The International Journal of Advanced Manufacturing Technology (2024) 132:2715–2733
https://doi.org/10.1007/s00170-024-13492-0

ORIGINAL ARTICLE

An experimental hybrid customized AI and generative AI chatbot human machine interface to improve a factory troubleshooting downtime in the context of Industry 5.0

Kahiomba Sonia Kiangala[1] · Zenghui Wang[2]

Received: 18 January 2024 / Accepted: 19 March 2024 / Published online: 3 April 2024

Abstract

Throughout industrial revolutions, equipment downtime mitigations have been one of the ultimate goals of most factories. Several tools, such as human machine interface (HMI) alarming systems or predictive maintenance schedules, assist in reducing system downtime but still depend on the operators' ability to swiftly retrieve, understand, and efficiently act upon reported failures. We propose the design of a hybrid experimental artificial intelligence (AI) and generative AI chatbot HMI that effectively extracts factory equipment conditions that are useful for troubleshooting and predictive maintenance analysis. We achieve these functions by feeding experimental factory-monitored data to the customized chatbot application tool running in its back-end, a Langchain agent linked to the OpenAI GPT−3.5 language model (LM) via OpenAI APIs. We design our chatbot front-end with Streamlit, an open-source web app. In the context of I5.0, our chatbot HMI uses personalized natural language, English, to interact with the operator, making the information extraction more understandable. We also integrate the generative AI capability of the GPT 3.5 LM that augments the factory data based on the loaded format to create a larger dataset for additional tasks like machine learning modelling. The experimental results show the accuracy of our customized chatbot HMI when retrieving data based on specific prompts and the advantages of a reduced troubleshooting time compared to operations in traditional factories, which are highly dependent on supervisors' interventions. Our study provides a valuable example of upgrading standard factory HMIs to I5.0-capable ones by implementing customized AI and generative AI chatbots within operational industrial environments.

Keywords Equipment downtime · Factory chatbot · Generative AI · Human machine interface (HMI) · Industry 5.0 · Troubleshooting

1 Introduction

For over a decade, the industrial sector has experienced a tremendous revolution with the advent of the Industry 4.0 (I4.0) paradigm [1–3]. This concept, also known as smart factory, focuses on implementing advanced and intelligent technological concepts such as artificial intelligence (AI) and machine learning (ML) [4] to improve production processes' efficiency and automation. Since 2021, a new industrial revolution, Industry 5.0 [5], has slowly been pathing its way into the industry. It builds on top of the existing I4.0 technological principles while shifting the attention toward smart manufacturing [6] systems that are more operator-centric (human-centred), sustainable, and resilient [7].

AI is the driving force behind the deployment of Industry 4.0. It plays a critical role in its implementation, bringing forth benefits such as the creation of automated tasks to improve system productivity and reduce human error, increased efficiency in decision-making, and prevention of system failure with data insights analysis (predictive maintenance); and the design of more resilient production systems reducing overall production costs, growing competitiveness,

✉ Kahiomba Sonia Kiangala
sokiangala@gmail.com

Zenghui Wang
wangzengh@gmail.com

1 Department of ElectricalEngineering, College of Sciences, Engineering and Technology (CSET), University of South Africa (UNISA), 28 Pioneer Ave, Florida Park, Roodepoort, Johannesburg 1710, Gauteng, South Africa

2 Department of Electrical and Mining Engineering, University of South Africa (UNISA), 28 Pioneer Ave, Florida Park, Roodepoort, Johannesburg 1710, Gauteng, South Africa

and increasing customer satisfaction [8]. Being built on similar technological principles, in Industry 5.0, AI carries the same promises and potential, focusing on strategically re-inserting the human factor within production processes to create more sustainable outcomes for factories and the planet. With its emphasis on human well-being, I5.0 also responds to the long pending concern of job losses in industries due to the introduction of AI-automated technologies and solutions in production processes. In addition to the advantages mentioned above, I5.0 aims to offer human-centered AI solutions that upskill operators to efficiently interact with machines and reduce waste and energy consumption, which is beneficial to the environment [9].

While looking at the great benefits of AI and Industry 5.0 in the manufacturing sector, monitoring the challenges emanating from their implementation is crucial to mitigate the risks of substantial failures and losses within factories during the execution phase. Some of these challenges are a call for skilled workers to understand the high level of new technological concepts, a high initial capital required with often unclear return on investment in the initial phase, a need for more data collection and security, and an ethical and legal responsibility in the implementation of these new technologies to avoid more job losses and outcome bias [10].

With the advent of I4.0 [11], manufacturing plants deal with more devices to monitor and a high amount of data to process [12]. In the context of I5.0, factories need to find the right balance between effective system monitoring, reducing equipment downtime and operators' growth and well-being. Techniques like predictive maintenance [13, 14], empowered by several ML algorithms such as convolutional neural network (CNN) support vector machine (SVM) classification [15], successfully mitigate system downtime by detecting potential equipment threats before they become fatalities. Beyond the predictive maintenance technological ability to keep systems running, I5.0 calls for an additional societal layer that personalized human-to-machine interaction employing natural language.

Chatbots are automated conversational tools currently utilized in various sectors [16, 17] to create low-cost, efficient, customized, intelligent interactive systems between humans and machines [18]. In recent years, generative AI [19], an AI branch that generates new data based on a specific input, has been the driving force behind chatbot creations. ChatGPT [20] is a powerful, ready-to-use generative AI large language model (LLM) tool enabling users to enter almost any kind of query and receive back insights in the requested format (texts, videos, or images). While ChatGPT and generative AI conversational models produce remarkable results in several academic and business areas [21], their integration still needs to catch up in the operational manufacturing sector [22].

We propose the design of an experimental hybrid customized chatbot HMI that combines AI and generative AI functions to:

1. Improve factory troubleshooting downtime by enabling easy and swift equipment information retrieval while keeping factory data secure (AI function of the chatbot)
2. Build a personalized HMI responding to natural language prompts to meet I5.0 human-centric requirements
3. Augment factory dataset by generating new data in the desired format to facilitate further ML model building and training purposes (generative AI function)

We structure the remaining sections of the paper as follows: In Section 2, we present a background and literature overview on some key concepts utilized in the research, such as the shift from I4.0 to I5.0, and summarized benefits and challenges of AI and I5.0, generative AI technologies, LLMs and ChatGPT, equipment failure downtime algorithms in factories, and the use of chatbots in manufacturing plants. In Section 3, we describe the theoretical design, the architectures, and the flowchart of the customized Chatbot HMI. Section 4 presents the experimental results of the hybrid chatbot, and we conclude the research in Section 5 with valuable insights on design considerations, limitations, and guidelines for future works. We present in Fig. 1 a process flow diagram of the manuscript.

2 Research background and literature review

2.1 From industry 4.0 to industry 5.0

In 2011, the industrial world embraced its fourth industrial revolution (4IR) [23], commonly known as Industry 4.0 or Smart Factory [24, 25], to a high-tech scheme introduced by the German government (Industrie 4.0). It promotes the implementation of advanced technological concepts that create intelligent and connected systems in the manufacturing environment to form cyber-physical production systems (CPPS) [26, 27]. The fourth industrial revolution came after three of its predecessors equally impacted the industrial sector in their eras. The first was characterized by a drastic shift from manual production lines to machine-boosted production powered by water or steam. The second one was initiated thanks to the discovery of electricity, which increased production lines' productivity with faster production processes. Integrating field-level controllers such as a programmable logic controller (PLC) that enabled automated production processes marked the beginning of the third industrial revolution.

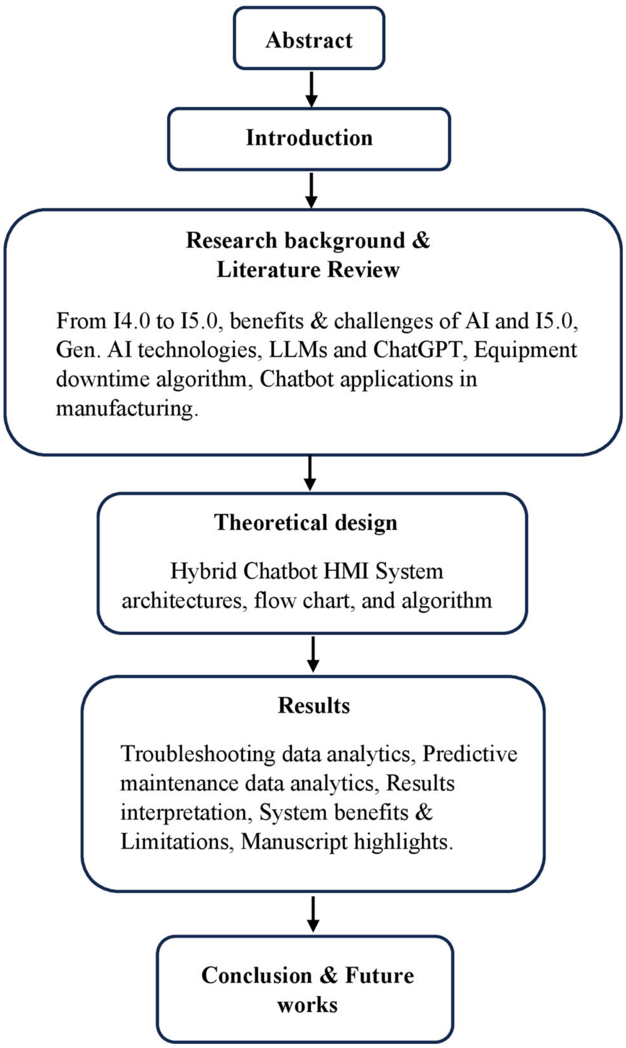


Fig. 1 Manuscript structure flow diagram

Under the 4IR, advanced technological concepts empower production processes to make more intelligent judgments and enable real-time data exchange between factory devices and stakeholders [28]. The Smart Factory concept also brings more flexibility in production, allowing mass production efficiency while simultaneously achieving personalized products [28, 29].

While I4.0 was effectively penetrating the industrial sector, a new revolution, the fifth industrial revolution, also known as Industry 5.0 (I5.0) [30, 31], came along. Industry 5.0 is denoted by a move of focus from advanced technological methods that aim to increase production efficiency and output to more societal objectives [32]. These societal objectives aim to put factory operators at the core of the production process to promote their growth and well-being. They also advocate for the respect of our planet's ecosystem while running manufacturing processes [15, 20]. Therefore, I5.0 came to build, on top of the existing I40 paradigm, the ability to consider our planet's sustainability and workers' social integration in production processes [33]. Figure 2 summarizes the key differences between I4.0 and I5.0, as highlighted by López, A. et al. [1].

2.1.1 Summarized benefits and challenges of Industry 5.0

AI, through some of its popular branches, such as ML, significantly impacts Industry 5.0. Often, when referring to the current industrial revolutions (Industry 4.0 and Industry 5.0) that created smart factories, autonomous systems, and a considerable level of digitalization, people talk about AI. It is understandable because AI is one of the primary tools driving the conceptualization and implementation of these new industrial revolutions within factories. Some of the benefits of Industry 5.0 include [8, 9]:

- Increased productivity: This is achieved by creating intelligent, highly automated, and decentralized systems.
- Improved decision-making: Data analytics is a key component in implementing solutions (abnormality detection, production forecast, predictive maintenance) that support decision-making in factories.
- Flexible and sustainable production systems: Flexible and scalable production systems facilitate product individualization and increase customer satisfaction and competitiveness.
- Reduced energy consumption and wastage are one of the essential goals of I 5.0. Unlike the previous industrial revolutions, I 5.0 takes to heart environmental responsibility and ecosystems to lessen harm to the planet and human beings.
- Human-centered activities: Industry 5.0 advocates smart re-insertion of the human factor in production pro-

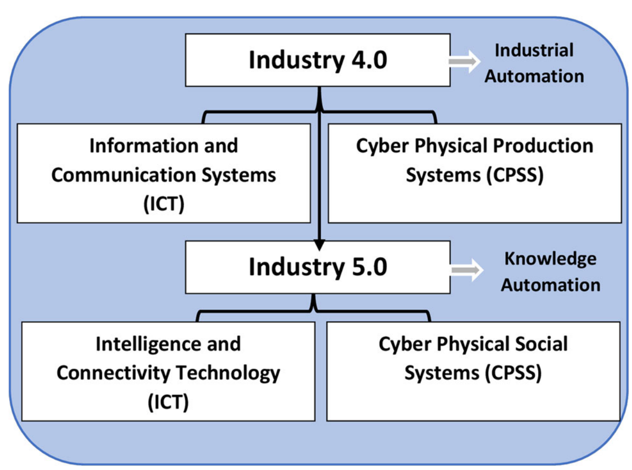


Fig. 2 Industry 4.0 versus Industry 5.0 adapted from López, A. et al. [1]

cesses to avoid job losses. It requires better collaboration between operators and machines.

Some of its challenges are [10]:

- Need for skilled workers: To understand the new technological concepts involved in I5.0, skilled and knowledgeable workers are needed.
- High investment required: Implementing I5.0 requires a high investment to transition between legacy systems and new technologies, offer proper training to workers, and hire skilled workers or experts.
- Data collection and security: A successful implementation of I 5.0 demands more data analytics, which requires even more protection and security.
- Ethics: While I5.0 promises to put human beings at the heart of its activities, it highly depends on human ethical values to create solutions that promote their own wellbeing while producing truthful results.

3 Brief overview on generative AI tools

Generative AI is a branch of AI that utilizes generative algorithms and models to create new content, such as texts, videos, sound, images, code, and other types of data based on a user's input [34]. One of the most recent considerable successes of generative AI was the ability to train its Generative Pretrained Transformer 3 (GPT-3) model on over 500 GB of data with over 100 billion parameters, achieving outstanding results for universal language description and interpretation. As per Brown, T.B. et al. [35], GPT-3 outperforms most pre-trained state-of-the-art language models. In November 2022, generative AI became more popular with the official launch of the generative pretrained transformer (GPT) chatbot application called ChatGPT by the company OpenAI. OpenAI utilized reinforcement learning with human feedback (RLHF) to fine-tune the GPT-3 model, which made the creation of ChatGPT possible [36]. ChatGPT was attractive to users because of its ability to answer queries in a human-like conversation format [37]. ChatGPT's ease of use and openness to the public have drastically impacted people's perception of the utilization of AI and ML, which were seen as technologies reserved for high-skilled developers and engineers. It was reported that in the last year, ChatGPT became a reference for various research and interactive requests on the internet. In fact, upon its release, ChatGPT reached over 100 million users worldwide [38].

Large language models (LLMs) are technologies used to generate texts exclusively in the generative AI field. For the past year, LLMs have also become a center of interest for several researchers for their ability to produce very precise human-like chats and conversations. For this reason, various customized applications opt to integrate pretrained LLMs in their back-end rather than trained models that are even time-consuming to build. Despite the explosion of ChatGPT's experimentation, research on the practical implementation of ChatGPT and generative AI techniques within the manufacturing environment remains scarce [19].

4 Factory equipment downtime estimation algorithm

Over the years, strategies such as predictive maintenance [39, 40] have helped factories improve equipment breakdown time by performing early machine failure detection before they become production-threatening. However, the method still depends on the operator's capacity to interpret the fault, understand the action required, and actively implement it without wasting more time.

There are various components and variables impacting factories' equipment downtime estimation. They are not fixed and can vary depending on a plant type or size. The following are samples of basic elements practical to approximate a small factory equipment downtime. Let us assume the following variables:

- Total number of factory equipment: μ,
- Number of a factory's maintenance operators: α,
- Average maintenance time per equipment: β,
- Buffer time to analysze and understand faults to start maintenance: Z.

The total number of equipment that failed in the plant, Δ, can be estimated by Eq. 1 as follows:

$$\Delta = \sum_{x}^{\mu}(x * y) \tag{1}$$

where $x \epsilon N^*$ and represents the factory equipment count increment and y is the equipment health status. We define y by the following:

$$y = \begin{cases} 0 & \text{if the equipment is healthy} \\ 1 & \text{if the equipment is faulty} \end{cases}$$

From the above variables, the overall factory's equipment downtime θ can be estimated by Eq. 2 as follows:

$$\theta = \frac{\Delta * \beta}{\alpha} + Z * y \tag{2}$$

By substituting Eqs. 1 in 2, θ can further be expanded in Eq. 3 as:

$$\theta = \frac{\sum_{x}^{\mu}(x * y) * \beta}{\alpha} + Z * y \tag{3}$$

From Eq. 3, we can understand that the more maintenance operators are involved in troubleshooting, the less equipment downtime is. Unfortunately, increasing the number of maintenance operators can be costly for the factory. In real factory operations scenarios, not all operators can be allocated to troubleshooting during production, as several other plant parts require attention.

A more cost-effective way to control θ would be to reduce the value of Z, the troubleshooting buffer time, which depends on the operators' or supervisors' ability to analyze and detect equipment issues. Our customized chatbot HMI aims to support operators in reducing Z by providing precise factory equipment data analytics necessary for troubleshooting.

5 Popular chatbots use cases in the manufacturing industry

The manufacturing industry has demonstrated its readiness to implement various AI applications within its production processes. However, chatbot applications have yet to become popular within the field. Wang, S. et al. [41] highlight three primary use cases for implementing chatbots in the manufacturing industry.

1. For training purposes: A well-known application of chatbots in the education sector is in the training and orientation area. Chatbots can give trainees or students the knowledge required to achieve their goals. By exchanging questions and answers, the chatbot system can determine the trainees' gaps and direct the information-feeding flow according to the audience's needs [42]. In factories, maintenance supervisors can apply this training capability to train new operators in assembly lines regularly. Chatbots will act as supervisors' experts to provide all required information to new operators based on their current system knowledge.
2. For factory problem recording and monitoring: Factory designers can integrate chatbots with advanced visual aid functions to provide answers regarding visually monitored conditions [43]. Although this method can appear costly, especially for small- to medium-sized enterprises, because of the corresponding cost related to licenses to enable visual functionalities (for example, currently acquiring the GPT 4 monthly subscriptions as the free GPT 3.5 does not have this capability yet) and other optical equipment to configure (cameras and the corresponding drivers) or install in the assembly lines, this task can be convenient for new operators to confirm process guidelines especially when dealing with complex activities that could confuse them.
3. For maintenance guidelines and assistance: With or without visual aids, chatbots can help operators perform equipment repairs and maintenance by elaborating on steps to follow to achieve the outcome. Using the visual aid or in a question-answer (QA) text format, the worker should give the precise and appropriate details for the system to identify the item to repair and the step from which the operators require assistance.

Our customized chatbot HMI combines a monitoring and maintenance use case (without visual aid) created from factory historical data to provide valuable insights on equipment health statuses for troubleshooting purposes.

6 Customized AI and generative AI chatbot HMI theoretical design

The customized AI and generative AI chatbot has two essential components: a front-end, which serves as the interface between the operator's requests and the factory data and a back-end, containing the system's brain power, the OpenAI GPT 3.5 language model (LM) to understand the operator's inquiries and advise on a suitable solution.

We design the factory's front-end with Streamlit [44], an open-source tool to develop data science and machine learning (ML) web applications. The factory's customized application has a security layer to it. It allows to safely process data in a separate platform without exposing it to open fields such as the web. Loading factory data straight in an open platform like ChatGPT would have this security risk. We use Python programming to create the front-end application in Streamlit and connect it to the back-end unit (GPT 3.5 LM model) using OpenAI APIs with an exclusive API key, ensuring more data security.

We store the factory data in a CSV file. It is a convenient format for most controllers, such as PLCs that can automatically save information in this format. Unfortunately, by default, LMs do not understand CSV file format. They understand texts. To resolve this issue, we incorporate a Langchain agent via Langchain libraries, which understand CSV, into our back-end unit. The Langchain agent becomes the tunnel linking our factory data to the LM. Therefore, we can train GPT 3.5 LM on the factory data, which becomes the primary source of information to answer users' prompts.

The chatbot HMI's goal is to improve the equipment downtime in Eq. 3 by decreasing "Z," the buffer time required for a plant maintenance supervisor to analyze overall data and retrieve faulty devices' information. Based on historical insights, the factory's general requirements would be to timeously:

- Find all faulty devices
- Find their locations
- Identify actions to take on the faulty devices
- Read the sensor value causing the fault
- Generate a report on faulty devices for more analytics

7 Customized AI and generative AI chatbot HMI flow diagram and architectures

In Fig. 3, we present the customized chatbot HMI process operation flow diagram. The flowchart has two sections: the front-end web app and the back-end LM (GPT 3.5) brain. Through the first section, the operator loads the factory data in CSV format to train the LM in the back-end and sends queries related to the loaded factory data. The web app receives operators' prompts and links them to the back-end to extract the corresponding information. It also displays the final solution obtained by the back-end at the end of the process in a natural language format. We illustrate the second section of the flow chart (the back-end thinking process) in Fig. 4. It is the system's brain that performs all the intelligent "thinking routines" to provide the most accurate answer to the operator.

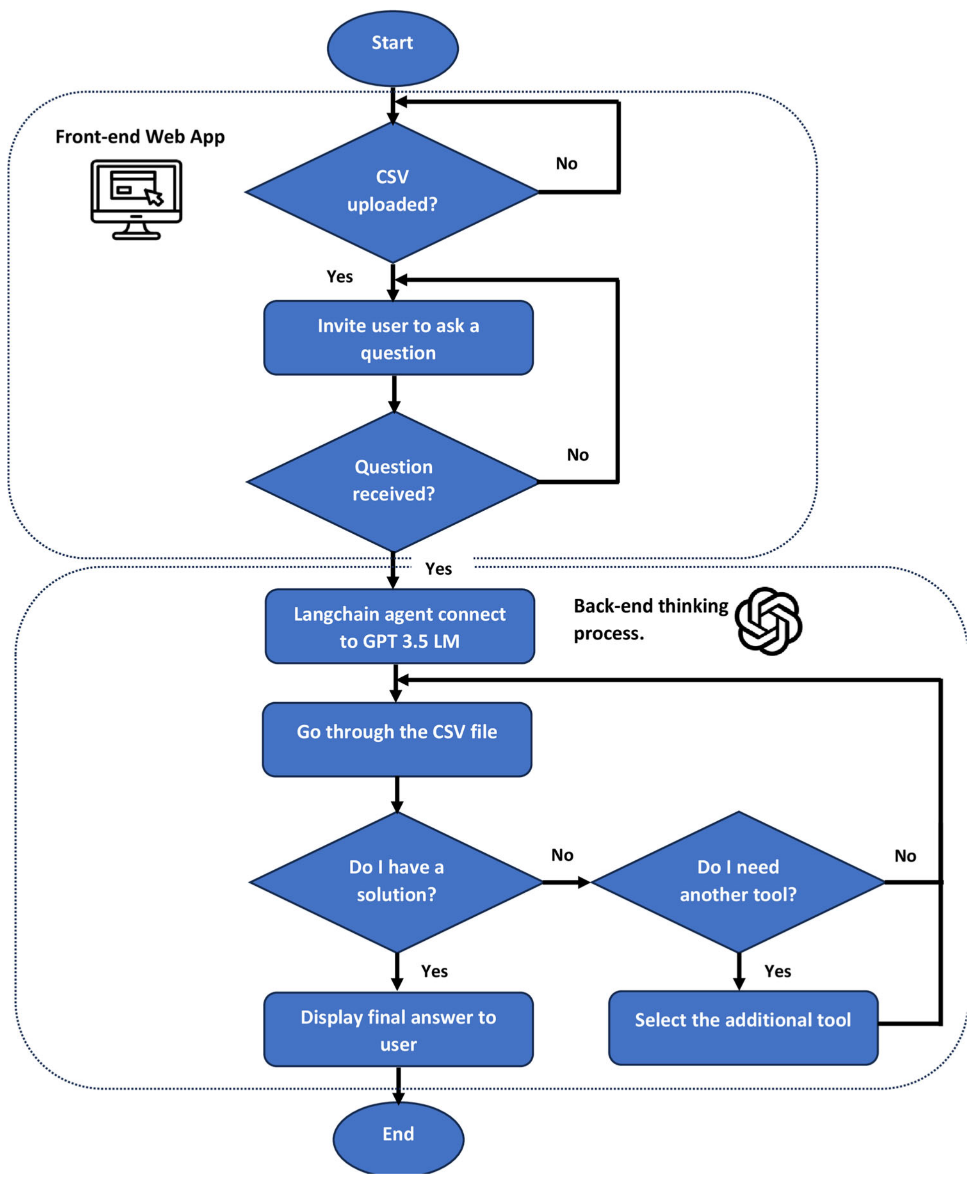


Fig. 3 Customized AI and generative AI chatbot HMI flow diagram

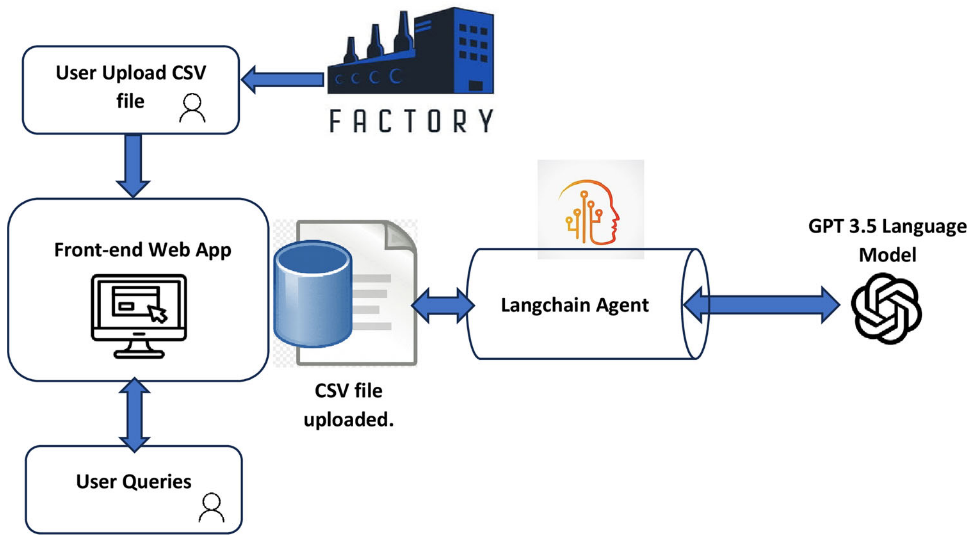


Fig. 4 Customized AI and generative AI chatbot HMI simplified architecture

We display in Figs. 5 and 6, the customized chatbot HMI web application front-end before and after loading the factory csv data respectively. Figure 5 shows the initial look of our customized chatbot HMI web application front-end before loading the factory data for training. It is the welcome page of the chatbot, the first page the user sees when launching the chatbot HMI application. It contains the following primary information:

- A title that summarizes the web application's function: **"HMI Maintenance ChatBot Assistant"**
- An instruction on the factory training data format: **"Upload the factory monitored data in CSV format."** The factory data to load in the application should be in a CSV file.
- An instruction on how to load the factory data: **"Drag and drop file here" and "Browse files."** A user can load the CSV file by dragging and dropping it straight to this page (to the area written **"Drag and drop file here")** or by browsing through the computer's location containing the file (by clicking on the "Browse" button.) It also includes the size limitation of the CSV file to load in the application (200MB): **"Limit 200MB per file • CSV"**
- Another descriptive note of the chatbot HMI back-end content which is: **"This is a customized Generative AI HMI chatbot for an experimental factory. It is running GPT 3.5 as its language model."** This note is visible on all the Chabot HMI front-end pages and does not appear in some figures in the manuscript (Figs. 8, 10, 11, 12, and 13) due to space limitations.

Note: The information mentioned above about the web application front-end has the same meaning in all other figures of the manuscript displaying our customized chatbot HMI front-end (Figs. 6, 8, 10, 11, 12, and 13).

Figure 6 presents the chatbot HMI web application front-end page after successfully loading a CSV file for training. In addition to the above information, it displays:

- The loaded CSV file name and size (73.7KB) as: **"Factory Equipment Statuses.csv 73.7KB"**
- An instructional prompt for the user to send queries about the loaded CSV file: **"Ask me questions about**

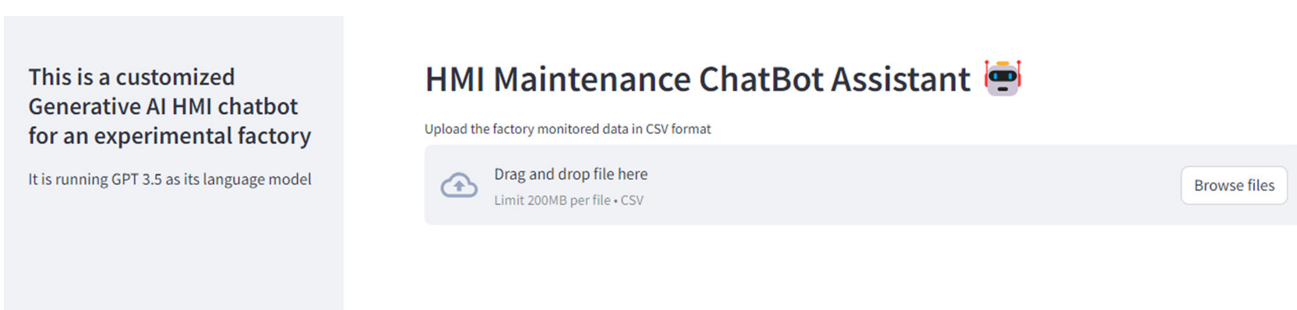


Fig. 5 Customized chatbot HMI web app front-end before loading a CSV file

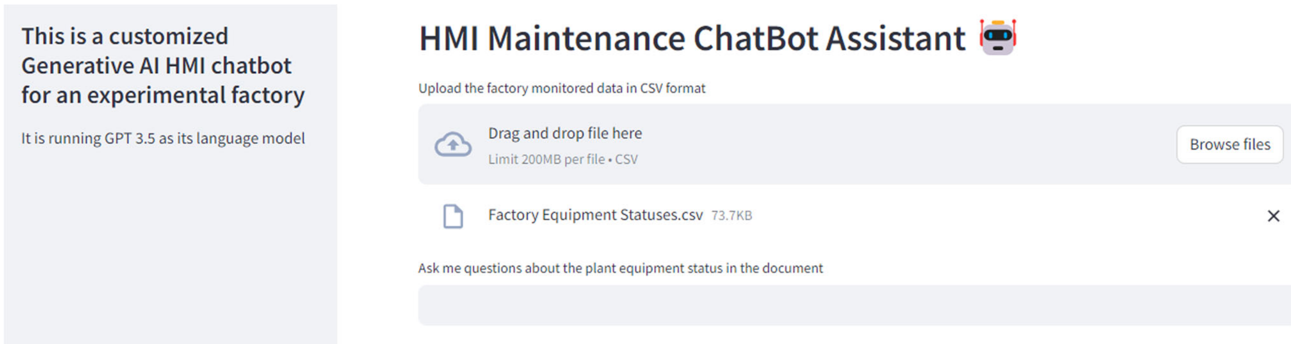


Fig. 6 Customized chatbot HMI web app front-end after loading a CSV file

the plant equipment status in the document." It is the same prompt in Figs. 8, 10,11, 12, and 13.

- An input text box area, just below the instructional prompt, where the user can type in the desired query. It has the same function in Figs. 8, 10, 11, 12, and 13.

Figure 4 is a simplified model of the customized chatbot HMI displaying the main system stakeholders: the factory data (in CSV format), the operator (loading data and sending prompts), the front-end web app receiving loaded data and prompts to transmit to the back-end, the Langchain agent playing the role of an interpreter between the loaded CSV data and the GPT 3.5 LM, and the GPT 3.5 LM itself.

Figure 7 shows the query/answer process architecture. It portrays how the Langchain agent, boosted by the GPT 3.5 LM, initiates a "thinking process" when receiving a prompt from the user. It goes through the CSV file and other tools made available by the LM until it finds the best answer for the operator.

Algorithm 1 is a translation of Fig. 7 describing the essential steps integrated into the customized chatbot HMI design. As mentioned in the previous section, we designed our customized chatbot (front-end and back-end) with Python programming language.

8 Experimental results

Note: We assume that a factory expert prepared the data used to train our system in an understandable format (example of data format in Table 1) after accurately learning the equip-

Fig. 7 Customized chatbot HMI query/answer process architecture

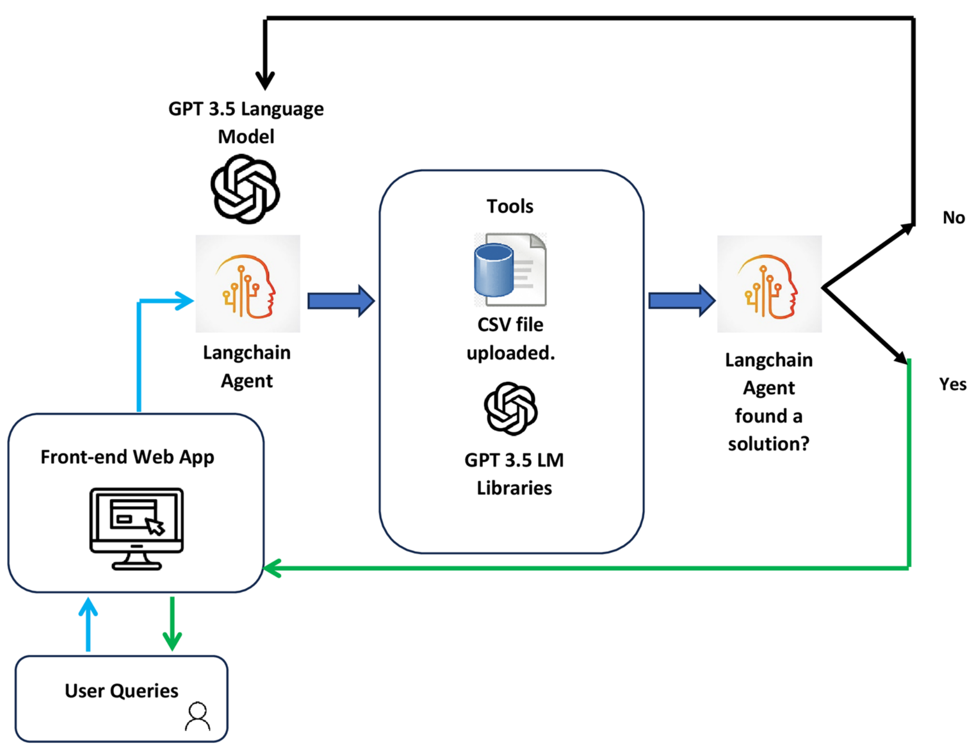


Table 1 Factory data CSV parameters and format

Parameters	Value	Comment
Equipment name	Conveyor1 to Conveyor850	This is the name of each equipment monitored in the factory. In this experimental factory we use conveyor numbers to identify them. In total there are 850 conveyors to monitor
Location	Furnace area, assembly area, packaging area, storage area, capping area	We perform our experiment on five main areas where the different conveyors are located
Sensor value	20–90	This is a numerical value of the conveyor sensor reporting a fault. The testing range is between 20 and 90, with 20 the minimum value (healthy state) and 90 the maximum (unhealthy state)
Status	No alarm, minor alarm, ON warning, alarm ON, high alarm ON	These are the equipment alarm level statuses from healthy to unhealthy
Action	Healthy—no action required. Check on the physical equipment for possible abnormalities. Start maintenance on next stop. Immediate stop	This field contains the recommended actions to take for each alarm status

ment's behavior and that its content is correct and accurate. Because of the token limitation a new dataset is used for testing the generative AI functions with five existing initial conveyors (from prompts 9 to 11 of Table 2).

8.1 Factory data equipment analytics for troubleshooting

We test the customized HMI chatbot's effectiveness using equipment data status from an experimental factory. Table 1 displays data parameters recorded for the equipment in the CSV file loaded for training the GPT 3.5 LM:

From the designed HMI chatbot front-end in Fig. 6, we run eight prompts to retrieve the most significant information for the plant troubleshooting based on the loaded dataset

Algorithm 1 Customized chatbot HMI process algorithm.

```
Install and import required libraries
Generate a factory.py script file
Set config. page for Streamlit web app
Set required title/subtitle for the page
Set LM (GPT 3.5) and Agent parameters
Prompt operator to load the csv file
if csv file is loaded then
    Prompt operator to ask a question
else
    Wait for operator's question
end if
if question is not empty then
    Start processing response timer
    Pass question to the CSV agent
    Print CSV agent response
    Print processing time
else
    Wait for a valid question
end if
Call Streamlit function to run factory.py
End
```

(AI function of the customized chatbot) in Table 1 and three prompts to generate a new dataset created by the chatbot with similar characteristics than the initial loaded data. It is the generative AI feature of the customized chatbot. Table 2 presents the prompts we ran, their answers, and processing time. Figure 8 displays prompt 8/answer 8 as read from the HMI chatbot.

As mentioned in the previous section, when receiving a prompt, our chatbot HMI goes through several steps before ensuring it has the final correct answer (the LM agent in the background as detailed in Fig. 7). We present in Fig. 9 the HMI chatbot thinking process for prompt number 7.

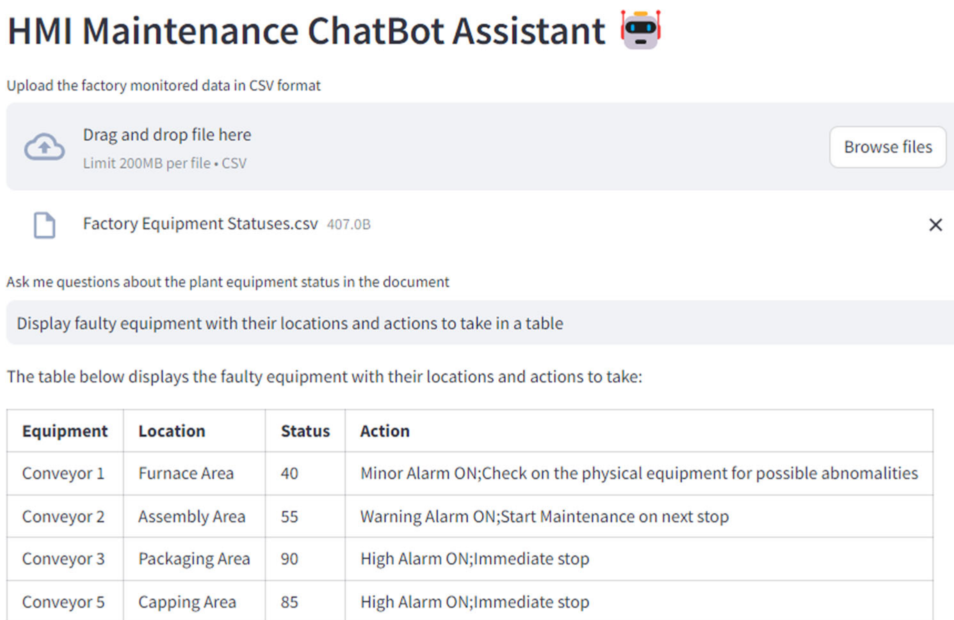
Figure 10 shows the generative AI capability of our chatbot that created five more factory equipment with similar characteristics to the loaded dataset. The dataset initially has five conveyors (from conveyor 1 to conveyor 5).

By analyzing the prompt results in Table 2, we notice that it takes less than 10 s to process and receive an answer from a single prompt. Therefore, on average, it would take less than 2 min to run the overall data analytics for the factory from our customized HMI chatbot for all 850 conveyors and extract the accurate troubleshooting insights to start maintenance processes, if required. From Eq. 3, our solution mitigates the total troubleshooting time by considerably reducing the dependency on Z, the analytics supervisor's waiting buffer time. The customized chatbot is easy to use and can be processed by a standard operator without solely relying on the supervisor's initial analytics. From Eq. 3, considering a very negligible value of Z, the overall value of θ can be further improved to Eq. 4:

$$\theta = \frac{\sum_{x}^{\mu}(x * y) * \beta}{\alpha} \tag{4}$$

Table 2 Customized chatbot HMI troubleshooting prompts, answers, and processing time results

Prompt number	Prompt	Answers	System processing time	Results (correct or false)
1	What equipment is in danger (without a question mark at the end)	The equipment in danger is Conveyor 3 and Conveyor 5	3.86 s	Correct
2	What equipment is in danger? (with a question mark at the end)	Agent stopped due to iteration limit or time limit	13.90 s	False
3	which equipment is at risk	Conveyor 1, Conveyor 2, Conveyor 3, Conveyor 5	5.12 s	Correct
4	why Conveyor 1, Conveyor 2, Conveyor 3, Conveyor 5 are at risk	Conveyor 1, Conveyor 2, Conveyor 3, and Conveyor 5 are at risk because they have a Status of 'Minor Alarm ON,' 'Warning Alarm ON,' or 'High Alarm ON'	5.90 s	Correct
5	why Conveyor 3 and Conveyor 5 are at risk?	Conveyor 3 and Conveyor 5 are at risk because their Sensor Value is 90 and 85 respectively, and their Status is High Alarm ON	5.31 s	Correct
6	Where are Conveyor 3 and Conveyor 5 located?	Conveyor 3 is located in the Packaging Area and Conveyor 5 is located in the Capping Area	3.28 s	Correct
7	Which action to take for Conveyor 3 and Conveyor 5?	The action to take for Conveyor 3 and Conveyor 5 is "Immediate stop"	5.94 s	Correct
8	Display faulty equipment with their locations and actions to take in a table	See Fig. 8	8.52 s	Correct
9[1]	Generate 5 more equipment with similar characteristics	5 more equipment with similar characteristics have been generated	6.65 s	Correct
10[1]	What are the 5 new equipment generated	The 5 new equipment generated are Conveyor 6, Conveyor 7, Conveyor 8, Conveyor 9 and Conveyor 10	5.46 s	Correct
11[1]	Show the 5 more equipment in a table	See Fig. 10	5.21 s	Correct



HMI Maintenance ChatBot Assistant

Upload the factory monitored data in CSV format

Drag and drop file here
Limit 200MB per file • CSV

Browse files

Factory Equipment Statuses.csv 407.0B

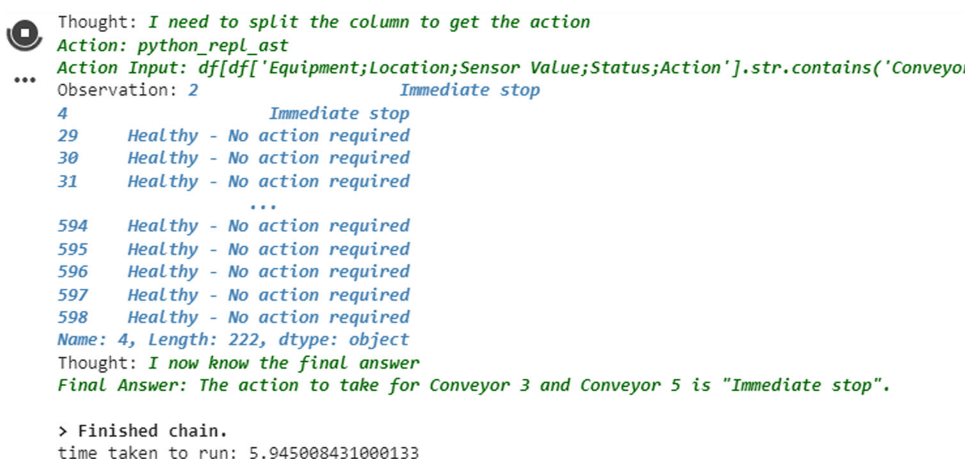
Ask me questions about the plant equipment status in the document

Display faulty equipment with their locations and actions to take in a table

The table below displays the faulty equipment with their locations and actions to take:

Equipment	Location	Status	Action
Conveyor 1	Furnace Area	40	Minor Alarm ON;Check on the physical equipment for possible abnomalities
Conveyor 2	Assembly Area	55	Warning Alarm ON;Start Maintenance on next stop
Conveyor 3	Packaging Area	90	High Alarm ON;Immediate stop
Conveyor 5	Capping Area	85	High Alarm ON;Immediate stop

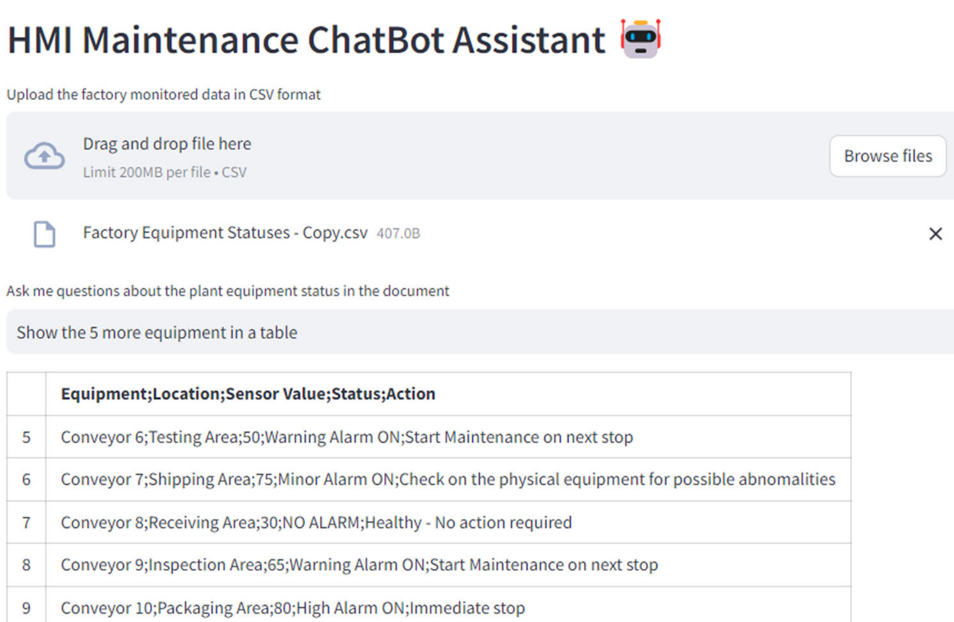
Fig. 8 Prompt and answer 8 results—AI function of HMI chatbot



```
Thought: I need to split the column to get the action
Action: python_repl_ast
Action Input: df[df['Equipment;Location;Sensor Value;Status;Action'].str.contains('Conveyo
Observation: 2                  Immediate stop
4                  Immediate stop
29     Healthy - No action required
30     Healthy - No action required
31     Healthy - No action required
                     ...
594    Healthy - No action required
595    Healthy - No action required
596    Healthy - No action required
597    Healthy - No action required
598    Healthy - No action required
Name: 4, Length: 222, dtype: object
Thought: I now know the final answer
Final Answer: The action to take for Conveyor 3 and Conveyor 5 is "Immediate stop".

> Finished chain.
time taken to run: 5.945008431000133
```

Fig. 9 Langchain Agent thinking process prompt number 7



HMI Maintenance ChatBot Assistant

Upload the factory monitored data in CSV format

Drag and drop file here
Limit 200MB per file • CSV

Browse files

Factory Equipment Statuses - Copy.csv 407.0B

Ask me questions about the plant equipment status in the document

Show the 5 more equipment in a table

	Equipment;Location;Sensor Value;Status;Action
5	Conveyor 6;Testing Area;50;Warning Alarm ON;Start Maintenance on next stop
6	Conveyor 7;Shipping Area;75;Minor Alarm ON;Check on the physical equipment for possible abnomalities
7	Conveyor 8;Receiving Area;30;NO ALARM;Healthy - No action required
8	Conveyor 9;Inspection Area;65;Warning Alarm ON;Start Maintenance on next stop
9	Conveyor 10;Packaging Area;80;High Alarm ON;Immediate stop

Fig. 10 Prompt and answer 11 results—generative AI function of HMI chatbot

Note: Because of the LM in the background, the prompt output language formulation may differ for the same prompts. It should not be an issue if the answer is correct.

8.1.1 Results interpretation

To better understand the significance of our results in Table 2 and Eq. 4 we numerically demonstrate, with some experimental data, how our proposed approach helps minimize equipment downtime during troubleshooting while respecting certain factory constraints. After computing these numerical values, we compare results in systems with and without our proposed chatbot troubleshooting HMI. Table 3 presents an experimental factory's parameters based on the equipment downtime estimation algorithm in Section 4.

We assume that in this experimental factory:

- The parameter symbolizing the average troubleshooting time per equipment β is the time the maintenance operator requires, in worst-case scenarios, to get equipment up and running after a fault. It excludes the analyzing time to pinpoint the faults.
- The buffer time Z to analyze and understand faults excludes the time required for the maintenance supervisors to move from their current location to the faulty area to start analyses.
- The constraint 1, $\alpha \leq 5$, means that in case of critical equipment downtime, the factory can increase the number of maintenance operators to speed up the process, but 5, in this experiment, is the maximum number of operators the plant can afford to deploy. Adding more maintenance operators would mean an extra cost for the factory.

Table 3 Experimental parameters numerical values

	Parameter	Experimental value
1	Number of equipment that failed,Δ	10
2	Number of factory's maintenance operators,α	3
3	Average troubleshooting time per equipment (worst case scenario), β	10 min
4	Buffer time to analyse and understand faults to start maintenance, Z	20 min
5	Equipment downtime,θ	To compute
6	Escalation time,Ω	60 min
7	Chatbot HMI query system processing time to display faulty equipment with actions to take and location like in Fig. 9 (worst case scenario), t	20 s
8	Constraint 1	$\alpha \leq 5$
9	Constraint 2	$\beta \leq 10$

- The constraint 2, $\beta \leq 10$, means that troubleshooting time per equipment should not exceed 10 min. In cases where the troubleshooting calls for more time than β, an escalation process starts implying more downtime, 1 h in this experiment.

To compare the impact of our solution in an experimental factory, we can expand Eq. 4 as follows:

$$\theta = \frac{\Delta * \beta}{\alpha} + Z * g + t * b + \Omega * d \tag{5}$$

With $g \in 0, 1$, $g = 0$ if our proposed chatbot HMI is implemented and $g = 1$ if it is not implemented.
With $b \in 0, 1$, $b = 0$ if our proposed chatbot HMI is not implemented and $b = 1$ if it is implemented.
With $d \in 0, 1$, $d = 0$ if there is no escalation and $d = 1$ if there is escalation.

We test the experimental parameters in the following cases:

- Case 1: System downtime with no escalation and no chatbot HMI

$$\theta = \frac{10 * 10}{3} + 20 * 1 + 20 * 0 + 60 * 0 \tag{6}$$

- Case 2: System downtime with no escalation, no chatbot HMI and more maintenance operators

$$\theta = \frac{10 * 10}{5} + 20 * 1 + 20 * 0 + 60 * 0 \tag{7}$$

- Case 3: System downtime with escalation and no chatbot HMI

$$\theta = \frac{10 * 10}{3} + 20 * 1 + 20 * 0 + 60 * 1 \tag{8}$$

- Case 4: System downtime with chatbot HMI and no escalation

$$\theta = \frac{10 * 10}{3} + 20 * 0 + 20 * 1 + 60 * 0 \tag{9}$$

- Case 5: System downtime with chatbot HMI and escalation

$$\theta = \frac{10 * 10}{3} + 20 * 0 + 20 * 1 + 60 * 1 \tag{10}$$

Table 4 presents a summary of computed system downtime for each case.

From the above numerical values, we notice that by implementing our customized chatbot HMI as a troubleshooting aid, the factory saves up to 20 min in retrieving all necessary

Table 4 Experimental system downtime values

	System setup	Downtime
Case 1	No escalation and no chatbot HMI	53 min
Case 2	No escalation, no chatbot HMI, and more maintenance operators	40 min
Case 3	Escalation and no chatbot HMI	113 min
Case 4	Chatbot HMI and no escalation	33 min 20 s
Case 5	Chatbot HMI and escalation	93 min 20 s

information that detects faulty equipment to locate and repair them. Although it may seem negligible, 20 min of downtime for a factory can be extremely costly when quantifying production losses. Moreover, our customized chatbot HMI considerably reduces the need for additional maintenance operators (which are expensive to the factory) to improve the downtime. As shown in Case 2, even by increasing the number of maintenance operators to the maximum allowed (5) without our chatbot, the factory downtime is still higher (40 min) than when implementing our chatbot with the minimum number of maintenance operators (33 min 20 s).

8.2 Predictive maintenance data analytics

We also implement the customized HMI chatbot to swiftly classify a conveyor motor fault based on several historical parameters. By reading parameter values, the chatbot establishes whether the current motor state is healthy or not and categorizes them into three groups: healthy, minor fault, and critical fault. Therefore, it provides flags to detect abnormalities before they become production-threatening.

We upload in the chatbot a data frame of 15,000 records of seven parameters presented in Table 5:

The predictive maintenance prompt sent to our customized chatbot contains the first six parameter values in Table 5 and outputs the last parameter (Fault Detected) for the predictive maintenance classification schedule. Table 6 presents prompt samples, answers, and processing time for the predictive maintenance analytics. Figure 11 displays the predictive maintenance prompt 1/answer 1 received from the HMI chatbot. Apart from the web application information detailed in Figs. 5 and 6, Fig. 11 displays the predictive maintenance file name and size (**parameters original – version 3.csv 0.6MB**), the first prompt of the predictive maintenance task in Table 6 (vibration speed 322, motor speed 120, current 7.3, belt tension 0.99, motor tension 1.6, temperature 13.78, what is the fault detected?) and the prompt result obtained after processing the query (CRITICAL FAULT) displayed underneath the input text box. The first prompt contains numerical values of the experimental factory predictive maintenance parameters in Table 5 (from 1 to 6) and inquires about the last parameter (7), the type of fault detected (CRITICAL FAULT in this case) based on the numerical values entered in the query. The background chatbot thinking steps and processing time for the predictive maintenance prompts would be like the one displayed in Fig. 9.
Note: Prompts 3 and 4 in Table 6 are the generative AI functions for the predictive maintenance dataset.

Table 5 Predictive maintenance parameters

	Parameters	Units / Value
1	Vibration speed	mm/s
2	Motor speed	m/s
3	Current	A (Amps)
4	Belt tension	N/m
5	Motor tension	N/m
6	Temperature	*C
7	Fault detected	Healthy, minor fault, or critical fault

Figures 12 and 13 are the two prompts (3 and 4) and answers for the generative AI function in predictive maintenance.

An exciting feature of our chatbot is that the order in which the parameters are entered in the system does not matter. It does not have to reflect the original data frame order. We can compare prompt 1 and prompt 2. Although parameters are in different order, they compute the same final answer: CRITICAL FAULT.

The proposed chatbot HMI, thanks to the OpenAI API, processes all queries timelessly, with an average processing time of 5 s in the worst case. It is a remarkable achievement comparing our chatbot solution execution time to a similar predictive maintenance system running a convolutional neural network (CNN) to create a classification model that predicts fault categories based on a 15,000-dataset records input [15].

Table 7 summarizes the CNN predictive maintenance system (hardware and software) key settings and the processing time reached on the dataset. Figure 14 shows the processing time for training the CNN predictive maintenance model based on three epochs, as presented by Kiangala, K.S. and Wang, Z. [15]. The total training time for the CNN model is 3180 s, approximately 1 h (53 min).

As we can observe in Table 6, the generative AI functions take longer (12.84 s at the worst case) than the troubleshoot-

Table 6 Customized chatbot HMI predictive maintenance prompts, answers, and processing time results

Prompt number	Prompt	Answers	System processing time	Results (correct or false)
1	Vibration speed 322, motor speed 120, current 7.3, belt tension 0.99, motor tension 1.6, temperature 13.78, what is the fault detected?	CRITICAL FAULT	4.08 s	Correct
2	Temperature 13.78, motor speed 120, belt tension 0.99, vibration speed 322, motor tension 1.6, current 7.3, what is the fault detected?	The fault detected is CRITICAL FAULT	2.48 s	Correct
3	Generate 5 more observations with similar characteristics	See Fig. 12	12.34 s	Correct
4	Generate 5 more observations with different fault detected	See Fig. 13	5.99 s	Correct

ing tasks dealing with data augmentation. It does not impact the factory's daily production process or troubleshooting actions as this function is implemented off-production.

The predictive maintenance analytics function of our customized chatbot HMI classifies the type of fault detected in a factory based on the received system parameters (parameters in Table 5). In a plant, a monitoring network could read device parameters through sensors and connect to our HMI system to give parameter data access. Our chatbot HMI would then, in real-time, establish whether the parameters imply a critical fault, a minor fault, or no fault at all. This chatbot function produces outcomes like a classifier ML model with the following benefits:

- A shorter training and testing phase processing time, unlike a state-of-the-art ML classifier (comparing Table 7 CNN overall processing time and Table 6 predictive maintenance processing time results).
- An easy web application front-end for data training with no high-level technical skill required. Unlike state-of-the-art ML models (CNN) that demand expert skill to train models, data training in our chatbot HMI is simple and consists of loading a CSV file (containing well-prepared data) in the web application.
- A natural language interface to interact with the system and perform predictive maintenance tasks with little training needed.

8.3 Summarized benefits of using our customized chatbot HMI over traditional HMIs

HMIs [27] are essential tools in the manufacturing sector. They allow operators to control, monitor, and interact with factory equipment to produce the desired outcome. In other words, HMIs help translate from factory machine languages to ones understandable by human operators [45]. In traditional manufacturing environments, a typical HMI consists of a few screens graphically displaying generic information, static or dynamic, such as machine and sensor statuses (healthy or faulty), system alarm conditions, or variable historical trends. It also contains control mechanisms in the form of software buttons or mechanical knobs linked to factory controllers. Traditional HMIs highly depend on operators' and supervisors' capacity to master manufacturing device operations.

With new industrial revolutions and increasing technological advancements, HMIs are becoming more complex due to the large number of factory devices to control and

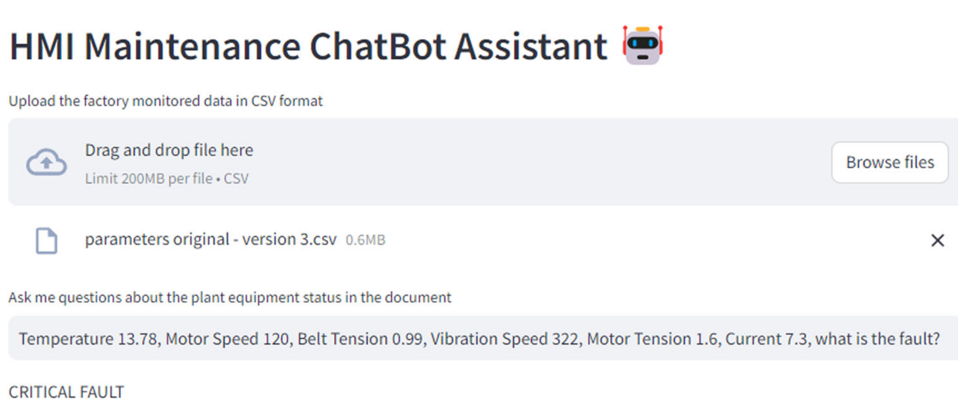


Fig. 11 Prompt and answer 1 result—predictive maintenance AI function of HMI chatbot

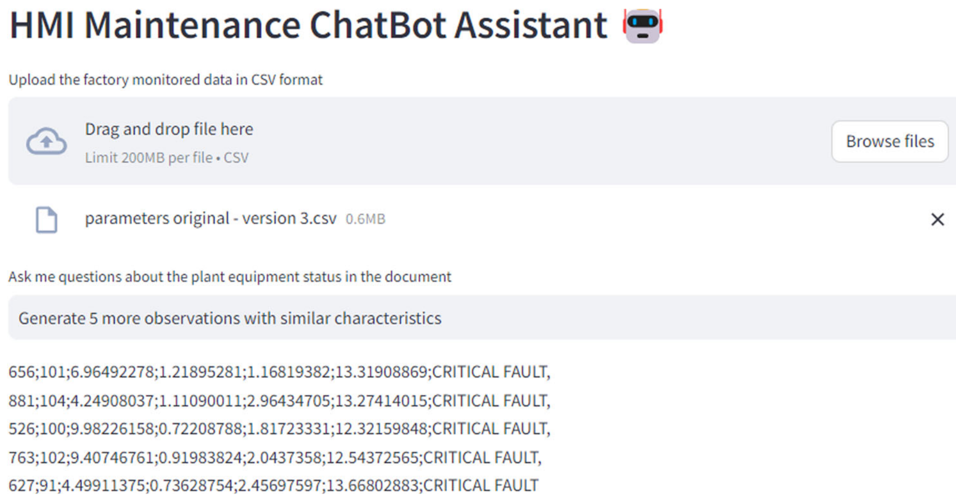


Fig. 12 Prompt and answer 3 predictive maintenance generative AI function

monitor [46]. Although software engineers are still trying to design supervisory control and data acquisition (SCADA) to show the best possible details to users, the system complexity requires special training and more efforts from operators to understand displayed information, icons, signs, interpret messages and act upon them.

In the context of Industry 5.0, the role of an HMI is completely changing [47] from simple SCADA control and monitoring tools to more interactive platforms that operators can utilize efficiently to interpret helpful data for the plant. We develop a customized chatbot acting as an I5.0 HMI with several benefits over traditional HMIs, such as the ability to:

- Learn historical information from the factory and concisely present it upon request. We achieve this advantage by training the GPT 3.5 LM with a factory CSV file database via a Langchain agent and incorporating the result into the customized chatbot HMI web application.
- Generate new datasets similar to the loaded historical data for AI purposes like ML modelling. This functionality is another benefit of the GPT 3.5 LM running in the customized chatbot back-end.
- Interact dynamically with operators in a natural language, English, for this project, making information retrieval smoother and reducing the need for extensive training.
- Act as a maintenance assistant, providing primary insights to troubleshoot faulty equipment therefore reducing the troubleshooting.
- Perform advanced tasks such as predictive maintenance analysis to detect equipment threats before they become fatalities.

8.4 Limitations

Our customized chatbot HMI solution is experimental as we integrate several new concepts and libraries currently being scrutinized and improved by developers and researchers to obtain acceptable results. It is also worth emphasizing that we built our customized solution based on the current

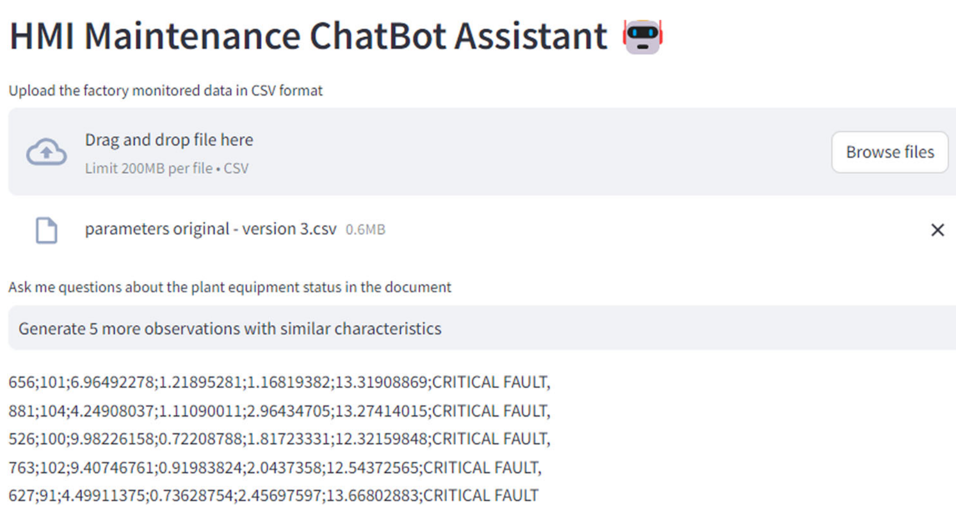


Fig. 13 Prompt and answer 4 predictive maintenance generative AI function

Table 7 Predictive maintenance CNN settings

Parameters	Values	Comment
Computer properties	CPU RAM: 8GB Processor: 64 bits	
Split ratio	0.8	Based on the 15,000 datasets, it means 12,000 data for training and 3000 data for validation
Libraries	Keras and tensorFLOW BACKEND	
Activation functions	ReLU	
Total processing time for 3 epochs	3180 s	See Fig. 13
6	Temperature	*C
7	Fault detected	Healthy, minor fault, or critical fault

GPT 3.5 LLMs functions. Hence, all its present drawbacks implicitly affect our system. The main limitations are as follows:

- Accessing the OpenAI APIs requires a billing fee based on the number of queries and tokens used. For this project, we spent approximately 20$ for all the testing and setup. The OpenAI pricing website (https://openai.com/pricing) provides detailed information. Some other options (Free, open-source APIs from different vendors) could be explored to eliminate this fee.
- Unlike the online version of ChatGPT, which is trained on millions of various data to provide broader answer ranges on several topics, we solely train our customized chatbot HMI on a limited range of factory data. The accuracy of answers depends on how well the factory data is initially structured and polished. When using a CSV file, in this case, columns and contents should be well-denominated

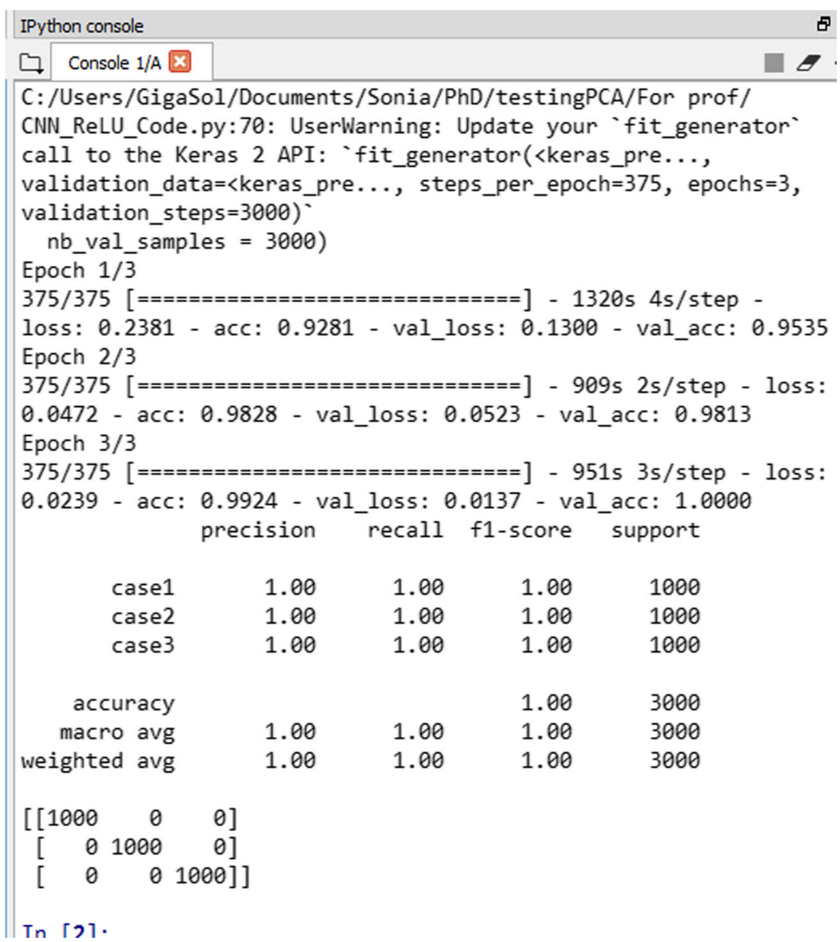


Fig. 14 CNN predictive maintenance output processing time [15]

for the LM to understand their meanings and respond to queries accordingly.

- As displayed in Table 2, we encountered a glitch in the system on prompt 2 where the LM could not find the correct answer to a question because there was a question mark at the end. The LM answers correctly on the same query in prompt 1. It is a limitation of some of the Langchain libraries currently still in experimental phases.
- We intend to create a solution that would be less complex to implement and cost-effective for small- to medium-sized manufacturing enterprises; hence, some of the paid advantages of the GPT models (GPT4) are not included in the system. A limited number of tokens (4096 provided by the GPT3.5 LM by default) are shared between our input queries and the execution time. It limits the amount of data we can process and the length of prompts we can input. For example, the number of new observations or equipment we can prompt the system to generate depends on the size of the current data file from which we are cloning the characteristics. For testing purposes, we reduced the amount of data when creating new observations or monitoring equipment. A solution to this problem would be paying for more tokens or subscribing to GPT 4.
- The dataset size to load from which the chatbot will still answer queries correctly is also conditioned by the token limit size, although our chatbot application can receive up to 200 MB of data.
- We have not automatically linked the factory data CSV file loading process to the chatbot app. The process is still done manually, and we will consider upgrading this task to automatic loading in future versions of the chatbot app.

8.5 Manuscript highlights

- System downtime is costly to factories.
- We design a generative AI chatbot HMI as a tool to alleviate system downtime.
- We customize and integrate the current GPT 3.5 LM functions in our system back-end.
- We obtain a responsive chatbot HMI extracting queried information from trained data.
- The GPT 3.5 LM drawbacks (number of tokens for queries) are a limitation of our tool.
- In I5.0, adding feedback loops between users and generative AI chatbot HMIs could improve decision-making accuracy.

9 Conclusion

In this research, we proposed a hybrid customized AI and generative AI chatbot HMI design for an experimental factory. Using the GPT 3.5 LM trained on factory data in its back-end, the customized chatbot provides operators, upon request, with the factory's equipment details helpful in troubleshooting in case of failure. Our chatbot can also generate new data with characteristics similar to the loaded data thanks to the generative AI feature of the GPT 3.5 LM in the system's brain. We design the chatbot front end with Streamlit, a web application that ensures security by separating the loaded factory data from the unsafe open web. We use additional LM frameworks like Langchain to integrate into the GPT 3.5 LM the ability to read CSV files. The experimental results we achieved by running our chatbot to extract equipment statuses and predictive maintenance data analysis show the system's accuracy in extracting information using natural language. We obtained an accuracy of 90.9% by computing 13 queries on the data retrieval for system troubleshooting and 100% for predictive maintenance analysis. The system processing time for troubleshooting prompts and answers is insignificant (less than 10 s on average), which eliminates the need for a waiting period for a maintenance supervisor to perform data analysis before troubleshooting, therefore improving the factory breakdown time. Our customized chatbot introduces the implementation of I5.0 HMIs to an I4.0 environment. It promotes interactive, personalized human-to-machine communication with natural language (with the operator at the system's center) over graphical and static interactions on traditional HMIs. Our research also contributes to boosting manufacturing SMEs' confidence in complying with current state-of-the-art technologies and industrial revolutions to remain competitive in the fast-growing industrial market.

In future chatbot app versions, we plan to improve the integration of generative AI chatbot HMIs in Industry 5.0 by creating feedback loops between users/operators and the system. This feature can empower the decision-making experience in factories by generating more accurate information based on user prompts. We would achieve this function by notifying the system of wrong or less accurate answers and rewarding better ones. We intend to make the system more autonomous by automating the data training process, saving the factory information in a server, and connecting it directly to the chatbot app. Having the factory data in a secure server would allow us to manipulate the generated outcome (emailing them automatically to supervisors or saving them in separate folders). We also plan on integrating visual aids like graphs and charts to improve the data analytics experience when troubleshooting factory equipment.

Acknowledgements This research is partially supported by the South African National Research Foundation (Grant Nos. AJCR230704126719 and 137951) and the South African Eskom Tertiary Education Support Programme.

Author Contributions Conceptualization: Kahiomba Sonia Kiangala and Zenghui Wang. Data curation: Kahiomba Sonia Kiangala. Formal analysis: Zenghui Wang. Funding acquisition: Zenghui Wang; Investigation, Kahiomba Sonia Kiangala and Zenghui Wang; Methodology, Kahiomba Sonia Kiangala and Zenghui Wang; Project administration, Zenghui Wang; Resources, Kahiomba Sonia Kiangala; Software, Kahiomba Sonia Kiangala; Supervision, Zenghui Wang; Visualization, Kahiomba Sonia Kiangala; Writing – original draft, Kahiomba Sonia Kiangala; Writing – review and editing, Zenghui Wang

Funding Open access funding provided by University of South Africa. This work was supported in part by the South African National Research Foundation under Grant 137951, and the South African Eskom Tertiary Education Support Program.

Availability of data and materials Not applicable

Code availability Available

Declarations

Conflict of interest The authors declare no competing interests.

Ethics approval Not applicable

Consent to participate Not applicable

Consent for publication Not applicable

References

1. López A, Estévez E, Casquero O, Marcos M (2023) A methodological approach for integrating physical assets in industry 4.0. IEEE Trans Ind Informat 19:9546–9554. https://doi.org/10.1109/TII.2022.3230714
2. Kim SH, Jeon JH, Aridi A, Jun B (2023) Factors that affect the technological transition of firms toward the industry 4.0 technologies. IEEE Access 9:1694–1707. https://doi.org/10.1109/ACCESS.2022.3233390
3. Haghnegahdar L, Joshi SS, Dahotre N (2022) From iot-based cloud manufacturing approach to intelligent additive manufacturing: industrial internet of things-an overview. Int J Adv Manuf Technol 119:1461–1478. https://doi.org/10.1007/s00170-021-08436-x
4. Shah M, Vakharia V, Chaudhari R al (2022) Tool wear prediction in face milling of stainless steel using singular generative adversarial network and lstm deep learning models. Int J Adv Manuf Technol 121:723–736. https://doi.org/10.1007/s00170-022-09356-0
5. Javed S, Javed JV, Deventer H, Mokayed, Delsing J (2023) A Smart Manufacturing Ecosystem for Industry 5.0 using Cloud-based Collaborative Learning at the Edge. Paper presented at the NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium, Miami, FL, USA. https://doi.org/10.1109/NOMS56928.2023.10154323
6. Jwo JS, Lin CS, Lee CH (2021) Smart technology–driven aspects for human-in-the-loop smart manufacturing. Int J Adv Manuf Technol 114:1741–1752. https://doi.org/10.1007/s00170-021-06977-9
7. Fraga-Lamas P, Varela-Barbeito J, Fernández-Caramés TM (2021) Next generation auto-identification and traceability technologies for industry 5.0: a methodology and practical use case for the shipbuilding industry. IEEE Access 9:140700–140730. https://doi.org/10.1109/ACCESS.2021.3119775
8. Moreira PA, Fernandes RM, Avila LV, Bastos LdSL, Martins VWB (2023) Artificial intelligence and industry 4.0? validation of challenges considering the context of an emerging economy country using cronbach's alpha and the lawshe method. Eng 4:2336–2351. https://doi.org/10.3390/eng4030133
9. Ghobakhloo M, Mahdiraji HA, Iranmanesh M, al (2024) From industry 4.0 digital manufacturing to industry 5.0 digital society: a roadmap toward human-centric, sustainable, and resilient production. Inf Syst Front. https://doi.org/10.1007/s10796-024-10476-z.
10. Adel A (2022) Future of industry 5.0 in society: human-centric solutions, challenges and prospective research areas. J Cloud Comp 11. https://doi.org/10.1186/s13677-022-00314-5.
11. Nasir V, Sassani F (2021) A review on deep learning in machining and tool monitoring: methods, opportunities, and challenges. Int J Adv Manuf Technol 115:2683–2709. https://doi.org/10.1007/s00170-021-07325-7
12. Kiangala KS, Wang Z (2021) An effective communication prototype for time-critical iiot manufacturing factories using zero-loss redundancy protocols, time-sensitive networking, and edge-computing in an industry 4.0 environment. Processes 9. https://doi.org/10.3390/pr9112084
13. Kiangala KS, Wang Z (2018) Initiating predictive maintenance for a conveyor motor in a bottling plant using industry 4.0 concepts. Int J Adv Manuf Technol 97:3251–3271. https://doi.org/10.1007/s00170-018-2093-8
14. Ouadah A, Zemmouchi-Ghomari L, Salhi N (2022) Selecting an appropriate supervised machine learning algorithm for predictive maintenance. Int J Adv Manuf Technol 119:4277–4301. https://doi.org/10.1007/s00170-021-08551-9
15. Kiangala KS, Wang Z (2020) An effective predictive maintenance framework for conveyor motors using dual time-series imaging and convolutional neural network in an industry 4.0 environment. IEEE Access 8:121033–121049. https://doi.org/10.1109/ACCESS.2023.3316116
16. Trivedi J (2019) Examining the customer experience of using banking chatbots and its impact on brand love: the moderating role of perceived risk. J Internet Commer 18:91–111. https://doi.org/10.1080/15332861.2019.1567188
17. Adikari A, De Silva D, Moraliyage H, Alahakoon D, Wong J, Gancarz M, Chackochan S, Park B, Heo R, Leung Y (2022) Empathic conversational agents for real-time monitoring and co-facilitation of patient-centered healthcare. Futur Gener Comput Syst 126:318–329
18. Loh SH, Jeuk VJ, Sim JJ, Yeap KH (2023) Implementation of artificial intelligence chatbot in semiconductor manufacturing to optimize overall equipment effectiveness. Paper presented at IEEE 13th International Conference on Control System, Computing and Engi-

neering (ICCSCE). https://doi.org/10.1109/ICCSCE58721.2023.10237164
19. Bilgram V, Laarmann F (2023) Accelerating innovation with generative ai: ai-augmented digital prototyping and innovation methods. IEEE Eng Manag Rev 51:18–25. https://doi.org/10.1109/EMR.2023.3272799
20. Wang FY, Yang J, Wang X, Li J, Han QL (2023) Chat with chatgpt on industry 5.0: learning and decision-making for intelligent industries. IEEE/CAA J Autom Sinica 10:831–834. https://doi.org/10.1109/JAS.2023.123552
21. Ray PP (2023) Chatgpt: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. Int Things Cyber-Phys Syst 3:121–154
22. Chen TY, Chiu YC, Bi N, Tsai RTH (2021) Multi-modal chatbot in intelligent manufacturing. IEEE Access 9:82118–82129. https://doi.org/10.1109/ACCESS.2021.3083518
23. Tripicchio P, D'Avella S, Unetti M (2022) Efficient localization in warehouse logistics: a comparison of lms approaches for 3d multilateration of passive uhf rfid tags. Int J Adv Manuf Technol 120:4977–4988. https://doi.org/10.1007/s00170-022-09018-1
24. Zuehlke D (2010) Smartfactory—towards a factory-of-things. Design and Installation of an Agent-Controlled Cyber-Physical Production System Using the Example of a Beverage Bottling Plant 34:129–138. https://doi.org/10.1016/j.arcontrol.2010.02.008
25. Fortoul-Diaz JA, Carrillo-Martinez LA, Centeno-Tellez A, Cortes-Santacruz F, Olmos-Pineda I, Flores-Quintero RR (2023) A smart factory architecture based on industry 4.0 technologies: open-source software implementation. IEEE Access 11:101727–101749. https://doi.org/10.1109/ACCESS.2023.3316116
26. Marschall B, Schleicher M, Sollich A, Becker T, Voigt T (2022) A methodological approach for integrating physical assets in industry 4.0. Design and Installation of an Agent-Controlled Cyber-Physical Production System Using the Example of a Beverage Bottling Plant 3:39–47. https://doi.org/10.1109/JESTIE.2021.3097941
27. Dafflon B, Moalla N, Ouzrout Y (2021) The challenges, approaches, and used techniques of cps for manufacturing in industry 4.0: a literature review. Int J Adv Manuf Technol 113:2395–2412. https://doi.org/10.1007/s00170-020-06572-4
28. Lu Y, Xu X, Wang L (2020) Smart manufacturing process and system automation – a critical review of the standards and envisioned scenarios. J Manuf Syst 56:312–325. https://doi.org/10.1016/j.jmsy.2020.06.010
29. Zhong RY, Xu X, Klotz E, Newman ST (2017) Intelligent manufacturing in the context of industry 4.0: a review. Engineering 3:616–630
30. Tran TA, Ruppert T, Eigner G, Abonyi J (2022) Retrofitting-based development of brownfield industry 4.0 and industry 5.0 solutions. IEEE Access 10:64348–64374. https://doi.org/10.1109/ACCESS.2022.3182491
31. Breque M, De Nul L, Petridis A (2021) Industry 5.0: towards a sustainable, human-centric and resilient European industry. Luxembourg, LU: European Commission, Directorate-General for Research and Innovation, 2021. Available Online https://research-and-innovation.ec.europa.eu/knowledge-publications-tools-and-data/publications/all-publications/industry-50-towards-sustainable-human-centric-and-resilient-european-industry_en. Accessed 06 Dec 2023
32. Committee EES (2021) Industry 5.0. Available Online https://ec.europa.eu/info/research-and-innovation/research-area/industrial-research-and-innovation/industry-50_en. Accessed 06 Dec 2023
33. Xu X, Lu Y, Vogel-Heuser B, Wang L (2021) Industry 4.0 and industry 5.0-inception, conception and perception. J Manuf Syst 61:530–535. https://doi.org/10.1016/j.jmsy.2021.10.006
34. Gupta M, Akiri C, Aryal K, Parker E, Praharaj L (2023) From chatgpt to threatgpt: impact of generative ai in cybersecurity and privacy. IEEE Access 11:80218–80245. https://doi.org/10.1109/ACCESS.2023.3300381
35. Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, al (2020) Language models are few-shot learners. https://doi.org/10.48550/arXiv.2005.14165
36. Ouyang L, Wu J, Jiang X, Almeida D, Wainwright CL, Mishkin P, Zhang C, Agarwal S, Slama K, Ray A, al (2022) Training language models to follow instructions with human feedback. https://doi.org/10.48550/arXiv.2203.02155
37. Gamage G, Kahawala S, Mills N, De Silva D, Manic M, Alahakoon D, Jennings A (2023) Augmenting industrial chatbots in energy systems using chatGPT generative AI. Paper presented at IEEE 32nd International Symposium on Industrial Electronics, Helsinki, Finland. https://doi.org/10.1109/ISIE51358.2023.10228101
38. Committee EES (2023) Number of ChatGPT Users. Available Online https://explodingtopics.com/blog/chatgpt-users. Accessed 29 Nov 2023
39. Feng M, Li Y (2022) Predictive maintenance decision making based on reinforcement learning in multistage production systems. IEEE Access 10:18910–18921. https://doi.org/10.1109/ACCESS.2022.3151170
40. Burmeister N, Frederiksen RD, Høg E, Nielsen P (2023) Exploration of production data for predictive maintenance of industrial equipment: a case study. IEEE Access 11:102025–102037. https://doi.org/10.1109/ACCESS.2023.3315842
41. Wang S, Wan J, Zhang D, Li D, Zhang C (2016) Towards smart factory for industry 4.0: a self-organized multi-agent system with big data-based feedback and coordination. Comput Netw 101:158–168
42. Amendaño-Murrillo S, Dután-Gómez C, Lema-Condo E, Robles-Bykbaev V (2020) Personal robotic assistants: a proposal based on the intelligent services of the IBM cloud and additive manufacturing. Paper presented at IEEE ANDESCON. https://doi.org/10.1109/ANDESCON50619.2020.9271971
43. Dersingh A, Srisakulpinyo P, Rakkarn S, Boonkanit P (2017) Chatbot and visual management in production process. In Proc Int Conf Electron Inf Commun, pp 274–277
44. Streamlit (2023) A faster way to build and share data apps. Available Online https://streamlit.io/. Accessed 07 Dec 2023
45. Mourtzis D, Angelopoulos J, Panopoulos N (2023) The future of the human–machine interface (hmi) in society 5.0. Future Int 15:3251–3271. https://doi.org/10.3390/fi15050162
46. Kumar N, Chan Lee S (2022) Human-machine interface in smart factory: a systematic literature review. Technological Forecasting and Social Change 174. https://doi.org/10.1016/j.techfore.2021.121284
47. Yang J, Liu T, Liu Y, Morgan P (2022) Review of human-machine interaction towards industry 5.0: human-centric smart manufacturing. 42nd Computers and Information in Engineering Conference (CIE). St. Louis, Missouri, USA, August 14–17. https://doi.org/10.1115/DETC2022-89711