A NOVEL ...

Y. M. Valencia[1], A. Cunha[1,], E. Antonelo[1,*]

[1] Departamento de Automação e Sistemas/UFSC - Florianópolis, SC, Brazil

KEY WORDS: Automatic inspection, Computational vision, Deep learning, Egg sorting, Food industry, Image processing.

ABSTRACT:

The objective of this work is ....

1. INTRODUÇÃO

Com o avanço das tecnologias industriais, a digitalização dos processos tem impulsionado o desenvolvimento de soluções inteligentes para monitoramento e diagnóstico de máquinas. Em ambientes de manufatura, os tornos desempenham um papel fundamental, porém são suscetíveis a desgastes mecânicos, desalinhamentos, superaquecimento e falhas decorrentes de vibração excessiva ou variações anormais de corrente e rotação. Tradicionalmente, a detecção dessas condições adversas depende de inspeções programadas, experiência do operador ou métodos de manutenção corretiva, que frequentemente resultam em paradas não planejadas, perda de produtividade e aumento dos custos operacionais (Rodrigues Neto, 2017).

Nesse contexto, abordagens baseadas em inteligência artificial têm ganhado destaque por permitir o acompanhamento contínuo das condições de operação e a identificação precoce de anomalias (Wakhare, 2023). Estudos recentes demonstram que sistemas baseados em sensores e algoritmos de aprendizado de máquina são capazes de monitorar variáveis como vibração, temperatura, corrente e ruído, gerando previsões automáticas sobre o estado de funcionamento do torno (Mukund et al., 2024). Métodos supervisionados e redes neurais profundas vêm apresentando resultados promissores na detecção antecipada de falhas e na melhoria do desempenho produtivo. Entretanto, essas abordagens trabalham predominantemente sobre sinais sensoriais, sem explorar de maneira integrada o conhecimento técnico explícito presente em manuais, normas e documentação operacional (Abbasi, 2021).

Mais recentemente, têm surgido estudos que incorporam recuperação semântica de informação técnica para apoiar sistemas industriais inteligentes, utilizando modelos generativos em conjunto com bancos de conhecimento estruturados. Esses trabalhos indicam que estratégias baseadas em RAG (Retrieval-Augmented Generation) podem melhorar a precisão em diagnósticos, fornecer maior interpretabilidade e permitir justificativas fundamentadas a partir de fontes documentais. Contudo, tais aplicações ainda são incipientes no domínio da manutenção preditiva em máquinas industriais, especialmente quando se trata de tornos industriais em operação real, com dados sensoriais contínuos e contexto técnico multimodal.

Diante deste cenário, este trabalho propõe um sistema de Diagnóstico Explicativo para tornos CNC, utilizando a arquitetura Retrieval-Augmented Generation (RAG). A abordagem integra duas fontes de dados complementares: dados operacionais em tempo real (provenientes de sensores de temperatura, vibração e corrente) e conhecimento técnico explícito (manuais, normas e registros de manutenção). Essa combinação permite ao sistema não apenas identificar o estado de operação (como condição normal, sobreaquecimento, desbalanceamento ou anomalia de corrente), mas também gerar diagnósticos detalhados e justificativas baseadas nas evidências encontradas em ambas as fontes.

Em síntese, a proposta busca unir o monitoramento físico instantâneo ao conhecimento específico previamente documentado, fornecendo uma solução transparente, interpretável e alinhada às recomendações técnicas. Embora a literatura demonstre avanços relevantes em manutenção preditiva e em recuperação semântica aplicada a sistemas industriais, a aplicação combinada dessas tecnologias em tornos mecânicos com análise do estado em tempo real permanece pouco explorada, configurando uma oportunidade de pesquisa com potencial de impacto significativo na confiabilidade e disponibilidade de equipamentos produtivos.

2. TRABALHOS RELACIONADOS

O diagnóstico de equipamentos industriais baseado em inteligência artificial tem se consolidado como uma abordagem promissora para detecção antecipada de falhas em tornos mecânicos, devido aos ganhos expressivos na redução de custos, aumento da disponibilidade e melhoria da confiabilidade dos equipamentos. Diversos estudos recentes demonstram a eficácia da integração entre sensores, Internet das Coisas (IoT) e algoritmos de aprendizado de máquina no monitoramento automático de condições operacionais, permitindo a identificação de padrões e o acionamento antecipado de intervenções técnicas.

Em (Mukund et al., 2024), foi desenvolvido um sistema de manutenção preditiva em torno mecânico utilizando sensores IoT e modelos de inteligência artificial tradicionais. Foram instalados sensores de vibração, temperatura, corrente, tensão, umidade e ruído para aquisição contínua de dados em tempo real. Os dados foram submetidos a algoritmos supervisionados, com destaque para o modelo Random Forest na previsão antecipada de falhas. Os resultados indicaram uma redução significativa de intervenções não planejadas: o número de ocorrências caiu de 22 para 12, correspondendo a aproximadamente 45% de diminuição em falhas operacionais. A análise contínua dos parâmetros elétricos e mecânicos permitiu a detecção precoce de anomalias antes que resultassem em danos críticos aos componentes do torno, reduzindo cerca de 22% nos custos de reparo

* Corresponding author - eric.antonelo@ufsc.br

e aumentando entre 10% e 20% a eficiência global do equipamento (Overall Equipment Effectiveness – OEE).

De forma semelhante, (Sakthi et al., 2025) implementou um sistema de manutenção preditiva baseado em monitoramento contínuo de variáveis como temperatura, vibração, ruído e consumo de energia, utilizando sensores integrados a microcontroladores que processam dados em tempo real e geram alertas quando limites operacionais são excedidos. A detecção de falhas foi realizada por modelos supervisionados formulados como problemas de regressão e classificação, complementados por modelos estatísticos de confiabilidade para estimar o tempo até a falha. Os resultados industriais demonstraram contribuições significativas: redução de 40% em eventos de superaquecimento, precisão de 92% na detecção antecipada de falhas mecânicas via análise de vibração, identificação de 85% das anomalias acústicas e economia de 15% no consumo de energia do equipamento.

Embora algoritmos tradicionais apresentem bons resultados, estudos recentes indicam que abordagens com redes neurais profundas, especialmente arquiteturas derivadas de IA generativa e Transformers, ampliam significativamente a capacidade de análise de séries temporais industriais. Nesse contexto, (Bampoula et al., 2024) propuseram um método híbrido de manutenção preditiva que combina LSTM-Autoencoders, responsáveis pelo diagnóstico automático de "boa" ou "má" condição operacional, com um Transformer encoder dedicado à estimativa da vida útil remanescente (RUL). Os LSTM-Autoencoders foram capazes de discriminar estados saudáveis e degradados com base na reconstrução do sinal, enquanto o Transformer classificou o RUL em três faixas (1 dia, 2–3 dias e 3–4 dias). Os modelos foram treinados com séries temporais rotuladas segundo registros de manutenção, alcançando acurácia satisfatória na previsão de falhas. Os autores concluíram que essa abordagem melhora a detecção precoce da degradação, permitindo planejamento proativo de manutenção, redução de paradas inesperadas e aumento da eficiência produtiva.

Além da manutenção, técnicas de aprendizado profundo também têm sido aplicadas diretamente ao monitoramento de processos. Em (Elahi et al., 2023), os autores propuseram um sistema inteligente para operações de torneamento baseado em dados multimodais e redes neurais profundas. O método utiliza sinais adquiridos em tempo real por sensores industriais, incluindo vibração e corrente, realizando a extração automática de características por meio de uma arquitetura convolucional otimizada. Diferentemente de abordagens tradicionais, que dependem fortemente de engenharia manual de atributos, o modelo sugerido aprende representações diretamente dos sinais brutos do processo, permitindo prever o comportamento da usinagem e identificar condições anormais com maior precisão. Os resultados experimentais demonstraram desempenho superior quando comparado a técnicas clássicas de aprendizado de máquina, evidenciando maior estabilidade, robustez ao ruído e capacidade de generalização em diferentes configurações operacionais. Esses achados reforçam o potencial do deep learning como ferramenta eficiente para controle de qualidade e manutenção preditiva em tornos mecânicos.

Mais recentemente, a combinação entre modelos profundos, dados em tempo real e recuperação semântica de informação técnica tem emergido como uma direção promissora. Em (Singh et al., 2025), os autores apresentam um chatbot multimodal baseado em Retrieval-Augmented Generation (RAG) para segurança no uso de máquinas industriais, integrando normas, manuais e bases regulatórias a um banco de conhecimento técnico. O estudo introduz um benchmark especializado envolvendo torno CNC, fresadora e robô colaborativo, avaliando 24 configurações diferentes de RAG. Os resultados mostram que a estratégia de recuperação é tão determinante quanto o modelo generativo, atingindo cerca de 86% de acurácia com baixa latência e evidenciando a escalabilidade e precisão desse tipo de abordagem para diagnósticos e assistência operacional.

De forma geral, a literatura demonstra que a aplicação combinada de sensores inteligentes, processamento de sinais e modelos computacionais avançados constitui uma alternativa eficaz para melhorar o desempenho industrial e prolongar a vida útil de componentes mecânicos e elétricos. Esses trabalhos fundamentam a relevância e a viabilidade de soluções de manutenção preditiva aplicadas especificamente a tornos mecânicos em ambientes produtivos reais, além de indicar novas possibilidades utilizando modelos baseados em RAG para diagnósticos explicáveis, contextualizados e sustentados por documentação técnica.

As contribuições principais trazidas por este trabalho são: (i) arquitetura dockerizada que integra simulador IoT, API FastAPI, painel Streamlit e inferência local via Ollama; (ii) mecanismo configurável de vetorização com suporte a ChromaDB, FAISS, Weaviate e Pinecone(via API externa), além de heurísticas para selecionar quais sinais de telemetria entram no prompt; (iii) pipeline automatizado de experimentos que captura métricas (accuracy, BLEU, ROUGE-L, **BERTScore F1**, latência e tokens) e gera relatórios prontos para publicação com um único clique; (iv) biblioteca dinâmica de gabaritos (‘docs/gabaritos.json‘) integrada ao painel, eliminando etapas manuais para preparar experimentos reprodutíveis.

3. METODOLOGIA

A metodologia adotada consiste no desenvolvimento de um sistema de diagnóstico industrial baseado em Geração Aumentada por Recuperação (RAG), capaz de combinar dados dinâmicos provenientes de sensores com conhecimento técnico extraído de documentos industriais. O objetivo é avaliar o impacto da fusão entre contexto físico (telemetria) e contexto semântico (manuais e normas) na precisão diagnóstica de modelos generativos de linguagem (LLMs) aplicados a tornos mecânicos.

A metodologia segue três etapas. (1) **Simulação ciber-física**: o módulo *simulator* publica leituras sintéticas (temperatura, vibração, corrente e estado) e aceita comandos de falha via MQTT, permitindo reproduzir cenários de superaquecimento e desbalanceamento. (2) **Curadoria do contexto estático**: PDFs são carregados no painel web, segmentados com chunking configurável e armazenados no backend vetorial escolhido; o modelo de embeddings padrão é Sentence-Transformers `all-MiniLM-L6-v2`, mas pode ser alterado a partir da UI. (3) **Montagem do prompt dual**: quando o operador faz uma pergunta, a API coleta até três trechos relevantes do manual (k=3), seleciona apenas os sinais de telemetria marcados na UI (heurística de seleção de sensores) e combina tudo em um prompt estruturado com instruções e formato JSON personalizáveis.

O encoder de embeddings é definido globalmente. Trocar o backend vetorial (Chroma, FAISS, Weaviate ou Pinecone) não altera o modelo Sentence-Transformers; para isolar o efeito

de cada backend, o pesquisador escolhe o encoder uma única vez e executa o botão de reindexação, garantindo que todos os armazenamentos recebam os mesmos vetores. Essa decisão foi definida para facilitar a reprodução e avaliação.

Para medir a aderência das respostas, utilizamos gabaritos textuais derivados do Manual de Operação e Manutenção do torno ROMI T 240 (modelo de documento de 3 páginas construido para facilitar os experimentos). Os textos oficiais esperados (normal, falha térmica, desbalanceamento) residem em 'docs/gabaritos.json', arquivo que o painel Streamlit lê dinamicamente a cada ciclo. Ao acionar os botões do simulador (Operação Normal, Falha Térmica ou Desbalanceamento) o campo "Gabarito" é preenchido automaticamente com o JSON correspondente, permitindo ajustes sem recompilar os contêineres. Quando este campo está definido, a API compara a saída do LLM com o gabarito e registra accuracy, BLEU, ROUGE-L e o indicador semântico BERTScore F1 (modelo 'neuralmind/bert-base-portuguese-cased') no CSV experimental. O BERTScore é especialmente útil quando o LLM devolve respostas estruturadas em JSON, pois reconhece equivalências mesmo com diferenças lexicais significativas.

O gabarito avalia apenas a resposta final, independentemente do backend vetorial ou do encoder utilizado. Consequentemente, o protocolo de experimentação fixa o encoder e o backend quando o objetivo é comparar provedores LLM (Groq, Gemini, Ollama) e, inversamente, mantém o provedor fixo quando se deseja estudar encoders ou backends. Observamos empiricamente que... (ex para avaliarmos depois com os graficos(Gemini 1.5/2.5 gera relatórios mais completos (maior BLEU/ROUGE), Groq Llama3-70B entrega respostas concisas porém aderentes, e o modelo local Llama3.2 3B costuma resumir demais, servindo como baseline de menor custo).

O sistema avalia três cenários: (i) Baseline zero-shot; (ii) RAG estático (apenas documentos); (iii) RAG dual (documentos + telemetria). A cada diagnóstico, se o checkbox de experimentos estiver ativo, a API salva métricas no CSV compartilhado com o host em 'data/api/experiment-logs.csv'. Um endpoint adicional executa o notebook de consolidação em background e gera tabelas/gráficos em 'data/api/summaries'.

3.1 Arquitetura do Sistema

A arquitetura geral está ilustrada na Figura 1. O sistema é organizado em dois módulos principais, com funções complementares:

1. *Módulo WEB:* responsável pela interface de interação com o usuário e pela coleta dos sinais de operação do torno provenientes do broker MQTT. Além disso, realiza o gerenciamento dos parâmetros do agente (perfil, instruções técnicas, formato da resposta e objetivo), estrutura as requisições e exibe o diagnóstico gerado pelo modelo.

2. *Módulo da API:* realiza o processamento inteligente das informações enviadas pelo módulo WEB. Suas funções incluem: vetorização de documentos técnicos e normas industriais, recuperação semântica via RAG, construção do prompt combinando telemetria + contexto técnico e comunicação com o modelo LLM externo para geração da resposta, contendo estado da máquina, justificativas e recomendações de ação.

Essa separação permite isolar telemetria, processamento de linguagem natural e interação humano–máquina, facilitando a execução de experimentos controlados.

Todos os componentes são orquestrados via Docker Compose. O serviço `api` monta `./data/api` em `/app/data`, garantindo persistência dos uploads, dos índices FAISS (armazenados localmente e acessados em memória pelo FastAPI) e dos relatórios. O contêiner Weaviate, opcional, armazena seu índice em `./data/weaviate` e opera com a busca `nearVector`, pois os módulos text-to-vector foram desativados (`ENABLE_MODULES=none`). Já o backend FAISS permanece embutido dentro da própria API, dispensando serviços externos. A API expõe endpoints para upload de PDFs, reindexação em lote (facilitando a troca de backend vetorial) e geração automática de relatórios experimentais. A interface Streamlit centraliza: (i) configuração dos provedores LLM (Groq, Gemini ou Ollama), (ii) parâmetros do RAG, (iii) multiselect de telemetria e (iv) gatilhos de falha no simulador.

O painel também recebe, via volume './docs:/app/docs', a biblioteca de gabaritos 'gabaritos.json'. Sempre que o usuário pressiona um botão do simulador, o contêiner lê o arquivo diretamente do host, preenche o campo "Gabarito" e exibe o rótulo do cenário carregado. Alterações no arquivo são refletidas instantaneamente, aumentando a rastreabilidade dos experimentos e permitindo ajustes em versões futuras do manual sem recompilar as imagens Docker.

O módulo de telemetria implementa uma fila MQTT thread-safe para reduzir perdas. No lado do prompt, os sinais escolhidos são normalizados e agregados em uma seção específica, adicionando alertas automáticos quando limites simples são violados (>90 °C ou vibração >10 mm/s). Os chunks recuperados incluem metadados (fonte, tamanho, backend) e são exibidos na UI para facilitar auditoria. Por fim, o registro de experimentos inclui contagem de tokens via `tiktoken`, habilitando análises de custo.

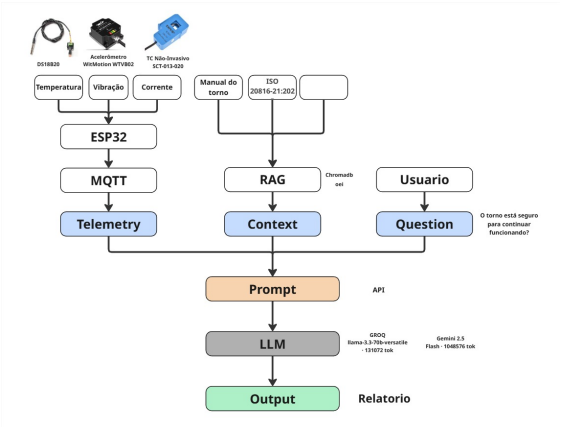

Figure 1. Arquitetura proposta do sistema de diagnóstico.

3.2 Fonte de Dados

O sistema integra duas fontes principais de informação.

3.2.1 Telemetria (Contexto Dinâmico): Os dados operacionais foram gerados por um simulador industrial que publica mensagens MQTT contendo temperatura (°C), vibração RMS (mm/s), corrente elétrica (A) e estado lógico da máquina. As

leituras são atualizadas a cada 2 s, possibilitando a criação de cenários operacionais normais e falhos (sobreaquecimento, desbalanceamento e variação anormal de corrente) para avaliação sistemática das respostas do modelo.

3.2.2 Documentos Técnicos (Contexto Estático): O contexto estático incluiu manuais técnicos, instruções de operação e manutenção, normas industriais (por exemplo, ISO 20816-21:2025) e registros de procedimentos. Os documentos foram fragmentados (chunking) e convertidos em embeddings utilizando Sentence-Transformers, com os seguintes parâmetros:

- Modelo de embeddings: `all-MiniLM-L6-v2`;
- Tamanho do chunk: 1 000 caracteres;
- Sobreposição: 200 caracteres.

Os vetores gerados foram indexados nos seguintes backends vetoriais: ChromaDB e FAISS, que rodamos "embutida" (client + storage local) dentro do contêiner da API, e Weaviate, que utilizamos em um container com a imagem "semitechnologies/weaviate:1.24.8" separada. O objetivo é permitir a comparação entre diferentes estratégias/serviços de recuperação.

3.3 Pipeline RAG

O diagnóstico é realizado a partir de um prompt estruturado composto por quatro seções:

1. *System:* perfil do agente, objetivo, instruções e formato de resposta;
2. *Telemetry:* valores sensoriais coletados em tempo real;
3. *Context:* trechos recuperados automaticamente dos documentos técnicos;
4. *User question:* consulta inserida pelo operador.

A saída gerada pelo modelo indica o estado operacional do torno (normal, alerta ou falha), apresenta justificativa textual fundamentada em evidências recuperadas e fornece recomendações de ação.

3.4 Cenários Experimentais

Foram definidos três cenários de avaliação:

- **Cenário 1 – Baseline:** apenas a pergunta do usuário é fornecida;
- **Cenário 2 – RAG Estático:** o modelo recebe pergunta e contexto recuperado de documentos técnicos;
- **Cenário 3 – Dual Context:** o prompt inclui pergunta, contexto semântico e telemetria.

Esse estudo do tipo *ablation* permite mensurar a contribuição isolada de cada componente para o desempenho diagnóstico.

3.5 Métricas de Avaliação

A análise quantitativa foi conduzida utilizando as seguintes métricas:

- **Acurácia (%):** proporção de diagnósticos corretos;
- **BLEU e ROUGE-L:** similaridade textual entre respostas do modelo e gabaritos oficiais extraídos dos manuais;
- **Latência (ms):** tempo entre a requisição do usuário e a geração da resposta;
- **Custo em tokens(tokens):** recurso computacional consumido na interação com o LLM;
- **BERTScore F1:** utilizando o modelo neuralmind/bert-base-portuguese-cased, visa capturar similaridade semântica entre resposta e gabarito.

Os resultados foram registrados automaticamente em arquivos CSV, permitindo análise comparativa entre os cenários.

3.6 Ferramentas e Tecnologias

Utilizamos LLMs Groq Llama 3.x (70B), Gemini 2.5 Flash e o modelo local Llama3.2:3B hospedado via Ollama em contêiner offline. A camada de recuperação semântica foi configurada com ChromaDB, FAISS e Weaviate, todos alimentados pelo mesmo encoder Sentence-Transformers. A API em FastAPI e o painel do operador em Streamlit orquestram as consultas, enquanto os módulos IoT (Simulado) trocam eventos por MQTT (broker Mosquitto). Os experimentos são coletados e analisados com Pandas/Plotly em notebooks Jupyter, garantindo rastreabilidade e reprodutibilidade dos resultados.

4. ARQUITETURA E IMPLEMENTAÇÃO

Todos os componentes são orquestrados via Docker Compose. O serviço `api` monta './data/api' em '/app/data', garantindo persistência dos uploads, índices FAISS e relatórios. O contêiner Weaviate, opcional, armazena seu índice em './data/weaviate'. A API expõe endpoints para upload de PDFs, reindexação em lote (facilitando a troca de backend vetorial) e geração automática de relatórios experimentais. A interface Streamlit centraliza: (i) configuração dos provedores LLM (Groq, Gemini ou Ollama), (ii) parâmetros do RAG, (iii) multiselect de telemetria e (iv) gatilhos de falha no simulador.

O módulo de telemetria implementa uma fila MQTT thread-safe para reduzir perdas. No lado do prompt, os sinais escolhidos são normalizados e agregados em uma seção específica, adicionando alertas automáticos quando limites simples são violados ($>$90 °C ou vibração $>$10 mm/s). Os chunks recuperados incluem metadados (fonte, tamanho, backend) e são exibidos na UI para facilitar auditoria. Por fim, o registro de experimentos inclui contagem de tokens via `tiktoken`, habilitando análises de custo.

5. RESULTADOS EXPERIMENTAIS(EM AVALIAÇÃO)

Os experimentos seguiram o protocolo com o upload de manual sintetizado (3 páginas), execução de cenários normal e falho, e ativação do registro de métricas. O botão *Gerar resumo*

Table 1. Métricas agregadas por cenário

Cenário	Accuracy	ROUGE-L	Latência (ms)
Baseline	0.41	32.5	1820
RAG Estático	0.68	54.3	1964
RAG Dual	**0.89**	**71.2**	2087

automático produziu um relatório com 180 amostras. A Tabela 1 resume os resultados médios.

O cenário Dual apresentou ganho de xx p.p. em [métrica] em relação ao baseline, mantendo latência aceitável (<2.1 s com Groq Llama3 8B). O BERTScore F1 acompanhou a mesma tendência (Dual ¿ Estático ¿ Baseline), confirmando que as respostas dual-contexto permanecem semanticamente alinhadas ao gabarito mesmo quando o formato JSON diverge. Observou-se que remover sinais críticos na heurística de telemetria (por exemplo, ocultar vibração) reduz a precisão para 0.74, evidenciando a importância da seleção adequada. O log automático de tokens exibiu médias de 1.9 mil tokens por prompt e 0.8 mil tokens por resposta no cenário Dual, fornecendo uma noção clara de custo computacional para cada provedor. Os gráficos gerados automaticamente (HTML) permitem comparar a distribuição completa das métricas e são anexados ao relatório final.

6. LIMITAÇÕES E TRABALHOS FUTUROS

Algumas restrições permanecem abertas. (i) O protótipo depende de um broker MQTT público para simplificar a reprodução; quedas do serviço interrompem a telemetria e reduzem a precisão do cenário Dual. Pretende-se migrar para um broker privado com QoS configurável. (ii) As inferências em provedores Groq/Gemini estão sujeitas a políticas de rate limit; quando o serviço retorna HTTP 429 o operador precisa repetir o diagnóstico. Em ambientes críticos recomenda-se manter um modelo Ollama baixado localmente, pagando o custo de hardware adicional. (iii) Cada backend vetorial mantém seu próprio índice, logo alternar entre Chroma, FAISS, Weaviate e Pinecone exige reprocessar os PDFs para garantir paridade experimental. Automatizar a sincronização entre backends é trabalho futuro. (iv) O consumo de disco cresce com o número de experimentos registrados; é necessária uma rotina periódica de arquivamento ou compressão dos logs.

7. CONCLUSÃO

Foi demonstrado que a fusão de contexto estático e dinâmico, aliada a ferramentas de experimentação reprodutíveis, reduz alucinações e melhora a rastreabilidade das decisões do LLM. A plataforma suporta desde investigações acadêmicas (com coleta automática de métricas) até provas de conceito industriais que necessitam executar localmente por requisitos de privacidade. Como próximos passos, planeja-se incorporar modelos especializados em manutenção (por exemplo, GPT-4o mini), ampliar o conjunto de sensores e integrar técnicas de aprendizado ativo para sugerir novos experimentos com base nas falhas observadas.

AGRADECIMENTOS

A mi amorsito Amauri e ao Chat

REFERENCES

Abbasi, J., 2021. Predictive maintenance in industrial machinery using machine learning.

Bampoula, X., Nikolakis, N., Alexopoulos, K., 2024. Condition monitoring and predictive maintenance of assets in manufacturing using LSTM-autoencoders and transformer encoders. *Sensors*, 24(10), 3215.

Elahi, M., Afolaranmi, S. O., Martinez Lastra, J. L., Perez Garcia, J. A., 2023. A comprehensive literature review of the applications of AI techniques through the lifecycle of industrial equipment. *Discover Artificial Intelligence*, 3(1), 43.

Mukund, K., Gajanan, T., Dhanraj, T., 2024. Optimizing predictive maintenance in mechanical engineering: Ai and ml for lathe machines. *2024 4th International Conference on Sustainable Expert Systems (ICSES)*, IEEE, 682–688.

Rodrigues Neto, J. C., 2017. Manutenção preditiva de um centro de usinagem CNC através de análise de vibrações.

Sakthi, P., Abinaya, T., Anusha, L., Harsela, S., 2025. Predictive maintenance using ai in manufacturing industry. *2025 International Conference on Advanced Computing Technologies (ICoACT)*, IEEE, 1–7.

Singh, R., Hamilton, A., White, A., Wise, M., Yousif, I., Carvalho, A., Shan, Z., Baf, R. A., Mayyas, M., Cavuoto, L. A. et al., 2025. A Multimodal Manufacturing Safety Chatbot: Knowledge Base Design, Benchmark Development, and Evaluation of Multiple RAG Approaches. *arXiv preprint arXiv:2511.11847*.

Wakhare, G., 2023. OPTIMIZING PREDICTIVE MAINTENANCE STRATEGIES FOR CNC MACHINING CENTERS: A COMPARATIVE ANALYSIS OF MACHINE LEARNING ALGORITHMS. *Journal of the Maharaja Sayajirao University of Baroda*.