries, in order to gauge the effect of this expert knowledge on hybrid performance. For Table 14, we omitted information about which types of perfumes were used in the experiments, given that this information is not always ready to hand, e.g., when monitoring new products for which the perfume type has not been encountered before.

It is immediately clear that the model setup using the ground truth (see Table 9) for the reaction phases outperforms the one with predicted phases (see Table 11), which nicely validates the impact of having reaction information with greater fidelity. Note, however, that, barring any expert input, Table 11 more accurately reflects a real-world scenario than Table 9.

In all cases, the data-driven alternative is outperformed on average by its hybrid counterparts. Especially in the early stages of the forecasting process—when we would indeed expect process knowledge to contribute more

RMSE for the Data-Driven Model								
Time Point	**FFFFFFFF**	**TFFFFFFF**	**TTFFFFFF**	**TTTFFFFF**	**TTTTFFFF**	**TTTTTFFF**	**TTTTTTFF**	**TTTTTTTF**
1	62.99727288							
2	83.92374671	62.81682893						
3	93.96282965	82.29720977	38.60885477					
4	116.6363935	107.3802886	71.66660945	48.76527736				
5	109.9400967	104.7742847	80.25303551	63.18467589	38.12533739			
6	94.17395687	92.38185398	74.16828758	63.91613437	48.52682644	40.07486071		
7	97.95281025	99.31722097	86.76151053	73.59619758	58.96693358	54.07689635	41.44427215	
8	90.20167855	96.06922412	82.47164967	72.30452077	60.6111443	56.45706205	48.8845562	30.75913017

Table 8: RMSE results for the Data-Driven Baseline.

RMSE for the Hybrid Model								
Time Point	**FFFFFFFF**	**TFFFFFFF**	**TTFFFFFF**	**TTTFFFFF**	**TTTTFFFF**	**TTTTTFFF**	**TTTTTTFF**	**TTTTTTTF**
1	62.33363896							
2	72.12591784	54.2970769						
3	79.25324496	67.04440373	38.64004932					
4	97.86911093	89.69901073	66.30912956	47.69714335				
5	100.9000889	89.84934778	76.59176102	64.42006626	43.90845849			
6	81.11635821	72.91844729	63.33927351	52.15414785	43.02184621	38.87023059		
7	85.80357477	74.3106411	74.73518457	64.85762456	56.60056507	54,12613485	42.37193769	
8	73.04566436	69.89247799	67.92594882	59.51703431	53.17362955	49.29780893	41.81103191	30.27720324

Table 9: RMSE results for the Hybrid Model (using the ground truth for reaction stages).

Difference in RMSE between the Data-Driven Baseline and the Hybrid Model (+10.69% AVG improvement)								
Time Point	**FFFFFFFF**	**TFFFFFFF**	**TTFFFFFF**	**TTTFFFFF**	**TTTTFFFF**	**TTTTTFFF**	**TTTTTTFF**	**TTTTTTTF**
0	**+1.05%**							
14	**+14.06%**	**+13.56%**						
28	**+15.65%**	**+18.53%**	-0.08%					
56	**+16.09%**	**+16.47%**	**+7,48%**	**+2.19%**				
84	**+8.22%**	**+14.24%**	**+4.56%**	-1.96%	-15.17%			
120	**+13.87%**	**+21.07%**	**+14.60%**	**+18.40%**	**+11.34%**	**+3.01%**		
150	**+12.40%**	**+25.18%**	**+13.86%**	**+11.87%**	**+4.01%**	-0.09%	-2.24%	
180	**+19.02%**	**+27.25%**	**+17.64%**	**+17.69%**	**+12.27%**	**+12.68%**	**+14.47%**	**+1.57%**

Table 10: Difference in RMSE results between the Data-Driven Baseline and the Hybrid Model (using the ground truth for reaction stages). Negative differences (minus sign) indicate that the Hybrid Model is under-performing *vis-à-vis* the Data-Driven Baseline. Positive differences (plus sign, in bold) indicate the opposite.

strongly to the model's ability to make accurate forecasts—the differences are significant. By contrast, in some of the later time steps, we see that the data-driven alternative sometimes outperforms the hybrid, suggesting that actual measurement info may outweigh expert knowledge. This finding indicates perhaps that different models, i.e., purely data-driven or hybrid, should be

RMSE for the Hybrid Model								
Time Point	**FFFFFFFF**	**TFFFFFFF**	**TTFFFFFF**	**TTTFFFFF**	**TTTTFFFF**	**TTTTTFFF**	**TTTTTTFF**	**TTTTTTTF**
1	62.37579518							
2	71.36649043	52.40583127						
3	75.70915414	64.95946848	39.32953002					
4	94.28357317	87.67074397	68.05354571	49.056567413				
5	96.10433983	90.72942069	79.0925433	63.561087305	43,31311463			
6	78.82588751	78.46155193	66.71639638	57.571268203	47.97742255	42.97009679		
7	82.50507092	75.74435232	73.73902822	69.7224074	58.1773889	53.179141001	41.13074968	
8	74.27537419	71.09881365	69.78261438	64.89763452	58.525878006	53.94605421	44.93750144	34.355606488

Table 11: RMSE results for the Hybrid Model (using predicted reaction stages).

Difference in RMSE between the Data-Driven Baseline and the Hybrid Model (+8.57% AVG improvement)								
Time Point	**FFFFFFFF**	**TFFFFFFF**	**TTFFFFFF**	**TTTFFFFF**	**TTTTFFFF**	**TTTTTFFF**	**TTTTTTFF**	**TTTTTTTF**
1	**+0.99%**							
2	**+14.96%**	**+16.57%**						
3	**+19.43%**	**+21.07%**	-1.87%					
4	**+19.16%**	**+18.35%**	**+5.04%**	-0.60%				
5	**+12.58%**	**+13.40%**	**+1.45%**	-0.60%	-13.61%			
6	**+16.30%**	**+15.07%**	**+10.05%**	**+9.93%**	**+1.13%**	-7.22%		
7	**+15.77%**	**+23.73%**	**+15.01%**	**+5.26%**	**+1.34%**	**+1.66%**	**+0.76%**	
8	**+17.66%**	**+25.99%**	**+15.39%**	**+10.24%**	**+3.44%**	**+4.45%**	**+8.07%**	-11.69%

Table 12: Difference in RMSE results between the Data-Driven Baseline and the Hybrid Model (using predicted reaction stages). Negative differences (minus sign) indicate that the Hybrid Model is under-performing *vis-à-vis* the Data-Driven Baseline. Positive differences (plus sign, in bold) indicate the opposite.

Difference in RMSE between the Data-Driven Baseline and the Hybrid Model, without expert knowledge (+7,11% AVG improvement)								
Time Point	**FFFFFFFF**	**TFFFFFFF**	**TTFFFFFF**	**TTTFFFFF**	**TTTTFFFF**	**TTTTTFFF**	**TTTTTTFF**	**TTTTTTTF**
1	-1.11%							
2	**+14.69%**	**+17.16%**						
3	**+16.00%**	**+22.65%**	**+3.88%**					
4	**+17.05%**	**+21.74%**	**+7.29%**	**+4.23%**				
5	**+12.10%**	**+17.18%**	**+4.95%**	**+1.37%**	-7,52%			
6	**+12.62%**	**+17.92%**	**+7.72%**	**+5.01%**	-3.14%	-14.54%		
7	**+12.97%**	**+22.33%**	**+13.18%**	**+2.16%**	-8.57%	-6.85%	-19.68%	
8	**+18.97%**	**+30.18%**	**+16.86%**	**+10.03%**	**+3.33%**	**+4.93%**	**+8.49%**	-29.45%

Table 13: Difference in RMSE results between the Data-Driven Baseline and the Hybrid Model, without expert knowledge (using predicted reaction stages). Negative differences (minus sign) indicate that the Hybrid Model is under-performing *vis-à-vis* the Data-Driven Baseline. Positive differences (plus sign, in bold) indicate the opposite.

applied at different points in time for optimal performance. Overall, the results clearly show that the hybrid approach is able to deliver results that are either on par with or superior to those of the baseline data-driven approach.

Difference in RMSE between the Data-Driven Baseline and the Hybrid Model, without perfume types (+4,91% AVG improvement)								
Time Point	**FFFFFFFF**	**TFFFFFFF**	**TTFFFFFF**	**TTTFFFFF**	**TTTTFFFF**	**TTTTTFFF**	**TTTTTTFF**	**TTTTTTTF**
1	-1.65%							
2	**+9.61%**	**+11.20%**						
3	**+10.70%**	**+17.19%**	-1,17%					
4	**+10.52%**	**+13.15%**	-0,67%	-4.17%				
5	**+9.64%**	**+14.14%**	**+4.12%**	**+5.11%**	-0.80%			
6	**+6.42%**	**+14.42%**	**+6.75%**	**+2.16%**	-5.82%	-8.02%		
7	**+6.84%**	**+22.02%**	**+11.21%**	**+4.90%**	-0,74%	-0.30%	-5.29%	
8	**+10.88%**	**+22.31%**	**+6.89%**	**+1.28%**	-5.78%	**+1.10%**	**+1.51%**	-12.99%

Table 14: Difference in RMSE results between the Data-Driven Baseline and the Hybrid Model, without perfume types (using predicted reaction stages). Negative differences (minus sign) indicate that the Hybrid Model is under-performing *vis-à-vis* the Data-Driven Baseline. Positive differences (plus sign, in bold) indicate the opposite.

When comparing Tables 12, 13, and 14, we can see that the average improvements of the hybrid model with respect to the data-driven alternative degrade as we remove important pieces of knowledge. Removing knowledge about the predicted reaction phases and appropriate viscosity ranges leads to a decrease of 1.46% in performance improvement, while removing knowledge of perfume types leads to a decrease of 3.66%. However, even these more constrained hybrid models still outperform the data-driven baseline, possibly owing to the extensive product composition information subsumed under various static states (as illustrated previously in Figure 10).

6.2. Use Case 2: Continuous Residual Acrylic Acid & Epoxy Forecasting

The evaluation results for Use Case 2: Continuous Residual Acrylic Acid & Epoxy Forecasting are given by Figures 14, 15, 16, 17, 18, and 19, each belonging to a specific evaluation experiment. Each of these figures shows two separate plots, one on the left for the data-driven baseline and one on the right for the hybrid model, for a specific experiment with its own unique identifier (i.e., EXP011, EXP020, EXP024, EXP069, EXP070, EXP904). Both plots show the ground truths for residual acrylic acid and epoxy, which are shown as black and blue bold curves, while the shaded areas reflect the averages of the forecasts +/- the standard deviations. The logic behind these plots relies on the fact that each time point is forecast multiple times successively, such that it thus becomes possible to calculate averages and standard deviations.

Based on the plots visualised in Figures 14, 15, 16, 17, 18, and 19, it is not immediately clear which model outperforms the other. We can, however, say with relative certainty that the hybrid model outperforms the data-driven

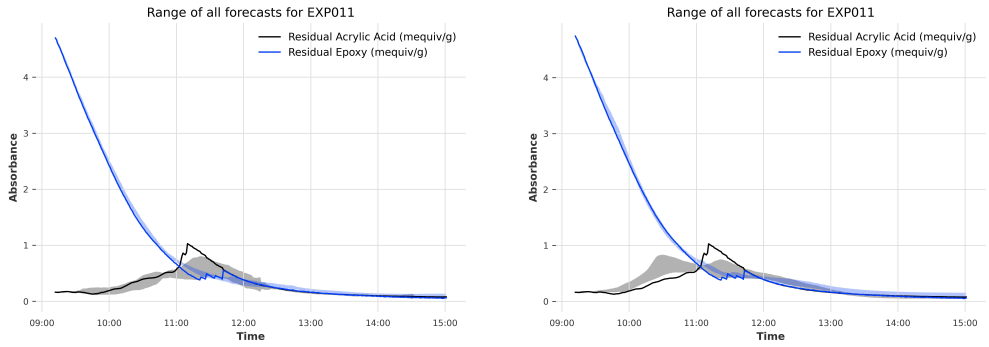


Figure 14: Forecast average plots for the Data-Driven Baseline (left) and the Hybrid Model (right) on EXP011.

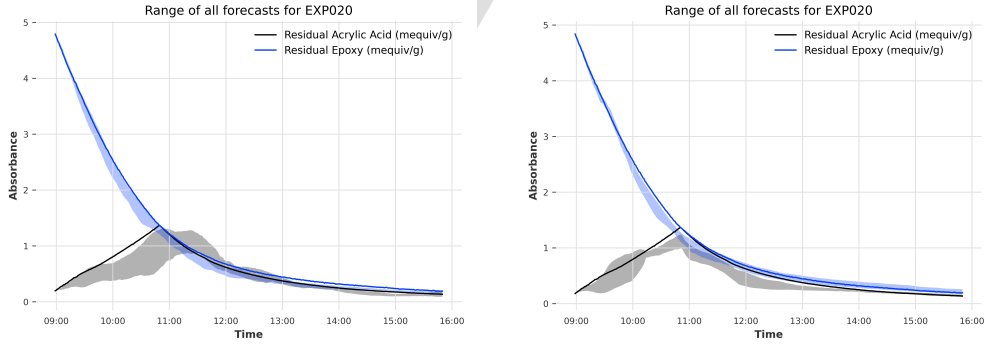


Figure 15: Forecast average plots for the Data-Driven Baseline (left) and the Hybrid Model (right) on EXP020.

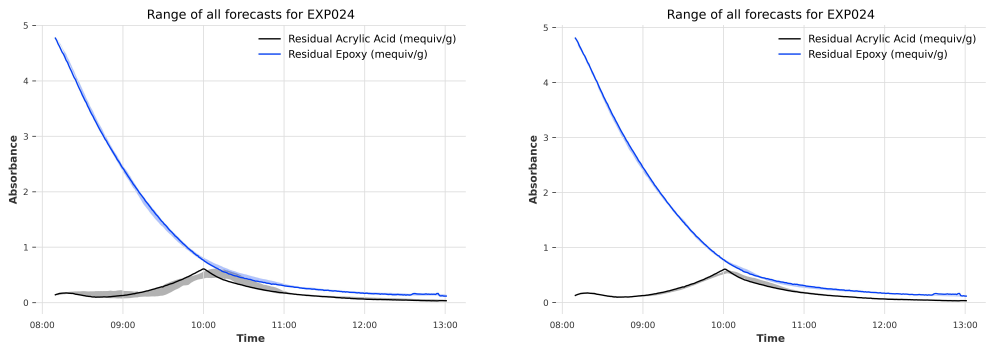


Figure 16: Forecast average plots for the Data-Driven Baseline (left) and the Hybrid Model (right) on EXP024.

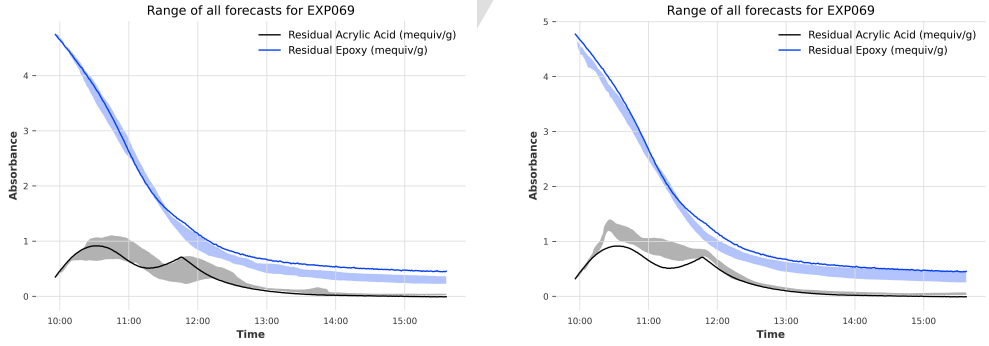


Figure 17: Forecast average plots for the Data-Driven Baseline (left) and the Hybrid Model (right) on EXP069.

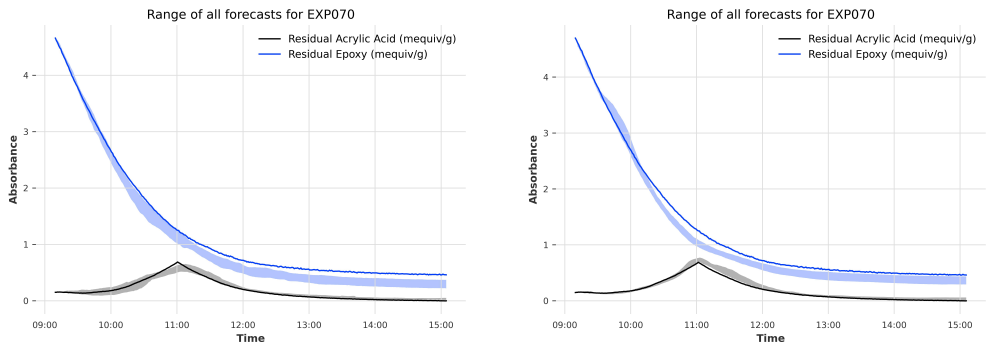


Figure 18: Forecast average plots for the Data-Driven Baseline (left) and the Hybrid Model (right) on EXP070.

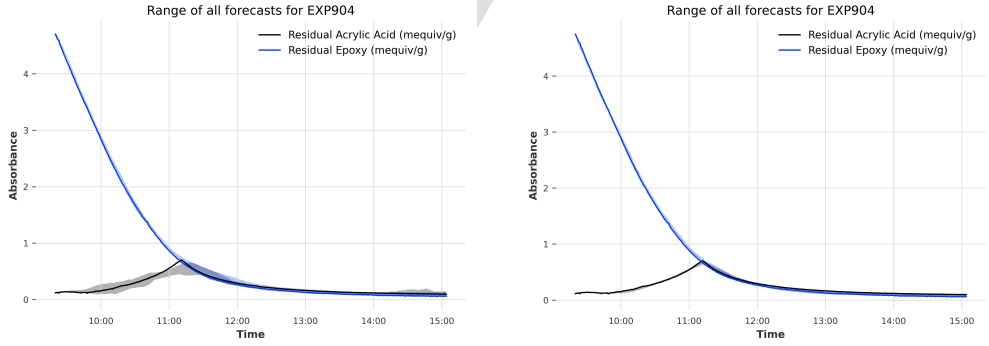


Figure 19: Forecast average plots for the Data-Driven Baseline (left) and the Hybrid Model (right) on EXP904.

baseline on at least EXP024 and EXP904, as it is able to closely track the ground truth for these experiments.

As for the other experiments, Table 15 offers a more in-depth look at the differences in terms of respective performances. The metrics that are being compared in Table 15 are the Mean Absolute Error (MAE) across all forecasts (for a given experiment) and the associated standard deviation (Std. Dev.) across all forecasts (for a given experiment). In the columns belonging to the hybrid results, we have also reported the percentage improvement with respect to the same metric for the baseline. For EXP904, the hybrid model clearly outperforms the baseline on all metrics and for both targets. The same goes for EXP024, thus confirming our earlier observations. Based on the same tabular results, we can also confirm that the hybrid outperforms the baseline on EXP020, though the discrepancies are less pronounced here. For EXP069 and EXP070, the hybrid shows only marginal improvements, while for EXP011 it falters on most metrics. On average, the hybrid outperforms the baseline on all metrics.

Comparison between Data-Driven Baseline and Hybrid Model								
Experiment	**Baseline MAE (Acrylic Acid)**	**Hybrid MAE (Acrylic Acid)**	**Baseline Std. Dev. (Acrylic Acid)**	**Hybrid Std. Dev. (Acrylic Acid)**	**Baseline MAE (Epoxy)**	**Hybrid MAE (Epoxy)**	**Baseline Std. Dev. (Epoxy)**	**Hybrid Std. Dev. (Epoxy)**
EXP011	0.0561	0.0869 (-54.767%)	0.0750	0.0786 (-4.751%)	0.0499	0.0579 (-15.910%)	0.0245	0.0199 (**+18.701%**)
EXP020	0.118	0.100 (**+15.220%**)	0.126	0.0911 (**+28.054%**)	0.0905	0.0801 (**+11.517%**)	0.0558	0.0498 (**+10.820%**)
EXP024	0.0380	0.0165 (**+56.620%**)	0.0381	0.0149 (**+60.905%**)	0.0283	0.0211 (**+25.371%**)	0.0201	0.00899 (**+55.374%**)
EXP069	0.0884	0.0931 (-5.374%)	0.0829	0.0983 (-18.597%)	0.136	0.129 (**+4.998%**)	0.0444	0.0527 (-18.701%)
EXP070	0.0358	0.0336 (**+6.307%**)	0.0250	0.0263 (-4.993%)	0.144	0.126 (**+12.469%**)	0.0395	0.0439 (-11.063%)
EXP904	0.0399	0.0241 (**+39.604%**)	0.0328	0.00964 (**+70.681%**)	0.0452	0.0332 (**+26.447%**)	0.0233	0.0116 (**+50.219%**)
AVG improvement		**+9.601%**		**+21.883%**		**+10.815%**		**+17.558%**

Table 15: Comparison between hybrid approach and baseline approach on a number of metrics.

6.3. KG-Driven Explanations

To demonstrate the principle of generating decision-level explanations by combining INK and SHAP, we will focus on specific forecasts for both use cases. To begin with, for Use Case 1: Intermittent Viscosity Forecasting, Figure 20 shows the various forecasts made by the hybrid model for one of the evaluation experiments. Here, the red dot encircles the forecast we want

to explain. In other words, we are interested in explaining the very last forecast of the experiment.

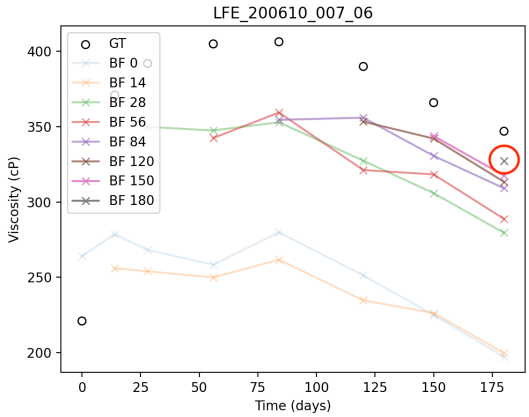


Figure 20: Hybrid forecasts for a selected experiment from Use Case 1: Intermittent Viscosity Forecasting. GT corresponds with the ground truth, while BF X corresponds with the successive forecasts made for a given information level X. Explanations have been generated for the forecast corresponding to the red circle.

In Figure 21, the explanations generated according to the procedure laid out in Subsection 4.2 have been mapped onto the TALK graph representing the experiment. The top salient features are visualised on a gradated colour scale, with light green indicating the least salient features and dark green indicating the most salient ones. The most important paths for the forecast in question can be rendered textually as follows, where each path has been unfurled by interposing connecting nodes wherever they are encountered:

1. Event 7 $\overset{hasPreviousEvent}{\longrightarrow}$ Event 6 $\overset{hasValue}{\longrightarrow}$ 366
2. Event 7 $\overset{hasPreviousEvent}{\longrightarrow}$ Event 6 $\overset{hasPreviousEvent}{\longrightarrow}$ Event 5 $\overset{hasValue}{\longrightarrow}$ 390
3. Event 7 $\overset{hasPreviousEvent}{\longrightarrow}$ Event 6 $\overset{hasPreviousEvent}{\longrightarrow}$ Event 5 $\overset{hasPreviousEvent}{\longrightarrow}$ Event 4 $\overset{hasValue}{\longrightarrow}$ 406
4. Event 7 $\overset{hasDetermineStructurantState}{\longrightarrow}$ DetermineStructurantState_0 $\overset{determinedBy}{\longrightarrow}$ StructurantAmountObservationStep_0 $\overset{hasResult}{\longrightarrow}$ 0.5
5. Event 7 $\overset{hasFreshnessState}{\longrightarrow}$ DetermineFreshnessState_0 $\overset{determinedBy}{\longrightarrow}$ NeatOilObservationStep_0 $\overset{hasResult}{\longrightarrow}$ 1.701

The explanation shown above accords with our intuition. When forecasting the final viscosity value, the previous 3 viscosity values play an outsize

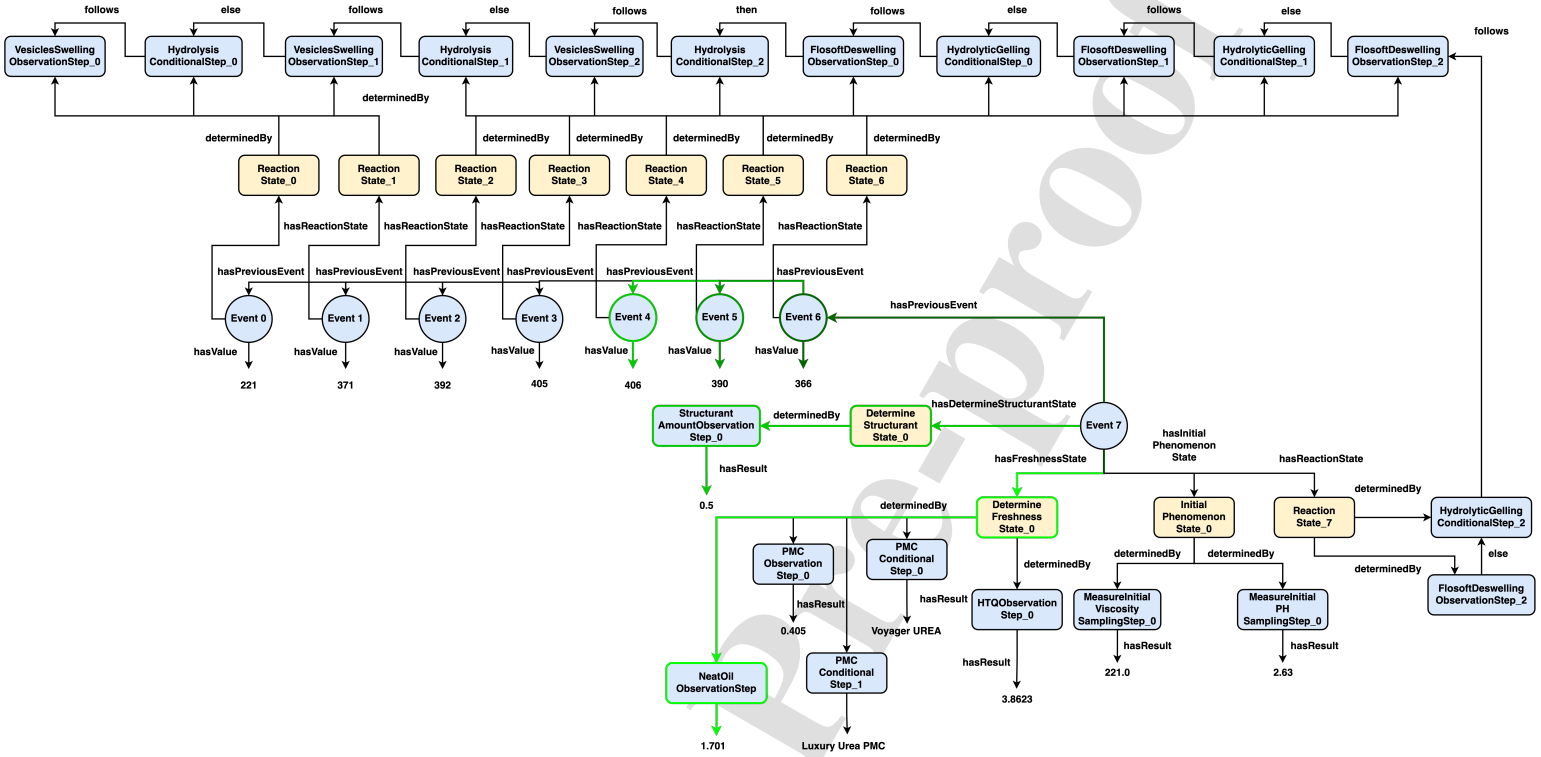


Figure 21: Explanation for hybrid forecast in Use Case 1: Intermittent Viscosity Forecasting. The colour gradation indicates the degree of importance of each path on a normalised scale, with dark green reflecting the highest importance and light green the lowest.

role in the prediction. However, even here at the end of the process, static information about the initial product composition, particularly the amount of perfume and the amount of structurant used in the formula, still plays an important role. Knowledge about the reaction phases preceding the final event did not, in this case, seem to have a significant impact on the model's decision, which makes sense given how close we are to the end of the process (where data characteristics will typically start to outweigh expert knowledge more strongly than at the beginning).

Next, for Use Case 2: Continuous Residual Acrylic Acid & Epoxy Forecasting, Figure 22 shows the various forecasts made by the hybrid model for one of the evaluation experiments, namely, EXP904. Here, the red vertical line indicates the forecast we want to explain. In other words, we are interested

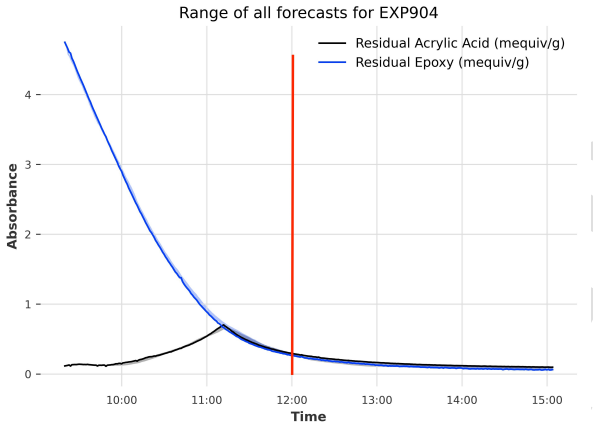


Figure 22: Hybrid forecasts for experiment EXP904 from Use Case 2: Continuous Residual Acrylic Acid & Epoxy Forecasting. Explanations have been generated for the forecast corresponding to the red vertical line.

in explaining a forecast taking place in the middle of the experiment. Recall that for Use Case 2: Continuous Residual Acrylic Acid & Epoxy Forecasting, each time point is associated with 120 subsequent forecasts (i.e., 60 acrylic acid and 60 epoxy forecasts need to be made). In Figure 23, we show only the explanation generated for the first epoxy forecast out of the total of 60 forecasts. The most important paths for the forecast in question can be rendered textually as follows, where each path has been unfurled by interposing connecting nodes wherever they are encountered:

1. Event 127 $\overset{hasEpoxyValue}{\longrightarrow}$ 0.220
2. Event 127 $\overset{hasPreviousEvent}{\longrightarrow}$ Event 126 $\overset{hasEpoxyValue}{\longrightarrow}$ 0.228
3. Event 127 $\overset{hasPreviousEvent}{\longrightarrow}$ Event 126 $\overset{hasPreviousEvent}{\longrightarrow}$ Event 125 $\overset{hasEpoxyValue}{\longrightarrow}$ 0.229
4. Event 127 $\overset{hasPreviousEvent}{\longrightarrow}$ Event 126 $\overset{hasTheoreticalEvent}{\longrightarrow}$ TheoreticalEvent_126 $\overset{hasDCEPValue}{\longrightarrow}$ -0.0193
5. Event 127 $\overset{hasProcessMonitoringState}{\longrightarrow}$ ProcessMonitoringState_127 $\overset{determinedBy}{\longrightarrow}$ EpoxySamplingStep_127 $\overset{hasResult}{\longrightarrow}$ 0.0929

The explanation shown above also accords with our intuition. The most important pieces of information informing the first epoxy forecast correspond with the most recently measured epoxy values. Aside from these measure-

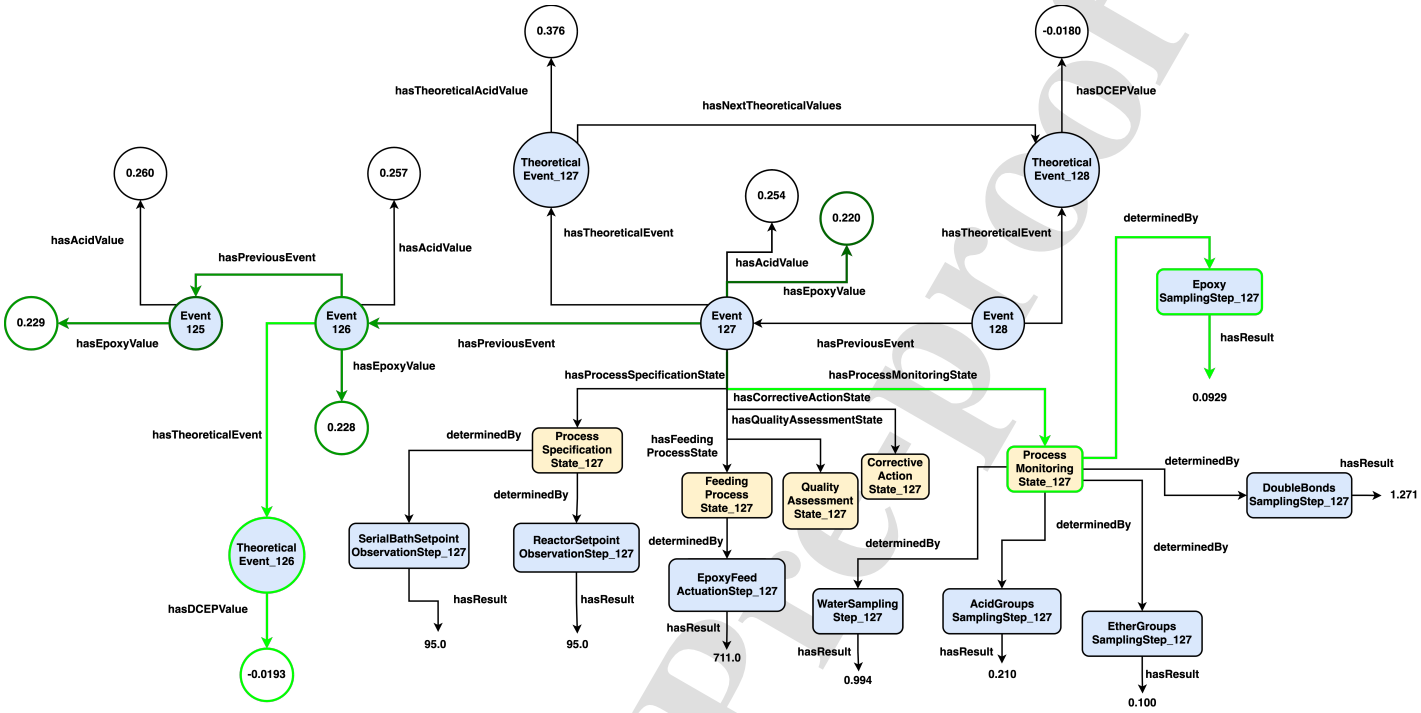


Figure 23: Explanation for hybrid forecast (epoxy) in Use Case 2: Continuous Residual Acrylic Acid & Epoxy Forecasting. The colour gradation indicates the degree of importance of each path on a normalised scale, with dark green reflecting the highest importance and light green the lowest.

ments, theoretical values seem to be influential, as well as specific monitored values pertaining to intermittent epoxy samples.

The examples provided above demonstrate what explanations generated by relying on INK and SHAP would look like in concrete situations. The explanations were rendered both in a textual format of interlinked nodes and relationships and in a more intuitive graphical format, making use of the TALK graph itself.

7. Overall Reflections

Following the evaluations conducted in the previous section, it is now possible to reflect on the challenges identified at the outset of this research paper, as well as the contributions made to address those challenges. Specifically,

the current research has attended to the aforementioned challenges in the following ways:

- Many hybrid AI approaches within PMC/PSE are domain-specific and usually limited to particular quantitative forms of expert knowledge. To address this, we designed a hybrid AI approach relying on KGs and KG embeddings to formally represent process knowledge potentially originating from diverse PMC/PSE subdomains, thus effecting a more generic solution capable of integrating a wide range of different modalities, including process flows.

- Most KG embedding techniques are not suitable for capturing the characteristics of industrial process knowledge, which usually includes numerical (sensory) data and long-range sequences or execution flows. By adapting the INK+TALK paradigm to a PMC/PSE setting, we were able to represent process knowledge in a graph format suited to event-specific neighbourhood embeddings which retain both numerical features and long-range sequences up to a configurable depth.

- Achieving greater degrees of decision transparency is a well-known challenge when applying ML to PMC/PSE problems. However, most KG embedding techniques assume a latent feature space, making the inputs derived from these techniques inherently opaque in terms of their interpretation. By leveraging INK's inherent transparency, together with TALK's efficient and informative organisation of process knowledge, we were able to integrate SHAP to identify the most salient features for any given decision and map them onto the TALK graphs from which they were extracted.

- We evaluated our methodology on two separate, real-world use cases situated in the chemical engineering domain, both of which require us to make forecasts about the further evolution of a given production process. A separate evaluation pipeline was set up for each use case, together with ML models trained on the corresponding downstream task. The hybrid approach involving KGs and INK+TALK was compared against an architecturally similar baseline data-driven approach which does not integrate any expert process knowledge. The hybrid approach was found to offer a competitive performance on all evaluation metrics, with average improvements of up to 8.57% and 10.21% for

the first and second use case respectively. For each use case, KG-driven explanations were generated at different points in the forecasting process, both in a textual format and in a more intuitive graphical format, illustrating the role of TALK graphs and KGs in general in making decision explanations more intelligible.

Potential drawbacks of the approach can also be identified. First, while the meta-ontology for PMC/PSE can be applied to different application domains, each domain KG still needs to be constructed. While KG generation techniques can be leveraged to expedite this part of the process, reliance on KGs involves an extra modelling cost that is integral to the proposed methodology. Having tools in place to assist in the modelling procedure is therefore instrumental to hybrid approaches for PMC/PSE. As stated earlier in this paper, it is possible to construct hybrid models without using KGs. Traditional forms of hybrid modelling mostly made use of data-driven approaches to inform mechanistic first principles models. While this can be effective in specific use cases, it precludes knowledge integration in a more generic fashion. Process workflows and other qualitative tacit knowledge that is difficult to capture using mathematical equations, require the symbolic infrastructure provided by KGs to be adequately represented. Conversely, traditional forms of hybrid modelling that try to augment data-driven performance with first principles models do so indirectly by influencing the learning procedure instead of integrating process knowledge directly as a learning input. While ideally the latter approach could be seen as complementary to our own KG-based approach, it usually involves extensive manual expertise to be set up for specific problem domains, as individual augmentations are tied to particular first principles models. Here, KGs offer a more generic form of data fusion, which is more task-agnostic even if, as stated previously, a non-negligible amount of effort needs to be invested in knowledge construction.

Second, depending on the complexity of the process knowledge we wish to model, the depth of the INK embeddings can be increased to accommodate longer-range sequences. However, this also increases the complexity of the input space, potentially leading to a dimensionality explosion. To mitigate this eventuality, which may be detrimental to predictive performance, as already discussed for the second use case, feature selection techniques like PowerShap can limit the size of the final feature space. At this point, we can also briefly discuss the algorithmic complexity of INK itself, and of the overall approach in which it is used. For a given TALK graph with N_f focal (i.e.,

event) nodes and an average neighbourhood size c, and for a given maximal extraction depth d, the worst-case time complexity is determined analogously to that of a bounded variant of the depth-first search algorithm. The way INK performs neighbourhood extraction follows the methodology of depth-limited search, which is itself analogous to iterative deepening but with only a single iteration up to a predetermined depth[2] [71, 72]. However, in the case of INK, the objective is not to locate and retrieve any particular element in the search space, meaning that the worst-case scenario will always be identical to the best-case scenario (i.e., search is performed exhaustively every time). The time complexity of iterative deepening is given by $O(b^{d'})$ with b the branching factor of the problem space and d' the depth of the solution of the problem [72]. The branching factor can be determined in a number of ways; but one common way of defining it is to use the average number of neighbours in the graph [73]. If we transpose this analysis to INK, we observe that the branching factor b is given by the average neighbourhood size c and that, given our prior remarks on INK's exhaustive way of "searching," d' is equivalent to d. If we were to apply INK to a given TALK graph, the time complexity scales linearly in function of the number of focal nodes, such that it is equal to $O(N_f c^d)$. If we define N_f^{total} as the total number of focal nodes in the data set for a given problem domain, then the total time complexity for extracting INK representations is given by $O(N_f^{total} c^d)$, where c is defined across all TALK graphs. To further reduce this overall complexity, and depending on the task at hand, it might be possible to augment the current neighbourhood extraction approach with properties of breadth-first search, by branching directly to neighbours with the most salient features after a given depth has been explored. Branches containing redundant info could even be pruned to avoid assimilating less informative or duplicative features. Note that the latter also addresses the problems associated with undesirable dimensionality increases.

Third, as we have observed during the evaluation, hybrid performance occasionally degrades towards the end of the forecasting procedure *vis-à-vis* the more straightforward baseline approaches. This suggests that the hybrid approach is not always able to sufficiently prioritise relevant sensor data when making its final predictions. A combined approach may be opted for

[2]Generally speaking, iterative deepening involves multiple iterations of depth-limited search, starting from depth one until the solution has been found.

to mitigate this, with data-driven predictions being preferred in later stages. Alternatively, a more clever state attribution scheme could be devised, where old static information gets phased out for events taking place later in the process.

Fourth, while combining INK, TALK, and SHAP enables hybrid decision transparency in the way we have shown, further improvements can be considered. For instance, contrastive and counterfactual explanations have been found to accord well with human understanding. As SHAP can be used to generate such explanations by mutating features and generating counterfactual answers, it would be worthwhile to investigate how they could be integrated into our graph-based approach [74]. Furthermore, by leveraging large language models (LLMs), it may be possible to augment the graph-based explanations with additional narrative support [75]. We would have to investigate whether generated narratives could be integrated directly into the graph-based structure, or whether it would be better to supplement the explanations we already have with explanations expressed in natural language. The use of LLMs to provide additional explanatory capabilities could be applied both to the non-contrastive, non-counterfactual, graph-based explanations we have presented in this paper, and to hypothetical contrastive and counterfactual explanations.

Finally, the proposed methodology was benchmarked on two datasets that are not publicly available as they contain proprietary information. These datasets were opted for insofar as they exhibit many of the desirable characteristics that are needed to evaluate the methodology proposed in this paper. While the vast majority of publications in the field of AI-driven PMC/PSE make use of closed datasets reflecting real manufacturing processes [2], some publicly available manufacturing datasets do exist [2, 76]. However, while the latter often include at least one of a variety of relevant input variables (e.g., process parameters, sensor data), such datasets usually lack an adequate representation of the procedural knowledge maintained by domain experts, or even, as far as we were able to ascertain, a clear description of the process flow itself across various executions [2]. Conversely, benchmarks that are often used to evaluate hybrid AI with Knowledge Graphs, such as MUTAG [77], KGBench [78], and KGrEaT [79] do not contain procedural or sequential knowledge. We should also note that, while at least one framework has been proposed to generate synthetic datasets integrating both process data and procedural knowledge [80], this framework does not seem to support detailed knowledge of the overall production process or the production setup,

limited as it is to modelling quantified process parameters and quality characteristics. While the two use cases employed in this paper are by no means optimal, they do have the benefit of satisfying the desired characteristics, making them ideally suited for evaluation purposes. Furthermore, these use cases reflect real-world (i.e., not synthetic) industrial processes, thus providing a valuable touchstone to test the methodology's validity and overall efficacy in realistic settings.

8. Conclusion

Throughout this work, we have proposed a generic methodology to solve various downstream tasks in PMC/PSE subdomains. Our methodology relies on the TALK paradigm to unify historical data and processual knowledge in a single data structure, together with the INK algorithm to extract embedded representations from different focal nodes. The TALK paradigm is flexible and generic enough to be extensible to a broad range of PMC/PSE application domains, where events can model any kind of prediction target. INK embeddings, in turn, can be used to train a large variety of ML models. By maintaining the transparency of individual feature dimensions, while simultaneously integrating process knowledge, our methodology lends itself especially well to decision-level explainability. The methodology as a whole has been evaluated on two separate use cases reflecting real-world PMC/PSE problems, where it was found to be on par with or outperform data-driven baselines on all metrics. The explainability enabled by pairing our methodology with the SHAP algorithm was also illustrated on these use cases. In all, we believe we have demonstrated both the feasibility of using INK+TALK to enable knowledge-driven PMC/PSE as well as its efficacy *vis-à-vis* data-driven alternatives on realistic use cases.

As regards future work, we would be interested in exploring how well the methodology generalises to a greater variety of subdomains, types of domain knowledge, and ML models. More extensive forms of explainability involving counterfactuals, contrastive explanations, and large language models could also be investigated. Additionally, as intimated in the previous section, we are interested in investigating how combined approaches, integrating both purely data-driven models and hybrid AI, could be devised, or alternatively, whether a more flexible form of state attribution for phasing out out-of-date information would be more feasible. Finally, we are currently designing an intuitive methodology with which domain experts could easily translate

their knowledge of a specific industrial process to a formal KG structure. We believe this would further stimulate the adoption of the hybrid methodology presented throughout this paper.

CRediT Authorship Contribution Statement

Michael Weyns: Conceptualization, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review and editing. **Thibault Blyau:** Data curation, Formal analysis, Investigation, Methodology, Validation, Visualization, Writing – review and editing. **Bram Steenwinckel:** Conceptualization, Methodology, Software, Writing – review and editing. **Filip De Turck:** Conceptualization, Supervision, Writing – review and editing. **Sofie Van Hoecke:** Conceptualization, Formal analysis, Supervision, Funding acquisition, Project administration, Resources, Writing – review and editing. **Femke Ongenae:** Conceptualization, Formal analysis, Investigation, Methodology, Validation, Supervision, Funding acquisition, Project administration, Resources, Writing – review and editing.

Declaration of Competing Interest

The authors declare that they have no competing interests.

Data Availability

We are not able to disclose any data used to make the evaluations presented throughout this paper, as it contains proprietary industrial process information by P&G and Allnex that we are not at liberty to share. However, we believe using these datasets for evaluation was crucial to demonstrate real-world industrial process applicability. We have no knowledge of open datasets with these properties. Furthermore, the code for the INK[3] and TALK [4] algorithms (including algorithms to enable explainability) underlying our proposed methodology have been made available on GitHub.

[3] `https://github.com/predict-idlab/INK`
[4] `https://github.com/predict-idlab/TALK`

Acknowledgements

Part of this research was supported through the Flemish Government (AI Research Program). Part of this research was also supported through the CHAI project[5], an imec.icon research project funded by imec and Agentschap Innoveren & Ondernemen (VLAIO), with Allnex, Procter & Gamble, Dotdash, and IDLab as partners. The work of Michael Weyns and Thibault Blyau was supported by the Research Foundation - Flanders (FWO) through their strategic basis research doctoral grants (1SD8821N and 1SH2C24N).

[5] https://www.imec.be/nl/vlaamse-innovatiemotor/samenwerking/imec-icon/projectportfolio/chai

References

[1] M. Soori, B. Arezoo, R. Dastres, Internet of things for smart factories in industry 4.0, a review, Internet of Things and Cyber-Physical Systems 3 (2023) 192–204.

[2] H. Tercan, T. Meisen, Machine learning and deep learning based predictive quality in manufacturing: a systematic review, Journal of Intelligent Manufacturing 33 (7) (2022) 1879–1905.

[3] K. M. Jablonka, C. Charalambous, E. Sanchez Fernandez, G. Wiechers, J. Monteiro, P. Moser, B. Smit, S. Garcia, Machine learning for industrial processes: Forecasting amine emissions from a carbon capture plant, Science Advances 9 (1) (2023) eadc9576.

[4] P. Martinez, M. Al-Hussein, R. Ahmad, Intelligent vision-based online inspection system of screw-fastening operations in light-gauge steel frame manufacturing, The International Journal of Advanced Manufacturing Technology 109 (3) (2020) 645–657.

[5] C.-Y. Hsu, W.-C. Liu, Multiple time-series convolutional neural network for fault detection and diagnosis and empirical study in semiconductor manufacturing, Journal of Intelligent Manufacturing 32 (3) (2021) 823–836.

[6] P. Daoutidis, J. H. Lee, S. Rangarajan, L. Chiang, B. Gopaluni, A. M. Schweidtmann, I. Harjunkoski, M. Mercangöz, A. Mesbah, F. Boukouvala, et al., Machine learning in process systems engineering: Challenges and opportunities, Computers & Chemical Engineering 181 (2023) 108523.

[7] M. R. Dobbelaere, P. P. Plehiers, R. Van de Vijver, C. V. Stevens, K. M. Van Geem, Machine learning in chemical engineering: strengths, weaknesses, opportunities, and threats, Engineering 7 (9) (2021) 1201–1211.

[8] M. Mowbray, M. Vallerio, C. Perez-Galvan, D. Zhang, A. D. R. Chanona, F. J. Navarro-Brull, Industrial data science–a review of machine learning applications for chemical and process industries, Reaction Chemistry & Engineering 7 (7) (2022) 1471–1509.

[9] Y. Hu, Y. Man, Energy consumption and carbon emissions forecasting for industrial processes: Status, challenges and perspectives, Renewable and Sustainable Energy Reviews 182 (2023) 113405.

[10] J. M. Aguiar-Pérez, M. Á. Pérez-Juárez, An insight of deep learning based demand forecasting in smart grids, Sensors 23 (3) (2023) 1467.

[11] C. C. Pantelides, J. G. Renfro, The online use of first-principles models in process operations: Review, current status and future needs, Computers & Chemical Engineering 51 (2013) 136–148.

[12] J. Sansana, M. N. Joswiak, I. Castillo, Z. Wang, R. Rendall, L. H. Chiang, M. S. Reis, Recent trends on hybrid modeling for industry 4.0, Computers & Chemical Engineering 151 (2021) 107365.

[13] A. Azarpour, T. N. Borhani, S. R. W. Alwi, Z. A. Manan, M. I. A. Mutalib, A generic hybrid model development for process analysis of industrial fixed-bed catalytic reactors, Chemical Engineering Research and Design 117 (2017) 149–167.

[14] H. Vedam, V. Venkatasubramanian, Pca-sdg based process monitoring and fault diagnosis, Control engineering practice 7 (7) (1999) 903–917.

[15] T. Bikmukhametov, J. Jäschke, Combining machine learning and process engineering physics towards enhanced accuracy and explainability of data-driven models, Computers & Chemical Engineering 138 (2020) 106834.

[16] N. Sharma, Y. Liu, A hybrid science-guided machine learning approach for modeling chemical processes: A review, AIChE Journal 68 (5) (2022) e17609.

[17] L. Rajulapati, S. Chinta, B. Shyamala, R. Rengaswamy, Integration of machine learning and first principles models, AIChE Journal 68 (6) (2022) e17715.

[18] X. Wilcke, et al., The knowledge graph as the default data model for machine learning. data sci. 1, 39–57 (2017) (2017).

[19] S. Bachhofner, E. Kiesling, K. Revoredo, P. Waibel, A. Polleres, Automated process knowledge graph construction from bpmn models, in:

International Conference on Database and Expert Systems Applications, Springer, 2022, pp. 32–47.

[20] M. Yahya, J. G. Breslin, M. I. Ali, Semantic web and knowledge graphs for industry 4.0, Applied Sciences 11 (11) (2021) 5110.

[21] Y. Wan, Y. Liu, Z. Chen, C. Chen, X. Li, F. Hu, M. Packianather, Making knowledge graphs work for smart manufacturing: Research topics, applications and prospects, Journal of Manufacturing Systems 76 (2024) 103–132.

[22] G. Buchgeher, D. Gabauer, J. Martinez-Gil, L. Ehrlinger, Knowledge graphs in manufacturing and production: a systematic literature review, IEEE Access 9 (2021) 55537–55554.

[23] G. Song, D. Fu, D. Zhang, From knowledge graph development to serving industrial knowledge automation: a review, in: 2022 41st Chinese Control Conference (CCC), IEEE, 2022, pp. 4219–4226.

[24] A. Banerjee, R. Dalal, S. Mittal, K. P. Joshi, et al., Generating digital twin models using knowledge graphs for industrial production lines, in: Workshop on Industrial Knowledge Graphs, co-located with the 9th International ACM Web Science Conference 2017, Association for Computing Machinery, 2017, pp. 425–430.

[25] G. Stavropoulou, K. Tsitseklis, L. Mavraidi, K.-I. Chang, A. Zafeiropoulos, V. Karyotis, S. Papavassiliou, Digital twin meets knowledge graph for intelligent manufacturing processes, Sensors 24 (8) (2024) 2618.

[26] M. Nickel, K. Murphy, V. Tresp, E. Gabrilovich, A review of relational machine learning for knowledge graphs, Proceedings of the IEEE 104 (1) (2015) 11–33.

[27] P. Ristoski, H. Paulheim, Rdf2vec: Rdf graph embeddings for data mining, in: International semantic web conference, Springer, 2016, pp. 498–514.

[28] B. Steenwinckel, M. De Brouwer, M. Stojchevska, F. De Turck, S. Van Hoecke, F. Ongenae, Talk: Tracking activities by linking knowledge, Engineering Applications of Artificial Intelligence 122 (2023) 106076.

[29] B. Steenwinckel, G. Vandewiele, M. Weyns, T. Agozzino, F. D. Turck, F. Ongenae, Ink: knowledge graph embeddings for node classification, Data Mining and Knowledge Discovery 36 (2) (2022) 620–667.

[30] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: Advances in neural information processing systems, Vol. 30, Curran Associates, Inc., 2017, pp. 4765–4774.

[31] S.-H. Wang, H. S. Pillai, S. Wang, L. E. Achenie, H. Xin, Infusing theory into deep learning for interpretable reactivity prediction, Nature communications 12 (1) (2021) 5288.

[32] D. Rodriguez-Granrose, A. Jones, H. Loftus, T. Tandeski, W. Heaton, K. T. Foley, L. Silverman, Design of experiment (doe) applied to artificial neural network architecture enables rapid bioprocess improvement, Bioprocess and Biosystems Engineering 44 (6) (2021) 1301–1308.

[33] M. S. Alhajeri, J. Luo, Z. Wu, F. Albalawi, P. D. Christofides, Process structure-based recurrent neural network modeling for predictive control: A comparative study, Chemical Engineering Research and Design 179 (2022) 77–89.

[34] G. De Giacomo, M. Lenzerini, et al., Tbox and abox reasoning in expressive description logics., KR 96 (316-327) (1996) 10.

[35] B. Zhou, J. Bao, Y. Liu, D. Song, Ba-ikg: Bilstm embedded albert for industrial knowledge graph generation and reuse, in: 2020 IEEE 18th International Conference on Industrial Informatics (INDIN), Vol. 1, IEEE, 2020, pp. 63–69.

[36] P. Wen, Y. Ma, R. Wang, Systematic knowledge modeling and extraction methods for manufacturing process planning based on knowledge graph, Advanced Engineering Informatics 58 (2023) 102172.

[37] H. Han, J. Wang, X. Wang, S. Chen, Construction and evolution of fault diagnosis knowledge graph in industrial process, IEEE Transactions on Instrumentation and Measurement 71 (2022) 1–12.

[38] B. Steenwinckel, P. Heyvaert, D. De Paepe, O. Janssens, S. Vanden Hautte, A. Dimou, F. De Turck, S. Van Hoecke, F. Ongenae, Towards adaptive anomaly detection and root cause analysis by automated

extraction of knowledge from risk analyses, in: 9th International Semantic Sensor Networks Workshop, Co-Located with 17th International Semantic Web Conference (ISWC 2018), Vol. 2213, CEUR-WS Team, 2018, pp. 17–31.

[39] B. Steenwinckel, D. De Paepe, S. V. Hautte, P. Heyvaert, M. Bentefrit, P. Moens, A. Dimou, B. Van Den Bossche, F. De Turck, S. Van Hoecke, et al., Flags: A methodology for adaptive anomaly detection and root cause analysis on sensor data streams by fusing expert knowledge with machine learning, Future Generation Computer Systems 116 (2021) 30–48.

[40] X. Liang, Q. Zhang, Y. Man, Z. He, Toward sustainable process industry based on knowledge graph: a case study of papermaking process fault diagnosis, Discover Sustainability 5 (1) (2024) 1–17.

[41] M. Nickel, V. Tresp, Tensor factorization for multi-relational learning, in: Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III 13, Springer, 2013, pp. 617–621.

[42] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: Advances in neural information processing systems, Curran Associates, Inc., 2013, pp. 2787–2795.

[43] W. Huang, J. Liu, T. Li, S. Ji, D. Wang, T. Huang, Fedcke: Cross-domain knowledge graph embedding in federated learning, IEEE Transactions on Big Data 9 (3) (2022) 792–804.

[44] Y. Lin, Z. Liu, H. Luan, M. Sun, S. Rao, S. Liu, Modeling relation paths for representation learning of knowledge bases, arXiv preprint arXiv:1506.00379 (2015).

[45] A. García-Durán, A. Bordes, N. Usunier, Composing relationships with translations, Ph.D. thesis, CNRS, Heudiasyc (2015).

[46] B. Yang, W.-t. Yih, X. He, J. Gao, L. Deng, Embedding entities and relations for learning and inference in knowledge bases, arXiv preprint arXiv:1412.6575 (2014).

[47] T. Trouillon, C. R. Dance, É. Gaussier, J. Welbl, S. Riedel, G. Bouchard, Knowledge graph completion via complex tensor factorization, Journal of Machine Learning Research 18 (130) (2017) 1–38.

[48] S. M. Kazemi, D. Poole, Simple embedding for link prediction in knowledge graphs, in: Advances in neural information processing systems, Curran Associates, Inc., 2018, pp. 4284–4295.

[49] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, M. Welling, Modeling relational data with graph convolutional networks, in: The semantic web: 15th international conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, proceedings 15, Springer, 2018, pp. 593–607.

[50] L. Cai, B. Yan, G. Mai, K. Janowicz, R. Zhu, Transgcn: Coupling transformation assumptions with graph convolutional networks for link prediction, in: Proceedings of the 10th international conference on knowledge capture, Association for Computing Machinery, 2019, pp. 131–138.

[51] C. Qian, L. Wen, A. Kumar, Tracewalk: Semantic-based process graph embedding for consistency checking, arXiv preprint arXiv:1905.06883 (2019).

[52] A. Narayanan, M. Chandramohan, R. Venkatesan, L. Chen, Y. Liu, S. Jaiswal, graph2vec: Learning distributed representations of graphs. arxiv 2017, arXiv preprint arXiv:1707.05005 (2017).

[53] D. Wang, P. Zhang, P. Deng, T. Li, Temporal knowledge graph embedding for metro flow analysis, in: Intelligent Management of Data and Information in Decision Making: Proceedings of the 16th FLINS Conference on Computational Intelligence in Decision and Control & the 19th ISKE Conference on Intelligence Systems and Knowledge Engineering (FLINS-ISKE 2024), World Scientific, 2024, pp. 501–508.

[54] J. M. Rožanec, P. Zajec, K. Kenda, I. Novalija, B. Fortuna, D. Mladenić, Xai-kg: knowledge graph to support xai and decision-making in manufacturing, in: International Conference on Advanced Information Systems Engineering, Springer, 2021, pp. 167–172.

[55] J. Rožanec, E. Trajkova, I. Novalija, P. Zajec, K. Kenda, B. Fortuna, D. Mladenić, Enriching artificial intelligence explanations with knowledge fragments, Future internet 14 (5) (2022) 134.

[56] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. McIlraith, S. Narayanan, M. Paolucci, B. Parsia, T. Payne, et al., Owl-s: Semantic markup for web services, W3C member submission 22 (4) (2004).

[57] H. Neuhaus, M. Compton, The semantic sensor network ontology, in: AGILE workshop on challenges in geospatial data harmonisation, Hannover, Germany, 2009, pp. 1–33.

[58] J. Morbach, A. Yang, W. Marquardt, Ontocape—a large-scale ontology for chemical process engineering, Engineering applications of artificial intelligence 20 (2) (2007) 147–161.

[59] D. Garijo, Y. Gil, A new approach for publishing workflows: abstractions, standards, and linked data, in: Proceedings of the 6th workshop on Workflows in support of large-scale science, Association for Computing Machinery, 2011, pp. 47–56.

[60] A. Annane, N. Aussenac-Gilles, M. Kamel, Bbo: Bpmn 2.0 based ontology for business process representation, in: 20th European Conference on Knowledge Management (ECKM 2019), Vol. 1, Academic Conferences and Publishing International Limited, 2019, pp. 49–59.

[61] S. Verstichel, F. Ongenae, L. Loeve, F. Vermeulen, P. Dings, B. Dhoedt, T. Dhaene, F. De Turck, Efficient data integration in the railway domain through an ontology-based methodology, Transportation Research Part C: Emerging Technologies 19 (4) (2011) 617–643.

[62] M. T. Ribeiro, S. Singh, C. Guestrin, "why should i trust you?" explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, Association for Computing Machinery, 2016, pp. 1135–1144.

[63] E. Pastor, E. Baralis, Explaining black box models by means of local rules, in: Proceedings of the 34th ACM/SIGAPP symposium on applied computing, Association for Computing Machinery, 2019, pp. 510–517.

[64] R. Guidotti, A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini, F. Giannotti, Local rule-based explanations of black box decision systems, arXiv preprint arXiv:1805.10820 (2018).

[65] M. T. Ribeiro, S. Singh, C. Guestrin, Anchors: High-precision model-agnostic explanations, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 32, AAAI Press, 2018.

[66] M. De Micco, D. Gragnaniello, F. Zonfrilli, V. Guida, M. M. Villone, G. Poggi, L. Verdoliva, Stability assessment of liquid formulations: A deep learning approach, Chemical Engineering Science 262 (2022) 117991.

[67] G. Merfeld, C. Molaison, R. Koeniger, A. E. Acar, S. Mordhorst, J. Suriano, P. Irwin, R. S. Warner, K. Gray, M. Smith, et al., Acid/epoxy reaction catalyst screening for low temperature (120 c) powder coatings, Progress in organic coatings 52 (2) (2005) 98–109.

[68] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, A. Gulin, Catboost: unbiased boosting with categorical features, Advances in neural information processing systems 31 (2018) 6638–6648.

[69] J. Verhaeghe, J. Van Der Donckt, F. Ongenae, S. Van Hoecke, Powershap: A power-full shapley feature selection method, in: Joint European conference on machine learning and knowledge discovery in databases, Springer, 2022, pp. 71–87.

[70] T. Blyau, M. Weyns, F. Ongenae, S. Verstockt, S. Van Hoecke, Physics-informed machine learning for quality forecasting in chemical production processes (unpublished results).

[71] R. R. Kumar, T. S. S. Vadlapatla, S. Azmeera, A. Komaraneni, A. Jula, Performance comparison of depth limited search and a* algorithm: A case study, in: E3S Web of Conferences, Vol. 391, EDP Sciences, 2023, p. 01140.

[72] R. E. Korf, Depth-first iterative-deepening: An optimal admissible tree search, Artificial intelligence 27 (1) (1985) 97–109.

[73] R. K. Devarapalli, A. Biswas, Processing large graphs with an alternative representation, in: Advances in Computers, Vol. 128, Elsevier, 2023, pp. 185–212.

[74] S. Rathi, Generating counterfactual and contrastive explanations using shap, arXiv preprint arXiv:1906.09293 (2019).

[75] A. Zytek, S. Pidò, K. Veeramachaneni, Llms for xai: Future directions for explaining explanations, arXiv preprint arXiv:2405.06064 (2024).

[76] J. Krauß, J. Dorißen, H. Mende, M. Frye, R. H. Schmitt, Machine learning and artificial intelligence in production: Application areas and publicly available data sets: Maschinelles lernen und kü nstliche intelligenz in der produktion: Anwendungsgebiete und öffentlich zugängliche datensätze, in: Production at the leading edge of technology: Proceedings of the 9th Congress of the German Academic Association for Production Technology (WGP), September 30th-October 2nd, Hamburg 2019, Springer, 2019, pp. 493–501.

[77] P. Ristoski, G. K. D. De Vries, H. Paulheim, A collection of benchmark datasets for systematic evaluations of machine learning on the semantic web, in: The Semantic Web–ISWC 2016: 15th International Semantic Web Conference, Kobe, Japan, October 17–21, 2016, Proceedings, Part II 15, Springer, 2016, pp. 186–194.

[78] P. Bloem, X. Wilcke, L. van Berkel, V. de Boer, kgbench: A collection of knowledge graph datasets for evaluating relational and multimodal machine learning, in: European Semantic Web Conference, Springer, 2021, pp. 614–630.

[79] N. Heist, S. Hertling, H. Paulheim, Kgreat: a framework to evaluate knowledge graphs via downstream tasks, in: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, 2023, pp. 3938–3942.

[80] R. Nordsieck, A. Schweizer, M. Heider, J. Hähner, Pdpk: A framework to synthesise process data and corresponding procedural knowledge for manufacturing, arXiv preprint arXiv:2308.08371 (2023).

Declaration of interests

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

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: