Using SLMs and LLMs for Diagnosis

Sophia Plank
Graz University of Technology, Austria

Franz Wotawa[1]
Institute of Software Engineering and Artificial Intelligence, Graz University of Technology, Austria

Abstract

Large Language Models (LLMs) have rapidly transformed how organizations approach complex problem solving, raising the question of whether they can also be applied to technical reasoning tasks. In this paper, we investigate whether LLMs can be used for system diagnosis, i.e., the localization of faults in systems comprising interacting components. In particular, we conducted an initial study using examples ranging from electric circuits to heating systems, evaluating 13 different language models of varying sizes with respect to the correctness and completeness of their diagnosis. The experimental evaluation revealed that language models with more parameters achieve superior performance.

2012 ACM Subject Classification Computing methodologies → Artificial intelligence; Computing methodologies → Machine learning; Computer systems organization → Maintainability and maintenance

Keywords and phrases LLM diagnosis performance, SLM diagnosis performance, comparative study

Digital Object Identifier 10.4230/OASIcs.DX.2026.14

Funding *Franz Wotawa*: The research described in this paper has been partially carried out within the FFG project FO999923190 SAGE – Scalable Agents for Facility Management and Energy Efficiency funded by the Austrian Federal Ministry for Climate Action, Environment, Energy, Mobility, Innovation and Technology (BMK).

1 Introduction

Agentic AI [2, 5] has gained a lot of attention in industry and practice aiming at combining AI and in particular large-language models (LLMs) to solve certain tasks like software development and programming [12]. Hence, the question is about the impact of LLMs on diagnosis. Last year, Sztyber-Betley and colleagues [11] raised a similar question and provided a study mainly focusing on support capabilities of LLMs in model-based diagnosis. Due to substantial improvements in LLMs' capabilities over the past years, it is timely to examine their diagnostic capabilities, focusing on the engineered systems of current LLMs, and to discuss the consequences for the research field of diagnosis.

Let us have a look at a recent interaction with the free version of ChatGPT[2], which we depict in Figure 1. The conversation started with asking ChatGPT about its diagnostic capabilities. After a positive reply, we provided a diagram of a simple electric circuit and observations (1). It is worth noting that the same circuit, including the observations, was used in previous model-based diagnosis literature to provide evidence for introducing fault models [10] and physical impossibilities [4]. Surprisingly, ChatGPT provided a sophisticated answer (2) that included only correct diagnoses. Besides the diagnosis indicating that one bulb is faulty, which was as the most likely explanation, ChatGPT also returns potential structural issues. Furthermore, ChatGPT was also able to provide means for distinguishing between alternative candidate diagnoses (3), which are also reasonable and well outlined.

[1] Corresponding author
[2] `https://chatgpt.com`

1

> Can you solve diagnosis problems?

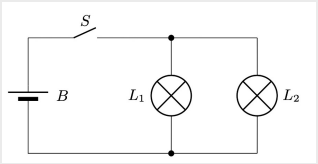
Yes—I can help work through diagnostic problems, depending on what you mean.

- **Medical symptoms:** I can help you think through possible causes, ask useful follow-up questions, and suggest when to seek care—but I can't replace a licensed clinician or give a definitive diagnosis.
- **Technical issues (car, computer, etc.):** I can often walk step-by-step through troubleshooting and likely causes.
- **Logical/academic "diagnosis" problems:** I can fully solve and explain those.

If you have a specific case in mind, describe the situation (symptoms, context, timing, anything relevant), and I'll help you reason through it.

> Here is a circuit.

Go ahead and share the circuit (image or description), and tell me what you're trying to diagnose or find —I'll walk through it with you.



2

>Assume the switch S to be closed. Bulb L1 is emitting light but L2 is dark.

With switch S closed, both bulbs L1 and L2 are connected **in parallel** across the battery B. In a healthy circuit, they should both light with the same voltage across them.
Since L1 is glowing but L2 is dark, that tells us:

- The battery and switch are working (because L1 lights).
- The top and bottom rails (wires) are intact at least up to .

So the problem must lie **in the branch containing** . The most likely diagnoses are:
1. **L2 is faulty (blown filament / open circuit)** — most common.
2. A **break in the wiring** leading to or from L2.
3. A **loose connection** at one of L2's terminals.

In all these cases, the key idea is the same:

👉 The L2 branch is an **open circuit**, so no current flows through it, and it stays dark.

Meanwhile, L1 still works because parallel branches operate independently.
If you want, I can show how to confirm which fault it is with simple tests.

3

> Yes, please tell me the tests for distinguishing the diagnoses.

Good—this is exactly how you'd troubleshoot it in practice. Since the fault must be an **open circuit in the L_2 branch**, the goal is to find *where* that open is.
Here are simple, decisive tests:

1) Swap the bulbs (quickest test)

- Put **L_2 into L_1's position**, and L_1 into L_2's position.

Results:

- If the same physical bulb still doesn't light → **L_2 is faulty (blown)**.
- If the problem stays in the same position (the right branch still dark) → wiring/connection issue in that branch.

2) Check voltage across L_2 (with a voltmeter)

Measure across the two terminals of L_2 with the switch closed.

Results:

- **Full battery voltage present** → wiring is fine → **bulb L_2 is blown**.
- **0 V (or very low)** → there's a **break in the wiring or connection** feeding L_2.

3) Continuity test of L_2 (power OFF)

Remove L_2 and test it with a multimeter (continuity/ohms mode).

Results:

- **No continuity (infinite resistance)** → filament broken → **bulb is faulty**.
- **Continuity present** → bulb is fine → problem is in wiring/connections.

.....

Figure 1 Question-answer interaction with ChatGPT April 2026 focusing on diagnosis and explanations of a simple electric circuit comprising a battery, a switch, and two bulbs connected in parallel.

Hence, for the provided simple example, ChatGPT was able to compute a reasonable diagnosis and to explain how to distinguish them. All of this functionality has been a research question within the model-based diagnosis community for decades. Why and how ChatGPT has these capabilities is not the focus of this paper, and would lead to coming up with certain speculative assumptions, like that ChatGPT uses a reasoning system behind it, or that the capabilities are mainly due to the simplified system, where a lot of publications are available that have been used to train ChatGPT's LLM. In contrast, we are interested in further elaborating the capabilities of available LLMs, considering electric circuits of varying complexity, and systems from other domains, such as heating systems in buildings. Furthermore, we want to clarify whether smaller models, i.e., small language models (SLMs), have similar capabilities. The latter would enable us to provide diagnostic systems based on smaller LLMs or SLMs, running on an ordinary computer rather than a dedicated server.

The contributions of this paper are:

- Coming up with a set of example problems to be used for evaluating the diagnosis capabilities of LLMs or similar Agentic AI systems.
- Conducting an experimental evaluation considering 13 different LLMs and SLMs and comparing their diagnosis capabilities. For the latter, we use correctness, completeness, and other measures.

We structure this paper as follows. In Section 2, we discuss realted work. In Section 3, we introduce the process, setup, and the underlying metrics for comparing diagnosis capabilities used in our experimental evaluation. We present and discuss the experimental results in Section 4. Finally, we conclude the paper.

2 Related Work

Model-based diagnosis derives fault candidates by contrasting a formal system model with observed behaviour. Its theoretical foundation was laid by Reiter [9], and Wotawa [13] gives a recent overview of the field. Its strength is precision, but it presupposes a model, and building one for each new system requires time and domain expertise. The approach taken here removes that prerequisite: the language models receive a plain text description of the system and reason about the fault directly.

More recently, language models have been applied directly to fault diagnosis. Sztyber-Betley et al. [11] ask, whether LLMs grasp the concepts of model-based diagnosis and focus on their use as a support tool within an existing diagnostic process. Qaid et al. [8] test four open-source models on machine fault diagnosis and find that they perform as well as established deep-learning methods. Lukens et al. [6] combine language models with technical manuals and show how important careful evaluation is for such systems. However, these works focus on large models and sensor- or document-based inputs.

Several studies examine models small enough to run on local hardware, asking how much capability is lost compared to larger models. Abdin et al. [1] show that careful, textbook-like training data can let a small model compete with much larger ones. This finding is directly relevant here, where Phi-2 and Phi-3 Mini differ a lot despite their similar size. Hasan et al. [7] systematically evaluate twenty models under 10B parameters on code generation and find that smaller models can stay competitive. Their setup resembles ours in spirit, but targets code generation with automatic metrics rather than fault diagnosis with manual scoring.

To the best of our knowledge, no existing study systematically evaluates small, locally-run models on text-based diagnosis of technical systems using manual, multi-category scoring. This paper closes that gap and clarifies where small language models can support technical fault diagnosis and where they still fall short.

3 Experimental Setup

This section provides an overview of the evaluation framework used in this paper to assess differences in the performance of language models on technical fault diagnosis tasks. The framework defines the overall experimental pipeline, including model selection, the construction of input scenarios, and the evaluation procedure. The framework is designed to ensure reproducibility and fairness across all models. To this end, all models receive the same prepared input, are prompted using the same templates, and are evaluated against the same scoring system. The circuits and fault scenarios used as inputs span a range of complexity levels, from simple electrical circuits to more complex heater systems, allowing for a better assessment of how the diagnostic capability changes with an increasing task difficulty. The evaluation is conducted by manually inspecting the model's response.

The set of models used for the experiments consists of 10 small language models (SLMs) and 3 large language models (LLMs), all open source and freely accessible. The set of models was chosen as diverse as possible in terms of providers and sizes. Different providers were chosen to avoid conclusions that are specific to a model family or training approach. The span of providers includes Google, Mistral AI, Meta, Alibaba, Microsoft, and Stability AI. Additionally, it was made sure to include multiple sizes from the same provider to assess the effect of model size on its diagnostic abilities. The three API-based models, Gemini 2.5 Flash, Mistral Large, and Llama 4 Scout 17B, serve as a reference point for current open-source LLM capabilities.

Table 1 Selected SLMs and LLMs for the experiment.

#	Model	Parameters	Provider
1	Gemini 2.5 Flash	n/a[a]	Google
2	Mistral Large	123B	Mistral AI
3	Llama 4 Scout 17B	17B	Meta
4	Mistral 7B v0.3	7B	Mistral AI
5	Phi-3 Mini 4K	3.8B	Microsoft
6	Llama 3.2 3B	3B	Meta
7	Qwen2.5 3B	3B	Alibaba
8	Phi-2	2.7B	Microsoft
9	Gemma-2 2B	2B	Google
10	StableLM-2 1.6B	1.6B	Stability AI
11	Qwen2.5 1.5B	1.5B	Alibaba
12	TinyLlama 1.1B	1.1B	Meta
13	Llama 3.2 1B	1B	Meta

a) Parameter count not publicly disclosed by Google.

As shown in Table 1, the selected models cover a wide range of sizes and origins, with a clear separation between the three API-based reference models and the ten locally executed SLMs.

The experimental setup consists of three main phases: circuit description preparation, prompt construction, and model execution.

The first phase converts the input images into text, since not all language models can process images directly. Each circuit description follows a consistent structure, it begins with a general *circuit description* of components and their connections, followed by the specific *fault scenario*. For more complex systems, such as the heater and powertrain circuits, additional categories describe their functionality in more detail, including circuit behavior, wiring and control and measurement signals.

The second phase designs a set of prompt templates that keep the prompting consistent across models. Three distinct templates exist depending on the circuit type: a general circuit prompt, a powertrain specific prompt and a heater specific prompt. All templates share the same structure. They open with a system role definition, such as *"You are an expert electrical engineer diagnosing a fault in a ..."*, followed by the corresponding circuit description, and they conclude with the task instruction *"Task: Analyze the fault scenario, what are the reasons for the fault? "*. The domain specific templates add a brief functional description of the circuit type to give the model some relevant background context.

In the third and final phase, the ten locally evaluated models are loaded into and executed through the LM Studio API (version 0.4.1). The three API-based models, Gemini 2.5 Flash, Mistral Large and Llama 4 Scout 17B, are queried through their respective external APIs: Google, Mistral, and Groq.

All models receive the same prepared inputs, with a temperature of 0.2 and a maximum output length of 800 tokens per response. The low temperature keeps the output focused on the most probable tokens without making it fully deterministic, so small variations are still possible. The output length is the same for every scenario to keep the conditions equal while still leaving room for a full diagnostic explanation. Each model runs three times per scenario, and each run is saved in a separate output directory for subsequent evaluation. Each of the three runs is scored independently across the six categories, so each run has its own set of scores. The per-scenario and per-circuit values reported in Section 4 are the averages over these runs.

```
Circuit description:
- Bulbs: L1, L2
- Switches: S
- Battery: B
- Wiring:
       - Battery B positive connected to switch S
       - Switch S connects to L1 and L2 in parallel
       - L1 and L2 connect back to battery negative

If the battery is not empty and the switch is closed both bulbs should emit light. If
    the switch is open, no light is emitted.

Fault Scenario:
Switch S is closed.
L1 emits light.
L2 is dark.

Identify the faulty component(s).
```

Listing 1 Input file for Circuit 1 Scenario 1

The circuit descriptions list components and connections explicitly, and wires are assumed to be intact; the models were therefore not expected to name broken wires as candidates. The prompt did not instruct the models to return subset-minimal diagnoses; minimality is a property of the expected diagnoses only, against which the responses are scored. Listing 1 shows the complete input for Circuit 1, Scenario 1 as an example. The three templates differ only in the system role definition and, for the domain-specific ones, in a short functional description of the circuit type; the task instruction is identical for all scenarios.

3.1 Input Circuits and Scenarios

In this section, we present the 7 underlying systems and the different diagnosis scenarios used in our experimental evaluation.

3.1.1 Circuit 1

Circuit 1 consists of two bulbs L_1 and L_2, one switch S and one battery B. If the battery is not empty and the switch is closed, both bulbs should emit light. If the switch is open, no light is emitted.

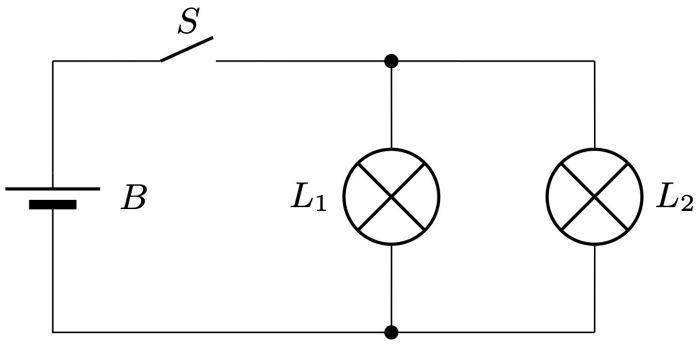


Figure 2 A simple two-bulb circuit used as Input Circuit 1.

Scenario 1: Let the switch S be closed and L_1 emit light while L_2 is dark. In this scenario the best explanation is that L_2 is broken or faulty.

Scenario 2: Let the switch S be closed, but both L_1 and L_2 be dark. In this scenario possible explanations are that the switch S itself is faulty, that both L_1 and L_2 are broken, or that the battery B is empty.

3.1.2 Circuit 2

Circuit 2 consists of the same components as Circuit 1 with an additional switch S_2. If one of the two switches is closed, then both bulbs emit light. If both switches are open, none of the bulbs should emit light.

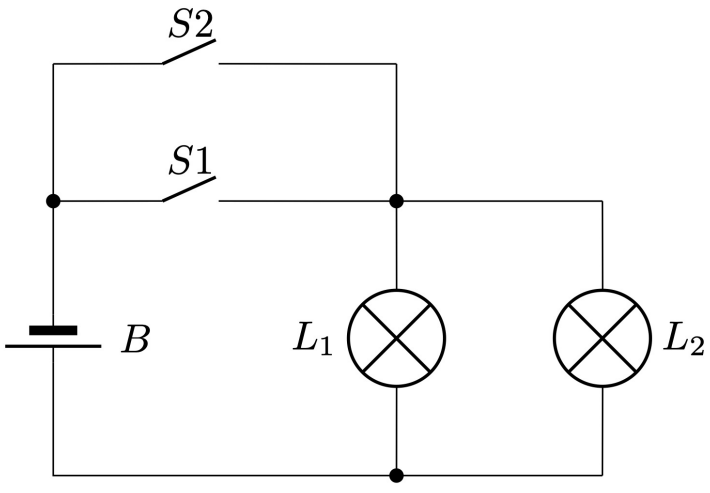


Figure 3 Input Circuit 2, an extension of Circuit 1 with a second parallel switch.

Scenario 1: Let switch $S1$ be open and $S2$ be closed. Bulb L_1 emits light but L_2 is dark. With $S1$ open, current flows through $S2$, making the circuit functionally equivalent to Circuit 1. Therefore the best explanation is that L_2 is broken or faulty.

Scenario 2: Let both switches $S1$ and $S2$ be closed, but both bulbs L_1 and L_2 are dark. In this scenario either both bulbs or both switches are broken, or the battery is empty.

3.1.3 Circuit 3

Circuit 3 consists of five bulbs L_1 to L_5, four switches $S1$ to $S4$ and one battery B. Each switch controls one or more bulbs: S_1 controls L_1, $S2$ controls L_2, $S3$ controls L_3 and L_4, and $S4$ controls L_5. Closing a switch lets its bulbs emit light; if all switches are open, none of the bulbs should emit light.

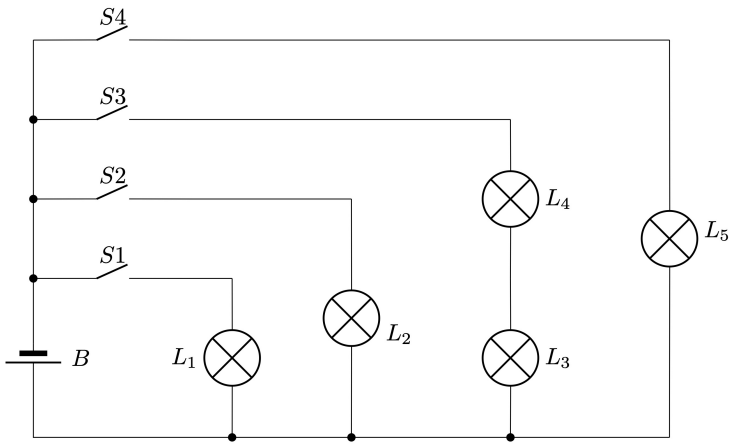


Figure 4 Input Circuit 3, the most complex electrical circuit with five bulbs and four switches.

Scenario 1: Let switches S_1 and S_3 be closed and S_2 and S_4 be open. Bulb L_1 emits light, while all other bulbs are dark. The fault therefore lies with S_3 or with the bulbs it controls: either S_3 is faulty, or both L_3 and L_4 are broken. Because L_3 and L_4 are in series, both must fail for that branch to stay dark.

Scenario 2: Let switches S_1 and S_3 be closed and S_2 and S_4 be open. None of the bulbs emit light. In this scenario the best explanation is that the battery B is empty.

3.1.4 Heater

Figure 5 shows a simple electric heater, which consists of a power supply V_P, a switch SW, an ammeter A, a heating resistor R_H, a temperature sensor T, and a controller *Ctrl*. The power supply connects to SW, which feeds a branch containing A, R_H, and T down to ground. The temperature sensor T measures the temperature at R_H and reports it to *Ctrl*. When the temperature drops below a minimum threshold, *Ctrl* closes SW, allowing current to flow through R_H and heat the system. Once the maximum threshold is reached, SW is opened and heating stops until the temperature falls again, repeating the cycle.

Fault Scenario: Switch SW is permanently closed and the temperature never reaches the maximum threshold. The faulty components in this case are one of three: T, R_H, or excessive heat loss. T may be faulty and report an incorrectly low temperature, causing

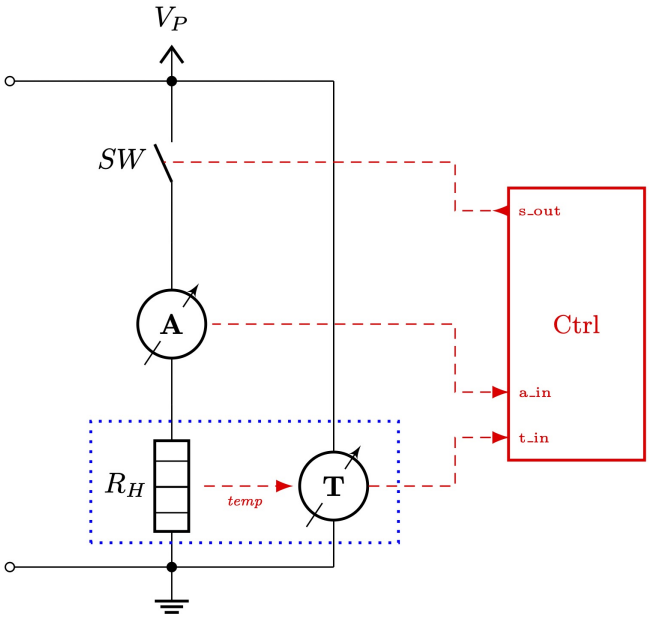


Figure 5 Input Heater, a single-loop heating circuit with a controller-operated switch.

Ctrl to never open SW. R_H may be degraded or partially failed, generating insufficient heat to reach the maximum threshold despite current flowing through it. Alternatively, the heat loss to the environment may be larger than expected, preventing the system from reaching the maximum temperature even with the heater running continuously.

3.1.5 Powertrain

The powertrain circuit shown in Figure 6 consists of a battery *bat* with internal resistance R_{bat}, an h-bridge composed of four switches sw_1 to sw_4, a motor branch with resistance R_{mot} and inductance L_{mot} in series, an electric motor M, and a mechanical load *Load*. The internal resistance of the battery, R_{bat}, models a voltage drop under high current conditions. The h-bridge controls the motor by switching the polarity of the voltage applied to it, enabling forward, backward, and stop operation. Not all switch configurations are valid; for instance, closing sw_1 and sw_2 simultaneously would create a short circuit from the battery's positive pole to ground. Table 2 shows the configurations assumed for this example.

Table 2 Switch configurations for the Powertrain example.

sw_1	sw_2	sw_3	sw_4	Behavior
on	off	off	on	`forward`
off	on	on	off	`backward`
off	on	off	on	`stop`

Note that there are additional valid configurations for the `stop` behavior. The h-bridge is controlled by using a mode variable that takes the values `forward`, `backward`, and `stop`; individual switch control is not possible. The motor converts electricity into mechanical energy characterized by the rotational angle ϕ and the torque T, while the load determines the torque the motor must handle, affecting its rotational speed accordingly.

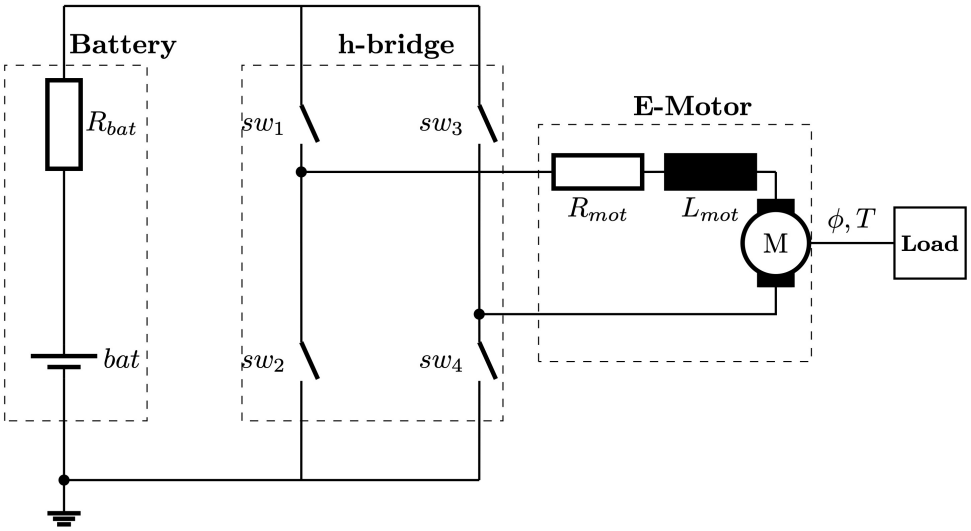


Figure 6 Input Powertrain, an H-bridge driving a motor branch with series resistance and inductance.

Fault Scenario: The h-bridge is set to forward mode, meaning sw_1 and sw_4 are closed while sw_2 and sw_3 are open, but the motor M does not rotate. As shown in Figure 6, current should flow from bat through R_{bat}, sw_1, R_{mot}, L_{mot}, M, and sw_4 to ground. A failure anywhere along this path can prevent rotation. The faulty component is therefore one of the following: an empty battery bat, a stuck-open switch sw_1 or sw_4, an open-circuit fault in R_{mot} or L_{mot}, or an electrical or mechanical failure in M itself.

3.1.6 Heater Circuit

The heater circuit shown in Figure 7 is a fluid-based heating system comprising two redundant parallel pumps $P1$ and $P2$, an inlet temperature sensor T_I, a radiator R, an outlet temperature sensor T_O, a motorized return valve V, and a check valve V_C. On a heating request, one pump drives fluid from the hot supply line through T_I, R, and T_O to the return valve V, which regulates flow rate and thus heat transfer to the room. The check valve V_C allows fluid to recirculate when V is closed. Under normal operation, the heat is partially emitted in R, resulting in $T_I > T_O$. With no pump running, there is no flow and $T_I = T_O$.

Scenario 1: Pump $P1$ is on and $P2$ is off, but $T_I = T_O = 80\,°\text{C}$. Since active pumping through R should produce $T_I > T_O$, the equal readings suggest either no heat transfer despite flow or unreliable sensor readings. The faulty component is therefore $P1$, T_I, T_O, or R: $P1$ might not actually be pumping despite being on, one of the sensors may report identical values despite a real temperature difference, or R may not be emitting heat despite flow.

Scenario 2: Under the same conditions as Scenario 1, now the flow through the radiator branch is confirmed. Since $P1$ is functioning and flow is present, the missing temperature difference must be due to a sensor fault or a failure in heat transfer. The faulty component is therefore T_I, T_O, or R, which might not be emitting heat despite flow.

Scenario 3: Under the same conditions as Scenario 1, no flow is detected despite $P1$ being on. Since the absence of flow fully explains $T_I = T_O$, the fault is isolated to the pump. The faulty component is therefore $P1$.

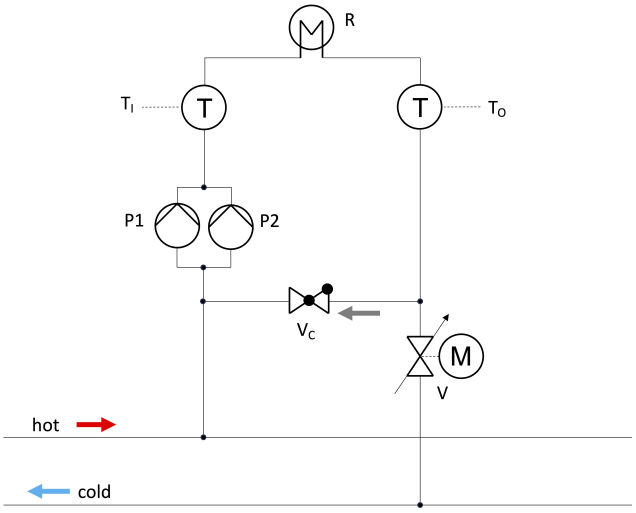


Figure 7 Input Heater Circuit, a fluid-based heating loop with two redundant pumps and a return valve.

3.1.7 Heater System

The heater system shown in Figure 8 consists of three independent parallel heating circuits A, B, and C, each supplied by a central oven o via a shared hot supply line. Each circuit contains a pump (pA, pB, pC), an inlet temperature sensor ($ts1A$, $ts1B$, $ts1C$), a heat exchanger (hcA, hcB, hcC), and an outlet temperature sensor ($ts2A$, $ts2B$, $ts2C$). When a pump is active, fluid flows through the heat exchanger, which releases part of its heat into the room. Under normal operation, the inlet temperature is higher than the outlet temperature ($temp1x > temp2x$). With no pump running, there is no flow and both sensors read the same temperatures. The three circuits operate independently, so if one fails, the others keep working.

Scenario 1: All three pumps are on. Circuits A and C behave normally, meaning $temp1A = 70\,°\text{C}$, $temp2A = 65\,°\text{C}$ and $temp1C = 70\,°\text{C}$, $temp2C = 67\,°\text{C}$, confirming heat transfer in both branches. But Circuit B shows no temperature difference, i.e., $temp1B = temp2B = 70\,°\text{C}$, indicating no heat transfer despite pB being active. Since Circuits A and C are operating correctly, a fault in the shared oven o or supply line can be excluded. Furthermore, since all three circuits show the same inlet temperature of $70\,°\text{C}$, the inlet sensor $ts1B$ can also be excluded. The fault is isolated to Circuit B with the following candidates: pB, which may not be pumping despite being on; $ts2B$, which reports identical values despite a real temperature difference; or hcB, which may not be emitting heat despite flow.

Scenario 2: Under the same conditions as Scenario 1, flow in Circuit B is additionally confirmed. Since pB is functioning correctly, the fault is narrowed down to $ts2B$ reporting identical temperatures despite a real difference or hcB, which may not be emitting heat despite flow.

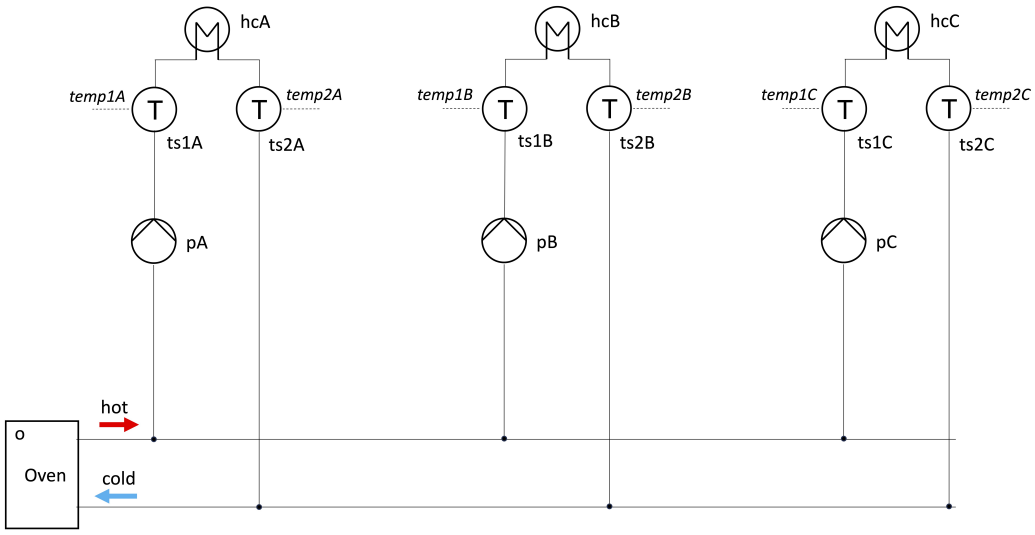


Figure 8 Input Heater System, three independent parallel heating circuits supplied by a shared oven.

3.2 Evaluation Metrics and Scoring

Model responses are evaluated manually across six categories, each scored on a scale from 0 to 2. Here 0 indicates poor performance, 1 indicates partial correctness, and 2 indicates good performance. The six categories are defined as follows:

Correctness measures whether the diagnosed fault corresponds to the expected diagnosis of the scenario, which is the set of subset-minimal candidates given in Section 3.1. A score of 2 is given if the response names all expected candidates and no others. A score of 1 is given if the response is partially correct, either because it misses some expected candidates or because it also blames a real but working component. A score of 0 is given if none of the named candidates is correct.

Completeness measures whether the explanation covers all relevant aspects of the diagnosis. A score of 2 is assigned if the response addresses all relevant components and fault causes, a score of 1 if some aspects are missing or insufficiently explained, and a score of 0 if the response is largely incomplete.

Reasoning measures whether the model explains how it arrives at its fault candidates. A step is faulty if it does not follow from the prompt or from an earlier step, or if it contradicts an earlier step. A score of 2 is given if the response explains every candidate step by step, starting from the given observations, and contains no contradiction. A score of 1 is given if an explanation is present but skips steps, or contains a faulty step that does not change the set of candidates. A score of 0 is given if candidates are named without any explanation, or if the response contradicts itself.

Hallucination measures whether the model introduces information not present in the prompt, i.e., it refers to a component, connection or measured value that does not appear in the circuit description or fault scenario, or contradicts it. A score of 2 is given if no such statement occurs; a score of 1 if such statements occur, but every named fault candidate is still justified by the prompt alone, so that removing them leaves the candidate set unchanged; and a score of 0 if at least one candidate depends on a hallucinated statement.

Uncertainty measures whether the model appropriately expresses uncertainty when multiple fault candidates are possible. A score of 2 is assigned if the model correctly acknowledges ambiguity and presents multiple plausible explanations, a score of 1 if uncertainty is only partially addressed, and a score of 0 if the model states a single definitive answer without justification in a scenario with multiple faulty components.

Clarity measures how understandable and structured the explanation is. A score of 2 is given if the response is well-organized and easy to follow, a score of 1 if the explanation is somewhat unclear or poorly structured, and a score of 0 if the response is difficult to understand due to structure and language.

To turn the six categories into a single comparable number, each response receives a weighted score. The weights reflect how important each category is for technical fault diagnosis. Correctness carries the highest weight (30%), because it directly captures whether the model identified the correct fault. Reasoning and Hallucination each count 20%, since sound logic and the absence of made-up information are equally important for a reliable diagnosis. Completeness receives 15%, as covering all fault candidates matters, but less than correctness itself. Uncertainty and Clarity receive 10% and 5%, as they add to the quality of a response but affect diagnostic accuracy less directly. The weighted score W of a response then follows as shown in Equation 1.

$$W = 0.30 \cdot c_{corr} + 0.15 \cdot c_{comp} + 0.20 \cdot c_{reas} + 0.20 \cdot c_{hall} + 0.10 \cdot c_{unc} + 0.05 \cdot c_{clar} \quad (1)$$

Here, each $c_i \in \{0, 1, 2\}$ is the score in the matching category, so the highest possible weighted score is 2.0. Two further measures support the comparison. The hit rate gives the percentage of scenarios in which a model reaches a correctness score of 2. The mean standard deviation $\bar{\sigma}$ measures how consistent a model is: for each of the six categories, the standard deviation of its scores across all runs is computed, and $\bar{\sigma}$ is the average of these six values. A lower $\bar{\sigma}$ indicates more stable scoring.

As a second, more robust comparison, a rank-sum metric is computed across all circuits. For each circuit, the models are ranked from 1 to 13, and the rank-sum R_m of model m is the sum of its ranks over all circuits, as defined in Equation 2.

$$R_m = \sum_{c=1}^{C} r_{m,c} \quad (2)$$

where $r_{m,c}$ is the rank of model m on circuit c, and C is the total number of circuits. A lower rank-sum means better overall performance. This ranking follows the same idea as the Friedman test, a non-parametric alternative to repeated-measures ANOVA [3]. Unlike the weighted score, the rank-sum is less sensitive to outliers, meaning a single bad result counts at most as rank 13, no matter how low the absolute score.

4 Experimental Results

This section summarizes the experimental results structured along two levels of analysis. The global analysis establishes an overall ranking of all evaluated models based on the weighted score W and the rank-sum metric R. The per-circuit analysis goes into detail of how the models performed in each circuit.

4.1 Global Analysis

Table 3 presents the average scores per evaluation category for each model, ordered by weighted score. The results show a clear three-tier hierarchy. Gemini 2.5 Flash and Mistral Large consistently lead across all categories, with no score below 1.74. They are followed

by the third API model, Llama 4 Scout 17B, which occupies an intermediate position, and after a notable gap, come the remaining SLMs. Across all models, clarity tends to be the strongest category, while correctness and completeness show the largest gap between top and bottom performers. This suggests that these categories are the primary differentiators of diagnostic performance.

Table 3 Averages per category for each model, ordered by ranking of weighted score.

#	Model	Correctness	Completeness	Reasoning	Hallucination	Uncertainty	Clarity
1	Gemini 2.5 Flash	1.77	1.79	2.00	1.74	1.97	2.00
2	Mistral Large	1.77	1.95	1.85	1.85	1.77	1.87
3	Llama 4 Scout 17B	1.26	1.31	1.59	1.92	1.54	1.92
4	Mistral 7B v0.3	1.00	1.15	1.38	1.26	1.44	1.79
5	Gemma-2 2B	1.00	1.08	1.23	1.13	1.28	1.87
6	Phi-3 Mini 4K	0.77	0.87	1.21	1.33	1.21	1.72
7	Qwen2.5 1.5B	0.90	0.85	1.00	1.00	1.03	1.67
8	Llama 3.2 3B	0.79	0.90	0.87	0.95	1.26	1.62
9	Qwen2.5 3B	0.69	0.77	0.82	1.05	0.85	1.67
10	StableLM-2 1.6B	0.44	0.26	0.54	0.82	0.82	1.44
11	Llama 3.2 1B	0.36	0.26	0.26	0.46	0.64	1.28
12	Phi-2	0.28	0.15	0.46	0.54	0.28	0.97
13	TinyLlama 1.1B	0.28	0.15	0.15	0.59	0.26	0.64

To complement the category averages, Table 4 presents the aggregated metrics used to derive the final ranking. The hallucination rate emerges as the strongest differentiator between model groups. While the top three models remain below 21%, most SLMs exceed 70% with Llama 3.2 1B even reaching 94.9%. This pattern is also consistent with the weighted scores W, suggesting that hallucination is a reliable indicator for the overall diagnostic quality. The mean standard deviation $\bar{\sigma}$ reinforces this grouping even further, with the top three models remaining below 0.47, while most SLMs exceed 0.6, indicating that SLM performance is not only lower on average, but also considerably less stable across circuits.

Table 4 Hit rate, hallucination rate, standard deviation $\bar{\sigma}$ and the weighted score W that were computed.

#	Model	Hit %	Hallucination %	$\bar{\sigma}$	W
1	Gemini 2.5 Flash	79.5	20.5	0.264	**1.846**
2	Mistral Large	76.9	15.4	0.353	**1.832**
3	Llama 4 Scout 17B	38.5	7.7	0.467	1.526
4	Mistral 7B v0.3	25.6	61.5	0.653	1.235
5	Gemma-2 2B	20.5	71.8	0.581	1.155
6	Phi-3 Mini 4K	7.7	59.0	0.622	1.076
7	Qwen2.5 1.5B	17.9	74.4	0.708	0.982
8	Llama 3.2 3B	2.6	87.2	0.612	0.944
9	Qwen2.5 3B	7.7	71.8	0.668	0.865
10	StableLM-2 1.6B	2.6	76.9	0.632	0.595
11	Llama 3.2 1B	7.7	94.9	0.593	0.418
12	Phi-2	0.0	84.6	0.584	0.385
13	TinyLlama 1.1B	0.0	76.9	0.544	0.314

The rank-sum results in Table 5 place the models in the same order as the weighted score for all 13 models. This agreement shows that the ranking is robust and independent of the chosen aggregation method and that no single circuit distorts the overall picture.

Table 5 Rank-Sum results across all circuits.

#	Model	C1	C2	C3	PT	H	HC	HS	Rank-Sum
1	Gemini 2.5 Flash	1	1	1	1	1	1	2	8
2	Mistral Large	4	2	2	2	2	2	1	15
3	Llama 4 Scout 17B	7	3	3	3	3	4	3	26
4	Mistral 7B v0.3	2	4	7	5	5	5	4	32
5	Gemma-2 2B	8	5	4	10	6	3	5	41
6	Phi-3 Mini 4K	6	8	5	7	4	6	6	42
7	Qwen2.5 1.5B	3	7	11	6	7	8	9	51
8	Llama 3.2 3B	9	10	6	4	9	7	7	52
9	Qwen2.5 3B	5	9	12	8	8	10	8	60
10	StableLM-2 1.6B	11	6	10	13	10	9	11	70
11	Llama 3.2 1B	10	11	8	11	12	12	12	76
12	Phi-2	13	13	9	9	13	11	10	78
13	TinyLlama 1.1B	12	12	13	12	11	13	13	86

4.2 Per-Circuit Analysis

Figure 9 shows the weighted score of every model on every system. On Circuit 1, the overall performance level is very high, which is consistent with the low complexity of the circuit. The gap between the API-based models and the SLMs is still small here, and several SLMs place ahead of Mistral Large and Llama 4 Scout 17B. On Circuit 2, which extends Circuit 1 with an additional parallel switch, this gap begins to widen. One observation is not visible in the scores: in Scenario 2 of Circuit 1, no model other than Gemini 2.5 Flash identified the theoretically valid, though unlikely, possibility of both L_1 and L_2 being faulty at the same time. All other models instead named only the most probable fault candidates.

At Circuit 3, the most complex of the three electrical circuits, the gap between the API-based models and the SLMs becomes dramatic. Gemini 2.5 Flash, Mistral Large, and Llama 4 Scout 17B all remain above 1.70, maintaining their strong performance from the previous circuits, while all remaining SLMs collapse to scores below 0.60. Model size alone does not explain this collapse: Mistral 7B v0.3, the largest of the locally executed models and fourth in both Circuit 1 and Circuit 2, drops behind Gemma-2 2B and Llama 3.2 3B. Qwen2.5 3B, TinyLlama 1.1B, and StableLM-2 1.6B achieve near-zero scores, indicating an almost complete failure to diagnose faults in this configuration.

On the domain-specific systems, the overall performance recovers noticeably across most models compared to Circuit 3. On the Heater, Gemini 2.5 Flash again achieves the maximum score, and the mid-range SLMs show a relatively gradual decline rather than a collapse. On the Powertrain, Llama 4 Scout 17B achieves its strongest result across all circuits, and Llama 3.2 3B and Mistral 7B v0.3 recover from their poor Circuit 3 performance. For the Heater Circuit, no model achieves a score of 2.00, with Gemini 2.5 Flash and Mistral Large leading at their weakest results so far. A notable result is Gemma-2 2B, which places ahead of both Llama 4 Scout 17B and Mistral 7B v0.3, suggesting that this model handles fluid-based diagnostic scenarios particularly well. On the Heater System, the most complex system evaluated, Mistral Large leads for the first time, while Gemini 2.5 Flash, which was not able to identify all faulty components, places third behind Llama 4 Scout 17B. Llama 3.2 1B and TinyLlama 1.1B achieve near-zero scores, indicating an almost complete failure to handle the complexity of this system.

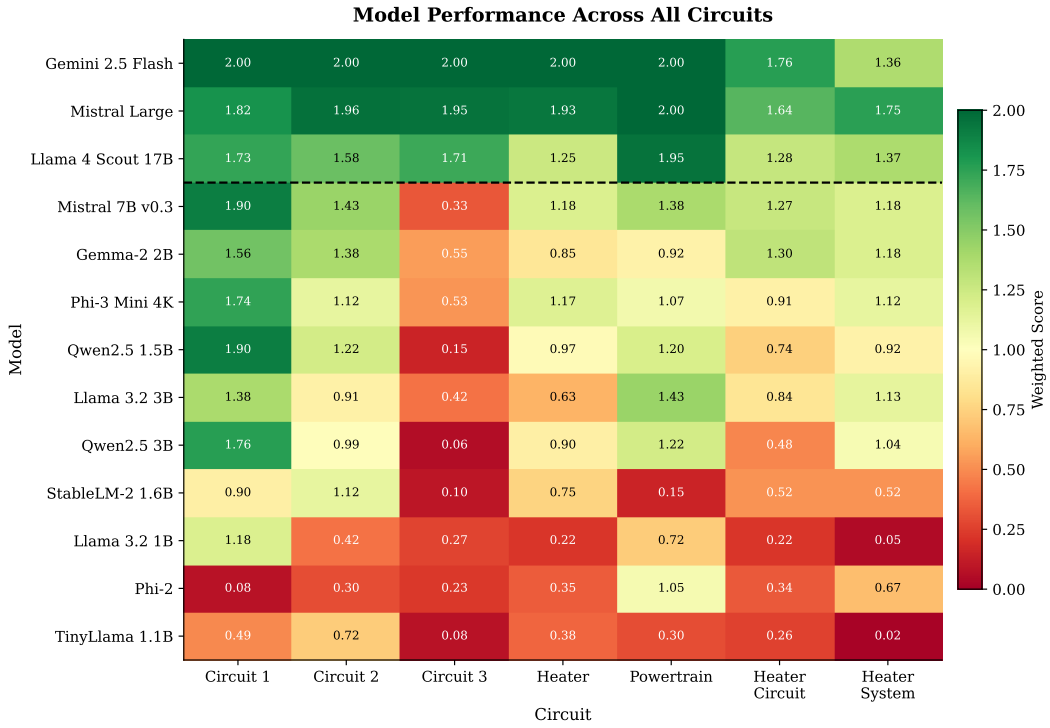


Figure 9 Weighted Score W of every model on every system. Systems are ordered by increasing complexity from left to right, models by overall weighted score from top to bottom.

4.3 Case Study: Phi-2 on Powertrain

The Powertrain is the only circuit in which Phi-2 reaches a weighted score above 1.0. Across all other circuits, the model scores between 0.08 and 0.67, but on Powertrain it achieves 1.05, performing better than several larger SLMs such as Gemma-2 2B and Llama 3.2 3B. This is particularly surprising given that the Powertrain circuit is structurally more complex than the simpler electrical circuits where Phi-2 failed almost completely.

A closer look at the model's responses suggests that this strong performance is not the result of genuine reasoning, but rather a reflection of the training data. In all three runs, Phi-2 produces a list of plausible fault causes such as low battery voltage, open switches, faulty motor windings, and high motor resistance. While these match several of the correct fault candidates, the responses also contain clear signs that the model is not actually diagnosing the given scenario. In one run, the response is followed by a section titled *"Follow-up Exercises"* with five additional questions and detailed solutions involving multimeter measurements. In the other two runs, the response ends with an unrelated *"Exercise 2"* about designing a control algorithm for a robotic arm. None of this content was requested by the prompt.

H-bridge and motor control circuits are common examples in introductory electrical engineering, and the structure of Phi-2's responses, a list of candidate faults followed by exercises and solutions, closely matches the typical layout of such textbooks. This is consistent with the training methodology of the Phi family, which explicitly relies on textbook-like training data [1]. In Phi-2's case, the high Powertrain score does not indicate strong diagnostic capability, but rather a fortunate overlap between the test scenario and the model's training data. This raises the question whether benchmark performance on familiar circuit types can be misleading, and reinforces the importance of including less common scenarios in evaluations of diagnostic capability.

5 Conclusions

In this paper, we present the results of an experimental evaluation of the diagnostic capabilities of large and small language models. In this study, we considered 13 different and freely available language models and 7 simple example systems from different domains. Well-known large language models performed best but still exhibit hallucinations and limitations in terms of correctness and completeness. This is worse for small language models. From our evaluation, we obtain that Gemini 2.5 Flash performs best in diagnosis, followed by Mistral Large and Llama 4 Scout 17B. The best small language model was Mistral 7B v0.3, followed by Gemma-2 2B and Phi-3 Mini 4K. Regarding the domains, large language models achieved better results for electric circuits than for heating system diagnosis. SLMs struggled to diagnose more complex electrical circuits.

There are some limitations of the study. First, we conducted the experiments only on smaller example systems. Second, we did the evaluation manually. Hence, there might be a bias in the obtained results. To prevent this, we defined the evaluation categories and applied them as closely as possible, trying to avoid any systematic bias that affects only one circuit or one set of language models. Finally, we limit ourselves to freely available models, not considering commercial ones. It is worth noting that, as shown in the example in the introduction (see Figure 1), ChatGPT was able to perfectly describe potential faults in the first electric circuit. Given the outcomes of other large language models, the reason might be the use of specialized implementations of diagnostic procedures. It would be worth conducting a specific study to determine how ChatGPT or similar large language models perform in diagnosis, considering parametrized circuits of larger sizes. We leave answering this question to future research.

Hence, future research has to cover: (i) extending the study considering more and larger systems from different domains, (ii) applying the study to commercial LLMs, and (iii) studying the effects of combining different diagnosis approaches like model-based diagnosis with different language models. This might be best for small language models that still offer textual communication but have limited diagnostic capabilities. Furthermore, such models can be installed locally and easily integrated into diagnostic processes without requiring communication with servers, thereby enabling data privacy to be more easily ensured.

References

1 Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini,

Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone, August 2024. arXiv:2404.14219 [cs.CL]. doi:10.48550/arXiv.2404.14219.

2 Deepak Bhaskar Acharya, Karthigeyan Kuppan, and B. Divya. Agentic ai: Autonomous intelligence for complex goals—a comprehensive survey. *IEEE Access*, 13:18912–18936, 2025. doi:10.1109/ACCESS.2025.3532853.

3 Milton Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32:675–701, 1937.

4 Gerhard Friedrich, Georg Gottlob, and Wolfgang Nejdl. Physical impossibility instead of fault models. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 331–336, Boston, August 1990. Also appears in Readings in Model-Based Diagnosis (Morgan Kaufmann, 1992). URL: http://www.aaai.org/Library/AAAI/1990/aaai90-051.php.

5 Soodeh Hosseini and Hossein Seilani. The role of agentic ai in shaping a smart future: A systematic review. *Array*, 26:100399, 2025. doi:10.1016/j.array.2025.100399.

6 Sarah Lukens, Matthew Bishof, Nadir Siddiqui, and Destiny West. An Evaluation Framework for Fault Diagnosis Using Technical Manuals in Retrieval-Augmented Large Language Models. *Annual Conference of the PHM Society*, 17(1), October 2025. doi:10.36001/phmconf.2025.v17i1.4549.

7 Md Mahade Hasan, Muhammad Waseem, Kai-Kristian Kemell, Jussi Rasku, Juha Ala-Rantala, and Pekka Abrahamsson. Assessing small language models for code generation: An empirical study with benchmarks. *Journal of Systems and Software*, 236:112815, June 2026. doi:10.1016/j.jss.2026.112815.

8 Hamzah A. A. M. Qaid, Bo Zhang, Dan Li, See-Kiong Ng, and Wei Li. FD-LLM: Large Language Model for Fault Diagnosis of Machines, December 2024. arXiv:2412.01218 [cs.AI]. doi:10.48550/arXiv.2412.01218.

9 Raymond Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1):57–95, April 1987. doi:10.1016/0004-3702(87)90062-2.

10 Peter Struss and Oskar Dressler. Physical negation — Integrating fault models into the general diagnostic engine. In *Proceedings* 11^{th} *International Joint Conf. on Artificial Intelligence*, pages 1318–1323, Detroit, August 1989. URL: http://ijcai.org/Proceedings/89-2/Papers/075.pdf.

11 Anna Sztyber-Betley, Elodie Chanthery, Louise Travé-Massuyès, Silke Merkelbach, Karol Kukla, Maxence Glotin, Alexander Diedrich, and Oliver Niggemann. Are Diagnostic Concepts Within the Reach of LLMs? In Marcos Quinones-Grueiro, Gautam Biswas, and Ingo Pill, editors, *36th International Conference on Principles of Diagnosis and Resilient Systems (DX 2025)*, volume 136 of *Open Access Series in Informatics (OASIcs)*, pages 2:1–2:20, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/OASIcs.DX.2025.2.

12 Huanting Wang, Jingzhi Gong, Huawei Zhang, Jie Xu, and Zheng Wang. Ai agentic programming: A survey of techniques, challenges, and opportunities, 2025. doi:10.48550/arXiv.2508.11126.

13 Franz Wotawa. Model-Based Diagnosis—European Perspectives and Contributions. *The European Journal on Artificial Intelligence*, 39(3):557–576, 2026. doi:10.1177/30504554251403464.