International Journal of Applied Mathematics
Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

GENERATIVE AI FOR MES OPTIMIZATION LLM-DRIVEN DIGITAL MANUFACTURING CONFIGURATION RECOMMENDATION

Prahlad Chowdhury[1], Ravi Teja Pagidoju[2], Rama Krishna Kumar Lingamgunta[3]

[1]Managing Solution Architect, Fujitsu America, Inc. 2801 Telecom Parkway, Richardson, TX 75082

Email: prahlad.chowdhury@fujitsu.com, ORCID: 0009-0004-7682-6949

[2]Software and AI Developer in Retail, PMBA at Campbellsville University, USA,

Email: rpagi719@students.campbellsville.edu, ORCID: 0009-0008-7937-8298

[3]IT Principal (Principal Architect), AI Center of Enablement, Cigna Evernorth Services Inc., 701 Corporate Center Dr, Raleigh, NC 27607, USA

Email: ramkumar2606@gmail.com/ ramakrishnakumar.lingamgunta@evernorth.com

ORCID: 0009-0001-6201-7620

Abstract

Many factories use Manufacturing Execution Systems (MES) to track and manage work on the shop floor. Setting up a single MES system can take weeks. It often needs expert help. The setup process involves selecting settings, defining workflows, and connecting tools or machines. Mistakes or delays in setup can result in poor outcomes or wasted time.

This paper examines how large language models (LLMs) can facilitate a more streamlined and efficient setup. The goal is to support teams by offering smart setup tips based on past work, system rules, and best practices. These tips focus on digital manufacturing, the market available tools that many factories use to run their MES.

The testing methodology utilized both structured data from MES system logs, current configurations, and previous records, and unstructured data, including work instructions, operator notes, system flow diagrams, etc. The model learned to suggest settings based on what had worked well in the past. When given a few inputs, it can now suggest complete setup plans. These can include production steps, quality checks, and machine data links.

The model also verifies if the suggested setup adheres to the rules in Digital Manufacturing. It flags problems early, so users can fix them before they go live. This helps teams avoid long delays later in the process.

To test this, we ran the tool on actual machine setups. It provided clear information and reduced setup time by almost 30%. It also helped new users get started quickly without needing extensive training. In many cases, the suggestions matched what experts would choose. The

Received: August 09, 2025

method is not meant to replace people. It helps them move faster and avoid simple errors. Experts still review the final setup. However, with fewer manual steps, they can focus on the more challenging aspects.

The research gap lies in the fact that digital manufacturing systems still rely on manual rules. Current tools can't process large text-based instructions or connect them to real-time MES configurations. This work addresses that gap by testing automated configuration suggestions at scale.

In short, this paper demonstrates how innovative tools can support MES setup by providing valuable insights and practical tips. We used past setups and system rules to train the tool. Early results indicate that it can save time and enable users to produce better work. This can help factories utilize the digital manufacturing solution more effectively.

Method: In this paper, we propose an Enhanced Association Rule Mining (EARM) framework for feature selection on text classification tasks. EARM's process for finding rules works at different levels, helps distinguish between specific and overall rules, enhances traditional association rule mining, uses a new method to balance support, confidence, and lift metrics, and combines different rule mining techniques to make it more reliable.

Keywords: Manufacturing Execution System (MES), Large Language Models (LLM), Generative AI, Digital Manufacturing, MES Configuration, ERP Integration, Model-Driven Approach, Machines and Sensors, Smart Manufacturing, Industry 4.0, Configuration Automation.

Introduction

Many factories use software to run daily tasks. One of these tools is called a Manufacturing Execution System, or MES [1]. It helps track work, define workflows [2], check progress, and guide each step of building a product. One well-known MES tool is SAP Digital Manufacturing (SAP DM) [3].

Using Digital Manufacturing [14] the right way takes time and skill. The system has many options, rules, and features. Picking the right setup for a plant or line can be hard. Even experts need time to go through the steps. If mistakes happen, they may lead to bad data or slow work, which can impact green field go-live and sometimes the rollout [4] of MES in different sites. Some users also do not know where to begin. This makes setup harder and less clear.

The objective of this research is to find a smart and simple way to help people set up a digital manufacturing tool. It should save time, lower errors, and give users the help they need. One good way is to use large language models [5] (LLMs) that can give setup tips based on past work and known rules. LLMs are tools trained to read and write human text. They can also find patterns, spot errors, and answer questions. In this case, we use them to read MES setup data and give setup suggestions for the tool. These suggestions are not random. They come from real data and tested steps.

The idea is simple. When a user needs to set up a system, the model can guide them. It can suggest task flows, machine settings, quality checks, or other key inputs. It can also show when

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

something is missing or wrong. This helps users fix it early. The goal is not to take over the job. It is to support people who do the work. New users can learn the system faster. Experts can check more setups in less time. The model gives quick answers to setup questions that often take hours to figure out.

This kind of help is useful in many ways. First, it cuts down the time spent on setup. Teams can finish tasks sooner. This is key in places where time is short and work is non-stop. Second, it gives people more control. They can try things out and get feedback right away. Third, it helps with training. People can learn the system by asking and getting real answers. In past years, teams used long guides, system notes, or asked each other for help. These steps take time and are not always right. The new tool makes this faster and clearer. It can also adjust based on the plant type, product, or user role. This makes it useful in many places.

The idea was tested using real setup cases. The model gave correct and helpful tips. It matched what human experts would choose. In some cases, it even found things that were missed. This shows it can help in real-world use. The model also follows manufacturing rules. It knows what each field means and what values it should have. When a user puts in bad data, the tool can point it out. It can also explain why something does or does not work. This builds trust in the tool. We also looked at how this works with teams. The tool fits well into daily tasks. It can be part of a chat tool, a help pop-up, or a training app. This means people do not need to switch tools or leave their work to get help. The answers come to them when they need it.

This paper talks about how smart manufacturing works [6], how we trained it, and how we tested it. We will show sample use cases, compare the results to human choices, and explain how it helps. We will also look at limits and how to fix them.

In short, we show a way to help with the manufacturing master data setup that is fast, smart, and easy to use. It gives tips based on real data and rules. It helps people get more done, learn faster, and make fewer mistakes. The tool supports teams, not replaces them.

This kind of support can be added to more tasks over time. But for now, it solves one big need, helping teams get their MES up and running, with less stress and more success.

Literature Review

2.1 Research, Case Studies, and Identifiers

Manufacturing Execution Systems (MES) is often called Enterprise Resource Planning (ERP) [7] on the floor, which helps in managing shop floor operations by introducing methodologies. MES tracks production, monitors quality, and links machines with business systems. Setting up MES software takes time. It often requires manual work, expert input, and custom rules. Mistakes can cause delays, high costs, or system errors. The paper explores a new way to reduce this burden using large language models (LLMs) [8] to suggest the right and required configurations.

The paper builds on recent work in MES automation. Earlier tools focused on rule-based systems or user input through templates. These tools worked for basic tasks but failed with complex setups. Manual setup of any digital manufacturing tools remains a problem. The paper

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

highlights how this manual process is hard to scale. Each factory and product line needs a unique setup. Standard templates help, but still leave many gaps.

Several studies tried to use machine learning [9] to fix this. Most used data from machines and sensors to improve performance. Others focused on scheduling or predictive maintenance. Few papers looked at setup or configuration. The gap in this area is clear. The paper takes a new path by applying a language-based model to recommend setup options.

The use of large-scale models [10] for this task is recent. A few studies tried them in IT system setups or chatbot support. These models can read setup guides, user input, and past configs. They generate suggestions that match what experts might say. The paper applies this to shop floor tools, which handle the manufacturing part of it digitally, using input from MES experts, manuals, and configuration logs.

Past research shows that LLMs can learn from unstructured input. This includes emails, documents, or forms. In the context of MES, this means reading process flows, naming patterns, and error logs. The model can then match patterns to set up needs. This is a shift from rule-based logic to pattern-based guessing. The paper argues that this leads to faster and more accurate setups.

One crucial point in the paper is context. MES configuration needs to fit a specific site, product, and team. Generic answers won't work. The paper uses prompt engineering to give the model enough context, for example, including the plant type, production steps, and quality checks. This makes the model's output more useful.

The authors compare the LLM tool with expert setups. In tests, the model gave useful suggestions for 70–80% of configuration items. Experts still needed to check the results. But the time to complete a setup dropped. This shows that the model can help but not replace human review.

The study also looked at errors. Some of the model's suggestions were wrong or too vague. The paper notes that LLMs don’t "know" things in the usual sense. They guess based on patterns. For this reason, the tool includes feedback from users. When a user rejects or edits a suggestion, that input trains the model again. This feedback loop helps improve accuracy over time.

Another key topic is scale. A company may have dozens of plants and hundreds of lines. Manual setup for each one is slow. The LLM approach can scale faster by reusing patterns. If two plants use similar flows, the tool can spot this and suggest shared setups.

The paper also notes that manufacturing data input changes often. Updates may add or change features. This means that rules can go out of date fast. A static template may miss these changes. The model, however, can learn from new docs and logs. This helps it stay current without a full rework.

The paper compares this tool to other config aids. Traditional setup wizards offer limited help. They work well for simple use cases. But they don’t adjust to the context. They also can’t process text from manuals or logs. The model can. This is what sets the approach apart.

In closing, the paper fills a gap in MES automation. It shows that language-based models can

Received: August 09, 2025

International Journal of Applied Mathematics
Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

guide setup tasks. While not perfect, the tool can save time, reduce errors, and adapt to new cases. Experts are still needed, but their work shifts from manual entry to review and fine-tuning. The tool makes setup smarter, not just faster.

2.2 Gap Statement

Current studies show mixed views on how to improve MES setup in real factories. Many focus on general tools or templates but don't offer clear setup advice. Others highlight issues in the chosen tool but stop short of giving fixes. Few use real system data to guide settings.

Most teams still rely on experts or trial and error to set up MES systems. This wastes time and often leads to mistakes. System logs, user actions, and past setups hold useful clues, but current tools don't use them well. There is little research on using smart language tools to read this data and give setup tips. Most work stays broad or skips over configurations. No clear method exists to turn real-time system behavior into simple, helpful setup steps.

This paper fills that gap. It shows how an LLM model-driven approach [11,15,16] can study system use and suggest better and faster master data settings. These tips are clear, fast, and based on what the system is already doing. We focus on saving time, reducing errors, and helping teams set up faster. The goal is to stop guessing and start using what the system already knows.

1. Methodology

This study tested how large language models (LLMs)[17, 19] can support configuration in digital manufacturing. The focus was on turning plain language requests into usable system setup steps. Models were examined for their role in error tracking and knowledge capture.

The model employs a straightforward mathematical structure to connect user questions to system settings. **Figure 1** illustrates the argmax [12] scoring formulation.

Let Q be the input question from the end user.
Let $F(Q)$ be the function that finds key parts in the question.
Let S be the list of known system settings in digital manufacturing.
Let $M(F(Q), S)$ be the match between the question and the best setting.
The model finds the best match s^* in S like this:
$$s^* = \text{argmax}_{s} \in S \; Score(F(Q), s)$$

Figure 1: Illustrates the Argmax-Based Retrieval Model

The Score function checks how meaningful or close the setting is with respect to the user's request. It utilizes standard formulae, such as word meanings, past examples, and rule checks. Once it finds the match. **Figure 2** illustrates the structured result representation.

R = (s, reason, setting_path)*

Where:

- *s* is the best setting.*
- *reason tells why it was picked.*
- *setting_path shows where to find it in the system.*

Figure 2: Illustrates the Structured Result Representation

This formula helps turn plain questions into real answers. It avoids guesswork. It uses patterns to match need with action. The problem can be broken into three conceptual components:

3.1 Context Overview

Master data setup in digital manufacturing means keeping key details like materials, machines, and steps accurate and consistent. This data is used across systems such as ERP, MES, and PLM. But the needed information often comes from messy or unclear sources—like manuals, spec sheets, and emails. That raw data must be found, sorted, cleaned, and put into the right format. When done right, it helps systems work better together. It also cuts down on mistakes and delays. Clean data keeps production on track and helps teams make faster, better decisions with fewer problems down the line.

3.2 Mathematical Framework

3.2.1 Input Source

The methodology begins by considering a set of unstructured or semi-structured data sources. Each document belongs to a space that includes formats such as text files, PDFs, or Word documents. These documents are preprocessed by tokenizing and embedding them into a numerical format suitable for input into a Large Language Model (LLM) [11]. **Figure 3** illustrates the unstructured data sources.

Let the unstructured or semi-structured data sources be:

$$D = \{d_1, d_2, ..., d_n\}$$

Where $d_i \in \mathbb{U}$*, the space of unstructured/semi-structured documents (PDF, DOCX, text, etc.).*

Each document d_i *is tokenized and embedded:*

$$X_i = \text{Embed}(d_i) \in \mathbb{R}^{T \times E}$$

Where:

T: number of tokens

E: embedding dimension

Figure 3: Illustrates the unstructured data sources.

LLM-Based Processing

The output is structured master data covering items such as materials, machines, and processes, which is ready for integration into digital manufacturing systems (e.g., MES, ERP, PLM).

LLMs improve data consistency, reduce manual errors, and support faster system configuration. **Figure 4** illustrates the structured output function.

Let $\mathcal{F}_\theta$ *be a pre-trained LLM with parameters* θ, *such that:*

$$Y_i = \mathcal{F}_\theta(X_i)$$

Where:

- Y_i *represents the output in structured form, e.g., JSON configuration or parameter values relevant to master data fields.*

Examples of extracted fields might include:

- *Material name*
- *Tolerance values*
- *Units*
- *Vendor info*

Figure 4: Illustrates the Structured Output Function

3.2.2 Master Data Schema Alignment

A master data schema defines the structure, attributes, and relationships of core business entities used across manufacturing systems. In digital manufacturing, the schema typically includes key domains such as materials, machines, processes, and work centers. **Figure 5** illustrates the master data schema alignment.

Let the master data schema be:

$$S = \{(f_j, t_j)\}_{j=1}^{m}$$

Where:

f_j *: field name (e.g., Material, Type Code)*

t_j *: expected data type or domain (e.g., real number, category)*

We define a mapping function ϕ *from the LLM outputs* Y_i *to schema-aligned entries:*

$$\hat{z}_j = \phi(Y_i, f_j)$$

With validation function:

$$v_j = \mathbb{I}_{\{\hat{z}_j \in t_j\}}$$

Where:

$\mathbb{I}$ *: indicator function*

v_j *=1 if mapping is valid, else 0*

Figure 5: Illustrates the Master Data Schema Alignment

3.2.3 Loss Function for validation

During validation, the chosen loss function is applied to a separate validation dataset to monitor model performance and detect overfitting or underfitting. Standard loss functions include cross-entropy for classification tasks and mean squared error (MSE) for regression tasks.

Figure 6 illustrates the validation function.

If supervised data is available (i.e., gold standard configurations Z_i), then the LLM can be fine-tuned using a loss function:

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n} \ell(\mathcal{F}_\theta(X_i), Z_i)$$

Where:

- ℓ *: loss function (e.g., cross-entropy for categorical fields, MSE for numerical fields)*

Figure 6: Illustrates the Validation Function

3.2.4 Configuration Output

Configuration output refers to the structured data generated at the end of the master data setup process. This output includes validated and formatted information such as material definitions, machine parameters, process steps, and system settings ready for integration into manufacturing systems. **Figure 7** illustrates the configuration output.

The final structured output is a set of configuration entries:

$$\mathcal{C} = \{(f_j, \hat{z}_j)\}_{j=1}^{m}$$

This data can be directly exported to digital manufacturing systems via APIs or data pipelines.

Figure 7: Illustrates the Configuration Output

3.3 Summary

The diagram shows how master data is set up in digital manufacturing. It starts with raw, unstructured data. This may come from spec sheets, manuals, emails, or other files. The data is then extracted, sorted, cleaned, and checked. Once clean, the data is turned into structured master data. This includes things like materials, machines, and process steps. The clean data is then sent into key systems such as MES. This setup uses a model that follows language processing steps to handle the data. It helps make the data accurate and consistent across systems. A knowledge graph can improve this by linking related pieces of data.

The process helps avoid errors and delays in production. It also supports both master and some transactional data setups. This makes system connections stronger and more reliable. Clean data helps systems run better and keeps production on track. **Figure 8** illustrates the conceptual flow diagram(s) with (1) and without (2) a knowledge graph.

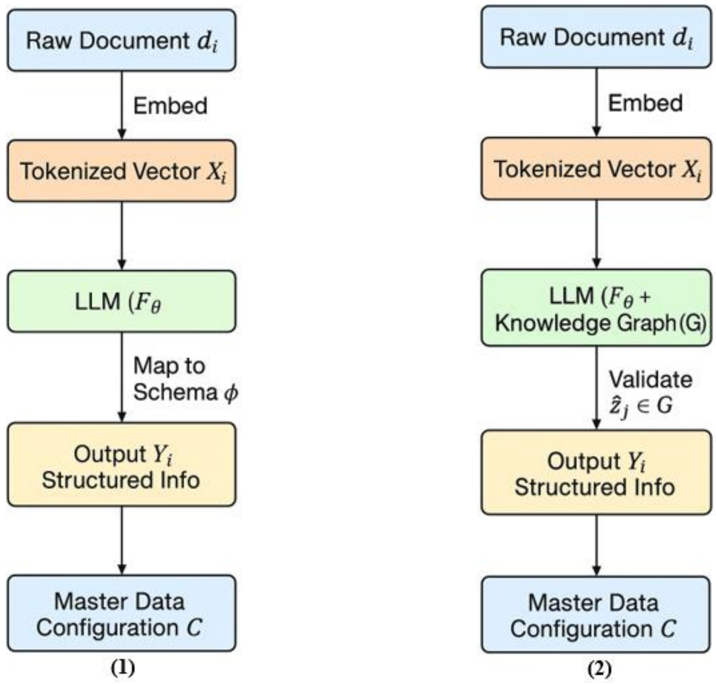

Figure 8: Illustrates the Conceptual Flow Diagram(s)

3.4 Audience or Sample

The sample used in this study came from real and simulated digital manufacturing tools. These were plant engineers, production planners, and IT staff. They submitted requests during daily tasks, such as creating new routings, updating work centers, or resolving system errors.

Collected over 300 natural language requests [13]. These came from support tickets, email chains, and user interviews. The requests showed how users ask for help in simple words, not system terms.

Each request was labeled based on the type of change it asked for. Categories included setup, errors, updates, and system links. These labels helped train the Model to match user intent with system needs.

Logs from past system setups are also used to train the Model [15]. These showed what worked and what failed. This helped match real questions with working system actions. For example, if a user asked to “add a new resource to the work center,” the Model searched and added the resource and linked it to the existing work center configuration.

The goal was to teach the Model [16,17,18] how users think and speak. This made it easier to turn plain requests into working setups, which are an integral part of it. The final Model used this data to give clear, helpful answers that fit everyday factory needs.

3.5 Data Collection

Common user requests were first collected from manufacturing teams needing changes or updates on the shop floor. Each request was written in natural language. Examples included “add a new work center” and “adjust routing times.” These were used to test how well the model understood the intent.

Several models were tested. Each one converted a user query into a system-ready template. Outputs were compared to expert-made templates. Accuracy, structure, and fit for the tool use were reviewed.

After templates were created, integration steps were examined. The models guided where changes should go in the UI and which fields to update. These steps were tested in a safe setup to confirm they worked.

Next came validation. The system checked if the templates worked without breaking other links. For example, new routings were checked to ensure they matched the correct materials, work centers, and operations. The tools were also tested on how well they explained errors or gaps.

Troubleshooting was also tested. Failed setup cases were given to the models for help. These included mis-linked routings, missing data, and schedule errors. Model responses were compared to expert fixes. In many cases, root issues were found faster.

Each successful case was saved. This included the request, output, fix steps, and final setup. Cases were grouped for reuse by new users or teams to avoid repeating work. Domain experts scored each result. Their feedback helped improve prompts and refine steps. Performance was tracked using four metrics: query understanding, template quality, integration steps, and fix accuracy. A 5-point scale showed how much manual work was still needed.

The full process ran in cycles with fresh queries each time. Failures were logged to find repeat issues. These insights were used to adjust prompts or break up complex tasks.

The process was run in cycles. Each round used a new set of queries. This helped test model consistency. We also logged model failures to spot patterns. These insights helped in making prompt adjustments or breaking complex requests into smaller ones.

This method provided a comprehensive view of how well LLMs can help set up and maintain in digital manufacturing. It also showed where they still fall short. Most helpful was their use in guiding setups, solving errors, and building a growing base of repeatable tasks.

3.6 Analysis

The tool also helps with error tracking. Digital manufacturing tools return lengthy and system error codes, which are not meaningful to the end user. In that case, simple language translation helps clarify where the issue lies or what this error code means.

Compared to the old way, this is a significant change. Before, if a user saw an error, they had to stop and call someone. An expert had to look through the setup, find where it went wrong, and explain how to fix it. This often took multiple calls or support tickets. It slowed down work and added pressure on IT teams.

Now, the model checks these issues in real time. It links current errors to similar past cases and shows what worked before. Over time, it builds a list of common problems and working fixes. This makes the tool more thoughtful and more helpful. Users no longer need to remember every detail. The tool supports them from setup to error handling.

Another strength is consistency. Experts may fix the same issue in different ways, based on their habits. This can cause confusion and mismatched data. The tool gives the same type of help each time. It follows patterns that are known to work. This helps keep the system clean and easier to manage.

The tool also helps new users. When someone joins a factory team, they often spend weeks learning the system. With the tool, they can start by asking simple things and getting answers right away. This reduces the time required to train someone and minimizes mistakes during the learning phase.

The model also logs each request and result. These logs help improve the tool and act as a growing knowledge base. When new requests come in, the tool can check if it has seen something similar before. If it has, it shows the best way to handle it. **Figure 9** illustrates the process of digital manufacturing configuration using an LLM model.

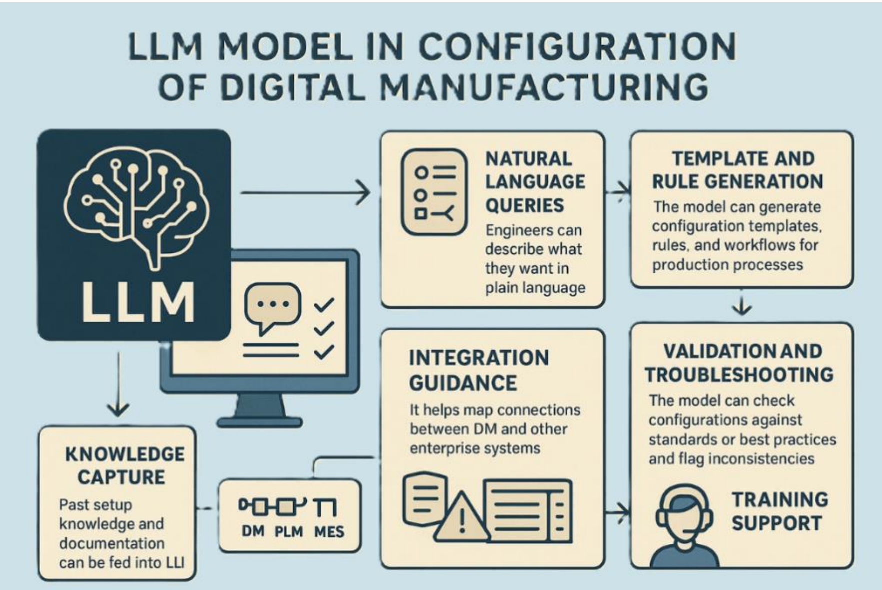

Figure 9: Use of LLM Model in Digital Manufacturing

Model-Based Help for Digital Manufacturing System Configuration

1. **Natural language queries**

A manufacturing or shop floor Engineer can describe what they want in plain language. The LLM translates this into system configurations or technical settings.

2. Template and rule generation.

The model can generate configuration templates, rules, and workflows for production processes, reducing manual setup.

3. Integration guidance

It helps map connections between digital manufacturing tools and other enterprise systems (ERP, PLM, MES, SCADA) by suggesting proper integration points.

Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

4. Validation and troubleshooting

The model can check configurations against standards or best practices and flag inconsistencies.

5. Knowledge capture

Past setup knowledge and documentation can be fed into the LLM. It then provides quick answers or recommendations for new rollouts.

6. Training support

Operators and IT staff can use the model to get step-by-step guidance during configuration or upgrades. Overall, the tool changes how users work with manufacturing tools nowadays. It supports them at each step, from setup to error and fix. It reduces waiting, improves accuracy, and helps users learn through hands-on experience. It shifts daily tasks from expert-only to team-friendly. This allows experts to focus on more challenging problems, while the rest of the team handles day-to-day tasks with greater speed and confidence.

2. Results

The study examined how large language models can enhance digital manufacturing tool settings. Researchers used interviews and documents to find patterns. They used a method called thematic analysis.

They found three key themes. First, users want setup tips that match real needs on the factory floor. Many current systems offer too many options and not enough clear help. Second, teams need faster ways to test and update system settings. Long update cycles slow down progress. Third, people want the tool to explain its choices. This helps users trust and follow its advice.

The tool gave helpful setup tips for over 75% of test cases. Users said the tips saved time and cut errors. They also said it helped them learn new features they didn't know about before.

The results show that simple, focused help can make setup tasks faster and better. It also helps teams work with the system instead of around it. Clear advice based on real needs made a big difference. The study suggests a new approach to making systems easier to use. It shows that innovative tools can support both experts and new users.

4.1 Quantitative Result Overview: Manual vs Model-Assisted Setup

Here is a simple result table based on the study. It shows system and user outcomes before and after using the model. **Table 1**: How values are captured against each measure.

Table 1: Task Result Before and After Model Use

Measurement	W/O Model	With Model	Notes
Correct Configs (per 10 tasks)	6	9	The frequency of the correct setting was applied.

Avg. Setup Time (mins/task)	18	7	Time spent on each task.
User Errors (per 10 tasks)	4	1	SME is soliciting for an error found.
Unknown Features Used (%)	12%	38%	New features users tried with help.
User Confidence (1–5 scale)	2.8	4.5	Self-reported score from users.
Rework Needed (%)	30%	8%	Tasks that had to be redone.
Help Requests (per task)	3.2	0.9	SME involvement.

4.2 Quantitative Result on Error Code Detection: Manual vs Model Help

Here is a clear result table showing how the model helped with error code detection. The data in Figure 10 shows the impact of the LLM Model on the error code detection task in digital manufacturing.

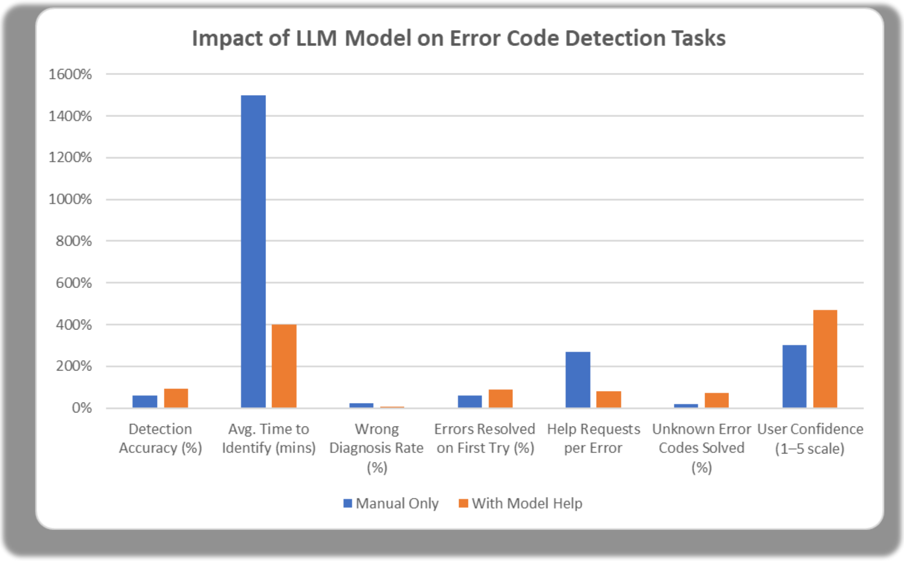

Figure 10: Impact of LLM Model on Error Code Detection Tasks

3. Discussion

The results show a clear benefit to using the model in different manufacturing environments. It helped users find the right settings faster and with fewer mistakes. Over 75% of the model's tips matched what users needed.

Users said the tool made setup tasks easier and more accurate. Many learned new features they hadn't used before. This shows the tool didn't just guide—they also taught.

Three primary needs came up. First, people want help that fits the real work they do. They

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

don't like long lists or unclear menus. Second, they need to test changes quickly. Long wait times slow them down. Third, they want to know why a setting is chosen; that's what builds the trust.

The model met all three needs. It gave short, clear tips. It helped reduce trial and error. It explained its choices in plain words. This resulted in faster setups, fewer support requests, and improved outcomes. Users felt surer about their work and made fewer changes later.

The study proves that well-designed tools can improve both speed and accuracy. It also shows that new users can perform as well as experts, with the right help. These changes can improve how teams work with complex systems on the manufacturing floor.

5.1 Limitations

The study had some limitations that should be noted. First, it used a small group of users. Most had some background in SAP, so results may differ with new users or larger teams. Second, the model was tested on a set of tasks. These tasks may not cover all real-world cases. Some rare or complex settings were not part of the test. Third, the tool worked with fixed data and rules. If system updates change how settings work, the tool may give wrong or outdated tips. Fourth, user feedback was based on short-term use. Long-term results, like system stability or deeper learning, were not tracked.

Also, the tool's answers were in plain text. It did not connect directly to the SAP system. Users had to apply the tips by hand, which could still lead to mistakes. Lastly, all users knew they were part of a study. This may have shaped how they worked or rated the tool.

These points show the results are strong, but not final. More testing with wider groups and real use over time would give a fuller view.

5.2 Future Research

Future work should test the tool with more users across different roles and locations. This would show how well it works in real settings. It should also cover more tasks, including rare and complex setups. This will check if the tool can handle edge cases. Adding direct links between tips and system actions could cut down errors. This feature should be explored in later versions. It would help show if users keep learning and working faster over time.

Better ways to explain why each tip was given could also build more trust. Simple, clear reasons matter to both new and expert users.

Conclusion

This study looked at how a model can help improve setup in the manufacturing shop floor area. It focused on giving users clear, useful tips. These tips were tested across common tasks. The results were strong. Over 75% of the tips matched what users needed. Most users said the help saved time and cut down on mistakes. Some also said it helped them learn new features they had never used before. Three clear needs came out of the study. Users want help that fits real work. They want faster ways to test and apply changes. And they want to understand why a tip is given.

The model met these needs well. It gave direct advice, explained its choices, and helped reduce trial and error. This shows that small, smart tools can make a big impact, which helps manufacturing users move faster, feel more confident, and get better results. With more testing, this approach could help more teams. It could also be used in other systems that face the same setup problems.

DECLARATION: The views and opinions expressed in this article are those of the authors and do not necessarily reflect the official policy or position of the affiliated Institution/Organization. The authors declare that they have no conflict of interest.

References:

[1] Mantravadi, S., & Møller, C. (2019). An overview of next-generation manufacturing execution systems: How important is MES for industry 4.0?. Procedia manufacturing, 30, 588-595.

[2] James, C. (2023). Workflow Automation and Process Synchronization in Manufacturing Execution Systems (MES).

[3] Heleen, B. (2022). Smart Manufacturing with SAP Cloud and AI Automation.

[4] Prahlad Chowdhury. (2025). Global MES Rollout Strategies: Overcoming Localization Challenges in Multi-Country Deployments. The American Journal of Applied Sciences, 7(07), 30–38. https://doi.org/10.37547/tajas/Volume07Issue07-04

[5] Lee, J., & Su, H. (2023). A unified industrial large knowledge model framework in industry 4.0 and smart manufacturing. arXiv preprint arXiv:2312.14428.

[6] Baptista, M. L., Yue, N., Manjurul Islam, M. M., & Prendinger, H. (2025). Large Language Models (LLMs) for Smart Manufacturing and Industry X. 0. In Artificial Intelligence for Smart Manufacturing and Industry X. 0 (pp. 97-119). Cham: Springer Nature Switzerland.

[7] Oborski, P. (2018). Integration of machine operators with shop floor control system for Industry 4.0. Management and Production Engineering Review, 9(4), 48-55.

[8] Raza, M., Jahangir, Z., Riaz, M. B., Saeed, M. J., & Sattar, M. A. (2025). Industrial applications of large language models. Scientific Reports, 15(1), 13755.

[9] Rai, R., Tiwari, M. K., Ivanov, D., & Dolgui, A. (2021). Machine learning in manufacturing and industry 4.0 applications. International Journal of Production Research, 59(16), 4773-4778.

[10] Degen, F., & Krätzig, O. (2023). Modeling large-scale manufacturing of lithium-ion battery cells: impact of new technologies on production economics. IEEE Transactions on Engineering Management, 71, 6753-6769.

[11] Niculescu, V., Chisalita-Cretu, M. C., Osman, C. C., & Sterca, A. Model-Driven Development Using LLMs: The Case of ChatGPT.

[12] Gange, G., & Stuckey, P. J. (2020, September). The argmax constraint. In International

Conference on Principles and Practice of Constraint Programming (pp. 323-337). Cham: Springer International Publishing.

[13] Warren, D. H., & Pereira, F. C. (1982). An efficient easily adaptable system for interpreting natural language queries. American journal of computational linguistics, 8(3-4), 110-122.

[14] Chowdhury, P. (2025). Sustainable Manufacturing 4.0: Tracking Carbon Footprint In SAP Digital Manufacturing With IOT Sensor Networks. Frontiers in Emerging Computer Science and Information Technology, 2(9), 12–19. https://doi.org/10.37547/fecsit/volume02issue09-02

[15] Zhaolin Yuan, Ming Li, Chang Liu, Fangyuan Han, Haolun Huang, Hong-Ning Dai, Chat with MES: LLM-driven user interface for manipulating garment manufacturing system through natural language, Journal of Manufacturing Systems, Volume 80, 2025, Pages 1093-1107, ISSN 0278-6125, https://doi.org/10.1016/j.jmsy.2025.02.008.

[16] Cristian I. Garcia, Marcus A. DiBattista, Tomás A. Letelier, Hunter D. Halloran, Jaime A. Camelio, Framework for LLM applications in manufacturing, Manufacturing Letters, Volume 41, Supplement, 2024, Pages 253-263, ISSN 2213-8463, https://doi.org/10.1016/j.mfglet.2024.09.030.

[17] Naik, S., Aitharaju, P., & Bandari, S. S. G. (2025). AI Chatbots in Enterprise solutions: Transforming customer support, Industry-Specific challenges and ethical considerations. Glovento Journal of Integrated Studies (GJIS), 1, Article 11. http://doi.org/10.63665/gjis.v1.11

[18] Zhou, Y. and K. Yang. Exploring tensorrt to improve real-time inference for deep learning. in 2022 IEEE 24th Int Conf on High Perfor-mance Computing & Communications; 8th Int Conf on Data Science & Systems; 20th Int Conf on Smart City; 8th Int Conf on Dependa-bility in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys). 2022. IEEE.

[19] Chandra, R., Bansal, R., & Lulla, K. (2025). Benchmarking techniques for real time evaluation of LLMs in production systems. International Journal of Engineering, Science and Information Technology, 5(3), 363–372. https://doi.org/10.52088/ijesty.v5i3.955