Large Language Model Agents as Prognostics and Health Management Copilots

Sarah Lukens[1], Lucas H. McCabe[1,2], Joshua Gen[1], Asma Ali[3],

[1] *LMI, Tysons, VA, 22102, USA*
sarah.lukens@lmi.org, lmccabe@lmi.org, joshua.gen@lmi.org

[2] *George Washington University, Washington, DC, 20052, USA*

[3] *GE Vernova, Chicago, IL, 60661, USA*
asma.ali1@ge.com

Abstract

Amid concerns of an aging or diminishing industrial workforce, the recent advancement of large language models (LLMs) presents an opportunity to alleviate potential experience gaps. In this context, we present a practical Prognostics and Health Management (PHM) workflow and self-evaluation framework that leverages LLMs as specialized in-the-loop agents to enhance operational efficiency without subverting human subject matter expertise. Specifically, we automate maintenance recommendations triggered by PHM alerts for monitoring the health of physical assets, using LLM agents to execute structured components of the standard maintenance recommendation protocol, including data processing, failure mode discovery, and evaluation. To illustrate this framework, we provide a case study based on historical data derived from PHM model alerts. We discuss requirements for the design and evaluation of such "PHM Copilots" and formalize key considerations for integrating LLMs into industrial domain applications. Refined deployment of our proposed end-to-end integrated system may enable less experienced and professionals to back-fill existing personnel at reduced costs.

1. Introduction

Industrial demographics have changed over time in several domains, in part due to shifting occupational preferences, shrinking generational cohorts, and lengthened professional careers (Silverstein, 2008). Additionally, corporate financialization, technological change, and industrial outsourcing have left engineering organizations with numerous workforce challenges that are not easily resolved by adapting hiring practices alone (Muellerleile, 2009; Greenberg, 2010). As a result, a so-called "experience gap" has caused concern in operational fields (Rovaglio, Calder, & Richmond, 2012). In particular, monitoring and maintenance of complex engineering systems typically requires the deployment of specialized personnel with sophisticated domain expertise, and such staff are in short supply. Although systemic approaches, such as large-scale programs to increase vocational training access, can be impactful, such strategies can be difficult for individual organizations to implement effectively. Instead, we consider whether recent digital innovation - particularly that of large language models (LLMs) - can help relieve these workforce pressures by supplementing less experienced maintenance and reliability professionals.

LLMs are (typically autoregressive) statistical models of token sequences, learned from large textual corpora (Chengwei Wei and Yun-Cheng Wang and Bin Wang and C.-C. Jay Kuo, 2024). In production, these models are often fine-tuned for instruction-following (Ouyang et al., 2022), whereby user-provided prompts induce a discrete distribution over output sequences (Sordoni et al., 2024). These so-called "instruction-tuned" models can serve as impressive conversational agents, but questions remain regarding effective application in industrial settings, including medicine (Thirunavukarasu et al., 2023), design and manufacturing (Makatura et al., 2023), and power engineering (Majumder et al., 2024).

The so-called "copilot framework" - where artificial intelligence (AI)-powered systems augment, rather than replace, existing workflows - offers an opportunity to meaningfully increase productivity by integrating LLMs alongside human personnel (Cambon et al., 2023). In conventional Prognostics and Health Management (PHM) workflows, real-time monitoring of industrial assets typically occurs in a Monitoring and Diagnostics (M&D) center and is driven by the outputs of sensor-based PHM models. When an alert is triggered, an M&D analyst determines if escalation to the plant's reliability and maintenance organization is necessary, providing recommendations for initiating appropriate actions, such as identifying possible fault causes and suggesting steps for troubleshooting. The maintenance organization then investigates the fault, writes the appropriate work order, and schedules and executes the required work. Efficient execution therein is a complex task requiring personnel with domain-specific expertise. We are interested in exploring applications of the copilot framework to the PHM domain, with the goal of alleviating industrial experience gaps.

1.1. Our contributions

In this work, we consider a potential AI copilot system for the maintenance and reliability domain. Our main contributions are as follows:

- We outline a framework for expediting PHM workflows using integrated, specialized LLM agents (Section 3.1).
- We propose a domain and use case-specific copilot evaluation rubric and provide a practical case study leveraging publicly-available marketing content detailing real use cases of a commercial PHM solution (Section 3.2).
- We examine the behavior of our PHM copilot. Our findings include that retrieval-augmented generation (RAG) with historical cases references measurably improves system performance in the context of likelihood to make recommendations based on observed, frequently occurring events (Section 4).

The remainder of this work is organized as follows: Section 2 provides background on the PHM copilot, its building block concepts, and a literature review of related work. Section 3 outlines the methodology for our prototype PHM copilot. The case study is divided into two parts: data preparation using a multi-agent framework and the PHM copilot itself. The results of the case study are presented in Section 4. Conclusions and discussion are presented in Section 5.

2. RELATED WORK

2.1. LLM-related technologies supporting copilot development

In this section, we review some concepts developed in the field of Generative AI and LLMs which are currently common building blocks used in designing copilot architectures. Note that as development in this area is rapidly evolving, over time this list will be subject to additions, modifications and enhancements.

Technical Language Processing (TLP). TLP refers to engineering approaches for tailoring natural language processing (NLP) tools to technical language data (Brundage, Sexton, Hodkiewicz, Dima, & Lukens, 2021; Dima, Lukens, Hodkiewicz, Sexton, & Brundage, 2021). Developing methods which adapt LLMs for engineering use cases in practical ways that meet specific requirements is one aspect of TLP. One common task in TLP has been failure mode classification which labels unstructured maintenance data with structured fields. This task involves identifying structured fields from short maintenance work order description such as the item, fault state and action taken, when possible (Hodkiewicz & Ho, 2016), (Lukens, Naik, Saetia, & Hu, 2019). Since descriptions can have zero, one or multiple possible labels, entity recognition or tagging have been common approaches (Sexton, Brundage, Hoffman, & Morris, 2017), (Bikaun & Hodkiewicz, 2021), (Sexton, Hodkiewicz, & Brundage, 2019). Recently, exploration for how to effectively utilize LLMs for failure mode classification has been conducted (Stewart, Hodkiewicz, Liu, & French, 2022), (Stewart, Hodkiewicz, & Li, 2023).

Retrieval Augmented Generation (RAG). Retrieval Augmented Generation (RAG) has emerged as a standard LLM paradigm (Lewis et al., 2020; Gao et al., 2023). In a RAG system, information retrieval over an external knowledge base is employed to improve LLM domain awareness and factuality. In its simplest form, a corpus of unstructured text is divided into smaller passages, encoded into vector representations using a text embedding model (Reimers & Gurevych, 2019; Ni et al., 2022), and organized into a vector database. Queries are encoded using the same model and compared against the database, typically using a vector index and approximate nearest-neighbor search algorithms (M. Wang, Xu, Yue, & Wang, 2021). Retrieved passages provide additional evidence for the LLM when considering the user's queries. Recently, more sophisticated approaches have emerged, such as incorporating a re-ranking step (Glass et al., 2022), self-reflection (Asai, Wu, Wang, Sil, & Hajishirzi, 2024), and considering graph community structure (Edge et al., 2024).

LLM Agents LLMs-as-agents is a recent paradigmatic advancement which generally refers to systems that allow LLMs to use tools or otherwise make function calls (Varshney, 2023). LLM-powered agents have been developed to support goal-completion in several domains, including web browsing (Deng et al., 2024), code debugging (Lee et al., 2024; Bouzenia, Devanbu, & Pradel, 2024), and social simulation (Park et al., 2023; Horton, 2023; Gürcan, 2024).

Multi-agent frameworks for LLM systems involve multiple models with specialized roles. While agents may use the same LLM, autonomous agents act independently based on

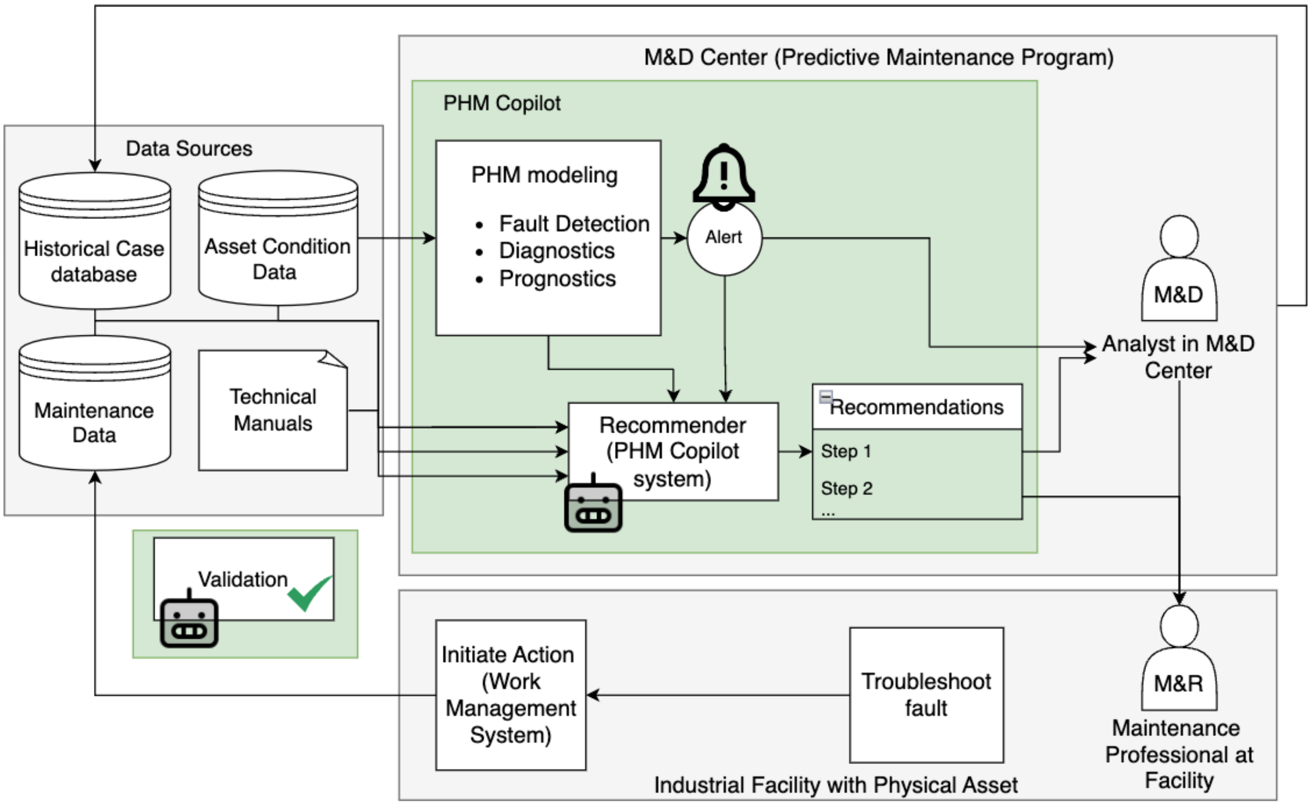

Figure 1. Conceptual illustration of a PHM Copilot as part of a comprehensive PHM system.

their roles, goals, and contexts. Such systems also require defined communication protocols and coordination mechanisms. This framework allows for the decomposition of complex problems into manageable tasks, with specialized agents handling individual aspects (Talebirad & Nadiri, 2023; X. Liu et al., 2024).

Evaluating LLM systems. LLMs are typically evaluated against static, standardized benchmarks, such as MMLU (Hendrycks et al., 2021), HellaSwag (Zellers, Holtzman, Bisk, Farhadi, & Choi, 2019), and TruthfulQA (Lin, Hilton, & Evans, 2022). These evaluations do not necessarily map directly to open-ended tasks or capture the subtleties of human preferences, however, motivating leader boards that crowd-source pairwise human comparisons (Chiang et al., 2024). Specific to engineering applications, DesignQA is a benchmark for multimodel LLMs specifically for their ability to understand and apply engineering requirements in technical documentation (Doris et al., 2024).

Soliciting human evaluations can be time-consuming and costly, so LLMs themselves have increasingly been employed as generative evaluators (Dubois et al., 2024). Strong LLM-as-judge systems can produce evaluations consistent with their crowd-sourced equivalents (Zheng et al., 2024), but are susceptible to judgement biases to varying degrees (Chen, Chen, Liu, Jiang, & Wang, 2024) and implicitly exhibit preference for their own generated text over those of other models and humans (Panickssery, Bowman, & Feng, 2024), presenting several challenges for automated LLM evaluation (Shankar, Zamfirescu-Pereira, Hartmann, Parameswaran, & Arawjo, 2024). More involved approaches, such as expert calibration (Y. Liu et al., 2024) and agent-based collaborative evaluation (Chan et al., 2024), have been developed to target these shortcomings.

2.2. Automation of Maintenance Troubleshooting

The concept of automating maintenance troubleshooting recommendations falls under the broader category of real-time suggestion systems, which are algorithms designed to suggest relevant information and provide prescriptive decision support across various applications, requiring well-populated knowledge frameworks (Lepenioti, Bousdekis, Apostolou, & Mentzas, 2020). In the industrial domain, real-time suggestion systems have been developed and can be considered a type of TLP task, providing actionable recommendations akin to chat-based tasks.

Many M&D centers have accumulated extensive databases of historical cases over the years. Recent studies have investigated using a TLP approach to this historical data for extracting knowledge relevant to maintenance and troubleshooting. Pau, Tarquini, Iannitelli, and Allegorico (2021) utilized NLP techniques to provide consistent troubleshooting insights in a Maintenance and Diagnostics (M&D) center (Pau, Tarquini, Iannitelli, & Allegorico, 2021). Their approach involved topic modeling and clustering to group cases, allowing for the extraction of valuable knowledge from the M&D center case data (Baker Hughes). This initiative aimed to support technical experts during troubleshooting activities, providing site operators with consistent technical insights and M&D operators with consistent recommendations to support junior personnel. Similarly, Sala and colleagues applied topic modeling (LDA) to historical records as part of their Product-Service Systems (PSS) offerings for manufacturing (Sala, Pirola, Pezzotta, & Cavalieri, 2022), (Sala, Pirola, Pezzotta, & Cavalieri, 2023), (Sala, Pirola, Dovere, & Cavalieri, 2019).

Peshave, Virani, Yang, and Saxena (2022) focused on evaluating vectorization approaches for short-text case titles using historical cases from an M&D center (Peshave et al., 2022). Their goal was to reduce the effort required from subject matter experts (SMEs). Trilla, Mijatovic, and Vilasis-Cardona (2022) utilized Task Learning Processes (TLP) for troubleshooting within Prognostics and Health Management (PHM) (Trilla, Mijatovic, & Vilasis-Cardona, 2022). Their approach extracted insights to advise maintenance teams on identifying the most probable root cause of problems. They developed a failure ontology based on failure modes and effects analysis, alongside a data-driven quality strategy called Return on Experience, to eliminate root causes and ensure sustainable improvements. Their work included developing "causality embeddings" between problems and root causes, differing from the conversational approach used in systems like ChatGPT.

Pires, Leitão, Moreira and Ahmad (2023) compared different recommendation systems for manufacturing operations, including a discrete event simulation model (digital twin) (Pires, Leitão, Moreira, & Ahmad, 2023). They deployed their digital twin in a case study of a battery pack assembly line in a university lab, focusing on recommending the optimal logistical scenario from a set of generated scenarios. This iterative process demonstrated improved user ratings over state-of-the-art recommendation systems. Addepalli, Weyde, Namoano, Ayodeji Oyedeji, Wang, Erkoyuncu, and Roy (2023) developed a knowledge extraction framework that provides information in response to degradation events by extracting historical degradation information from full-text papers (Addepalli et al., 2023).

LLMs to assist in maintenance troubleshooting

Recent advancements have explored the integration of large language models (LLMs) into maintenance troubleshooting workflows related to PHM. Vidyaratne, Lee, Kumar, Watanabe, Farahat, and Gupta developed an LLM-augmented pipeline to extract content from product manuals and organize it into troubleshooting tree structures. Their framework used LLMs to process unstructured text and create systematic guides for diagnosing and resolving issues in industrial equipment (Vidyaratne et al., 2024). Trilla, Yiboe, Mijatovic, and Vitrià presented a proof of concept for industrial-grade smart troubleshooting through causal technical language processing. Their approach leverages causal associations in text data used to determine the root cause of a problem and provide an unbiased estimation of the most likely potential solution and employs LLMs to represent technical knowledge and assist experts in diagnosing industrial asset issues (Trilla, Yiboe, Mijatovic, & Vitrià, 2024).

Kohl, Eschenbacher, Besingerand and Ansari propose a LLM-based chatbot for improving maintenance planning and operations which combines LLMs with knowledge graphs in a flexible, modular systems (Kohl, Eschenbacher, Besinger, & Ansari, 2024). A use-case scenario is presented in the railway industry, demonstrating the use of the chatbot in maintaining a cooling system. Similarly, Ferdousi, Hossain, Yang, and El Saddik propose DefectTwin, which integrates LLMs with digital twin for visual railway defect inspection. (Ferdousi, Hossain, Yang, & Saddik, 2024).

D. Li, H. Li, J. Li, H.W. Li, Wang, Minerva, Crespi and K.C. Li combined blockchain with LLMs in a PHM application, using sensor data and maintenance logs. The blockchain component ensured data security, while LLMs assisted in classifying equipment failures into "No Failure," "Minor Failure," and "Major Failure" categories (Li et al., 2024). Lukens and Ali further evaluated zero-shot LLM performance for troubleshooting, highlighting areas for future research (Lukens & Ali, 2023).

3. METHODOLOGY

3.1. PHM copilot

By "copilot," we refer generically to a system powered by LLMs designed to help address complex cognitive tasks (Ren, Zhan, Yu, Ding, & Tao, 2024). A conceptual illustration of what a PHM Copilot as part of a comprehensive PHM system could look like is shown in Figure 1. The major elements of a PHM system include: (1) data collection; (2) predictive modeling capabilities; (3) initiating actions based on the data and model outputs; and (4) validating if the predictions were correct (Hodkiewicz, Lukens, Brundage, & Sexton, 2021).

Toward this end, we propose a PHM copilot system with the following key components:

- a real-time predictive maintenance sensor system;
- a data store of historical records, including past failure

modes and their corresponding anomalous sensor readings;

- a Recommender agent responsible for (1) reviewing sensor reports that are flagged as anomalous and (2) constructing a structured list of troubleshooting steps; and
- an Evaluator agent tasked with validating the Recommender's investigative plan.

The Recommender and Evaluator agents are LLMs induced to return structured recommendation or evaluation objects, rather than open-ended responses, using function-calling. This technique can reduce costs and avoid the need for additional text processing steps in an automated workflow (Eleti & Kilpatrick, 2023). Our implementation relies on OpenAI's *gpt-3.5-turbo-0125* endpoint (Achiam et al., 2023), but the modular framework allows for alternative models, as well. Together, the PHM copilot and human personnel collaborate to identify failure modes as quickly, while prioritizing less invasive investigative steps first. For each alert from the predictive maintenance sensor system, the recommender agent is sent the following prompt:

> A sensor system identified the following warning(s): [observed], pertaining to the asset [asset]. Write a list of [step number] steps, to be executed in sequential order, by a maintenance professional in order to identify the casual failure mode. These steps should be as atomic as possible. Our goal is to identify the failure mode as quickly as possible, while prioritizing low-invasiveness steps early on, as well.

where [observed] and [asset] are replaced with cleaned sensor observations and asset type labels, respectively. [step number] is set to 10.

The Evaluator agent has access to each incident's true failure mode and is tasked with determining if the steps in the Recommender's plan would identify the true failure if carried out. The evaluator is sent the following prompt:

> A sensor system identified the following warning(s): [observed], pertaining to the asset [asset].
> This prompted a thorough manual investigation, revealing the following failure mode: [failure mode].
> Without knowing the true failure mode, the following sequential investigative plan was proposed: [plan].
> For each step in the sequential investigative plan, assess whether or not a trained maintenance professional would explicitly discover the given true failure mode in the course of performing that step in isolation and provide reasoning. If the step would not discover the failure mode, represent that step with False. If it would discover the given failure mode, represent it with a True. For example, for a failure mode of 'punctured inner tube', and steps of { step_1: ‘check tire pressure gauge’, step_2:‘examine inner tube for punctures.’}, you should represent it as {step_1:False, step_2:True}

where [observed], [asset], [failure mode], and [plan] are replaced with the sensor observation(s), applicable asset class, plan produced by the recommender agent, and the ground truth failure mode(s), respectively.

The Evaluator produces a boolean for each of the ten steps in the Recommender's plan: true if that step would catch the ground truth failure, and false otherwise. The final output of the Evaluator is a list of ten boolean values, which correspond to each step in the given plan. The Evaluator validates each step and each plan independently, allowing for multiple steps in a plan to potentially reveal a true failure mode. In Evaluating cases with multiple failure modes, the Evaluator evaluates the plan once per failure mode.

3.2. Case study with commercial PHM system records

GE Vernova offers a PHM solution which uses an anomaly detection algorithm based on multivariate pattern recognition (Herzog, 2014), (Herzog, Hanlin, Wegerich, & Wilks, 2005). The company publicizes historical use cases - including sensor anomalies detected by the PHM model, the root cause of the fault, and the corrective actions taken - on its marketing web page (GE, 2024). Table 1 illustrates textual data provided from an example case. The cases include categorical fields for filtration (Asset, Industry, Market), images which show the detected anomalous behavior, and free text fields (Observation, Cause and Value).

To process the historical cases from the website for our automated workflow, we employ two additional LLM agents, following the same structured generation via function-calling scheme as the Recommender and Evaluator:

Observation Agent is responsible for reviewing the unstructured case text and returning a structured object describing

Table 1. Example of the text from a case on the GE Vernova website, with extracted observation and failure mode.

Field	Case text
Title	Increased bearing temperatures on a gas turbine exposed
Asset	Gas Turbine
Industry	Power
Market	Latin America
Observed	Beginning in November, GE Digital's Asset Performance Management detected a deviation on a gas turbine at a combined-cycle plant. Specifically, the journal bearing temperature increased from 215°F (101°C) to 245°F (118°C). GE Digital's Industrial Managed Services team added this item to the weekly report for discussion with the customer.
Cause	After the alert from GE Digital‘s Industrial Managed Service team, the customer discovered a bearing misalignment. After the customer aligned the bearing, the journal bearing temperature returned to 208°F (97°C).
Value	Due to the early notification from GE Digital's Industrial Managed Service team, the customer was able to align the bearing. Overheating of the bearing could have resulted in damage to the bearing, which could have led to repair costs, loss of production, and a trip. GE Digital's Industrial Managed Service team was able to verify the maintenance actions were successful by observing the actual values return to the model-predicted estimate. This catch is estimated to have avoided approximately $129,600 in costs. Avoided costs are based on North American average production loss.
Observed	A deviation was detected on a gas turbine at a combined-cycle plant. The journal bearing temperature increased from 215°F to 245°F.
Failure Mode	Bearing misalignment

observed physical conditions. For each observation, the observation agent is given the following prompt:

> You are helping to structure text by only returning the observed behavior of the sensor system from the full description. Anything regarding who did the observing, like GE Digital, or regarding Asset Performance Management, the Managed Services team and discussions with the customer are out of scope and not an observed behavior of the sensor system. Structure output in sentences with periods, keeping words like detected, found or identified and include facility (ex:combined cycle plant, oil and gas processing, offshore platform, coal power plant, mining, etc) and asset (ex: motor, compressor, steam turbine, gas turbine).. The full description is: [observed]

The marketing web page data contains information that is not useful for this case study such as details on the GE team, cost savings, and customer interaction. An ideal response from the observation agent will strip the reported text to its machine observation. It is key that none of the important information about the system's observation itself is removed. An example of an extracted response is shown in Table 1.

Failure Mode Extraction (FME) Agent. The FME agent, tasked with failure mode classification, is responsible for processing the asset, cause, and value fields to produce a structured list of failure modes. In this context, a "failure mode" refers to the physical cause of asset failure that needs to be identified during troubleshooting. Many case descriptions also include observed faults which are symptoms or consequences of the initial cause. While identifying these secondary faults is important for tasks such as maintenance planning, the focus of this use case is on identifying the primary cause. Diagnosing a symptom rather than the cause may not necessarily lead to the correct diagnosis or corrective action. Under this contextual definition of failure mode, we expect one or two primary faults per case. For each case, the FME agent is given the following prompt:

> You are helping identify the physical cause or failure mode which contributed to a detected anomaly on a [asset]. From the following unstructured cause and value statements, can you return the failure mode or modes in a structured form. An ideal failure mode contains information about two things: 1. Physical part(s) or component(s) and 2. States or condition of the physical object which caused the fault. If no failure mode is in the description, return ”no failure mode stated”. There can be multiple failure modes, separate these with a semi-colon. Do not return actions taken. Do not return observations recorded from sensors such as increased temperature or decreased pressure. The cause is: [cause] and the value is: [value]

We utilize the processed case records to illustrate system behaviors and understand practical implementation requirements for the PHM copilot. A high level schema of the system and overview is shown in Figure 2. Ultimately, 394 historical cases were extracted from the GE Vernova website, covering 36 asset classes, predominantly involving rotating assets such as turbines, pumps, generators, and engines in process manufacturing industries (oil and gas, chemicals, mining, etc.). Some cases lacked structured fields; nine assets were completed by our SME based on contextual information. The distribution of cases among asset classes is uneven. Out of the 36 asset classes, 10 (28%) had more than 10 cases each, collectively representing 82% of the dataset (328 cases). The asset classes with the highest number of cases include compressors, gas turbines, pumps, combustion turbines, boiler feed pumps, steam turbines, reciprocating en-

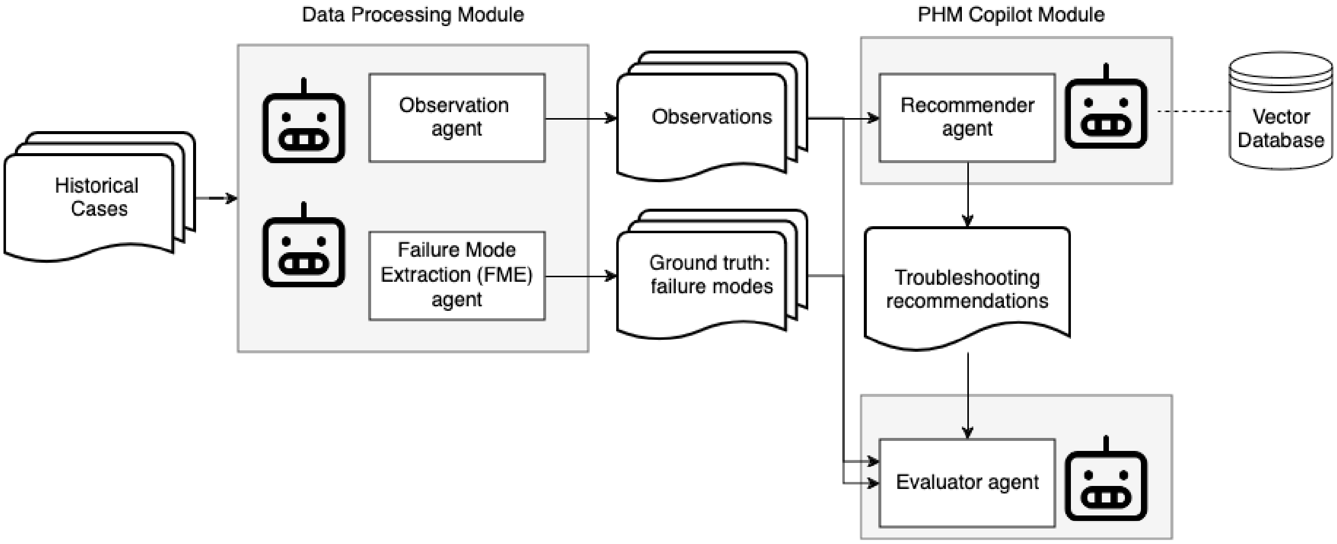

Figure 2. High level schema of the case study with the LLM agents used.

gines, generators, Heat Recovery Steam Generators (HRSG), and jet engines.

3.3. RAG experiment

We briefly examine the potential of retrieval-augmented generation (RAG) to enhance system performance. The Recommender agent is designed with two configurations: a baseline with no augmentation ("LLM-only"), and a variation where the Recommender is given access to semantically similar historical records ("RAG"). To support this comparison, the data is randomly split into a collection and a test set of 100 and 294 records, respectively. The records in the collection set are organized into a local *txtai* vector database (Mezzetti, 2020) using the pre-trained *all-MiniLM-L6-v2* encoder model (Sentence Transformers, 2021), a down-sized implementation of MiniLM (W. Wang et al., 2020). Therein, historical records are represented by fixed-length vectors representing the records' positions in *all-MiniLM-L6-v2*'s learned latent space.

In the RAG configuration of the Recommender agent, incoming alerts are queried against the vector database to identify the 10 most semantically-similar historical records pertaining to the same asset. Semantic similarity is assessed by pairwise vector comparison; given vectors $\mathbf{u}$ and $\mathbf{v}$ (vector representations for the alert and a historical record, respectively), we calculate cosine similarity:

$$\text{similarity}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}, \quad (1)$$

i.e., the L2-normalized dot product between $\mathbf{u}$ and $\mathbf{v}$. The following is then appended to the Recommender's prompt:

> Here are some previous examples that may or may not be relevant for this case: [context]

where [context] is replaced with the retrieved historical records.

4. RESULTS

Subject Matter Experts (SMEs) were involved to validate model behavior. The primary SME, a GE Vernova author with over ten years of experience in commercial software development and support, particularly with the software generating the historical case data, conducted most of the evaluations. A second author with ten years of experience in Asset Performance Management software and a Certified Maintenance and Reliability Professional also reviewed the SME assessments.

4.1. Observation and FME agents

To validate the results of the observation agent, 50 agent's cleaned observation responses were randomly sampled. 46 out of the 50 (92%) of the responses from the observation agent contained all of the relevant information in the observation, In other words, for the SME-reviewed sample, the agent returned the most complete response it could 92% of the time. And in all 50, the marketing tone was dropped and the observations read like passive technical descriptions. An example where the observation agent dropped relevant information is shown in Table 2. In this example, most noteworthy is the missing domain context in the original description which was not present in the response.

As the results were reviewed with the SME, it was observed

Table 2. Example of the Observation Agent in which context such as the domain was dropped when translating GE's online description to a usable engineering description of the observed behavior. The agent changes the tone and purpose of the text to focus on the observation itself, as opposed to superfluous information about GE and the date.

	Text
Input	On July 15th, GE Digital's Asset Performance Management detected a potential performance/balance flow issue on a boiler feed pump at an combined-cycle power plant. On August 16, the bearing vibrations on the outboard bearing of the boiler feed pump increased from values of 1.3 mils up to values as high as 1.88 mils. GE Digital's Industrial Managed Services team added this item to the weekly report for discussion with the customer.
Output	The bearing vibrations on the outboard bearing of the boiler feed pump increased from values of 1.3 mils up to values as high as 1.88 mils.

Table 3. Accuracy metrics for the SME reviewed failure modes predicted by the FME agent grouped by failure mode category.

LLM extracted failure mode category	Percent of Cases Reviewed	Percent with desired label	Percent with desired label or Partial
1 Failure Mode	6.2%	92%	100%
2 Failure Modes	7.1%	30%	100%
3+ Failure Modes	100%	0%	89%
No Failure Mode Stated	100%	7%	13%

that in more complicated cases, such as involving faults arising from a complex chain of events, the FME agent was more likely to err and return all possible listed faults (including symptoms of the fault), list information instructed not to list in the prompt, or list avoided faults. In some cases it did not state a failure mode. To address this issue, we determined that some of the data required hand labeling post-hoc. Rather than hand labeling all of the data, we used a process for identifying which failure modes to review. The criteria for selecting data to review and hand labeling was:

- **3+ Failure Modes:** If a case has more than 2 LLM extracted failure modes (28 cases)
- **No Failure Mode Stated:** If the LLM suggested "no failure mode stated" (16 cases)
- **2 Failure Modes:** Random sample of cases with 2 identified failure modes (10 cases)
- **1 Failure Mode:** Random sample of cases with 1 identified failure mode (13 cases)

A total of 67 cases were reviewed by SMEs for failure modes, resulting in 51 failure mode labels being overwritten by SME labeling. A partial score option was available and typically used in cases when the desired failure mode was identified, but additional, extraneous failure modes were also returned. During hand-labeling, these extraneous failure modes were deleted. Ultimately, each of the 394 cases ended up with 0, 1, or 2 failure mode labels.

The accuracy metrics split by failure mode group are reported in Table 3. The high percentage of correct or partially correct scores (third column of Table 3) indicates that the LLM often correctly identified the failure mode, but in some cases added unnecessary additional failure modes.

4.2. Recommender agent

The Recommender produced 294 plans recommending 10 steps for each case to solve the issue from the asset class and observation.

The Evaluator agent is used to asses the efficiency of the Recommender, i.e., how early, if at all, personnel following the Recommender's troubleshooting steps would discover the true failure mode. Example output for the case shown in Table 1, pertaining to bearing misalignment on a gas turbine, is shown in Table 4. In the RAG configuration, the recommendation to check for bearing alignment occurred earlier than in the LLM-only setting (Step 5 and Step 2, respectively).

Overall, the Evaluator assesses the Recommender as on average catching more failure modes (96.0%) and doing so in fewer steps (2.49) when employing RAG than without (94.0% and 3.49, respectively), as illustrated in Table 5 and Figure 3.

4.3. Evaluator agent

Our proposed system involves an Evaluator agent, following existing LLM-as-judge patterns. It is not guaranteed, however, that the Evaluator is itself a reliable judge. We evaluated the Evaluator via spot-checking, employing a Subject Matter Expert (SME) to manually review recommendations across 10 randomly selected cases covering 12 failure modes, for 20 sets of recommendation steps covering 24 failure modes total (Figure 4).

The Evaluator assessed that the Recommender's troubleshooting steps would lead to the discovery of the failure mode in 95.8% of the manually-checked cases; the SME, on the other hand, found a failure mode discovery rate of 70.8%. Agreement on which step would reveal the failure mode, however, was only 25% (Figure 4).

5. DISCUSSION

We outline a simplistic LLM agent-based AI copilot framework to understand the scope, requirements and considerations for realistic implementation in the PHM domain. Our

Table 4. Sample responses of the Recommender (LLM-only and RAG) and Evaluator agents for the example case in Table 1.

Step No.	Recommender - LLM only	Evaluator LLM only	Recommender - RAG	Evaluator RAG
Step 1	Inspect the gas turbine for any visible signs of damage or wear such as leaks, corrosion, or loose components.	FALSE	Check for any visible signs of damage or wear on the gas turbine components, focusing on the journal bearing area.	FALSE
Step 2	Check the lubrication system of the gas turbine to ensure proper functioning and adequate lubricant levels.	FALSE	Inspect the alignment of the gas turbine components, specifically the journal bearing alignment.	**TRUE**
Step 3	Review the operating logs and maintenance records of the gas turbine to identify any recent maintenance activities or abnormalities.	FALSE	Review historical data of the gas turbine to identify any previous deviations or trends in temperature changes in the journal bearing.	FALSE
Step 4	Conduct a vibration analysis on the gas turbine to detect any abnormal vibrations or resonance that could indicate a potential issue.	FALSE	Analyze the cooling system of the gas turbine to ensure proper functioning and circulation of coolant around the journal bearing.	FALSE
Step 5	Inspect the journal bearings of the gas turbine for signs of wear, misalignment, or lack of lubrication.	**TRUE**	Conduct a thermal imaging scan of the journal bearing area to identify any areas of abnormal temperature rise.	FALSE
Step 6	Perform a thermographic inspection of the gas turbine to identify any hotspots or temperature anomalies.	FALSE	Inspect the lube oil system to check for any issues with lubrication that may be impacting the journal bearing temperature.	FALSE
Step 7	Check the cooling system of the gas turbine to ensure proper cooling and ventilation to prevent overheating.	FALSE	Review the control valves and cooling fans related to the journal bearing to ensure they are functioning correctly.	FALSE
Step 8	Inspect the inlet air filters of the gas turbine to ensure they are clean and not causing any restriction to airflow.	FALSE	Perform vibration analysis on the gas turbine to detect any abnormal vibrations that may be causing the temperature deviation in the journal bearing.	FALSE
Step 9	Conduct a borescope inspection of the gas turbine internals to check for any blockages, fouling, or damage.	FALSE	Check the insulation around the inner barrel tunnel to ensure there are no gaps or issues causing heat build-up in the journal bearing area.	FALSE
Step 10	Review the control system and operational parameters of the gas turbine to ensure they are within normal operating limits and settings.	FALSE	Verify the alignment and condition of the fuel nozzles to rule out any potential issues affecting the journal bearing temperature.	FALSE

Table 5. Output of the evaluator agent across 294 cases and 399 failure modes, comparing two models for recommender agent - LLM-only and RAG-based. Results are normalized across both failure mode and case.

Metric	LLM-only	RAG
No. Failure Modes Caught	375	383
% Failure Modes Caught	94.0%	**96.0%**
No. Cases all Failure Modes Caught	271	278
% Cases all Failure Modes Caught	92.2%	**94.6%**
Mean No. of Steps	3.49	**2.49**
Standard Deviation	3.40	2.28
Median	1.0	1.0

findings identify several areas of priority for developing a more sophisticated practical system.

Data Preparation. The Observation Extraction task was relatively straightforward for the LLM and generally performed well. Although content the SMEs found important was occasionally omitted, the agent performed well overall while executing significantly faster than manual text cleanup.

The FME task was less straightforward, especially in more complex cases. When the description contained content organized as a list of failure modes such as a sequence of cascading events or avoided faults, the FME agent struggled to select the desired fault(s) (cause fault or occurred fault in this example) and tended to select every possible fault in the description. The differences in performance between the Observation and FME extraction tasks highlight the importance of domain knowledge in failure mode extraction.

PHM Copilot. Our experiment explored providing the Recommender domain knowledge via retrieved historical cases, and our results indicate that doing so improves system performance. Historical cases, however, do not provide sufficient coverage to serve as sole exemplars; instead, an approach that also retrieves passages from relevant technical manuals, P&ID diagrams, or textbooks may be preferable.

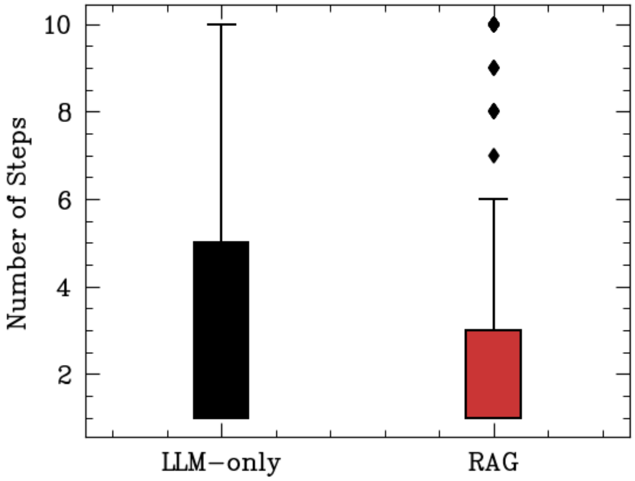

Figure 3. Box plot comparison of the number of steps (for failure modes identified) in which a failure mode would have been caught following the recommendations of the recommender agent, according to the evaluator agent. Cases where the failure mode were never discovered at all are excluded.

Agreement between the Evaluator agent and SME reviewers was low; future work should explore integrating domain knowledge on the Evaluator side, as well. We emphasize that a strong Evaluator can itself lead to a strong Recommender. Due to the stochasticity of LLM outputs, a self-checking procedure whereby the Recommender generates several troubleshooting plans and the Evaluator selects the best among them has the potential to improve Recommender performance without making changes to the Recommender.

5.1. Limitations and Future work

The limitations in this case study are organized to highlight areas for future work and key design considerations for adapting LLM technology to PHM applications.

Case Study Data. The diversity of the different asset types in the case study dataset was beneficial for identifying high level requirements, but in practice, each asset type, its usage and manufacturer specific design elements are specific inputs for health monitoring. This variability means that different assets or systems in similar operational contexts may require distinct technical manuals and specifications with relevant physical system information.

Further, the case study data consists of cleaned success stories, which may not be representative of realistic asset condition data streaming to a predictive modeling system. This dataset does not accurately represent a class-imbalanced environment, where 99.99% of observations are healthy and false positives are the predominant type of alert.

Information Retrieval and Reference Data. The retrieval strategy employed for the Recommender Agent's RAG configuration is simplistic, relying on semantic similarity alone. Depending on the use case, more sophisticated approaches may be applied to retrieve the most relevant context. For future system development, an appropriate evaluation processes is needed to ensure the accuracy and relevance of the documents retrieved by the system.

While relying solely on historical cases as a reference dataset can provide insights into commonly recurring faults and help avoid past mistakes, there are limitations. Rare but high-consequence faults may not be adequately represented. Additionally, using only historical cases overlooks valuable resources such as technical manuals, P&ID diagrams, and relevant textbooks. Recent work has developed approaches for using LLMs to assist in extracting content from technical manuals to assist in maintenance troubleshooting (Vidyaratne et al., 2024).

The LLM used in this work is natively unimodal. Realistic PHM applications, however, may include data best expressed in non-text modes, such as images or audio. Recently, multimodel LLMs have been explored for engineering design (Doris et al., 2024), (Picard et al., 2023), (Ferdousi et al., 2024). Future work should focus on identifying how to adapt context- and asset-specific technical content which may be multi-modal in nature.

Performance Evaluation. Our evaluator agent served as a placeholder to ensure system validation was explicitly included in the PHM co-pilot. In this study, the evaluator agent simply checked whether a specific failure mode would be identified by the recommendations. However, other performance metrics, such as assessing if the troubleshooting steps are in order of increasing invasiveness, may also be important.

In addition to enhancing the existing evaluation approaches, future work should focus on expanding the coverage of evaluated responses. While manually review of model outputs demands significant SME bandwidth, it allows for deeper analysis such as comparing performance across different physical systems. More broadly, there is a need for standardized approaches to evaluate and benchmark LLM performance for industrial applications.

Operational Constraints. The case study was performed on personal laptops using a shared repository. However, for an operational system, it is important to consider additional components including legal, cybersecurity and aspects of Responsible AI. For instance, access to LLMs like GPT-3.5 are readily accessible via APIs. However, using these APIs could result in data leakage through prompts sent to the hosting company. Protecting sensitive information, such as export control data in a nuclear power plant, necessitates secure, controlled environments for deploying pre-trained LLM's,

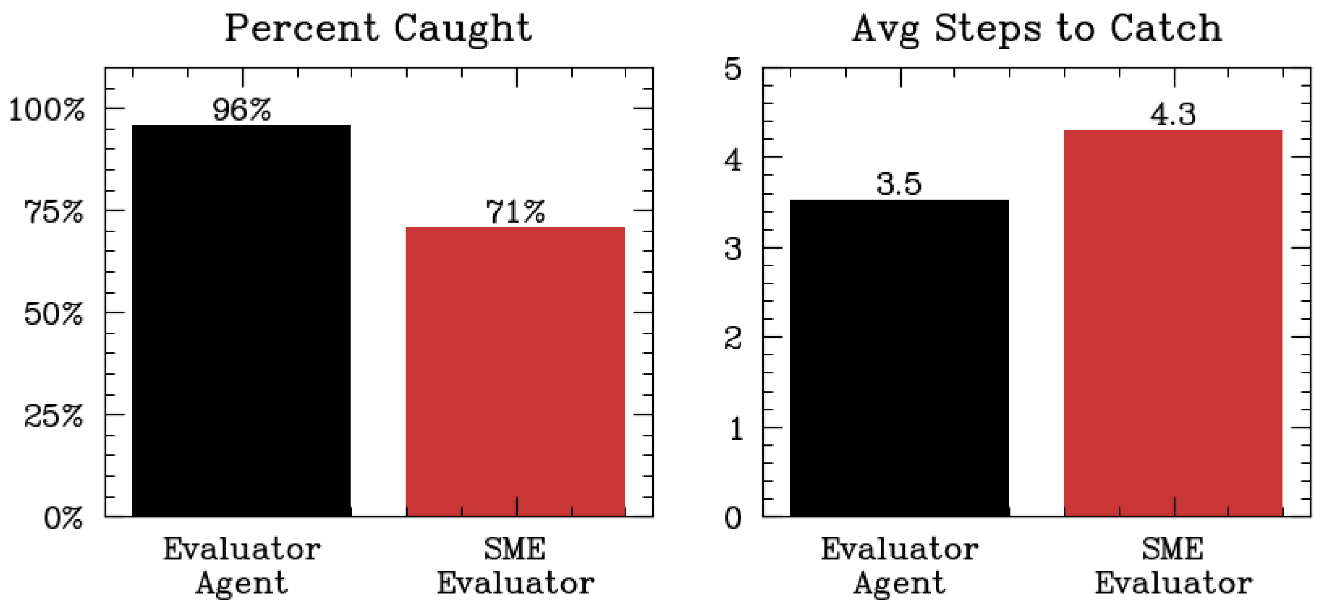

Figure 4. Comparison between the Evaluator agent (output evaluated by LLM) and the SME evaluation (output evaluated by SME) for a subset of cases covering 24 failure modes.

which presents additional challenges in model and technology selection and system design.

Operational constraints also involve data collection, storage and handling of PHM data. Additionally, human-centered design is important, as involving end-users in the development process ensures the tool meets practical needs. These considerations provide a starting point and highlights key areas for further development with industrial requirements in mind.

6. CONCLUSION

A PHM copilot was implemented using open-source case data, as a proof-of-concept to explore the potential for tailoring LLMs to PHM tasks. At a high level, two primary use case areas were identified where LLMs demonstrate value: (1) as tools for data quality improvement, where data may be insufficient for the desired analytics, and (2) as a tool for developing a prescriptive layer on an already mature prescriptive model to assist in decision support and recommendations.

A significant challenge in this study was the resource-intensive process of manual SME reviews, which limited our ability for deeper exploration such as incorporating additional performance metrics and integrating additional data sources such as technical manuals for retrieval. For future development in this area, we suggest incorporating simpler, more gradable tasks upfront to streamline experimentation and allow for more efficient evaluation. It may be also beneficial to modify the design of the PHM recommender system to mitigate LLM limitations, such as their non-deterministic nature, to enhance applicability in industrial settings.

ACKNOWLEDGMENT

This work was financially supported by LMI Applied Research and Partnerships.

NOMENCLATURE

PHM	Prognostics and Health Management
TLP	Technical Language Processing
AI	Artificial Intelligence
LLM	Large Language Model
M&D	Monitoring and Diagnostics
P&ID	Piping and Instrumentation Diagram
M&R	Maintenance and Reliability
SME	Subject Matter Expert
RAG	Retrieval-Augmentation Generation
FME	Failure Mode Extraction
HRSG	Heat Recovery Steam Generator

REFERENCES

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., ... others (2023). GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Addepalli, S., Weyde, T., Namoano, B., Oyedeji, O. A., Wang, T., Erkoyuncu, J. A., & Roy, R. (2023). Automation of knowledge extraction for degradation analysis. *CIRP Annals*, *72*(1), 33–36. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0007850623000070` doi:

https://doi.org/10.1016/j.cirp.2023.03.013

Asai, A., Wu, Z., Wang, Y., Sil, A., & Hajishirzi, H. (2024). Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations.* Retrieved from https://openreview.net/forum?id=hSyW5go0v8

Bikaun, T., & Hodkiewicz, M. (2021). Semi-automated Estimation of Reliability Measures from Maintenance-Work Order Records. In *PHM Society European Conference* (Vol. 6, pp. 9–9).

Bouzenia, I., Devanbu, P., & Pradel, M. (2024). RepairAgent: An Autonomous, LLM-Based Agent for Program Repair. *arXiv preprint arXiv:2403.17134*.

Brundage, M. P., Sexton, T., Hodkiewicz, M., Dima, A., & Lukens, S. (2021). Technical language processing: Unlocking maintenance knowledge. *Manufacturing Letters*, *27*, 42–46.

Cambon, A., Hecht, B., Edelman, B., Ngwe, D., Jaffe, S., Heger, A., … Teevan, J. (2023). Early LLM-based Tools for Enterprise Information Workers Likely Provide Meaningful Boosts to Productivity. *Microsoft Research. MSR-TR-2023-43*.

Chan, C.-M., Chen, W., Su, Y., Yu, J., Xue, W., Zhang, S., … Liu, Z. (2024). ChatEval: Towards Better LLM-based Evaluators through Multi-Agent Debate. In *The Twelfth International Conference on Learning Representations.*

Chen, G. H., Chen, S., Liu, Z., Jiang, F., & Wang, B. (2024). Humans or LLMs as the Judge? A Study on Judgement Biases. *arXiv preprint arXiv:2402.10669*.

Chengwei Wei and Yun-Cheng Wang and Bin Wang and C.-C. Jay Kuo. (2024). P. *APSIPA Transactions on Signal and Information Processing*, *13*(2), -. Retrieved from http://dx.doi.org/10.1561/116.00000010 doi: 10.1561/116.00000010

Chiang, W.-L., Zheng, L., Sheng, Y., Angelopoulos, A. N., Li, T., Li, D., … others (2024). Chatbot Arena: An Open Platform for Evaluating LLMs by Human Preference. *arXiv preprint arXiv:2403.04132*.

Deng, X., Gu, Y., Zheng, B., Chen, S., Stevens, S., Wang, B., … Su, Y. (2024). Mind2Web: Towards a Generalist Agent for the Web. *Advances in Neural Information Processing Systems*, *36*.

Dima, A., Lukens, S., Hodkiewicz, M., Sexton, T., & Brundage, M. P. (2021). Adapting natural language processing for technical text. *Applied AI Letters*, *2*(3), e33.

Doris, A. C., Grandi, D., Tomich, R., Alam, M. F., Cheong, H., & Ahmed, F. (2024). DesignQA: A Multimodal Benchmark for Evaluating Large Language Models' Understanding of Engineering Documentation. *arXiv preprint arXiv:2404.07917*.

Dubois, Y., Li, C. X., Taori, R., Zhang, T., Gulrajani, I., Ba, J., … Hashimoto, T. B. (2024). Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, *36*.

Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., … Larson, J. (2024). From Local to Global: A Graph RAG Approach to Query-Focused Summarization. *arXiv preprint arXiv:2404.16130*.

Eleti, H. J., Atty, & Kilpatrick, L. (2023, June). *Function Calling and Other API Updates.* https://openai.com/index/function-calling-and-other-api-updates/. (Accessed: 2024-06-25)

Ferdousi, R., Hossain, M. A., Yang, C., & Saddik, A. E. (2024). Defecttwin: When llm meets digital twin for railway defect inspection. *arXiv preprint arXiv:2409.06725*.

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., … Wang, H. (2023). Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

GE. (2024). *Remote Monitoring Powered by Digital Twins.* https://www.ge.com/digital/industrial-managed-services-remote-monitoring-for-iiot/. (Accessed: April 1, 2024)

Glass, M., Rossiello, G., Chowdhury, M. F. M., Naik, A., Cai, P., & Gliozzo, A. (2022, July). Re2G: Retrieve, Rerank, Generate. In M. Carpuat, M.-C. de Marneffe, & I. V. Meza Ruiz (Eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 2701–2715). Seattle, United States: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2022.naacl-main.194 doi: 10.18653/v1/2022.naacl-main.194

Greenberg, E. S. (2010). Labor Unions at Boeing: Reflections on Our Findings in'Turbulence: Boeing and the Future of American Workers and Managers'(Yale Press, 2010) and Developments Since Its Publication.

Gürcan, Ö. (2024). LLM-Augmented Agent-Based Modelling for Social Simulations: Challenges and Opportunities. *HHAI 2024: Hybrid Human AI Systems for the Social Good*, 134–144.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., & Steinhardt, J. (2021). Measuring Massive Multitask Language Understanding. In *International Conference on Learning Representations.*

Herzog, J. P. (2014, February 25). *Monitoring system using kernel regression modeling with pattern sequences.* Google Patents. (US Patent 8,660,980)

Herzog, J. P., Hanlin, J., Wegerich, S. W., & Wilks, A. D. (2005). High performance condition monitoring of aircraft engines. In *Turbo Expo: Power for Land, Sea, and Air* (Vol. 46997, pp. 127–135).

Hodkiewicz, M., & Ho, M. T.-W. (2016). Cleaning historical maintenance work order data for reliability analysis. *Journal of Quality in Maintenance Engineering*.

Hodkiewicz, M., Lukens, S., Brundage, M. P., & Sexton, T. (2021). Rethinking Maintenance Terminology for an Industry 4.0 Future. *International Journal of Prognostics and Health Management*, *12*(1).

Horton, J. J. (2023). *Large Language Models as Simulated Economic Agents: What Can We Learn from Homo Silicus?* (Tech. Rep.). National Bureau of Economic Research.

Kohl, L., Eschenbacher, S., Besinger, P., & Ansari, F. (2024). Large language model-based chatbot for improving human-centricity in maintenance planning and operations. In *PHM Society European Conference* (Vol. 8, pp. 12–12).

Lee, C., Xia, C. S., Huang, J.-t., Zhu, Z., Zhang, L., & Lyu, M. R. (2024). A Unified Debugging Approach via LLM-Based Multi-Agent Synergy. *arXiv preprint arXiv:2404.17153*.

Lepenioti, K., Bousdekis, A., Apostolou, D., & Mentzas, G. (2020). Prescriptive analytics: Literature review and research challenges. *International Journal of Information Management*, *50*, 57–70.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., … others (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Advances in Neural Information Processing Systems*, *33*, 9459–9474.

Li, D., Li, H., Li, J., Li, H.-W., Wang, H., Minerva, R., … Li, K.-C. (2024). Blockchain-enabled large language models for prognostics and health management framework in industrial internet of things. In *International conference on blockchain, metaverse and trustworthy systems, blocksys' 2024*.

Lin, S., Hilton, J., & Evans, O. (2022, May). TruthfulQA: Measuring How Models Mimic Human Falsehoods. In S. Muresan, P. Nakov, & A. Villavicencio (Eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics* (pp. 3214–3252). Dublin, Ireland: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2022.acl-long.229` doi: 10.18653/v1/2022.acl-long.229

Liu, X., Yu, H., Zhang, H., Xu, Y., Lei, X., Lai, H., … others (2024). AgentBench: Evaluating LLMs as Agents.

Liu, Y., Yang, T., Huang, S., Zhang, Z., Huang, H., Wei, F., … Zhang, Q. (2024, May). Calibrating LLM-Based Evaluator. In N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, & N. Xue (Eds.), *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)* (pp. 2638–2656). Torino, Italia: ELRA and ICCL. Retrieved from `https://aclanthology.org/2024.lrec-main.237`

Lukens, S., & Ali, A. (2023). Evaluating the Performance of ChatGPT in the Automation of Maintenance Recommendations for Prognostics and Health Management. In *Proceedings of the Annual Conference of the PHM Society* (Vol. 15).

Lukens, S., Naik, M., Saetia, K., & Hu, X. (2019). Best Practices Framework for Improving Maintenance Data Quality to Enable Asset Performance Analytics. In *Proceedings of the Annual Conference of the PHM Society* (Vol. 11).

Majumder, S., Dong, L., Doudi, F., Cai, Y., Tian, C., Kalathil, D., … Xie, L. (2024). Exploring the capabilities and limitations of large language models in the electric energy sector. *Joule*, *8*(6), 1544–1549.

Makatura, L., Foshey, M., Wang, B., HähnLein, F., Ma, P., Deng, B., … others (2023). How Can Large Language Models Help Humans in Design and Manufacturing? *arXiv preprint arXiv:2307.14377*.

Mezzetti, D. (2020). *txtai: the all-in-one embeddings database.* Retrieved from `https://github.com/neuml/txtai`

Muellerleile, C. M. (2009). Financialization takes off at Boeing. *Journal of Economic Geography*, *9*(5), 663–677.

Ni, J., Hernandez Abrego, G., Constant, N., Ma, J., Hall, K., Cer, D., & Yang, Y. (2022, May). Sentence-T5: Scalable Sentence Encoders from Pre-trained Text-to-Text Models. In S. Muresan, P. Nakov, & A. Villavicencio (Eds.), *Findings of the Association for Computational Linguistics: ACL 2022* (pp. 1864–1874). Dublin, Ireland: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2022.findings-acl.146` doi: 10.18653/v1/2022.findings-acl.146

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., … others (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, *35*, 27730–27744.

Panickssery, A., Bowman, S. R., & Feng, S. (2024). LLM Evaluators Recognize and Favor Their Own Generations. *arXiv preprint arXiv:2404.13076*.

Park, J. S., O'Brien, J., Cai, C. J., Morris, M. R., Liang, P., & Bernstein, M. S. (2023). Generative Agents: Interactive Simulacra of Human Behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (pp. 1–22).

Pau, D., Tarquini, I., Iannitelli, M., & Allegorico, C. (2021). Algorithmically Exploiting the Knowledge Accumulated in Textual Domains for Technical Support. In *PHM Society European Conference* (Vol. 6, pp. 12–12).

Peshave, A., Aggour, K., Ali, A., Mulwad, V., Dixit, S., & Saxena, A. (2022). Evaluating Vector Representations of Short Text Data for Automating Recommendations of Maintenance Cases. In *Proceedings of the Annual Conference of the PHM Society* (Vol. 14).

Picard, C., Edwards, K. M., Doris, A. C., Man, B., Giannone, G., Alam, M. F., & Ahmed, F. (2023). From Concept to Manufacturing: Evaluating Vision-Language Models for Engineering Design. *arXiv preprint arXiv:2311.12668*.

Pires, F., Leitão, P., Moreira, A. P., & Ahmad, B. (2023). Reinforcement learning based trustworthy recommendation model for digital twin-driven decision-support in manufacturing systems. *Computers in Industry*, *148*, 103884.

Reimers, N., & Gurevych, I. (2019, November). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In K. Inui, J. Jiang, V. Ng, & X. Wan (Eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (pp. 3982–3992). Hong Kong, China: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/D19-1410` doi: 10.18653/v1/D19-1410

Ren, Z., Zhan, Y., Yu, B., Ding, L., & Tao, D. (2024). Healthcare Copilot: Eliciting the Power of General LLMs for Medical Consultation. *arXiv preprint arXiv:2402.13408*.

Rovaglio, M., Calder, R., & Richmond, P. (2012). Bridging the Experience Gap - How do we migrate Skills and Knowledge between the Generations? . In *Computer Aided Chemical Engineering* (Vol. 30, pp. 1407–1411). Elsevier.

Sala, R., Pirola, F., Dovere, E., & Cavalieri, S. (2019). A dual perspective workflow to improve data collection for maintenance delivery: an industrial case study. In *Advances in Production Management Systems. Production Management for the Factory of the Future: IFIP WG 5.7 International Conference, APMS 2019, Austin, TX, USA, September 1–5, 2019, Proceedings, Part I* (pp. 485–492).

Sala, R., Pirola, F., Pezzotta, G., & Cavalieri, S. (2022). NLP-based insights discovery for industrial asset and service improvement: an analysis of maintenance reports. *IFAC-PapersOnLine*, *55*(2), 522–527.

Sala, R., Pirola, F., Pezzotta, G., & Cavalieri, S. (2023). Improvement of maintenance-based Product-Service System offering through field data: a case study. *Production & Manufacturing Research*, *11*(1), 2278313.

Sentence Transformers. (2021). *all-MiniLM-L6-v2: Sentence transformers model.* `https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2`.

Sexton, T., Brundage, M. P., Hoffman, M., & Morris, K. C. (2017). Hybrid datafication of maintenance logs from AI-assisted human tags. In *2017 IEEE International Conference on Big Data* (pp. 1769–1777).

Sexton, T., Hodkiewicz, M., & Brundage, M. P. (2019). Categorization Errors for Data Entry in Maintenance Work-Orders. In *Proceedings of the Annual Conference of the PHM Society* (Vol. 11).

Shankar, S., Zamfirescu-Pereira, J., Hartmann, B., Parameswaran, A. G., & Arawjo, I. (2024). Who Validates the Validators? Aligning LLM-Assisted Evaluation of LLM Outputs with Human Preferences. *arXiv preprint arXiv:2404.12272*.

Silverstein, M. (2008). Meeting the challenges of an aging workforce. *American Journal of Industrial Medicine*, *51*(4), 269–280.

Sordoni, A., Yuan, E., Côté, M.-A., Pereira, M., Trischler, A., Xiao, Z., ... Le Roux, N. (2024). Joint Prompt Optimization of Stacked LLMs using Variational Inference . *Advances in Neural Information Processing Systems*, *36*.

Stewart, M., Hodkiewicz, M., & Li, S. (2023). Large language models for failure mode classification: an investigation. *arXiv preprint arXiv:2309.08181*.

Stewart, M., Hodkiewicz, M., Liu, W., & French, T. (2022). MWO2KG and Echidna: Constructing and exploring knowledge graphs from maintenance data. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 1748006X221131128.

Talebirad, Y., & Nadiri, A. (2023). Multi-Agent Collaboration: Harnessing the Power of Intelligent LLM Agents. *arXiv preprint arXiv:2306.03314*.

Thirunavukarasu, A. J., Ting, D. S. J., Elangovan, K., Gutierrez, L., Tan, T. F., & Ting, D. S. W. (2023). Large language models in medicine. *Nature Medicine*, *29*(8), 1930–1940.

Trilla, A., Mijatovic, N., & Vilasis-Cardona, X. (2022). Towards Learning Causal Representations of Technical Word Embeddings for Smart Troubleshooting. *International Journal of Prognostics and Health Management*, *13*(2).

Trilla, A., Yiboe, O., Mijatovic, N., & Vitrià, J. (2024). Industrial-grade smart troubleshooting through causal technical language processing: a proof of concept. *arXiv preprint arXiv:2407.20700*.

Varshney, T. (2023, Nov 30). *Introduction to LLM Agents.* `https://developer.nvidia.com/blog/introduction-to-llm-agents/`.

Vidyaratne, L., Lee, X. Y., Kumar, A., Watanabe, T., Farahat, A., & Gupta, C. (2024). Generating troubleshooting trees for industrial equipment using large language models (llm). In *2024 ieee international conference on prognostics and health management (icphm)* (pp. 116–

125).

Wang, M., Xu, X., Yue, Q., & Wang, Y. (2021). A Comprehensive Survey and Experimental Comparison of Graph-Based Approximate Nearest Neighbor Search. *Proceedings of the VLDB Endowment*, *14*(11), 1964–1978.

Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., & Zhou, M. (2020). MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. *Advances in Neural Information Processing Systems*, *33*, 5776–5788.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., & Choi, Y. (2019, July). HellaSwag: Can a Machine Really Finish Your Sentence? In A. Korhonen, D. Traum, & L. Màrquez (Eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 4791–4800). Florence, Italy: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/P19-1472` doi: 10.18653/v1/P19-1472

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., ... others (2024). Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. *Advances in Neural Information Processing Systems*, *36*.

BIOGRAPHIES

Sarah Lukens is a Data Science Fellow at LMI. Her interests are focused on data-driven modeling for reliability applications by combining modern data science techniques with current industry performance data. This work involves analyzing asset maintenance data and creating statistical models that support asset performance management (APM) work processes using components from natural language processing, machine learning, and reliability engineering. Sarah completed her Ph.D. in mathematics in 2010 from Tulane University with focus on scientific computing and numerical analysis. Sarah is a Certified Maintenance and Reliability Professional (CMRP).

Lucas H. McCabe is a Data Science Fellow at LMI, where he focuses on problems in machine learning, graph analytics, and computational social science. He received his B.A. in Mathematics and Computer Science from Rutgers University (2018), M.S. in Applied and Computational Mathematics from Johns Hopkins University (2020), and is a Ph.D. candidate in Computer Science at The George Washington University. He has been named a DARPA Riser, Luminary Awardee (LMI), and Bernstein Scholar (Rutgers Institute for Quantitative Biomedicine).

Joshua Gen is a Data Scientist at LMI focusing primarily on Advanced Analytics and AI. Josh has a Master's in Data Science from the University of Virginia, as well as a Bachelor's in Statistics and Economics also from the University of Virginia. Josh is passionate about applying recent developments in AI, particularly LLMs, to assist with complex processes. Josh works primarily on LMI's AI products and tools helping to develop and deploy state-of-the-art machine learning models in secure environments. He has assisted in creating AI products leveraging LLM's and other data science techniques now in use by multiple government entities. He is also a developer in LMI's Forge research group, contributing to data science prototypes.

Asma Ali a Senior Staff Analytics Engineer with GE Vernova where she is a technical team lead for the Analytics & Data Science team. Asma earned a Bachelors Degree in Biomedical Engineering from University of Connecticut and a Masters Degree in Mechanical Engineering from University of IL while working full-time. Asma has 15+ years of industry experience focusing on software solution design, new product development and deploying process improvements, etc. Over the years, she worked closely with industrial domains like Power Generation, Oil & Gas, Mining & Metals, Aviation, Chemicals, Automotive, and more. Asma developed several dozen Digital Twins for Industrial Assets and Analytics for the APM product line. Currently, Asma is leading the efforts on Work Order Automation utilizing technical language processing for GE Vernova and providing GE Research with Power domain-related industrial expertise.