Agentic Anomaly Detection for Shipping

Alexander Timms *
Department of Bioengineering
Imperial College London
United Kingdom

Abigail Langbridge *
Dyson School of Design Engineering
Imperial College London
United Kingdom

Fearghal O'Donncha
AI for Sustainable Industry
IBM Research Europe
Dublin, Ireland
feardonn@ie.ibm.com

Abstract

Operational decision making in the shipping industry exemplifies a real-world challenge that extends beyond single tasks and static conditions. We introduce an agentic LLM system designed to enhance anomaly detection (AD) and maintenance processes within this highly dynamic domain, involving multi-persona stakeholder interactions. The method leverages the intrinsic knowledge and reasoning abilities of LLMs, augmented by a suite of external tools to reason on the severity of anomalies detected by an out-of-the-box AD tool. Our approach achieves this by considering environmental factors, interconnected system dynamics extracted from a knowledge graph, and broader operational parameters. Evaluations on large-scale shipping data demonstrate that our method effectively reasons about multimodal data, distilling complex system dynamics into operational insights. This represents the first agentic application in an open-world maritime environment.

1 Introduction

The global shipping industry, comprising over 100,000 vessels facilitates the transport of 90% of the world's goods, yet it also contributes 3% of global greenhouse gas emissions, which have risen by 20% in the past decade [2, 18]. With the International Maritime Organization (IMO) aiming to reduce emissions by 20% by 2030 and 70% by 2040 [10], enhancing shipping efficiency has become essential for achieving environmental goals.

However, these ambitions are complicated by an ageing global fleet, where over half of the vessels are older than 15 years, making them difficult to retrofit yet premature for scrapping [2]. Amid these constraints, digitalization and AI offer transformative potential in areas such as route optimization, fuel efficiency, and predictive maintenance [6]. While platforms like Maersk's TradeLens and Rolls-Royce's autonomous ship initiatives have made strides, the industry's complexity and regulatory hurdles demand more scalable, integrated solutions accessible to a broad range of industry persona [5, 13].

Key priorities include, seamless data integration across fragmented systems, real-time monitoring for predictive maintenance, scalability for diverse fleets, and transparent adherence to regulatory compliance across global frameworks. Additionally, solutions must provide automation and configurability to address the varied needs of industry personas without relying on specialised expertise.

*Work completed while at IBM Research Europe

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

2 Related work

Reinforcement Learning (RL) has achieved outstanding success in domains like Atari [11], Chess, and Go [16], typically within single-agent environments. However, industrial systems require agents to interact with multiple entities and understand the broader consequences of their actions on the overall system, highlighting the need for more complex, multi-agent approaches.

Recent research demonstrates the power of augmenting LLMs with external tools, such as retrieval augmentation [24], mathematical reasoning tools [8, 14], and code interpreters [3, 20]. These advancements enable LLMs to dynamically select and configure external tools to solve more complex reasoning tasks, as seen in systems like HuggingGPT [15], which uses language as a flexible interface to leverage multiple tools.

Further developments in planning algorithms integrate LLMs for enhanced decision-making in industrial applications. Auto-GPT [21] and ReAct [22] automate task planning and problem-solving by interacting with external systems, while Language Agent Tree Search (LATS) [23] combines LLM reasoning, acting, and planning, incorporating feedback from dynamic environments. These techniques represent significant progress in using LLMs to handle the complexity and variability of industrial processes.

3 Methods

Our solution leverages agent-based LLMs to enhance decision-making and planning for industrial asset management and maintenance. This architecture incorporates a reasoning and planning agent that decomposes tasks into subtasks and coordinates various specialised agent tools to execute these subtasks. An LLM-driven agentic approach in shipping enables diverse personas—such as fleet owners, ship owners, captains, engineers, and technicians—to seamlessly explore various aspects of the ship's systems and access tailored information.

3.1 Agentic Tools for Industrial Asset Management

We developed a suite of tools specifically designed for industrial applications, enhancing agentic decision-making by providing a comprehensive view of real-time ship operations along with contextual insights. These tools integrate structured information from a domain-specific knowledge graph, real-time sensor data, and technical specifications, including potential failure data. By leveraging LangGraph[2], the framework's modular and adaptable design allows it to meet diverse operational requirements and significantly improve decision-making. We demonstrate the method on real-world shipping data, showcasing its practical applicability and scalability.

Figure 1 illustrates the architecture of the system, integrating agentic reasoning with a domain-specific knowledge graph and context data to identify relevant components and sensor data to monitor performance. The system utilises LLM-as-a-judge capabilities to evaluate tool selections and ensure that the chosen tools align with the context and intent of the user query, optimising decision-making and anomaly detection.

3.1.1 Knowledge Graph Exploration

Knowledge graphs (KG) help integrate diverse data sources, provide context, and support applications such as semantic data integration, knowledge discovery, and predictive analytics and forecasting.

Our implementation utilised an Apollo Server connected to a Neo4j database, built around a predefined GraphQL schema [19]. Given the broad scope of applications pertinent to ship operations, the KG functions as an operational metamodel or semantic layer, bridging actual system operations with higher-level use cases supported by the data. This includes ship performance metrics, operational requirements, performance benchmarks, environmental conditions, and the corresponding setpoints required to achieve operational objectives. Additionally, it encompasses optimization strategies such as weather routing and port arrival management

While the KG can represent many aspects of ship operations, efficiencies, and predictive analytics (e.g. model management), we focused on the relationships between the vessel, its components, and

[2] `https://langchain-ai.github.io/langgraph/`

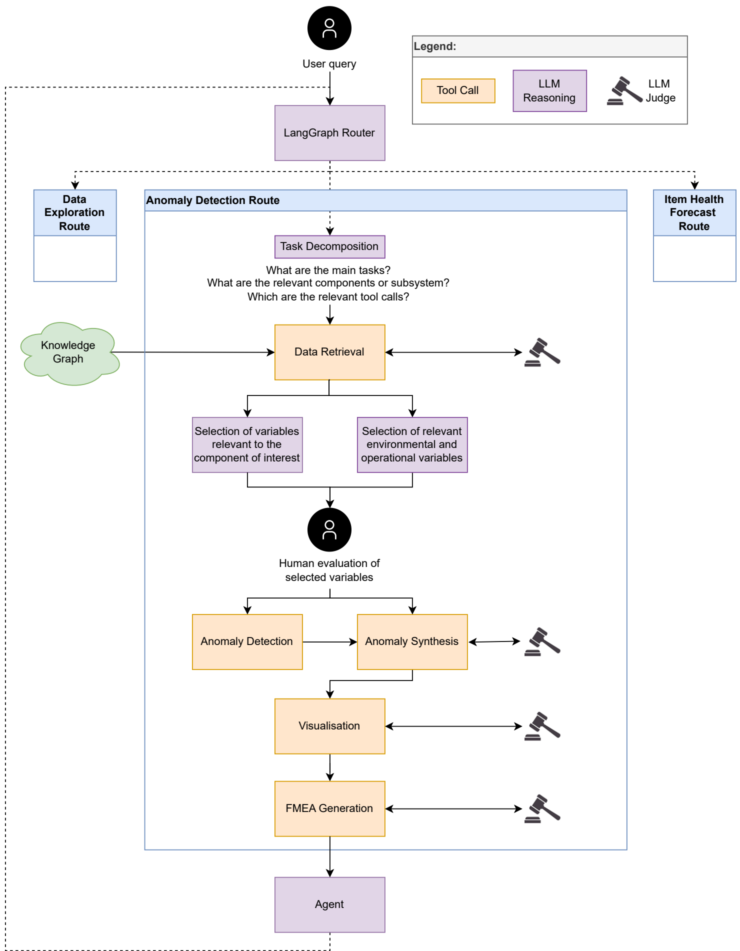


Figure 1: Flow diagram showing the user interaction with the agentic system, highlighting the Anomaly Detection Route. The router is used for path selection.

associated operational variables. Components such as the main engine, auxiliary engine, and cargo system are linked to variables like temperature, pressure, and performance metrics, creating a network that reflects the ship's functional state. Importantly in the context of anomaly detection, it allows greater visibility into how components can be influenced by exogenous factors such as sea-state or navigation patterns.

3.1.2 Anomaly Detection for Industrial Asset Management

Anomaly detection for industrial asset management is a challenging problem: operating conditions extremely dynamic due to fluctuating loads and environmental factors, there can be complex interactions between subsystems and components, while industrial labelled data are scarce. High false positive rates can lead to high maintenance costs and undermine confidence in the condition-based approach. Further, interpreting the results of many anomaly detection methods is non-trivial, and often requires operator input and judgement to manually flag alerts.

In this work, we extend the approach proposed by [7]. This online, unsupervised anomaly detection approach leverages an efficient formulation of the optimal transport (OT) problem in one dimension to detect anomalies in noisy, seasonal time series data. A key benefit of this approach is the built-in counterfactual explanations for each anomaly, which are derived from the optimal transport plan.

Based on these explanations, we modify the binary anomaly prediction in [7] (OT-AD) to leverage the counterfactual explanations explicitly. In lieu of labels to tune against, we set the parameters for OT-AD to $n_{\text{ref}} = 36$, $n_{\text{test}} = 24$, $n_{\text{buf}} = 48$ for all data. Considering the factual point y and its counterfactual $\hat{y}$, we can approximate the severity of the anomaly by considering the percentage change between these two, $\gamma = \frac{y-\hat{y}}{y}$. Both the magnitude and direction of γ can be used to reason about each anomaly, boosting the interpretability of the method.

3.1.3 Anomaly Synthesis Based on Intrinsic Knowledge

Reflecting the interconnected nature of many industrial systems, anomalies are often detected in multiple data streams concurrently. Often, an anomaly that originates in the environment, or a component of some distant subsystem (the *source*) propagates through many other downstream components. However, the relationships that govern the spread of anomalies through components are not made explicit in the KG. As such, we leverage the internal knowledge and reasoning abilities of the LLM to produce candidate sources for a given subsystem based on available data. We then present these sources to the operator for human evaluation.

Once the sources have been validated, we conduct an anomaly detection step on each source as in Section 3.1.2. Now, for each anomaly in the component of interest, we can compare the score γ to the coincident anomaly score in each source γ_{source}. If $\gamma_{\text{source}} = 0$, i.e. there is no anomaly in the source, we can attribute all of γ to the target anomaly. Similarly, for $|\gamma_{\text{source}}| < |\gamma|$, then, while the source perturbation may have influenced the component anomaly, we cannot attribute all of the anomaly to the source. In this case, we adjust the anomaly score $\gamma = \gamma - \gamma_{\text{source}}$. Conversely, if $|\gamma_{\text{source}}| \geq |\gamma|$, then we can reason that the component of interest is only anomalous due to the perturbation in the source, and label this period non-anomalous, i.e. $\gamma = 0$.

3.1.4 Failure Modes and Effects Analysis (FMEA) Tool

FMEA is a systematic method used to identify and assess potential failure modes within a system, process, or product. It focuses on understanding the causes and effects of each failure mode and evaluating their potential impact [17]. FMEA helps prioritise risks based on factors such as severity, occurrence, and detectability, enabling preventive measures to be taken. Typical information included in an FMEA consists of the component or system location, potential failure modes, causes of failure, severity of the failure, and the resulting effects of the failure on the system [17]. In industrial sectors like shipping, Failure Mode and Effects Analysis (FMEA) is often mandated by classification societies, including the American Bureau of Shipping [1]. Alternatively, when an FMEA for an asset or component doesn't exist it can be generated using LLMs [9].

3.1.5 LLM as a Judge

As the number of tools available to an LLM agent increases, discerning the correct tools and the sequence of their usage becomes more complex. Classical methods rely on orchestration frameworks like LangGraph, utilising the LLM's pretrained knowledge to grasp the context of the query. Additionally, the LLM can be augmented with predefined rules or logic that map specific query types to corresponding tools, streamlining tool selection and usage.

However, LLMs continue to face challenges in accurately selecting and invoking the appropriate external tools for specific tasks [15]. We implemented an LLM-as-a-judge agent to evaluate tool usage in relation to the user query and the associated subtasks. The agent reviewed the LLM's tool calls, assessing whether the completed actions aligned with the task's objectives or if additional tool invocations were necessary to fully meet the requirements.

4 Experiments

We leveraged real-world data from a 9-year-old, 330-meter crude oil tanker, collected continuously over one year from approximately 400 sensors. These sensors monitored critical ship components such as the main engine, electrical systems, navigational data, weather, cargo holds, and the fuel system. The relationships between components and their sensor data were represented by the knowledge graph [19]. Additionally, external factors like weather conditions, ship draught, and speed were considered for their influence on overall performance. The Llama 3 70B Instruct model was used for all experiments.

Our experimental setup aimed to evaluate the capacity of agentic systems to handle complex queries about ship components and subsystems. Through workshops with stakeholders from four major shipping companies, we identified over 40 user stories highlighting the industry's digitalisation requirements across multiple stakeholders. These include tools or methods that can help stakeholders detect machinery degradation, predict performance metrics under varying operational and environmental conditions, standardise and integrate data from diverse sources, and assist with maintenance scheduling and cargo handling.

The user stories underscored the need for agentic solutions capable of accurately interpreting user intent and surfacing relevant information. Ship-specific and domain-specific data integration is essential, reinforcing the importance of external tools. The clear distinction in user needs further supports the use of a high-level routing mechanism like LangGraph to analyse query types and select the appropriate tools or paths.

Agentic Reasoning for Shipping

User Query: Can you help me understand anomalies in the ship's engine over the past year. I would like to view anomalies and receive recommendations or suggestions to address these anomalies as part of a condition based maintenance strategy

Reasoning: The user query requires multiple tool calls to fulfill their request. They want to detect and graph anomalies for the main engine, and if relevant, generate a maintenance strategy to address anomalies. To accomplish this, we need to call the **get_data_knowledge_graph**, **get_anomalies**, and **get_anomalies_graph** tools to process anomalies.

```
Tool Call: {
  'name': 'get_data_knowledge_graph',   # tool name
  'args': {
    'component': 'main engine',  # user query relates to engine
    'months_of_data': 12 # user query past year of data
  }}
```

```
Tool Call:  {'name': 'get_anomalies_graph', 'args': {
 'sensors': ['me_tc1_rpm'],
 'filter_anomaly_by_component':['sog','er_ambient_pressure']}}
```

Reasoning: The user has asked to detect and graph anomalies for the main engine and then generate a work order for any anomalies. The tools calls that have been made are **get_data_kg** and **get_anomalies_graph**. We must now use the information on the detected anomalies that were filtered by speed over ground and engine room ambient pressure to recommend a maintenance strategy. Leveraging industry best practices, I will generate a work order providing details on the anomaly and maintenance strategy. Additional tool calls are required to fulfill the user's request. I need to call the **get_fmea** and **create_work_order** tools to evaluate anomalies and generate work order

```
Tool Call:  {'name': 'get_fmea', 'args': {
                'component': 'main engine' }}
```

```
Tool Call: {'name': 'create_work_order', 'args': {
 'component': 'main engine', 'fmea': [generated_fmea],
 'anomalies': [generated_anomalies]}},
```

Reasoning: The tools called have already answered the user's query. The **get_anomalies_graph** tool has generated an anomaly graph for the main engine filtered by speed over ground and engine room parameters, and the **create_work_order** tool has generated a work order for the anomalies found. No more tool calls are required.

Route to Agent: All steps have been completed, now we need to summarise activities to the user. A visualisation of the anomalies have been prepared for the end users and associated work orders were created for situations where detected anomalies were not explainable by external factors such as environmental conditions or speed of the vessel

While we are unable to share the data used in our experiments, the results presented here offer valuable insights for the domain. Specifically, they highlight the effective use of agentic frameworks and external tools in complex, open-world applications, where user requirements involve multi-persona interactions and multiple pathways to solutions. These findings can help inform the design and implementation of similar systems, emphasizing the adaptability of agentic approaches in meeting diverse and dynamic user needs.

5 Results

The implementation of the agentic framework, orchestrated through LangGraph, demonstrated strong capabilities in navigating multiple paths tailored to different user requirements and contexts. Each path was designed to handle specific tasks, such as data exploration, asset health forecasting, and anomaly detection in ship systems. These are illustrated in Figure 1.

The anomaly detection path was perhaps the most complex, requiring the agentic framework to detect, visualise, and report anomalous conditions within ship systems. We assessed the effectiveness of the agentic framework in accurately resolving the query's context and intent to meet the user's needs. This process involved identifying relevant ship components or subsystems, pertinent sensor data, and reasoning on the potential impact of the detected anomalies. Of particular interest were the system's reasoning capabilities, especially in selecting and utilising the correct tools while factoring in key operational considerations.

5.1 Agentic Workflow

The agent interprets the query by analysing intent, relevant components, and information on any specific data or trends to consider. The text box on the previous page summarises the reasoning trajectories. For the identified component, the agent is aided by a knowledge graph detailing entity relationships, alongside external data informed by the LLM's pretrained knowledge. This highlights the utility of combining structured knowledge with insight from the LLM on operational aspects. Subsequent tool calls relate to aspects such as anomaly detection, visualisation, generation of work orders or maintenance strategies, and integration of external knowledge from an FMEA system. At each tool call, an LLM-as-a-judge agent assess the status and reasons whether additional tool calls are required to address user query.

LLM Response

To monitor the performance of the main engine, I'll select three sensors that provide a comprehensive view of the engine's operation, considering both environmental and operational data. Here's my revised selection:

Step 1: Identify the most critical engine performance metric
I'll choose a sensor that directly measures the main engine's performance. Let's select:

- **ME Turbocharger 1 - RPM**: This sensor provides a direct measurement of the main engine's speed, which is a critical indicator of its performance.

Step 2: Consider operational data that impacts engine performance
Operational data, such as the ship's speed and load, can significantly impact the main engine's performance. Let's select a sensor that provides insight into these conditions:

- **Speed Over Ground**: This sensor measures the ship's speed, which affects the main engine's load and performance.

Step 3: Choose a sensor that provides environmental context
Environmental conditions, such as ambient pressure, can also impact the main engine's performance. Let's select a sensor that provides insight into these conditions:

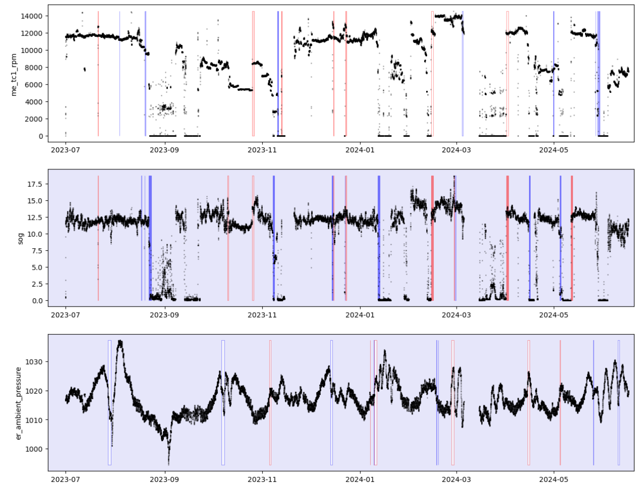
- **E/R Ambient Pressure**: This sensor measures the ambient pressure in the engine room, which can affect the engine's efficiency and performance.

These three sensors provide a well-rounded view of the main engine's performance, considering both operational and environmental factors. By monitoring these sensors, you can gain insights into the engine's speed, load, and efficiency, as well as the environmental conditions that affect its performance.

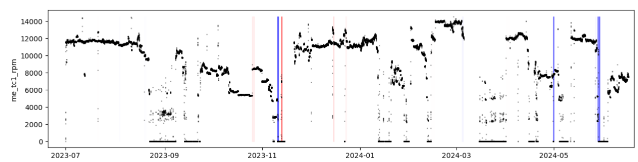
5.2 Knowledge Graph Exploration

Knowledge graphs enhance LLMs by complementing their pre-trained knowledge with structured, factual external data sources [12]. This is particularly beneficial when reasoning about domain-specific applications or when handling real-time data is critical [4]. In our implementation, we focused on the subject-predicate-object structure, commonly referred to as triples, which includes entities such as the vessel component (e.g., auxiliary engine), the variable being measured (e.g., auxiliary engine power output), and the relationship between them (e.g., RELATED_TO).

Beyond direct relationships between ship components and their measurements, vessel performance is also influenced by external factors like speed or weather conditions. These indirect influences are not typically encoded in an entity-relationship knowledge graph and require an LLM agent to interpret relevant exogenous factors. This approach allows the LLM agent to not only identify all variables directly related to ship components through the knowledge graph but also query and reason



(a) Anomalies detected for `me_tc1_rpm` alongside the environmental and operational sources identified in by the LLM, `sog` and `er_ambient_pressure` (shaded lavender).



(b) Anomalies for `me_tc1_rpm` filtered by the two anomaly sources, above.

Figure 2: Anomalies for `me_tc1_rpm` before and after the anomaly synthesis (filtering) step described in Section 3.1.3. Anomalous periods are highlighted in red or blue based on their direction, with paler colours corresponding to lower values of γ.

across other operational and environmental variables. The LLM's reasoning ability helps identify a subset of variables pertinent to the specific component being analyzed. For example, the LLM response (presented on the previous page) identifies the relevant sensor data for monitoring main engine performance, considering both direct measurements and operational contexts.

5.3 Anomaly Detection and Synthesis

Figure 2a shows the anomalies detected by the OT-AD tool [7] for the main engine turbocharger component, along with the anomalies associated with two potential sources suggested by the LLM. It's clear from these results that a number of anomalies in the turbocharger RPM are due to variations in the vessel's speed, resulting in a large number of false positives.

Figure 2b shows the result of our synthesis operation, where the severity of detected anomalies is moderated by concurrent anomalies in the source features. This demonstrates the effectiveness of the method for filtering anomalies due to external factors, reducing the overhead required by human operators.

5.4 Work Order Generation

A key challenge in industrial asset maintenance is distinguishing between spurious anomalies and events requiring human attention. Using the anomaly detection procedure outlined in Section 5.3, we filtered out insignificant events, allowing the system to focus on extracting meaningful insights. This process involved multiple tool calls: first, to identify the relevant Failure Mode and Effects Analysis (FMEA) for the specific ship component (Section 3.1.4), and second, to evaluate the potential root causes of the anomalies based on potential failures. This multi-step approach ensured that each anomaly was contextualised, enabling informed decisions on whether manual intervention was necessary. Figure 3 presents an example snapshot highlighting information surfaced to the user about

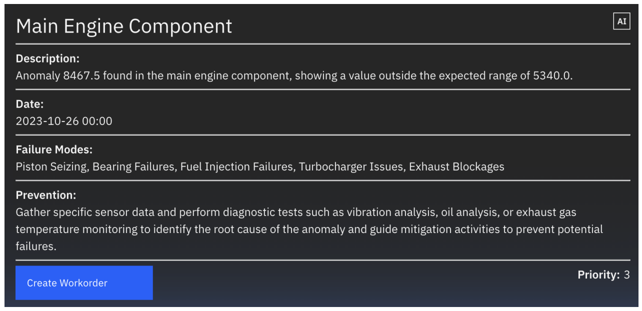


Figure 3: An example of a work order generated based on a detected anomaly. The agent reasons on the severity of the anomaly and uses pretrained knowledge to evaluate potential root causes or associated failure modes.

the asset, anomaly, and potential failure modes. It details a description of the anomaly and reasoning on which potential failure modes this could relate to.

6 Discussion

Agentic workflows for anomaly detection in shipping face key limitations, particularly around explainability and trust. While LLMs can automate tool calls, they often struggle to provide clear reasoning behind decisions, which can reduce industry confidence. This is particularly true in complex, interconnected systems such as shipping subject to multiple regulatory obligations. Additionally, the LLM's ability to make correct tool calls and pass accurate parameters is not guaranteed, potentially leading to errors in anomaly detection. The reliance on an LLM's internal knowledge to relate ship components further complicates the process, and while retrieval-augmented generation (RAG) can help, it still requires improvement. Enhancing the robustness of this approach will depend on refining decision-making processes and integrating real-time feedback to adapt to changing operational environments. Enhancing agentic observability is particularly critical

6.1 Broader Impacts (Societal)

Decarbonising the shipping industry is a critical societal goal, with efficiency improvements through digital twin technologies playing a key role in achieving it. However, the inherent complexity of ship systems, vessel heterogeneity, and the diverse range of operational challenges make it difficult to scale digital twin solutions, often requiring significant in-house data science expertise. Agentic LLMs, when augmented with external tools, can significantly accelerate progress towards these objectives by automating and streamlining decision-making processes across various ship types and operational scenarios, making digital twin capabilities more scalable and accessible through a generic natural language interface.

7 Conclusion

This paper presented an agentic LLM-based framework for anomaly detection in shipping, leveraging real-time data integration, knowledge graphs, and external tools to support complex decision-making in industrial asset management. By evaluating on real-world data, we demonstrated the ability to streamline the anomaly detection process, and provide contextual insights through dynamic tool selection. This approach offers a scalable solution to the challenges of digitalisation in the maritime sector, addressing issues such as machinery degradation, data integration, and operational efficiency. Future work will focus on enhancing explainability, improving tool orchestration, and integrating additional predictive analytics to further support decarbonisation and automation efforts in shipping.

Acknowledgements

Research described in this paper has received funding from the European Union's Horizon Europe Research and Innovation program under grant agreement no. 101056799.

References

[1] Guidance notes on failure mode and effects analysis (FMEA) for classification. Technical report, American Bureau of Shipping, ABS Plaza, 1701 City Plaza Drive, Spring, TX 77389 USA, 2015.

[2] Review of maritime transport 2023. Annual report, United Nations Conference on Trade and Development, New York, 2023. Available online: `https://unctad.org/publication/review-maritime-transport-2023`.

[3] L. Gao, A. Madaan, S. Zhou, U. Alon, P. Liu, Y. Yang, J. Callan, and G. Neubig. Pal: Program-aided language models. In *International Conference on Machine Learning*, pages 10764–10799. PMLR, 2023.

[4] L. Hu, Z. Liu, Z. Zhao, L. Hou, L. Nie, and J. Li. A survey of knowledge enhanced pre-trained language models. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[5] M. Jovanovic, N. Kostić, I. M. Sebastian, and T. Sedej. Managing a blockchain-based platform ecosystem for industry-wide adoption: The case of tradelens. *Technological Forecasting and Social Change*, 184:121981, 2022.

[6] T. Katsoulakos, G. Tsiochantari, F. O'Donncha, E. Kaklamanis, A. Maccari, and M. Mucharski. Shipping digital twin landscape. In *State-of-the-Art Digital Twin Applications for Shipping Sector Decarbonization*, pages 1–25. IGI Global, 2024.

[7] A. Langbridge, F. O'Donncha, J. Rayfield, and B. Eck. Optimal transport for efficient, unsupervised anomaly detection on industrial data. In *IEEE International Conference on Big Data (Big Data)*, 2024.

[8] P. Lu, B. Peng, H. Cheng, M. Galley, K.-W. Chang, Y. N. Wu, S.-C. Zhu, and J. Gao. Chameleon: Plug-and-play compositional reasoning with large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

[9] K. Lynch, F. Lorenzi, J. D. Sheehan, D. Kabakci-Zorlu, and B. Eck. FMEA builder: Expert guided text generation for equipment maintenance. In *AI for Critical Infrastructure Workshop@ IJCAI-24*, 2024.

[10] R. MEPC. 2023 IMO strategy on reduction of GHG emissions from ships. 2023.

[11] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[12] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, and X. Wu. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[13] Rolls-Royce Marine. Autonomous ships. the next step. *Rolls-Royce plc*, 2016.

[14] T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, L. Zettlemoyer, N. Cancedda, and T. Scialom. Toolformer: language models can teach themselves to use tools. 2023. *arXiv preprint arXiv:2302.04761*, 2023.

[15] Y. Shen, K. Song, X. Tan, D. Li, W. Lu, and Y. Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36, 2024.

[16] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

[17] D. H. Stamatis. *Failure mode and effect analysis*. Quality Press, 2003.

[18] M. R. Stuchtey, A. Vincent, A. Merkl, M. Bucher, P. M. Haugan, J. Lubchenco, and M. E. Pangestu. Ocean solutions that benefit people, nature and the economy. In *The Blue Compendium: From Knowledge to Action for a Sustainable Ocean Economy*, pages 783–906. Springer, 2023.

[19] T. Tsaousis. Deliverable 2.1: Transferable DT4GS architecture. Technical report, Inlecom, 2023.

[20] X. Wang, S. Li, and H. Ji. Code4struct: Code generation for few-shot structured prediction from natural language. *arXiv preprint arXiv:2210.12810*, 3, 2022.

[21] H. Yang, S. Yue, and Y. He. Auto-gpt for online decision making: Benchmarks and additional opinions. *arXiv preprint arXiv:2306.02224*, 2023.

[22] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

[23] A. Zhou, K. Yan, M. Shlapentokh-Rothman, H. Wang, and Y.-X. Wang. Language agent tree search unifies reasoning acting and planning in language models. *arXiv preprint arXiv:2310.04406*, 2023.

[24] Y. Zhuang, Y. Yu, K. Wang, H. Sun, and C. Zhang. Toolqa: A dataset for llm question answering with external tools. *Advances in Neural Information Processing Systems*, 36:50117–50143, 2023.