Human Aspects of Advanced Manufacturing, Production Management and Process Control, Vol. 184, 2025, 102–111
https://doi.org/10.54941/ahfe1006454

MetaBPL: Fault Detection in Business Logic Systems

Gregorios A. Katsios[1], Diego Manzanas Lopez[2], Benjamin Ryjikov[1], Samuel Merten[1], and Daniel Balasubramanian[2]

[1]Leidos Inc, Reston, VA 20190, USA
[2]Vanderbilt University, Nashville, TN 37235, USA

ABSTRACT

Ensuring the integrity and efficiency of business process logic systems is critical for industries such as manufacturing, infrastructure, and logistics. Even minor vulnerabilities can lead to costly operational disruptions or security breaches. While traditional fault detection methods rely on statistical quality assurance and auditing, they are reactive and time-consuming, and even their diagnosis requires downtime. To address these challenges, we introduce MetaBPL, an automated framework that leverages a Large Language Model (LLM) within a Retrieval-Augmented Generation (RAG) architecture for proactive fault detection and analysis. MetaBPL systematically identifies discrepancies, assesses their impact, and generates corrective recommendations. By improving the precision and scalability of fault detection, this approach enhances business process security and operational resilience while reducing reliance on extensive domain expertise.

Keywords: Human systems integration, Systems engineering, Systems modelling language

INTRODUCTION

Business logic systems form the backbone of operations for large-scale companies and defense-critical workflows across industries such as manufacturing, infrastructure, and logistics. These systems orchestrate processes (ranging from production planning and quality management to auditing and regulatory compliance) by integrating a diverse set of software systems and natural language artifacts. A company's Quality Management (QM) system often spans multiple platforms, such as Manufacturing Execution Systems (MES) for process modelling, Enterprise Resource Planning (ERP), for material tracking, and a combination of digital and paper-based logs for QM activities. Additionally, these processes are often governed by internal protocols and external standards, such as ISO-9000, further complicating the landscape. The heterogeneity of these artifacts, coupled with unformalized domain knowledge and implicit human requirements, presents significant challenges for ensuring the integrity and security of business operations.

Vulnerabilities and faults in these systems can have widespread repercussions. A single product recall might exceed ten million USD in losses, while unplanned downtime in a manufacturing line could cost upwards

of one million dollars per hour. Although statistical quality assurance and auditing can help prevent future faults, incorporating real-time, large-scale analysis could provide even greater insights and responsiveness. Moreover, the manual identification and rectification of discrepancies in such complex environments require a level of domain expertise that is not always readily available. As a result, there is an urgent need for proactive and automated solutions that can identify and analyze faults in business logic systems before they escalate into major operational failures.

To address these challenges, we propose MetaBPL, a framework that integrates a state-of-the-art LLM agent with a RAG architecture specifically tailored for fault detection in business processes. MetaBPL operates by first retrieving relevant context from a vector database, comprising artifacts such as regulatory excerpts, industry standards, and internal documentation. The system then dynamically generates fault detection queries based on business logic-specific prompts. This dual approach allows the system not only to identify discrepancies in complex rule hierarchies but also to offer detailed fault analyses and corrective recommendations that incorporate both formalized domain knowledge and elements of human intuition.

To facilitate a robust evaluation, we design an automated pipeline that generates organizational documentation (such as Travelers, Work Instructions, Bills of Material, Standard Operating Procedures, etc.) typically used in internal business operations. Such documentation is often proprietary and therefore difficult to access and share for evaluation purposes. Our generative pipeline leverages a Large Language Model (LLM) to automatically produce these documents based on a set of predetermined definitions and properties.

To ensure the generated content remains consistent across documents, we implement a feedback loop that revises the generated content and avoids producing material that conflicts with previously generated outputs. Using this approach, we built a small-scale benchmark comprising of seven sets of products, with each set containing ten organizational documents (such as Bill of Materials (BOM), Procurement and Supplier Management Policy, Work Instructions, Assembly Line Processes, Traveler Documents, Quality Assurance (QA) Policy, etc.). For each set, we deliberately inject faults, selected from our established fault taxonomy, to create a ground truth for systematic evaluation.

Our evaluation is based on three metrics: fault detection rate (percentage of total faults found by MetaBPL), the correctness of LLM-generated responses (balancing completeness and hallucination), answer accuracy with respect to provided context, and the context retrieval quality that measures the accuracy and conciseness of the retrieved context. Our evaluation, conducted on a diverse corpus of synthetic organizational documents, demonstrates the scalability and precision of our approach.

RELATED WORK

Fault identification and prevention is an effort that spans the entire lifespan of a manufacturing process. Even stable, tested and well-designed systems

can be disrupted by unanticipated events or interactions (Brecher et al., 2009). Our approach grounds itself in three related aspects of manufacturing: manufacturing standards and design, software modelling, and AI-based enhancements.

Aligned with standards and design, we note the impact of documents such as the ISO-9000 series documents responsible for outlining standards for quality assurance (Marquardt et al., 1999). Since its original publication, researchers have examined these standards from functional (Rogala et al., 2021) and economic perspectives (Clougherty et al., 2014). These documents serve as a global foundation for quality management systems; and in part, we attribute our system's analytic and generative capabilities to these standardization efforts.

To ensure compliance, as well as to address the challenges of high-volume and complex manufacturing processes, companies have come to rely on software systems for Manufacturing Execution Systems (MES) and Enterprise Resource Planning (ERP). These systems have been created to help in record-keeping, analysis, and organization of manufacturing processes (Shojaeinasab et al., 2022). Over time, they have undergone enhancements to both their interoperability and the granularity of their representation, with a long-term objective of creating a digital twin (Attaran et al., 2023; Jin et al., 2024). While current tooling allows for highly detailed and synchronized models (Cimino et al., 2019), there is ongoing research in enhancing these aspects (Rogala et al., 2021; He et al., 2021). General techniques for using these systems for the automation of tasks such as optimization and fault discovery (Nguyen et al., 2016) are yet to be widely adopted. While researchers have worked to develop fault ontologies (Liu et al., 2019), their usage has been largely application-specific and focused on post-facto analysis (Rajpathak et al., 2020).

The integration of artificial intelligence into manufacturing processes has grown significantly in recent years. Computer vision and related technologies are widely used for various tasks, including anomaly detection, counterfeit identification, and assembly planning (Zhou et al., 2022). Traditional natural language processing (NLP) methods have also been applied, particularly for semantic anomaly detection. Busch et al., (2024) employed seq2seq models to detect anomalies within business processes and analyze the specifics of undesired behavior. Additionally, Sola et al. (2022) compiled a large dataset of business process logic (primarily consisting of BPMN files) designed to support generative AI methods in automating various aspects of business workflows. More recently, large language models (LLMs) and other generative AI capabilities have been explored for applications such as product design (Liang et al., 2023), human-in-the-loop system development (Makatura et al., 2024), and knowledge management systems (Kernan et al., 2024).

METHODOLOGY

Fault Detection

The fault detection process (illustrated in Figure 1) begins with the ingestion of a business or manufacturing document, which is then parsed into smaller,

manageable chunks. Each chunk undergoes an initial background retrieval step, where dynamic queries are used to extract relevant background knowledge from a vector database (pre-populated). The dynamic queries are formed based on the document's contents. This contextual information helps determine whether the document chunk aligns with established operational standards or exhibits potential anomalies.

Next, the system enters the discrepancy detection phase. Here, a language model (LLM) is prompted with both the document chunk and the retrieved background snippets. The prompt explicitly instructs the LLM to assess whether any discrepancies or abnormalities exist between the document chunk and the background context. If a discrepancy is found, the LLM provides a brief explanation describing its nature. This explanation is then incorporated into a subsequent prompt that classifies the identified issue by comparing it against a predefined set of known faults. By including the discrepancy explanation in the classification step, the system ensures the LLM has sufficient context to make an informed and accurate fault identification.

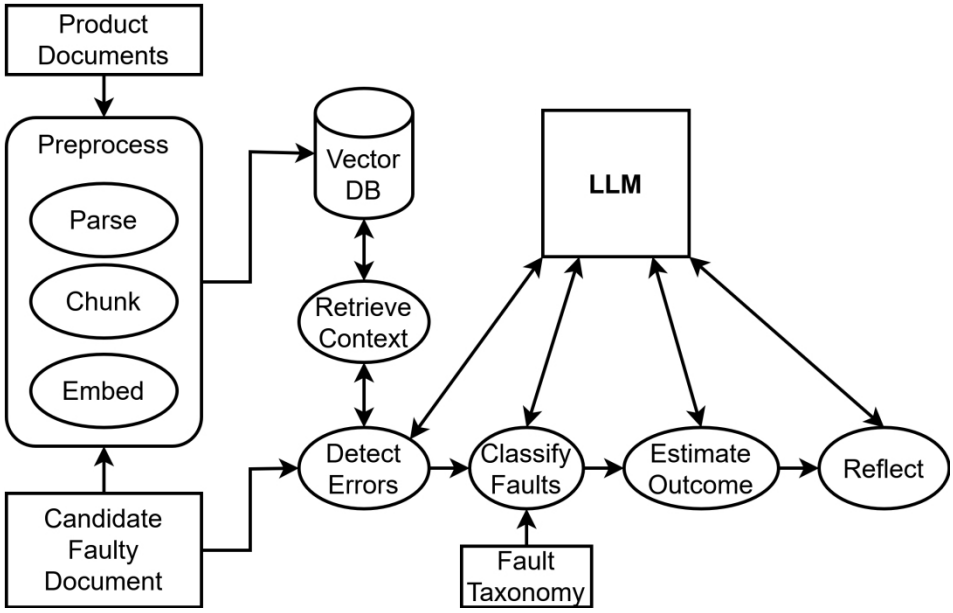

Figure 1: Fault detection pipeline. Background product documents are used to check the candidate document for discrepancies. The discrepancies are classified to specific faults against our fault taxonomy.

Once a fault is identified, the process moves to the severity and impact assessment phase. The LLM evaluates the fault's potential consequences on the documented process, describing the worst-case scenario while also rating the fault's severity, urgency, and importance using predefined categories (none, low, medium, or high). This combined quantitative and qualitative assessment ensures a well-rounded evaluation of the associated risks.

To further enhance reliability, the system incorporates a self-reflection step. At this stage, the LLM is presented with all relevant information

from previous steps, including the document chunk, background snippets, discrepancy explanation, fault classification, and impact assessment. The LLM then reviews its own analysis, assigns a confidence score to indicate how certain it is of its conclusions, and provides a brief rationale for its confidence level. This self-reflection step helps to validate the consistency of the pipeline while offering an additional layer of quality control by prompting the LLM to reassess its reasoning.

This structured pipeline systematically identifies, classifies, and evaluates faults within complex business and manufacturing documents. By integrating background retrieval, discrepancy detection, fault classification, impact assessment, and self-reflection, the system ensures a thorough and reliable approach to assessing potential document issues. We depict our fault detection pipeline in Figure 1.

Data Synthesis

The data synthesis system is a modular pipeline (illustrated in Figure 2) that automates the generation of interdependent production documents for a product. It leverages a local Large Language Model (LLM) hosted on an Ollama[1] server to generate documents based on dynamically constructed prompts. These prompts incorporate hand-written general document definitions, as well as automatically generated contextual definitions dependent on the user's specified product. The two types of document definitions jointly specify the purpose, content, and structure of the document, ensuring consistency across the production process.

A key feature is the use of a vector database (Chroma[2]) and a Sentence Transformer model[3] to embed and store document content and metadata. This enables efficient retrieval of semantically relevant context for subsequent documents. Instead of relying on static queries referencing only the product name and document type, the system formulates dynamic queries using comprehensive document definitions, ensuring accurate and context-aware document generation.

The pipeline follows an optimized sequence, beginning with foundational documents like the Bill of Materials (BOM) and the Procurement and Supplier Management Policy. These provide critical details on parts, materials, and suppliers, serving as references for later documents such as Work Instructions, Assembly Line Processes, Traveler Documents, Quality Assurance (QA) Policy, etc. Explicitly defined dependencies allow the system to integrate relevant information from previous documents, maintaining consistency throughout production documentation.

To enhance integrity, the system performs consistency checks. After generating a document, it retrieves related documents from the vector database and prompts the LLM to assess alignment with existing content. If inconsistencies are found, an automated revision process ensures adherence to required standards. We present the data synthesis pipeline in Figure 2.

[1] https://ollama.com/

[2] https://github.com/chroma-core/chroma

[3] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

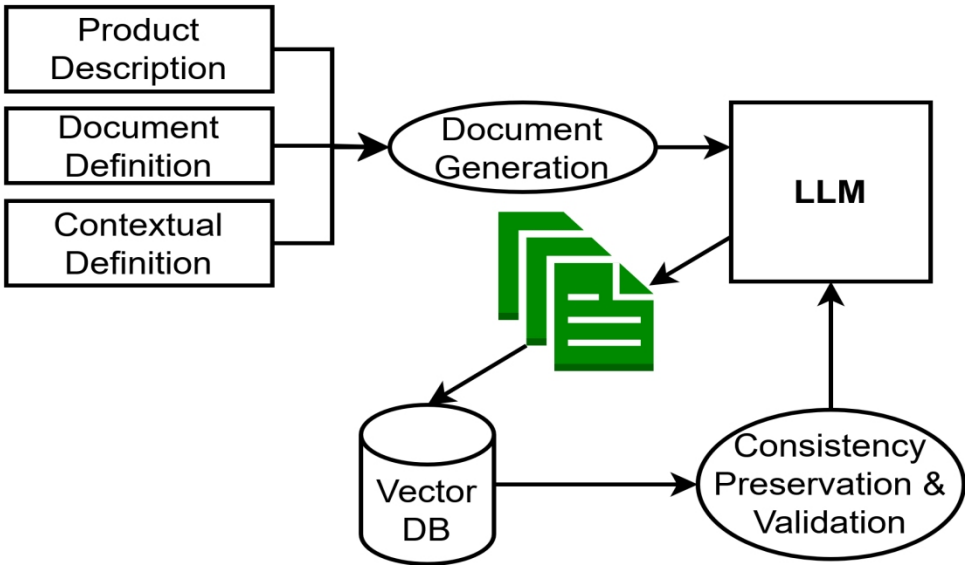

Figure 2: Data synthesis pipeline. 'Product description' and 'document definition' are hand-written materials. 'Contextual definition' is automatically produced in a prior step.

EVALUATION

We evaluate the effectiveness of MetaBPL using three core metrics:

1. **Fault Detection Rate:** The percentage of faults correctly identified, representing an intuitive classification accuracy for fault performance.
2. **Fault Explanation Performance:** A measure of how well the model balances completeness and conciseness when explaining detected faults. We compute TF-IDF-based cosine similarity and ROUGE (Lin 2004) scores (ROUGE-1, ROUGE-2, and ROUGE-L) between the LLM-generated explanation and the human-written ground truth. Cases where the fault is not detected are not penalized.
3. **Context/Location Retrieval Metrics:** A measure of the accuracy and conciseness of the retrieved context used for fault detection. Similar to fault explanation performance, we assess this using TF-IDF-based cosine similarity and ROUGE scores.

To evaluate MetaBPL, we construct a benchmark corpus of artificially generated test cases with controlled fault injection. This synthetic internal documentation includes structured documents such as Bill of Materials, Quality Control (QC) Policies, Work Instructions, and Travelers. Documents generated via our approach have an average of 330 words. Our pipeline ensures consistency across these documents, while errors are manually introduced based on an established fault taxonomy. This structured approach enables a thorough analysis of fault detection performance.

Table 1: Fault taxonomy including fault names and their descriptions.

Fault	Description
Loss of Provenance	Missing records of data usage, resource allocation, or process history, preventing traceability and auditing.
Ambiguous Reference to Object	A resource is referenced inconsistently, causing uncertainty about its correct usage.
Reference to Undefined Object	A resource is mentioned but does not exist in official records, making it unusable.
No Confirmation	A process runs without verifying outcomes, making errors undetectable and irreversible.
Missing Work Instructions	A workflow step exists but lacks execution details, leaving users without guidance.
No Termination	A process cycles indefinitely due to invalid transitions or unresolved conditions.
Incomplete Workflow	A process lacks necessary steps or elements, preventing successful completion.
Resource Leakage	More resources are used than needed, causing inefficiency and waste.
Unreferenced Object	A resource exists in records but is never used in any workflow step.
Unsatisfied Requirements	A documented requirement is missing from the final process or product.
Conflicting Requirements	Two or more requirements contradict each other, making them impossible to satisfy.
Incomplete Requirements	Requirements lack necessary details, leading to implementation gaps.
Inconsistent Work Instructions	Instructions contain contradictions, creating confusion in execution.
Duration Check Violation	A time limit exists but is not enforced, allowing tasks to exceed deadlines.
No Execution	A process step exists but is never triggered or used.
Resource Depletion	Required resources run out before a task can be completed.
Underconstrained Duration	Task durations lack constraints, leading to scheduling uncertainties.
Unused Object	A resource is collected but never used in any workflow step.

The benchmark consists of seven test cases, each covering a diverse range of fault types. Across all test cases, there are 29 faults spanning 18 unique fault types, including deadlocks, one-way functions, unsatisfied requirements, unused materials, and references to undefined materials. For detailed descriptions of these fault types, see Table 1.

RESULTS

We provide an overall summary of the results (Table 2) from our automated fault detection pipeline to demonstrate that MetaBPL can help improve the accuracy and efficiency of fault detection in complex business environments.

MetaBPL detects 44.8% of injected faults (13/29), demonstrating strong fault detection. Fault explanations show partial alignment with human-written ones (cosine similarity: 0.2457, ROUGE-1: 0.3525), indicating moderate relevance. Fault location accuracy is higher (cosine similarity: 0.5133, ROUGE-1: 0.4718), effectively pinpointing faulty sections. However, retrieved context quality is weaker (cosine similarity: 0.1701, ROUGE-1: 0.2434), highlighting the need for better supporting information selection.

Using LLaMa 3.2 8B (Dubey et al., 2024) on a remote AWS server, our pipeline processes the benchmark dataset in under 2.5 hours, averaging approximately 5 minutes per input document, i.e. organizational documentation containing faults.

Our evaluation also reveals several false positives, though we did not conduct a detailed error analysis. Our assessment focuses primarily on cases where the model correctly detected the manually injected faults. However, from a cursory review, most false positives appeared reasonable, likely due to the nature of our benchmark. Since the dataset consists of artificially generated organizational documentation rather than human-written text, some flagged issues may still reflect plausible inconsistencies.

These findings suggest that while MetaBPL performs reasonably well in fault localization, improvements in explanation clarity and context retrieval could enhance its overall effectiveness while maintaining efficient processing times.

Table 2: Evaluation results along our core metrics.

Fault Metrics	Explanation	Location	Context
Total Faults: 29	Cos. Sim: 0.2457	Cos. Sim: 0.5133	Cos. Sim: 0.1701
Faults Found: 13	ROUGE-1: 0.3525	ROUGE-1: 0.4718	ROUGE-1: 0.2434
Found%: 44.8	ROUGE-2: 0.1601	ROUGE-2: 0.4156	ROUGE-2: 0.1367
	ROUGE-L: 0.2549	ROUGE-L: 0.4521	ROUGE-L: 0.1917

CONCLUSION

MetaBPL represents a scalable, automated solution for detecting and analyzing faults in business process logic systems. By leveraging an LLM-powered RAG architecture, the framework enables proactive identification of discrepancies, structured severity assessment, and intelligent corrective recommendations. Our evaluation highlights its effectiveness in reducing reliance on manual fault detection while ensuring greater precision and operational resilience.

Additionally, the system demonstrates efficient processing times, analyzing our benchmark dataset in under 2.5 hours (approximately 5 minutes per fault) using LLaMa 3.2 8B (Dubey et al., 2024) on a remote AWS server.

While MetaBPL performs well in fault localization, results indicate that context retrieval and explanation clarity could be further improved to enhance the overall accuracy of automated fault analysis. Future work will focus on expanding fault taxonomies, refining self-reflection mechanisms,

and integrating real-time monitoring capabilities to further strengthen business logic security and improve interpretability.

Additional future work will extend MetaBPL into a neuro-symbolic fault detection pipeline. Business Process Model Notation (BPMN)[1], extended to outline interactions between ERP and MES systems, provides a structured language to model processes. We will leverage NLP techniques to extract such models of processes from given documents. These models, with formal specifications of properties that lead to faults, are fed to SMT Solvers and BDD-based model checkers. This approach will integrate RAG-LLMs' ability to restructure data with symbolic methods that provide a high-assurance approach to catching faults that decrease the risk of false positives or negatives due to LLM hallucinations.

ACKNOWLEDGMENT

The material presented in this paper is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under contract number N6523624C8008. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of DARPA.

REFERENCES

Attaran, M. and Celik, B. G. (2023) 'Digital Twin: Benefits, use cases, challenges, and opportunities', *Decision Analytics Journal*, 6, p. 100165. Elsevier.

Brecher, C., Esser, M. and Witt, S. (2009) 'Interaction of manufacturing process and machine tool', *CIRP Annals*, 58(2), pp. 588–607. Elsevier.

Busch, K., Kampik, T. and Leopold, H. (2024) 'XSemAD: Explainable Semantic Anomaly Detection in Event Logs Using Sequence-to-Sequence Models', in *Business Process Management: 22nd International Conference, BPM 2024, Krakow, Poland, September 1–6, 2024, Proceedings*. Springer-Verlag, Berlin, Heidelberg, pp. 309–327.

Cimino, C., Negri, E. and Fumagalli, L. (2019) 'Review of digital twin applications in manufacturing', *Computers in Industry*, 113, p. 103130. Elsevier.

Clougherty, J. A. and Grajek, M. (2014) 'International standards and international trade: Empirical evidence from ISO 9000 diffusion', *International Journal of Industrial Organization*, 36, pp. 70–82. Elsevier.

Guan, W., Cao, J., Gao, J., Zhao, H. and Qian, S. (2024) 'DABL: Detecting Semantic Anomalies in Business Processes Using Large Language Models', *ArXiv*, abs/2406.15781.

He, B. and Bai, K.-J. (2021) 'Digital twin-based sustainable intelligent manufacturing: a review', *Advances in Manufacturing*, 9(1), pp. 1–21. Springer.

Hussain, M. and Khanam, R. (2024) 'In-depth review of yolov1 to yolov10 variants for enhanced photovoltaic defect detection', *Solar*, 4(3), pp. 351–386. MDPI.

Jin, C. and Liu, R. (2024) 'An Evaluation of Capabilities, Benefits, and Challenges of Developing Digital Twin Models for Sustainable Development', *Human Factors in Design, Engineering, and Computing*, 159(159). AHFE Open Access.

Kernan Freire, S., Wang, C., Foosherian, M., Wellsandt, S., Ruiz-Arenas, S. and Niforatos, E. (2024) 'Knowledge sharing in manufacturing using LLM-powered tools: User study and model benchmarking', *Frontiers in Artificial Intelligence*, 7, p. 1293084. Frontiers Media SA.

Liang, R., Agnesina, A., Pradipta, G., Chhabria, V. A. and Ren, H. (2023) 'CircuitOps: An ML Infrastructure Enabling Generative AI for VLSI Circuit Optimization', in *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, pp. 1–6. IEEE.

Liu, B., Wu, J., Yao, L. and Ding, Z. (2019) 'Ontology-based fault diagnosis: A decade in review', in *Proceedings of the 11th International Conference on Computer Modeling and Simulation*, pp. 112–116.

Makatura, L. et al., (2024) 'How Can Large Language Models Help Humans in Design and Manufacturing? Part 1: Elements of the LLM-Enabled Computational Design and Manufacturing Pipeline', *Harvard Data Science Review*, Special Issue 5. The MIT Press. Available at: https://hdsr.mitpress.mit.edu/pub/15nqmdzl.

Marquardt, D. W. and Juran, J. M. (1999) *The ISO 9000 family of international standards*. McGraw-Hill.

Nguyen, D. T., Duong, Q. B., Zamai, E. and Shahzad, M. K. (2016) 'Fault diagnosis for the complex manufacturing system', *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 230(2), pp. 178–194. SAGE Publications.

Rajpathak, D., Xu, Y. and Gibbs, I. (2020) 'An integrated framework for automatic ontology learning from unstructured repair text data for effective fault detection and isolation in automotive domain', *Computers in Industry*, 123, p. 103338. Elsevier.

Rogala, P. and Wawak, S. (2021) 'Quality of the ISO 9000 series of standards-perceptions of quality management experts', *International Journal of Quality and Service Sciences*, 13(4), pp. 509–525. Emerald Publishing Limited.

Shojaeinasab, A., Charter, T., Jalayer, M., Khadivi, M., Ogunfowora, O., Raiyani, N., Yaghoubi, M. and Najjaran, H. (2022) 'Intelligent manufacturing execution systems: A systematic review', *Journal of Manufacturing Systems*, 62, pp. 503–522. Elsevier.

Singh, M., Fuenmayor, E., Hinchy, E. P., Qiao, Y., Murray, N. and Devine, D. (2021) 'Digital twin: Origin to future', *Applied System Innovation*, 4(2), p. 36. MDPI.

Sola, D., Warmuth, C., Schäfer, B., Badakhshan, P., Rehse, J. and Kampik, T. (2022) 'SAP Signavio Academic Models: A Large Process Model Dataset', in Montali, M., Senderovich, A. and Weidlich, M. (eds) *Process Mining Workshops. ICPM 2022. Lecture Notes in Business Information Processing*, vol. 468. Springer, Cham.

Zhou, L., Zhang, L. and Konz, N. (2022) 'Computer vision techniques in manufacturing', *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 53(1), pp. 105–117. IEEE.