ONGOING: A Human-readable, Model-enriching, Continuous Technician Knowledge Modeling Framework

Adrien Bolling[1], Sylvain Kubler[2]

[1,2] *Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg*
adrien.bolling@uni.lu
sylvain.kubler@uni.lu

ABSTRACT

Industry 5.0 reframes manufacturing around human-centric concerns: resilient operations, safe work, and decisions people can understand and contest. For PHM, that means elevating human-based features: competency, recency of practice, mentoring links, explainability, and fair exposure, rather than relying only on sensors or opaque models. Today those signals sit in ticket logs and massive databases, making them hard to audit, transfer, or reuse at scale. We suggest ONGOING, a representation layer framework that turns unstructured maintenance text into a human-auditable Knowledge Grid and a complex but modular feature vector, independent of any particular embedding model or projector. At its core, the grid tracks technician experiences by incrementing a part of the Knowledge Grid whenever tickets are resolved. Two mechanisms capture more advanced dynamics: knowledge transfer between people (e.g., mentorship) via a convex blend of Knowledge Grids, and neighborhood propagation that diffuses experience increases to semantically adjacent tasks through a Gaussian kernel. From each grid we derive interpretable features, such as hypervolume, sparsity, or maximum knowledge, that summarize knowledge distribution more accurately for better downstream use (e.g., dispatching optimizer models, LLMs, production forecast models). We implement the framework on a partner company's data, and deploy an instance at-scale (50000 tickets, 100 technicians) in real-time, using a multilingual sentence encoder and a toroidal SOM for ticket embedding. On our deployed instance, we designed a technician recommendation use-case. A maintenance expert study with human feedback over 55 real tickets found that grid-based recommendation were judged more pertinent than a scalar-based and a vector-based knowledge modeling approaches. Crucially, dispatchers could articulate rationales from visible grid neighborhoods and feature attributions, preserving interpretability. Beyond dispatch support, the Knowledge Grid enables training planning (identify coverage gaps), fairness monitoring (avoid single-point failure through over-reliance on "heroes"), and promotes workload balancing.

1. INTRODUCTION

Modern maintenance organizations increasingly accumulate unstructured, high-volume ticket logs in their ERP. These logs contain extensive documentation about workers' preferences, skills, and expert knowledge. Yet most workforce tooling still reduces expertise to a primitive aggregated knowledge scalar, coarse skills matrix, or checklists maintained by supervisors, which are hard to keep current and provide weak signals for data-driven planning or automation (e.g., assignment, training) (Hadiyanto & Anggoro, 2025).
In parallel, AI systems for PHM are maturing. LLM agents and graph models are now increasingly being considered for operations support, but integration remains fragile unless human expertise is represented in a way that is both interpretable to operators and useful to learning systems (Lukens, McCabe, Gen, & Ali, 2024). We argue that representing technician knowledge as an auditable spatial representation over ticket semantics paves the way from ticket databases to operator interpretability and ML readiness.

This paper addresses the representation problem through the aspect of Technician Knowledge Modeling. We introduce a method-agnostic framework that turns historical databases, and real-time streams of maintenance tickets into a human-readable and feature-rich label and category-agnostic representation of technician knowledge. Concretely, we embed ticket texts and project them onto a fixed n-dimensional grid, yielding a knowledge grid whose geometry (what cells are occupied, how mass is distributed, how far the extremes lie) provides complex and precise metrics, features, and visualization capabilities that humans can audit and that downstream models can consume. Unlike optimization-centric HR allocation methods (IP/Hungarian/heuristics) that solve who-does-what-when, the focus of this framework is modeling: produce the most faithful, stable, and explainable description of what a technician knows and how that knowledge evolves. The framework is generic to embedding and projection choices, but we also provide an instantiation of the framework already deployed in a real-life full-scale production-line setting, both as a way to validate our approach and to provide examples of implementation choices.

The knowledge we model is intentionally abstract: there is no fixed “unit of expertise.” Instead, a technician’s knowledge is a function of how often they have solved similar tasks (i.e., density in regions of the grid). We supplement this relationship through two additional knowledge variation mechanisms: a neighboring-knowledge propagation mechanism, to account for the phenomenon where a technician knowledgeable about certain tickets will also likely be somehow proficient in semantically close tickets, and a knowledge transmission mechanism, to account for knowledge transfer between technicians, when one technician may be deployed to teach another about a certain type of machine or failure. These additional mechanisms allow us to finally build and update a Knowledge Grid for each technician, representing their knowledge through the whole ticket logs space.
From this Knowledge Grid we can extract several features, e.g., sparsity, inter-extremum distance, hypervolume, maximum density, cluster count and much more. For organizations already exploring ticket automation, this bridges a known gap: surveys show strong progress on ticket allocation, but limited attention to complex, human-readable knowledge representations that feed those models (Zangari, Marcuzzo, Schiavinato, Gasparetto, & Albarelli, 2023).

Immediate use-cases appear from this Knowledge Modeling: enrichment of ticket allocation model’s input features, training and team-making support to locate areas of knowledge too few technicians are trained on, avoiding cases where a technician becomes ultra-specialized which could be detrimental to their well-being, or identifying recruitment needs in a growing team for example.

We deliberately separate framework from instantiation. The framework defines clear interfaces (tickets, pre-embedding, dimensionality reduction, knowledge features) which guarantees human-readability, and ML-readiness. The real-life instantiation we report uses a specific multilingual encoder and a toroidal SOM, but these components may be replaced if a site’s data or constraints differ. This separation is practical: it lets sites keep their Enterprise Resource Planning (ERP), Computerized Maintenance Management System (CMMS), and it gives PHM teams freedom to evolve models without rewriting downstream dashboards or planners.

This work advances human–AI integration for PHM by converting growing and opaque ticket tables into human-readable maps and machine-useful features, creating an integration layer between shop-floor expertise and learning systems, and putting the focus back on human operators by representing their knowledge as model-trainable features, aligning with emerging human-centric PHM guidance that stresses transparency and usability for maintenance workers (Denu, David, Mangione, & Landry, 2024).

We are explicit about boundaries. There is a cold-start for new technicians; ticket text quality matters; and very high-resolution grids increase storage pressure (compute is not the bottleneck). None are blockers in practice: cold-start can borrow from fleet priors, low-signal text can be stabilized via templates or weak ontologies, and grid resolution can be right-sized per site. Finally, we emphasize that our contribution is not an optimizer. We complement optimizers and dispatchers by supplying better quality human-knowledge representations and features. In that sense, the knowledge grid is a missing layer in the PHM stack: it treats the human as a first-class system with a complex state, not just a resource with availability.

Contributions :

- **Framework:** A general framework for continuous, human-readable representation of technician knowledge. Per-technician Knowledge Grids derived from ticket text, interpretable by supervisors yet usable as features for ML.
- **Open Implementation :** A publicly available Python Implementation of the Framework’s core mechanics: Knowledge Grids, their Knowledge learning/forgetting model, and a collection of associated features. (Available at https://github.com/AdrienBolling/ONGOING)
- **Industrial Evidence:** Industrial deployment evidence of a use-case of technician assignment at plant scale (100 techs, 50000 tickets), favored by dispatchers and maintenance managers over current baseline methods of scalar and vector-based representations, validated through a prospective comparison over 55 tickets
- **Integration Guidance:** Integration guidance covering map sizing, SOM implementation choices, and operational considerations.

Section 2 reviews related work on the Human Resources Allocation Problem (HRAP) and the literature about Human Knowledge Modeling in AI. Section 3 formalizes the framework and mathematically defines the knowledge modeling mechanisms and a collection of knowledge features. Section 4 details the factory implementation, its environmental setup, and details the validation process by defining chosen technician allocation use-case and the comparison baselines, and reporting the results. Section 5 discusses limitations and deployment concerns. Finally Section 6 concludes with implications related to PHM’s human-AI integration agenda.

2. RELATED WORK

2.1. Human resources allocation problem

Traditionally, HRAP has been addressed using operations research techniques, such as linear programming, Hungarian methods, genetic algorithms, and ant colonies to list a few (Bouajaja & Dridi, 2017). However, these methods often require simplifications and assumptions that may not hold in dynamic, complex, and stochastic environments. Recent findings (Ruiz-Rodríguez, Kubler, Robert, & Le Traon, 2024) suggest that increasingly realistic environments, needed for efficient real-life deployment, can't be solved through meta-heuristics or mathematical approaches. Over the last few years however, these limitations are being challenged through the use of more AI-centric methods, such as Reinforcement Learning (RL) (Lv, Jiang, Wu, & Zhao, 2024) (Muklason et al., 2024) (Bolling & Kubler, 2024) (Platten, Macfarlane, Graus, & Mesbah, n.d.), Graph Neural Networks (Platten et al., n.d.) (Lu, Ye, Chen, & Hentenryck, 2025) (Nguyen, Truong, & Tran-Thanh, 2025) (Zhang et al., 2024), or LLM-based approaches (Wasi, 2024) (Iso, Pezeshkpour, Bhutani, & Hruschka, 2025).

The growing need for human-centric approaches in Industry 5.0 (Khanna, Kumari, & Karim, 2024) highlights the lack of representative human-based features, in particular the question of the representation of knowledge.
Accurately representing the skill, and by extension the knowledge of a technician is crucial to the correct training and implementation of any dispatching, assignment, or routing system. Moreover, any advanced scheduling system considers and expectation of the time a ticket will take to be treated, which has been shown to be directly linked to the experience of the technician (Jaber, Givi, & Neumann, 2013).

2.2. Human knowledge modeling

Table 2 presents a taxonomy of a recent representative sample of AI-oriented papers implementing a model of a human's knowledge. We identify five major fields of interest, not necessarily exclusionary : Education, Human Resources (HR), HRAP, Industrial, and Scheduling.
Looking at the type of representation currently present in the AI literature, we notice 4 types of approaches, some similar in nature :

- **Scalar-based:** This representation type is as simple as it gets. All knowledge of the technician is aggregated under a single value, making no distinction between areas of knowledge. Although undoubtedly lightweight, easy to implement, and likely to fit a vast range of situations, this model fails to grasp non-trivial features of a human's knowledge.

- **Vector-based:** Vector-based approaches are a definite improvement over scalar-based approaches. The core idea is to segment the knowledge of our technician in several predefined categories. Such categories will usually greatly improve the usability of the model, however they often lack proper granularity, and require the data to be previously labeled into these categories, which may be a long and expensive process when an organization decides to upgrade its infrastructure and switch to more detailed categories.

- **Matrix-based:** Matrix-based approaches can be considered a variation of the usual vector-based approach, as these matrices will usually be built as the cross-product of two sets of categories of different semantics (e.g., "electrical/mechanical" and the type of machine).

- **Neural Network-based:** Neural Network-based approaches usually rely on a memory-able architecture (e.g., Gated Recurrent Unit (GRU) cells, Long Short Term Memory (LSTM) units, or Recurrent Neural Networks (RNN)-like networks in general). The ticket text or an embedding of it will be fed through the RNN cell, but a definite measure of knowledge won't be outputted. It is only implicitly stored in the RNN cell which output will be a non-human-readable representation of technician knowledge.

In practice, current state-of-the-art approaches offer a hard trade-off between human-readability of a technician's knowl-

Table 1. Acronyms used in Table 2.

Notation	Description
Edu	Education
HR	Human Resources
HRAP	Human Resources Assignment Problem
Indus	Industrial
Sche	Scheduling
Stu	Student
Emp	Employee
Tech	Technician
Dyn	Dynamic
Sta	Static
L	Learning
F	Forgetting
Vec	Vector
Mat	Matrix
Sca	Scalar
RNN	Recurrent Neural Network
GNN	Graph Neural Network
Opt	Optimization
Pip	Pipeline
Heur	Heuristic
RL	Reinforcement Learning
Ana	Analysis
ML	Machine Learning
DP	Dynamic Programming
Sim	Simulation
MA	Memetic Algorithm
DL	Deep Learning
TS	Tree Search

Table 2. Summary table of the litterature review under the following criteria : Field, type of Human agent, type of Knowledge modeled, Evolution of the knowledge if applicable, Representation of the knowledge, Number of Categories of knowledge if applicable, the type of model the knowledge serves as an Input to.

Article	Field	Human	Know.	Var.	Repr.	Num. Cat.	Input
(Liu et al., 2019)	Edu	Stu	Dyn	L, F	Vec	-	RNN
(Liang, Peng, Pu, & Wu, 2022)	Edu	Stu	Dyn	L, F	Mat	-	RNN
(Hashemifar & Sahebi, 2025)	Edu	Stu	Dyn	L, F	RNN	-	RNN
(Wasi, 2024)	HR	Emp	Sta	-	Vec	10 - 20	GNN
(Lee & Ahn, 2020)	HR	Emp	Sta	-	Vec	10 - 20	Opt
(Alvarez, Mohammed, & Lastra, 2025)	HR	Tech	Dyn	L	Vec	5 - 10	Pip
(Zhang et al., 2024)	HRAP	Tech	Sta	-	Vec	-	GNN
(Gonçalves, Alvelos, & Moura, 2025)	HRAP	Emp	Sta	-	Vec	-	Opt
(Muraretu, Ilie, & Ilie, 2017)	HRAP	Emp	Sta	-	Vec	5 - 10	Heur
(Lv et al., 2024)	HRAP	Emp	Sta	-	Vec	-	RL
(Tien & Prabhu, 2018)	Indus	Tech	Dyn	L, F	Sca	1	Ana
(Long-fei, Nakamura, & Kondo, 2020)	Indus	Tech	Dyn	L	Vec	-	ML
(Szwarc & Golińska-Dawson, 2024)	Indus	Tech	Dyn	L, F	Sca	1	Opt
(Muklason et al., 2024)	HRAP	Emp	Sta	-	Vec	2 - 5	RL
(Henao, Mercado, & González, 2023)	Indus	Tech	Dyn	L, F	Vec	-	Opt
(Ranasinghe, Senanayake, & Grosse, 2024)	Indus	Tech	Dyn	L, F	Sca	1	Ana
(X. Chen, Li, Lin, & Ding, 2024)	Indus	Tech	Sta	-	Vec	5 - 10	DP
(Denu et al., 2024)	Indus	Tech	Dyn	L	Sca	1	Sim
(Xu, Xie, & Hall, 2025)	Indus	Tech	Dyn	L, F	Sca	1	Opt
(Saber, Leyman, & Aghezzaf, 2024)	Indus, HRAP	Tech	Sta	-	Vec	5 - 10	Heur
(Han & Gong, 2025)	Indus, HRAP	Tech	Dyn	L, F	Sca	1	MA
(Imran Hasan Tusar & Sarker, 2024)	Indus, Sche	Tech	Sta	-	Vec	5 - 10	Opt
(Heuser & Tauer, 2023)	Indus, Sche	Tech	Dyn	L, F	Vec	-	Opt
(Safaei & Kiassat, 2018)	Sche	Tech	Sta	-	Vec	5 - 10	Heur
(Stein, Hildebrandt, & Thomas, 2024)	Sche	Tech	Sta	-	Sca	1	RL
(Felberbauer, Gutjahr, & Doerner, 2019)	Sche	Emp	Sta	-	Vec	5 - 10	Opt
(Yang, Li, Luo, Li, & Wen, 2025)	Sche	Tech	Dyn	L, F	Vec	1	Heur, Sim
(Z. Chen, De Causmaecker, & Dou, 2023)	Sche	Emp	Sta	-	Vec	2 - 5	DL, TS

edge model (so planners/auditors can reason about it) and suitability as an input feature for modern downstream complex learning systems (GNNs for assignment/routing, LLM agents for tool-use, RL for dispatch). Interpretable models (scalar, vector, matrix) are easy to audit and map to real skill taxonomies, but they saturate fast, missing cross-skill interactions, recency effects, task-specific transfer, complex human learning and forgetting behaviors, and context. Latent models (RNN/GRU/LSTM states, transformer embeddings, graph embeddings) capture those effects and usually win on predictive/control metrics, but they're opaque and harder to govern. Complex explainability methods need to be employed to even start to crack the black box.

To bridge this gap we introduce ONGOING: a Technician Knowledge Modeling Framework serving as a compromise between the human-readability of scalar, vector, and matrix based approaches, and the strengths of powerful but costly methods such as LLMs, RNNS, transformers and GMMs, namely their ability to provide a continuous, category-free representation of knowledge, and to provide downstream learning models with high quality input features characterizing the technician knowledge.

3. MODEL AND FRAMEWORK

3.1. Overview

ONGOING is a method-agnostic framework that turns historical and streaming maintenance tickets into a knowledge representation for each technician and team. The pipeline is simple by design, as shown in Figure 1: tickets are extracted from the company ERP, then encoded, projected through dimensionality reduction, and fed into each technician's own Knowledge Grid.

These Knowledge Grids can then be audited by human operators, or characterized as interpretable features and machine-ready descriptors to feed downstream tasks, such as assignment, scheduling, or neural network training. ONGOING separates representation from optimization: it does not schedule work, it supplies complex human-knowledge features and maps to whatever scheduler or predictor a site already trusts. The framework exposes clean interfaces and supports streaming updates for day-to-day stability, with periodic retraining only when the underlying production process undergoes significant material changes (e.g., a new machine type).

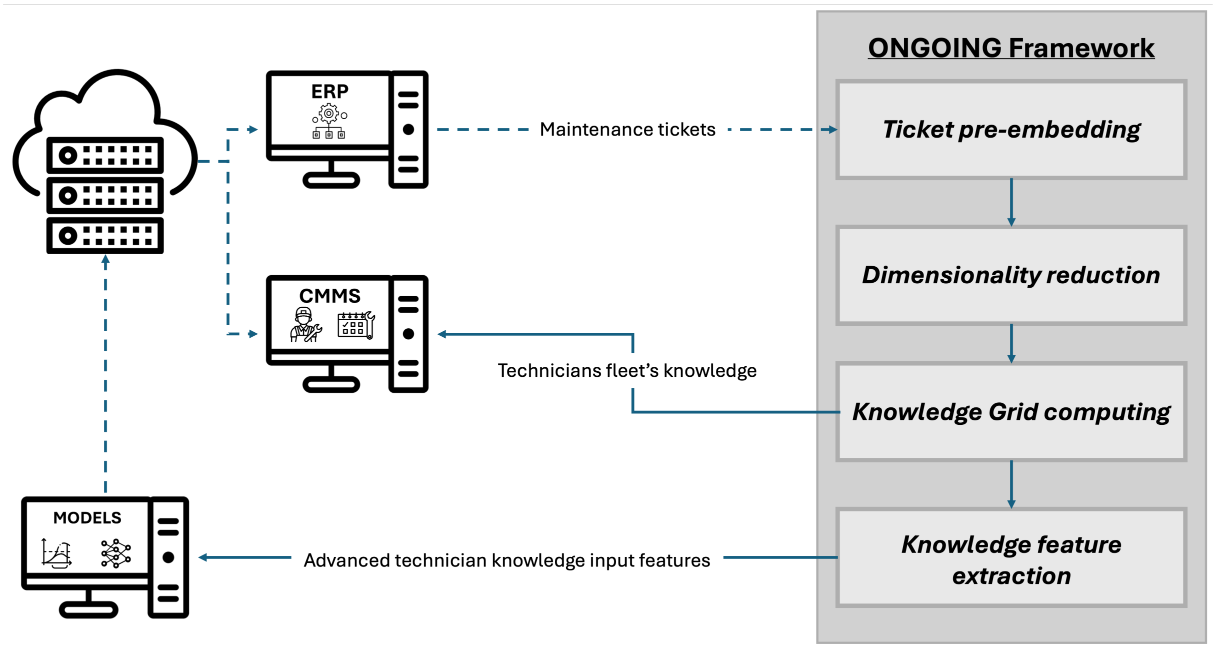

Figure 1. Workflow diagram of the proposed knowledge modeling framework

3.2. Knowledge modeling

3.2.1. Representing Knowledge: the Learning Forgetting Curve Model

The base model of knowledge in our model is borrowed from (Jaber et al., 2013), where knowledge is modeled as the time saved when performing a task, relative to the time it would take someone to perform this task for the first time, depicted in Figure 2.
In this way, the learning curve can be expressed by Eq. 1 :

$$T_x = T_1 x^{-b} \tag{1}$$

where T_x is the time to produce the xth unit, T_1 is the time to produce the first unit, x is the cumulative production, and b is the learning exponent ($0 \leq b < 1$, where $b = -\frac{\log(LR)}{\log(2)}$ and LR is the learning rate measured in percentage.

The forgetting curve can be considered a mirror image of the learning curve, as expressed by Eq. 2

$$\hat{T}_x = \hat{T}_1 x^f \tag{2}$$

where $\hat{T}_x$ is the time for the xth unit of lost experience of the forgetting curve, $\hat{T}_1$ is the intercept of the forgetting curve, x is the amount of output that would have been accumulated if interruption did not occur, and f is the forgetting exponent.

From this definition, we can define K as :

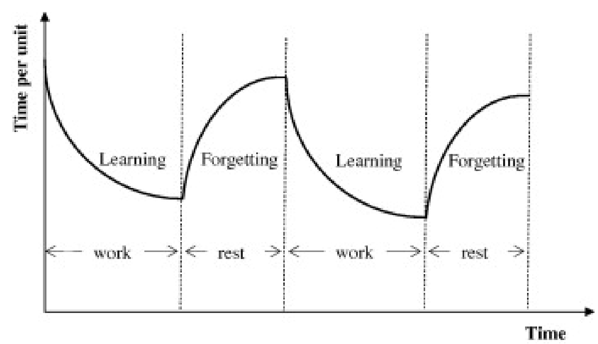

Figure 2. Behavior of the learning-forgetting process over time

$$K_i = T_{x_i} - T_1 \tag{3}$$

where K_i is the knowledge of the i-th technician W_i, and T_{x_i} is the time the technician W_i takes to treat the task. We can assume a fixed placeholder $T_1 = 1$ across all technicians and tasks to standardize knowledge scaling across the Knowledge Grid. Our knowledge then becomes intrinsically linked to ticket recurrence.

In practice, it is recommended to normalize knowledge even further across the technician fleet. To avoid unhealthy performance comparison between technicians, that could very well be stemming from bias in the database construction rather than a true skill gap, we will be normalizing each technician's knowledge across the grid between 0 and 1, with 1 represent-

ing the place where they have the most knowledge, and 0 the place where they have the least knowledge.
By scaling the knowledge like so, we don't capture their absolute knowledge levels but rather their distribution of knowledge, which is a much more ethical metric. In particular, aggregated Knowledge Grids, such as sector-level Knowledge Grids, will use pre-normalized technician Knowledge Grids, to prevent any form of absolute comparison.

3.2.2. Knowledge Grids

A knowledge grid is a n-dimensional representation of the knowledge of a technician. It is modeled as a n-d array, where each dimension is a dimension of $z(u)$ defined in Eq. 4:

$$z(u) = red(emb(u)) \tag{4}$$

where red is the abstract function representing the dimensionality reduction component, emb is the abstract function representing the pre-embedding component, and u is the original maintenance ticket.

We can then define G_i the Knowledge Grid representing technician W_i, to finally get $K_i(u) = G_i(z(u))$ the knowledge of technician W_i for the ticket u. A heatmap representing a 2-d Knowledge Grid is given in Figure 3. Although human-readability is not immediate on bare grids such as this one, they can easily be enriched through processes such as clustering and 3-d visualization. Examples are provided in Appendix under Figure 7, Figure 8, and Figure 9. These figures have been placed in Appendix to avoid violating guidelines regarding 3-d visualizations, and are as such considered optional.

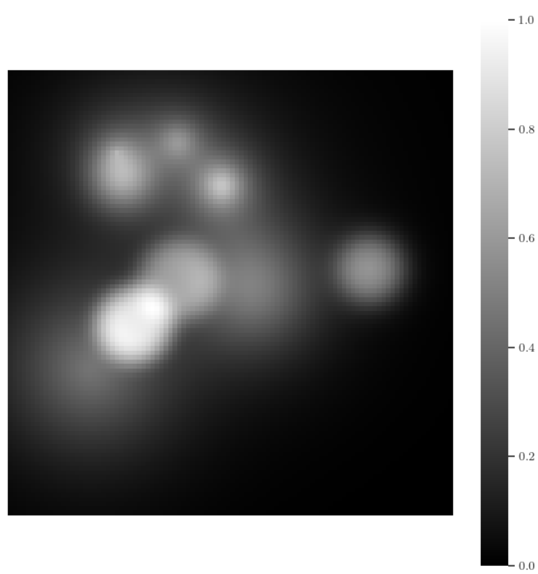

Figure 3. Visualization of a 2-d Knowledge Grid through a heatmap.

3.3. Additional Knowledge mechanisms

1. **Knowledge transfer:** considering maintenance experts' knowledge, we assume that a technician T_i may transfer part of his knowledge to another technician T_j through supervision, a mechanism mentioned by (María et al., n.d.) as a cornerstone of knowledge in organizations. We represent it as a weighted average between the two local knowledge, the one assigned and the one supervising. This average is weighted by a parameter τ representing how much of the information is transferred. This parameter is a global parameter.

$$K_b = (K_b^p + \Delta K_b)(1-\tau) + K_a^p \times \tau \tag{5}$$

where K^p is the knowledge prior to the increase.

2. **Knowledge propagation:** through the analysis of psychological works (Singley & Anderson, 1989), (Thorndike & Woodworth, 1901) and the analysis of maintenance experts, we assume that knowledge about a certain maintenance operation can be transferred to neighboring operations. This could be translated in several ways : knowledge about a machine, about the brand, about the type of failure. To represent this behavior, each increase in a technician's knowledge grid will be propagated in a neighborhood using a Gaussian kernel K with the following

$$\text{Let } G^{\Delta}(y) = \begin{cases} \Delta K & \text{if } y = z(u) \text{ , the ticket treated} \\ 0 & \text{else} \end{cases} \tag{6}$$

and

$$\text{Let } s = \frac{\Delta K}{\sum convolve(G^{\Delta}, K)} \tag{7}$$

in

$$G^{new} = G^{old} + convolve(G^{\Delta}, K) \times s \tag{8}$$

3.4. Knowledge features

Auxiliary notation. Let $\mathcal{G}$ denote the set of grid cells. Each cell $g \in \mathcal{G}$ has a fixed representative location (e.g., center) $r_g \in R^n$ in the reduced space. Let $\gamma : R^n \to \mathcal{G}$ be the cell-assignment rule that maps a point z to a cell $\gamma(z)$. For technician W_i, define the occupancy

$$N_i[g] = \sum_{u \in \mathcal{U}_i} \mathbf{1}\{\gamma(z(u)) = g\}, \quad T_i = \sum_{g \in \mathcal{G}} N_i[g]$$

and finally $S_i = \{g \in \mathcal{G} : N_i[g] > 0\}$ as well as the normalized density $p_i[g] = N_i[g]/T_i$. Consistently, the knowledge returned by the grid for a ticket u can be taken as

$$K_i(u) = N_i[\gamma(z(u))] \quad \text{or} \quad K_i^{\text{norm}}(u) = p_i[\gamma(z(u))].$$

Hypervolume (coverage span). The axis-aligned hypervolume covered by the occupied cells in the reduced space is

$$\mathrm{HV}_i \;=\; \prod_{k=1}^{n} \Big(\max_{g \in S_i} r_g^{(k)} \;-\; \min_{g \in S_i} r_g^{(k)} \Big), \tag{9}$$

optionally normalized by the grid's total span $\widehat{\mathrm{HV}}_i = \mathrm{HV}_i \Big/ \prod_{k=1}^{n} \big(\max_{g \in \mathcal{G}} r_g^{(k)} - \min_{g \in \mathcal{G}} r_g^{(k)} \big)$. It measures how broadly the technician's knowledge extends across the reduced space.

Interquartile range (IQR of cell densities). Let $\mathcal{V}_i = \{ p_i[g] \;:\; g \in S_i \}$ be the set of nonzero cell densities. If $Q_1(\mathcal{V}_i)$ and $Q_3(\mathcal{V}_i)$ are the 25th and 75th percentiles, then

$$\mathrm{IQR}_i \;=\; Q_3(\mathcal{V}_i) \;-\; Q_1(\mathcal{V}_i). \tag{10}$$

Higher values indicate greater concentration (heterogeneity) among occupied cells.

Maximum knowledge (peak specialization). The peak knowledge level is

$$\mathrm{MaxK}_i \;=\; \max_{g \in \mathcal{G}} N_i[g], \qquad \widehat{\mathrm{MaxK}}_i \;=\; \max_{g \in \mathcal{G}} p_i[g], \tag{11}$$

capturing the most frequent region of solved tickets.

Sparsity (coverage complement). A simple sparsity measure (bounded in $[0, 1]$) is the complement of occupancy coverage:

$$\mathrm{Sparsity}_i \;=\; 1 \;-\; \frac{|S_i|}{|\mathcal{G}|}. \tag{12}$$

An "effective" sparsity that accounts for uneven densities is the complement of the participation ratio:

$$\mathrm{Sparsity}_i^{\mathrm{eff}} \;=\; 1 \;-\; \frac{\big(\sum_{g \in \mathcal{G}} p_i[g]\big)^2}{\sum_{g \in \mathcal{G}} p_i[g]^2} \cdot \frac{1}{|\mathcal{G}|}. \tag{13}$$

Novelty of a ticket. Given a new ticket $u^\star$ with embedding $z^\star = z(u^\star)$, define a chosen distance $d : R^n \times R^n \to R_{\geq 0}$ (e.g., Euclidean) on the reduced space. Novelty relative to technician W_i is the distance to the nearest *occupied* cell:

$$\mathrm{Novelty}_i(u^\star) \;=\; \min_{g \in S_i} \; d\big(z^\star, r_g\big). \tag{14}$$

A scale-free variant divides by a technician-specific reference distance (e.g., median historical assignment distance), to stabilize thresholds across technicians.

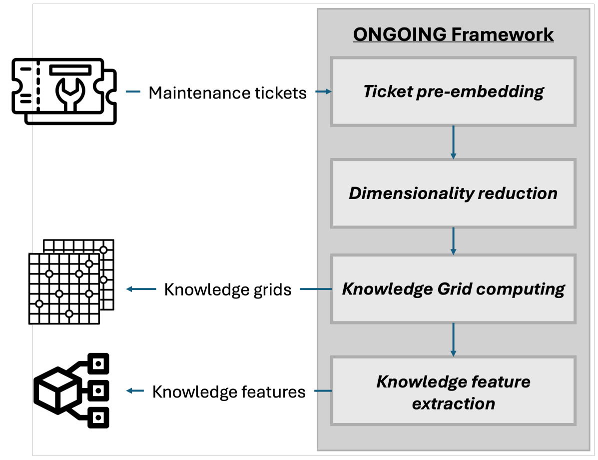

Figure 4. Summary diagram of implementation building blocks

4. EXPERIMENTS

Our framework aims to model knowledge, a metric for which there is no recorded ground-truth, nor standardized tests that could serve as a benchmark. One could argue that skill tests exist in social science literature, however such tests only address one specific category of skill at a time, whereas our approach aims to model knowledge in a continuous, category-free manner. As such, our chosen method of benchmarking has been a real-life, at scale implementation and comparison, in the production plant of one of our partner companies, a first-tier cutting tools manufacturer. The results will be defined as the satisfaction rates given by maintenance managers when comparing maintenance workers' knowledge modeled by both our baselines and the ONGOING framework, through a certain use-case scenario.

The following section details how we instantiated each part of our framework, defined in Figure 4, as well as the data we used, and a description of the environment we deployed our solution in. To preserve the anonymity of both the individuals and the company data, ranges will be preferred over exact figures during the description of the environment, and the data sample will be anonymised.

4.1. Environmental Setup

The environmental setup chosen to deploy our framework is a production plant of one of our manufacturing partners. More precisely, we aim to model the knowledge of a fleet of maintenance technicians spanning several sectors of production. An approximate scale for every core parameter of the environment is given in Table 3

Table 3. Approximate scale of the production line setup parameters

Parameter	Approximate scale
Num. Technicians	100
Num. Sectors	20
Num. Type Machines	150
Num. Machine Manufacturers	40
Num. Tickets	50000
Tickets Time Span (years)	5

Table 4. Schematic and altered sample of proprietary data

Field	Example Value
Short Desc.	"robot stuck"
Machine ID	"HAUSER S55-400"
Long Desc.	"During second phase, robot getting stuck into abutment : see error message 41."

4.2. Baselines

Among the methods identified in Table 2, the two traditional approaches of representing knowledge as a **scalar** and as a **vector** were chosen as baselines. Both RNN and matrix representations will be left out of the comparison as they don't appear to be suited to the current use-case and thus can not be used as baselines. In particular, RNN encodings don't explicitly provide a readily accessible knowledge level, and any matrix representation could have been assimilated to a vector-based representation in our case.
For fairness purposes, both the **scalar** and **vector**-based approaches will be considered dynamic, with the same Learning and Forgetting mechanisms used by the ONGOING framework, as defined in Subsection 3.2.

4.3. Data

As stated previously, company data is not to be released, as the databases of maintenance tickets logs are deemed to contain sensitive data about both employees of the company, and proprietary manufacturing processes.
The format of the data however, as well as cherry-picked anonymised and translated examples is provided to serve as a reference point for potential future users of the framework.

Although other features of the maintenance ticket logs may be used in heuristics in other parts of the full workflow, the current schematic of sanitized ticket data used is shown in Table 4

4.4. Ongoing Implementation

4.4.1. Pre-embedding

We use a multilingual sentence transformer, *distiluse-base-multilingual-cased-v2* from (Reimers & Gurevych, 2019), as the pre-embedding layer because our tickets are free-text and span multiple languages (primarily French, with some English and German). This model produces language-agnostic 512-dimensional sentence embeddings that align semantically similar tickets across languages, while staying compact enough for efficient storage, fast streaming updates, and clean dimensionality reduction into our knowledge grids. Given the unstructured nature of our tickets, as well as human-related constraints such as spelling mistakes, no unified vocabulary, nor standardized text structure for either problem or solution formulation, an NLP encoder is the right first layer. In a case where tickets are highly categorized with consistent taxonomies, a simpler statistical approach (e.g., Multiple Correspondence Analysis, MCA) on categorical features could have been a reasonable first baseline.

4.4.2. Dimensionality reduction

We use a Self-Organizing Map (SOM) for dimensionality reduction for its ability to preserve topological neighborhoods: semantically similar tickets remain adjacent after projection, which is exactly what our neighbor-based knowledge propagation needs. A SOM is an unsupervised neural projection in which each input vector is matched to a neuron of the map: the best-matching unit (BMU). This BMU's weight vector is then updated with the input vector, finally resulting in a 2-d map whose geometry reflects the structure of the original space. (Kohonen, 2001) Focusing on topology lets us aggressively compress high-dimensional embeddings into a 2-d map while discarding fine-grained geometry associated to semantics we do not need for the knowledge grid, and the SOM's vector-quantization behavior naturally produces clusterable, well-structured maps that help downstream analysis (Vesanto & Alhoniemi, 2000).
To mitigate the classic border effect (edge units having fewer neighbors), we train a toroidal SOM, the grid wraps around so opposite edges are adjacent, which is a standard option in established SOM tooling and explicitly reduces boundary artifacts (Mount & Weaver, 2011). Finally, to avoid a few units monopolizing assignments, we add a frequency-sensitive penalty so over-winning neurons are temporarily discouraged, promoting a more balanced distribution across the map.

4.4.3. Knowledge Grid computing

The choice of the SOM results in a 2-d space for the reduced ticket logs, which will be matched by the Knowledge Grids structure, resulting in human-readable grids. Due to the number of ticket logs at our disposal (50000), each grid has been empirically determined, through trial-and-error to have a shape of 100 by 100, allowing for enough room for clusters of data to appear, but not so much that it would result in a mostly empty grid. A suggested approach to finding a correct grid shape in relation to your data, is that 100 by 100 seems to be a soft ceiling of interpretability vs performance for high amounts of ticket logs (more than 10000), whereas 10 by 10 is the minimum that holds significance and interpretability for low amounts of data (less than 1000). For amounts of ticket logs in the 1000 to 10000 range, exploring options such as 30 by 30, or 50 by 50 is a strong compromise.

A working Python package has been made publicly available at https://github.com/AdrienBolling/ONGOING for reproducibility and testing purposes. The core logic is written entirely in JAX, and leverages its Just-In-Time (JIT) compiling capabilities.

4.4.4. Knowledge feature extraction

As stated previously, a working Python package is available for testing. This package ships with several Knowledge Features as defined in Section 3.4 :

- Hypervolume
- Inter-Quartile Range
- Maximum Knowledge
- Sparsity
- Ticket Novelty

In addition to these features, the specific structure of the data, namely the production sector-specific nature of technicians, allowed us to build sector-level Knowledge Grids by aggregating the corresponding technicians' Knowledge Grids.
Ticket clustering was then performed on these higher-level grids, to identify clusters of tickets, which created the possibility of analysing the distribution of knowledge among the fleet of technicians.

4.5. Validation through a use-case: Technician recommendation

To validate our Knowledge Modeling Method against the baselines, we design a simple but solution-oriented use-case with the company. Given a fleet of technicians, and an incoming maintenance ticket: suggest the technician most suited for the task. A core difficulty lies here, the most suited technician may not always be the most knowledgeable.

- **Scalar-based knowledge baseline :**
  The scalar-based baseline works as follows: assuming a fully instantiated and trained fleet of technicians, each of them having their knowledge modeled by a scalar, we extract three individuals: the best technician overall, the worst technician overall, the median technician overall.
  As it is not possible to extract more fine-grain information about the type of knowledge a technician possesses, our best approach is to use common statistical features to answer with these three cases that represent usual managers' approaches to technician allocation: "Do I want to treat the ticket as fast as possible, moderately fast to save my best technicians for more urgent matters, or do I want to train my least performing technicians".
- **Vector-based knowledge baseline :**
  The vector-based baseline implementation works as follows: we categorize each ticket according to the type of machines it is related to (in the order of 150 categories as stated in Table 3). Then, assuming a fully instantiated and trained fleet of technicians, we restrict each technician's knowledge to the domain of the ticket (here the type of machine of the maintenance ticket). The rest of the setup is strictly analogous to the scalar-based baseline, applied to this restricted knowledge.
- **ONGOING Framework :**
  Our solution using the ONGOING Framework (summarized in Figure 5) works as follows: assuming a fully instantiated and trained implementation, we project the given maintenance ticket into the knowledge grid via the ONGOING NLP encoder and SOM. From the resulting region, we extract each technician's local feature vector (e.g., metrics already maintained by ONGOING). We then cast technician selection as a multi-objective comparison and compute a Pareto front where technicians are the candidate solutions and the extracted features are the objectives. If a technician strictly Pareto-dominates all others, we recommend that single, fully dominating choice. Otherwise, we return the non-dominated set, a shortlist of partially dominating candidates, so supervisors can decide with full transparency.

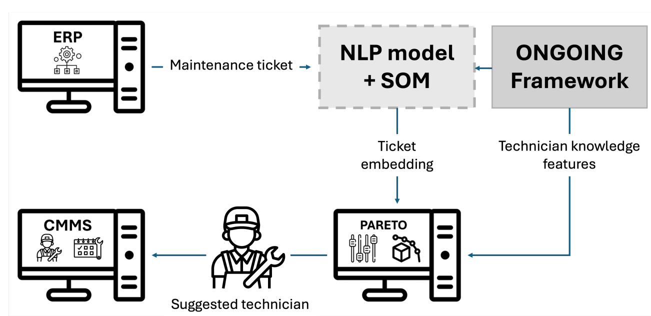

Figure 5. Summary diagram of ONGOING solution to the real-life validation use-case

4.6. Results

We gathered evaluations of each baselines' suggestions through a total of 55 maintenance tickets, separate from the training set, extracted from the ERP's stream, these tickets will not be provided for the same privacy reasons as the testing set.

Table 5. Comprehensive results of the use-case expert evaluation

Method	Pertinent	Non-Pertinent
Scalar-based	7	48
Vector-based	17	38
ONGOING	*41*	14

Each suggestion was submitted to binary evaluation "Pertinent" or "Non-Pertinent" by Maintenance experts usually in charge of ticket dispatching. The results are presented in Figure 6 as a bar plot, and precisely reported in Table 5.

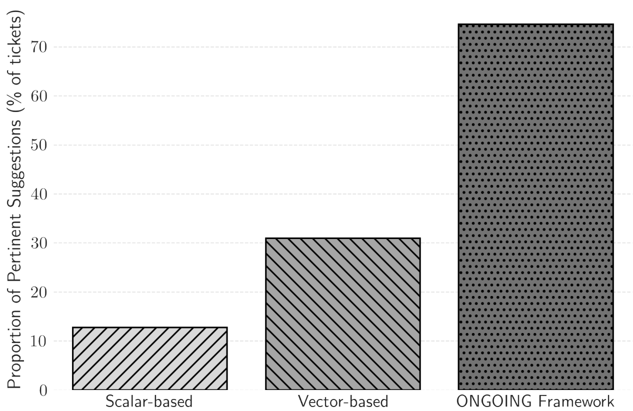

Figure 6. Comparative bar plot of the use-case experiment results.

Results show a more positive response to ONGOING's modeling of knowledge through the use-case of technician recommendation. The number and complexity of features that can be extracted allow for more transparency and a more informed choice, along with more flexibility in the objectives of the selection.

Even though the validation phase seems underwhelming at first glance, it is to be noted that this method was actually benchmarked on a very qualitative output : human knowledge representation, for which no know ground truths or standardized benchmarks exist. The only way for us to validate the ability of our framework to model technician knowledge in a satisfying way is to get human feedback from maintenance experts about a real-life testing deployment, a usually hardly feasible operation.

5. LIMITATIONS AND ETHICAL CONCERNS

Even though this framework could look like a drop-in solution for some, several technical limitations and ethical risks may arise in case of a careless real-life deployment. Below is a list of currently identified pitfalls, with concrete mitigation strategies for deployment.

5.1. Technical limitations

- **Storage capabilities:** The nature of the Knowledge Grids makes it so that the need storage space scales exponentially with the number of dimensions the final embeddings will get. Even 5-d 100 by 100 grids of 32-bit floats are considerably expensive in storage. If need be, to avoid this bottleneck, we recommend re-implementing the Knowledge Grids with either a sparse-matrix architecture, or a fragmented architecture that can be progressively loaded in RAM.
- **Data quality and coverage:** Ticket text is known to be messy, short, multilingual, filled with specific jargon and abbreviations, inconsistent, and riddled with mistakes. Certain tasks may also be severely under-reported (e.g., quick routine fixes, where a technician would spend more time creating the maintenance report than treating the issue), thus skewing technician grids and knowledge distribution.
  To mitigate this issue several strategies are available: enforce minimal ticket structure through properly defined ERP fields, normalize language and domain vocabulary, filter outliers, and periodically rebalance under-logged areas.
- **Representation error from projection:** A 2-d toroidal SOM preserves neighborhood structure only approximately. Dense regions can collapse, sparse regions overstretch, and cluster boundaries can be misleading on the map edges despite the torus. To mitigate these issues, report trustworthiness/continuity metrics, keep "map size vs. data volume" within validated bounds, and perform sanity-check with alternative projections (UMAP/PCA) during validation.
- **Cold-start and sparse technician histories:** New technicians (or new domains) start with empty or highly uncertain grids, which can unfairly depress recommendations. To mitigate these risks, initialize with prior knowledge about a technician's qualifications, and put in place some heavy mentorship updates through documented cowork.
- **Attribution, not availability nor optimization:** The grid reflects what was logged, not availability, location, or safety constraints. Using it alone for assignment will fail in real operations. To mitigate this pitfall, combine this framework with standard constraints (shift, location, certifications), and treat the grid as a capability signal, not an optimizer.

5.2. Ethical risks

- **Historical bias and feedback loops:** If past assignments favored certain technicians or teams, the grid can entrench that pattern: "the rich get richer," while others lose exposure and growth. To prevent that, lean towards training-oriented usage, introduce rotation policies, exploration bonuses for under-exposed technicians, and fairness constraints. Monitor exposure and opportunity metrics across seniority, shift, site, and contract type. Do not use grids as de-facto performance scores.

- **Worker impact and misuse:** A human-readable grid can be misread as a ranking or used punitively in HR decisions. That's outside this system's scope. This must absolutely be prevented: introduce a formal policy: to not use this Knowledge Modeling for disciplinary, pay, or hiring decisions. Use for training design, staffing balance, and decision support only. Provide appeal channels when a technician disagrees with their grid. Keep the intended safeguard in place of not modeling the absolute knowledge of a technician but rather his knowledge distribution.
- **Consent and transparency:** Technicians should understand what the grid is, what it isn't, and how it affects day-to-day work. To ensure transparency and informed consent, introduce mandatory on-boarding, explainer materials, opt-in pilots when feasible, and in-tool "why this recommendation" explanations. Employees are more likely to put their trust in a tool they understand fully.

6. CONCLUSION

We introduced the Knowledge Grid, a representation layer that turns noisy, multilingual ticket narratives into an interpretable continuous and category-free Knowledge Grid for each technician and team, and into complex and diverse features that downstream policies can consume. The grid is method-agnostic (any sentence embedder; any projector), but the instantiated pipeline: multilingual embeddings, toroidal SOM, conserved neighborhood updates, proved sufficient to support real-life at-scale usage. Two mechanisms capture how knowledge actually moves on the shop floor: transfer between people and propagation across semantically adjacent tasks. Together, they reduce cold-start pain, highlight latent specialization, and make implicit expertise quantifiable.

Through a concrete use-case of technician assignment extended from our instantiated solution, we showed that grid-based technician suggestions are overwhelmingly preferred by dispatchers over common baselines, such as vector-based representations. Crucially, the representation remains transparent: decision rationales can be traced to visible neighborhoods and to a small but extendable set of features (hypervolume, sparsity, overlap, recency density), enabling planners to contest or support recommendations with evidence. This is the key contribution for PHM: not a black-box optimizer, but a durable, human-readable knowledge representation that existing workflows can plug into.

This framework however suffers from certain limitations. First, our evidence, although promising, is thinly reported and should be extended through larger-scale testing phases at the plant-level through additional use-cases. Second, the current implementation, although valuable due to a successful at-scale deployment, suffers from the privacy it implies, a next step will be to instantiate this framework on publicly-available maintenance ticket logs datasets. Future work will include establishing full ablation studies to report on the relative usefulness of each part of our framework, alternative projectors will be explored and additional features and constraints will be formulated as lightweight policies on top of the implementation, to ensure an ethical solution.

Notably the grid-based approach enables priority-aware dispatch shortlists that remain human-auditable, training plans by exposing coverage gaps, and fairness monitoring via exposure trajectories. These are concrete ways to reduce response time, spread expertise, and avoid single-point failure through over-reliance on a few experts.
In summary, the Knowledge Grid makes technician knowledge measurable, transferable, and explainable. It is a pragmatic step toward PHM decision support that respects human expertise while unlocking model-readiness from the constant stream of maintenance text.

REFERENCES

Alvarez, A. L., Mohammed, W. M., & Lastra, J. L. M. (2025, December). Human professional skills assessment based on a modified learning curve model. *Production & Manufacturing Research*, *13*(1), 2525935. doi: 10.1080/21693277.2025.2525935

Bolling, A., & Kubler, S. (2024). Knowledge-aware and Learning-focused Multi-Objective Multi-agent Reinforcement Learning for Maintenance Technician Assignment.

Bouajaja, S., & Dridi, N. (2017, July). A survey on human resource allocation problem and its applications. *Operational Research*, *17*(2), 339–369. doi: 10.1007/s12351-016-0247-8

Chen, X., Li, K., Lin, S., & Ding, X. (2024). Technician routing and scheduling with employees' learning through implicit cross-training strategy. *International Journal of Production Economics*, *271*(C). (Publisher: Elsevier)

Chen, Z., De Causmaecker, P., & Dou, Y. (2023, March). A combined mixed integer programming and deep neural network-assisted heuristics algorithm for the nurse rostering problem. *Applied Soft Computing*, *136*, 109919. doi: 10.1016/j.asoc.2022.109919

Denu, M., David, P., Mangione, F., & Landry, A. (2024, January). Towards Human-Centric Digital Simulation: Guidelines to Simulate Operators Skills Acquisition and Health in Circular Manufacturing Systems. *IFAC-PapersOnLine*, *58*(19), 445–450. doi: 10.1016/j.ifacol.2024.09.252

Felberbauer, T., Gutjahr, W., & Doerner, K. (2019, June). Stochastic project management: Multiple projects with multi-skilled human resources. *Journal of Scheduling*, *22*(3), 271–288. (arXiv:1812.00664 [math]) doi: 10.1007/s10951-018-0592-y

Gonçalves, L., Alvelos, H., & Moura, A. (2025). Human Resource Allocation Problem Considering Aptitude, Work Experience and Level of Criticality of Workstations. In R. Marczewska-Kuzma, L. Hadas, & P. Golinska-Dawson (Eds.), *Implementation of Circular Economy in Supply Chains and Production Systems* (pp. 161–177). Cham: Springer Nature Switzerland. doi: 10.1007/978-3-031-88926-4$_1$2

Hadiyanto, H., & Anggoro, Y. (2025). A Skill Matrix Chart Case Study in Global Manufacturing. *Systems Research and Behavioral Science*, *n/a*(n/a). (_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/sres.3128) doi: 10.1002/sres.3128

Han, K., & Gong, W. (2025). Memetic algorithm based on non-dominated levels for flexible job shop scheduling problem with learn-forgetting effect and worker cooperation. *Comput. Ind. Eng.*, *200*(C). doi: 10.1016/j.cie.2024.110845

Hashemifar, S., & Sahebi, S. (2025, May). *Personalized Student Knowledge Modeling for Future Learning Resource Prediction.* arXiv. (arXiv:2505.14072 [cs]) doi: 10.48550/arXiv.2505.14072

Henao, C. A., Mercado, Y. A., & González, V. I. (2023, November). Multiskilled personnel assignment with k-chaining considering the learning-forgetting phenomena. *International Journal of Production Economics*, *265*, 109018. doi: 10.1016/j.ijpe.2023.109018

Heuser, P., & Tauer, B. (2023). Single-machine scheduling with product category-based learning and forgetting effects. *Omega*, *115*(C). (Publisher: Elsevier)

Imran Hasan Tusar, M., & Sarker, B. R. (2024, October). Technician assignment in multi-shift maintenance schedules in an offshore wind farm. *Renewable Energy Focus*, *51*, 100616. doi: 10.1016/j.ref.2024.100616

Iso, H., Pezeshkpour, P., Bhutani, N., & Hruschka, E. (2025, April). Evaluating Bias in LLMs for Job-Resume Matching: Gender, Race, and Education. In W. Chen, Y. Yang, M. Kachuee, & X.-Y. Fu (Eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: Industry Track)* (pp. 672–683). Albuquerque, New Mexico: Association for Computational Linguistics. doi: 10.18653/v1/2025.naacl-industry.55

Jaber, M. Y., Givi, Z. S., & Neumann, W. P. (2013, July). Incorporating human fatigue and recovery into the learning–forgetting process. *Applied Mathematical Modelling*, *37*(12), 7287–7299. doi: 10.1016/j.apm.2013.02.028

Khanna, P., Kumari, J., & Karim, R. (2024, June). Human-Centric PHM in the Era of Industry 5.0. *PHM Society European Conference*, *8*(1), 7–7. doi: 10.36001/phme.2024.v8i1.4121

Kohonen, T. (2001). *Self-Organizing Maps* (Vol. 30; T. S. Huang, T. Kohonen, & M. R. Schroeder, Eds.). Berlin, Heidelberg: Springer. doi: 10.1007/978-3-642-56927-2

Lee, D., & Ahn, C. (2020, June). Industrial human resource management optimization based on skills and characteristics. *Computers & Industrial Engineering*, *144*, 106463. doi: 10.1016/j.cie.2020.106463

Liang, Y., Peng, T., Pu, Y., & Wu, W. (2022, March). HELP-DKT: an interpretable cognitive model of how students learn programming based on deep knowledge tracing. *Scientific Reports*, *12*(1), 4012. (Publisher: Nature Publishing Group) doi: 10.1038/s41598-022-07956-0

Liu, Q., Huang, Z., Yin, Y., Chen, E., Xiong, H., Su, Y., & Hu, G. (2019, June). *EKT: Exercise-aware Knowledge Tracing for Student Performance Prediction.* arXiv. (arXiv:1906.05658 [cs]) doi: 10.48550/arXiv.1906.05658

Long-fei, C., Nakamura, Y., & Kondo, K. (2020, September).

Modeling User Behaviors in Machine Operation Tasks for Adaptive Guidance. arXiv. (arXiv:2003.03025 [cs]) doi: 10.48550/arXiv.2003.03025

Lu, J., Ye, T., Chen, W., & Hentenryck, P. V. (2025, October). Boosting Column Generation with Graph Neural Networks for Joint Rider Trip Planning and Crew Shift Scheduling. *Transportation Research Part E: Logistics and Transportation Review*, *202*, 104281. (arXiv:2401.03692 [math]) doi: 10.1016/j.tre.2025.104281

Lukens, S., McCabe, L. H., Gen, J., & Ali, A. (2024, November). Large Language Model Agents as Prognostics and Health Management Copilots. *Annual Conference of the PHM Society*, *16*(1). doi: 10.36001/phmconf.2024.v16i1.3906

Lv, B., Jiang, J., Wu, L., & Zhao, H. (2024, December). Team formation in large organizations: A deep reinforcement learning approach. *Decision Support Systems*, *187*, 114343. doi: 10.1016/j.dss.2024.114343

María, A., Palma, L., Cecilia, L., Bárcena, S., Delgado, R., Del, R., … Campo, M. (n.d.). The ideas generation process and the role of the learning curve: simulating the wealth of knowledge in organizations.

Mount, N. J., & Weaver, D. (2011). Self-organizing maps and boundary effects: quantifying the benefits of torus wrapping for mapping SOM trajectories. *Pattern Anal. Appl.*, *14*(2), 139–148. doi: 10.1007/s10044-011-0210-5

Muklason, A., Kusuma, S. D. R., Riksakomara, E., Premananda, I. G. A., Anggraeni, W., Mahananto, F., & Tyasnurita, R. (2024, January). Solving Nurse Rostering Optimization Problem using Reinforcement Learning - Simulated Annealing with Reheating Hyperheuristics Algorithm. *Procedia Computer Science*, *234*, 486–493. doi: 10.1016/j.procs.2024.03.031

Muraretu, I., Ilie, S., & Ilie, M. (2017, 01). Initial results on the effectiveness of a skill-based approach to human resource allocation. *Annals of the University of Craiova*, *14*, 19.

Nguyen, D. H., Truong, T. Q. A., & Tran-Thanh, L. (2025). Faster, Larger, Stronger: Optimally Solving Employee Scheduling Problems with Graph Neural Networks. In W. Buntine, M. Fjeld, T. Tran, M.-T. Tran, B. Huynh Thi Thanh, & T. Miyoshi (Eds.), *Information and Communication Technology* (pp. 141–151). Singapore: Springer Nature. doi: 10.1007/978-981-96-4285-$4_1$2

Platten, B., Macfarlane, M., Graus, D., & Mesbah, S. (n.d.). Automated Personnel Scheduling with Reinforcement Learning and Graph Neural Networks.

Ranasinghe, T., Senanayake, C. D., & Grosse, E. H. (2024). Effects of stochastic and heterogeneous worker learning on the performance of a two-workstation production system. *International Journal of Production Economics*, *267*(C). (Publisher: Elsevier)

Reimers, N., & Gurevych, I. (2019, August). *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks.* arXiv. (arXiv:1908.10084 [cs]) doi: 10.48550/arXiv.1908.10084

Ruiz-Rodríguez, M. L., Kubler, S., Robert, J., & Le Traon, Y. (2024, August). Dynamic maintenance scheduling approach under uncertainty: Comparison between reinforcement learning, genetic algorithm simheuristic, dispatching rules. *Expert Systems with Applications*, *248*, 123404. doi: 10.1016/j.eswa.2024.123404

Saber, R. G., Leyman, P., & Aghezzaf, E.-H. (2024, January). Modeling the Integrated Flexible Job-Shop and Operator Scheduling in Flexible Manufacturing Systems. *IFAC-PapersOnLine*, *58*(19), 343–348. doi: 10.1016/j.ifacol.2024.09.235

Safaei, N., & Kiassat, C. (2018, March). *A Swift Heuristic Method for Work Order Scheduling under the Skilled-Workforce Constraint.* arXiv. (arXiv:1803.01252 [cs]) doi: 10.48550/arXiv.1803.01252

Singley, M. K., & Anderson, J. R. (1989). *The transfer of cognitive skill* (No. 9). Cambridge, Mass: Harvard University press.

Stein, J., Hildebrandt, F. D., & Thomas, B. W. (2024, September). *Learning State-Dependent Policy Parametrizations for Dynamic Technician Routing with Rework.* arXiv. (arXiv:2409.01815 [cs]) doi: 10.48550/arXiv.2409.01815

Szwarc, E., & Golińska-Dawson, P. (2024, January). Robust Scheduling of Multi-Skilled Workforce Allocation: Job Rotation Approach. *Electronics*, *13*(2), 392. (Number: 2 Publisher: Multidisciplinary Digital Publishing Institute) doi: 10.3390/electronics13020392

Thorndike, E. L., & Woodworth, R. S. (1901). *Classics in the History of Psychology – Thorndike & Woodworth (1901a).*

Tien, K.-W., & Prabhu, V. (2018). Modeling the Influence of Technician Proficiency and Maintenance Strategies on Production System Performance. In *IFIP Advances in Information and Communication Technology* (pp. 47–54). Cham: Springer International Publishing. (ISSN: 1868-4238, 1868-422X) doi: 10.1007/978-3-319-99707-0_7

Vesanto, J., & Alhoniemi, E. (2000, May). Clustering of the self-organizing map. *IEEE Transactions on Neural Networks*, *11*(3), 586–600. doi: 10.1109/72.846731

Wasi, A. T. (2024). HRGraph: Leveraging LLMs for HR Data Knowledge Graphs with Information Propagation-based Job Recommendation. In *Proceedings of the 1st Workshop on Knowledge Graphs and Large Language Models (KaLLM 2024)* (pp. 56–62). Bangkok, Thailand: Association for Computational Linguistics. doi: 10.18653/v1/2024.kallm-1.6

Xu, S., Xie, F., & Hall, N. G. (2025, September). Sequencing with learning, forgetting and task similarity. *Eu-*

ropean Journal of Operational Research*, *325*(3), 400–415. doi: 10.1016/j.ejor.2025.03.002

Yang, W., Li, S., Luo, G., Li, H., & Wen, X. (2025, April). A Real-Time Human–Machine–Logistics Collaborative Scheduling Method Considering Workers' Learning and Forgetting Effects. *Applied System Innovation*, *8*(2), 40. (Publisher: Multidisciplinary Digital Publishing Institute) doi: 10.3390/asi8020040

Zangari, A., Marcuzzo, M., Schiavinato, M., Gasparetto, A., & Albarelli, A. (2023, September). Ticket automation: An insight into current research with applications to multi-level classification scenarios. *Expert Systems with Applications*, *225*, 119984. doi: 10.1016/j.eswa.2023.119984

Zhang, Z., Yao, W., Li, F., Yu, J., Simic, V., & Yin, X. (2024, November). A graph neural network-based teammate recommendation model for knowledge-intensive crowdsourcing. *Engineering Applications of Artificial Intelligence*, *137*, 109151. doi: 10.1016/j.engappai.2024.109151

APPENDIX

Below are Figure 7, Figure 8, and Figure 9. Both Figure 8, and Figure 9 represent technician-level Knowledge Grids, whereas Figure 7 represents a sector-level Knowledge Grid. All Knowledge Grids have been anonymised, cluster summary labels have been removed due to sensitive information. However the clustering remains, showing possibilities of progressively enriching individual Knowledge Grids. In addition, we clearly notice different profiles of technicians at a glance between Figure 8, and Figure 9. Figure 8 models a very specialized technician, whereas Figure 9 models a jack-of-all trades profile.

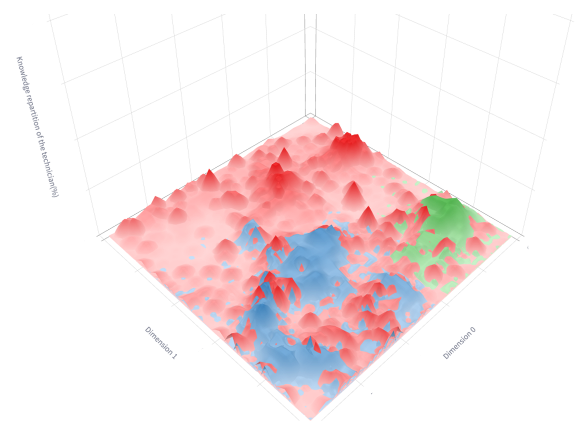

Figure 7. Anonymised visualization of a 2-d sector-level Knowledge Grid aggregate with 3-d representation.

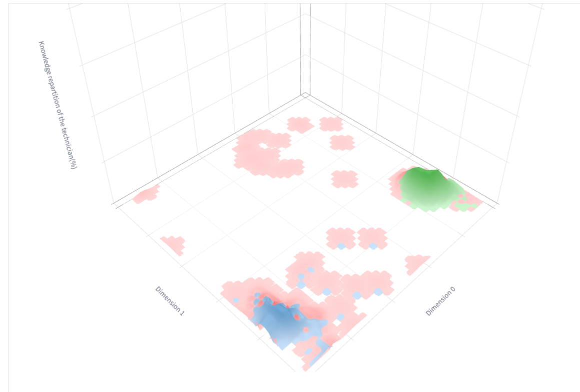

Figure 8. Anonymised visualization of a 2-d Knowledge Grid with 3-d representation belonging to a specialized archetype of technician.

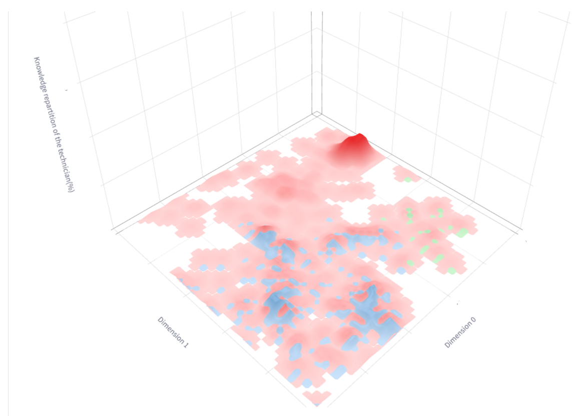

Figure 9. Anonymised visualization of a 2-d Knowledge Grid with 3-d representation belonging to a generalist archetype of technician.