Diagnóstico de avarias mecânicas para automóveis ligeiros – Artigo Científico

1140406 André Sousa
1140858 Carlos Moutinho
1240144 Carla Henriques
1201247 Miguel Ramos
1200618 Jorge Cunha

Instituto Superior de Engenharia de Porto - R. Dr. António Bernardino de Almeida 431, 4249-015 Porto, Portugal

Este artigo apresenta o desenvolvimento de um sistema pericial em PROLOG para diagnosticar avarias mecânicas em automóveis ligeiros. Descreve-se a metodologia adotada, a arquitetura do sistema e os resultados obtidos, destacando como a aplicação de técnicas de programação lógica e inteligência artificial contribui para diagnósticos precisos e eficientes em veículos modernos.

Palavras-chave: PROLOG, sistema pericial, diagnóstico, carro.

1 Introdução

Nos últimos anos, o avanço da inteligência artificial e do PROLOG tem impulsionado o desenvolvimento de sistemas periciais para diagnósticos de avarias em automóveis. Com a crescente complexidade dos veículos, essas tecnologias proporcionam diagnósticos precisos e automáticos de falhas mecânicas e elétricas.

Em PROLOG, uma base de conhecimento (BC) e um motor de inferência (MI) simulam o raciocínio humano, facilitando a atualização contínua dos diagnósticos e a compreensão do processo pelo utilizador.

2 Estado da Arte

2.1 Metodologia de pesquisa

Para encontrar fontes atualizadas e focadas no diagnóstico de carros com sistemas periciais em Prolog, foi feita uma pesquisa usando as seguintes *queries* nas plataformas Google Scholar, IEEE Xplore, ResearchGate & Academia.edu: "Prolog expert systems automotive diagnostics ", "logic programming in Prolog for vehicle diagnostics" "expert systems for car diagnostics using Prolog", "automotive diagnostics expert systems artificial intelligence", "Logic-based automotive diagnostics PROLOG", "OBD expert systems automotive diagnostics", "Knowledge representation inference automotive diagnostics", "Metaknowledge expert systems automotive", "OBD data uncertainty management PROLOG", "Automotive expert systems for diagnostics".

2.2 Lógica

A lógica proposicional e ferramentas como tabelas de verdade são fundamentais para representar e avaliar condições complexas em sistemas de diagnóstico [1]. As fórmulas de Horn simplificam inferências ao permitir a verificação de condições com uma única premissa verdadeira, ideais para deduções rápidas [2]. Técnicas de inferência e unificação ajudam o sistema a validar diagnósticos com base em dados conhecidos, tornando a resolução lógica mais precisa e automatizada [3].

2.3 Conceitos teóricos – PROLOG

PROLOG é uma linguagem de programação lógica usada para modelar fatos e regras e criar sistemas inteligentes [4]. Seus conceitos principais incluem fatos (informações básicas do sistema), argumentos (características dos fatos) e operadores (que relacionam fatos, formando estruturas complexas). A recursividade e o uso de listas são fundamentais: a recursividade permite que regras se refiram a si mesmas, facilitando a solução de problemas complexos; a manipulação de listas, com métodos nativos, torna a gestão de dados dinâmica e essencial para algoritmos de busca e classificação [5]. Esses conceitos ajudam a estruturar uma rede lógica de sintomas e falhas, atualizar dados de diagnóstico e explorar diferentes cenários de forma ágil e eficiente.

2.4 Sistemas Periciais em PROLOG

Sistemas periciais em PROLOG utilizam princípios de lógica para simular o raciocínio humano na resolução de problemas especializados. Com uma estrutura que inclui uma BC e MI, esses sistemas armazenam e utilizam dados inter-relacionados por regras lógicas, permitindo deduções precisas [6]. A BC é composta por fatos, que representam dados observáveis, e regras que estabelecem relações lógicas, frequentemente em forma de implicações (fórmulas de Horn), facilitando o processo inferencial e a construção de uma rede interligada de conhecimento [6]. O MI aplica as regras através de unificação e resolução lógica para derivar conclusões, identificando falhas ao verificar se os sintomas correspondem a falhas específicas [7]. O metaconhecimento é uma característica avançada que otimiza a inferência e ajuda a lidar com incertezas, ajustando diagnósticos com base em novas informações, o que é essencial para sistemas que lidam com falhas influenciadas por múltiplos fatores.

2.5 Sistemas Periciais OBD

Os Sistemas Periciais OBD em PROLOG armazenam fatos e regras que representam leituras de sensores, como temperatura do motor e emissões de gases, conectando esses dados a diagnósticos específicos [8]. As regras estruturadas em fórmulas de Horn criam uma rede de conhecimento para detetar anomalias e sugerir diagnósticos com eficiência [9]. O MI aplica essas regras usando unificação e resolução lógica para combinar variáveis e identificar problemas com base nos sintomas apresentados [10]. Ele compara

códigos de erro com dados dos sensores, permitindo diagnósticos precisos mesmo em cenários complexos [11]. O metaconhecimento é uma vantagem, ajustando diagnósticos conforme dados contextuais, como histórico de manutenção e condições operacionais, o que aumenta a precisão [12]. Além disso, um módulo de explicação embutido melhora a transparência, mostrando a lógica por trás das inferências e aumentando a confiança do utilizador [13].

2.6 Problemas comuns em sistemas periciais OBD

Sistemas periciais OBD enfrentam alguns desafios. A variabilidade dos sintomas pode resultar em diagnósticos ambíguos, pois diferentes problemas podem produzir sintomas semelhantes [14]. A complexidade dos sistemas automotivos modernos aumenta o número de regras necessárias, tornando a inferência mais lenta e exigindo maior capacidade de processamento [15]. A incerteza dos dados sensoriais, decorrente de desgaste, falhas intermitentes ou interferências externas, também é um obstáculo, levando alguns sistemas a integrarem técnicas probabilísticas para lidar com esses níveis de incerteza [16]. Além disso, as constantes mudanças nas normas de emissões e tecnologias automotivas exigem atualizações frequentes dos sistemas. Por fim, embora os módulos de explicação sejam úteis para profissionais, eles podem ser difíceis de interpretar para utilizadores leigos, que podem não compreender termos técnicos [17].

3 Sistema Pericial de Diagnóstico de Avarias Mecânicas para Automóveis Ligeiros

O desenvolvimento do MI em PROLOG culminou na criação de um sistema pericial capaz de diagnosticar avarias mecânicas em automóveis ligeiros de forma eficiente e precisa. A implementação do sistema foi guiada por decisões estratégicas, principalmente devido às restrições de tempo impostas pelos prazos, o que impactou o escopo e a abordagem do projeto.

3.1 Factos, Regras & Metaconhecimento

No sistema, os factos iniciais incluem os veículos disponíveis e os valores de referência dos componentes. Além disso, há dois tipos de factos adicionais: os relacionados com as perguntas a serem feitas aos utilizadores e as respostas fornecidas. Os factos são criados no formato facto(N, pergunta(Veiculo, Resposta)), onde Resposta pode ser "Sim", "Não" ou um valor numérico, e facto(N, proximo_teste(Veiculo, Pergunta)), que indica uma nova pergunta a ser feita. O valor N representa o número do facto e Veículo refere-se ao identificador do veículo na lista de opções.

As regras no sistema mapeiam as respostas dos utilizadores para gerar a próxima pergunta. Elas seguem o formato “regra ID se LHS então RHS”, onde ID é o número da regra, LHS são as condições que ativam a regra e RHS são os factos a serem criados quando a regra é disparada.

O metaconhecimento é representado por facto_dispara_regras(pergunta(Veiculo, Resposta), [1]), onde pergunta é a questão respondida pelo utilizador. Veiculo e Resposta mantêm o significado que nos factos, e o número no *array* indica as regras que devem ser ativadas.

3.2 Motor de inferência & Módulos de explicação

O MI analisa os factos existentes e verifica se algum dispara uma regra, criando factos quando isso ocorre. Para manter uma interação contínua com o utilizador, este motor tem características especiais. Além de criar factos a partir de regras, ele utiliza métodos que identificam se os novos factos são perguntas ou respostas, adaptando sua abordagem. Se for uma pergunta, o motor a apresenta ao utilizador; se for uma resposta, busca criar o próximo facto. O sistema ajusta as perguntas com base nas respostas dadas, continuando o processo até encontrar um diagnóstico, momento em que o motor para de fazer perguntas.

O sistema inclui três módulos de explicação: “como”, “porque” e “porque não”. O módulo “como” detalha o percurso lógico até o diagnóstico. O “porque” explica um facto, apresentando a pergunta e resposta anteriores que levaram àquela questão. Já o “porque não” mostra todas as perguntas que deveriam ter sido feitas para alcançar determinado facto, até à pergunta que impediu sua realização.

4 Arquitetura e Metodologias

4.1 Fluxograma para o conjunto de carros & regras

Devido ao tamanho/complexidade do fluxograma, as mesmas encontram-se no repositório do projeto.

4.2 Diagrama de sequência que representa todo o fluxo da interação utilizador – motor de inferência em PROLOG

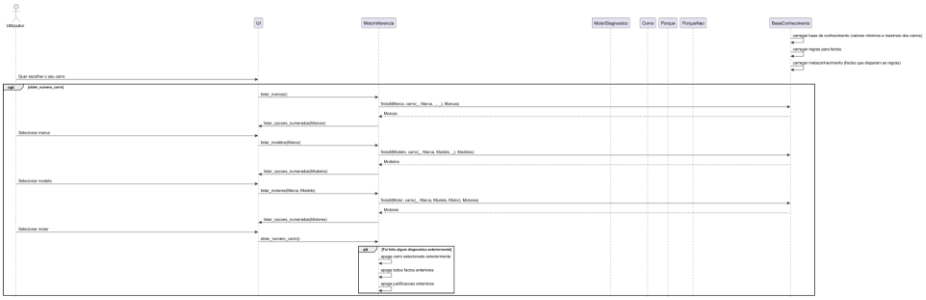

Fig. 1. Diagrama de sequência para a escolha do carro.

Fig. 2. Diagrama de sequência para o motor de inferência e diagnóstico.

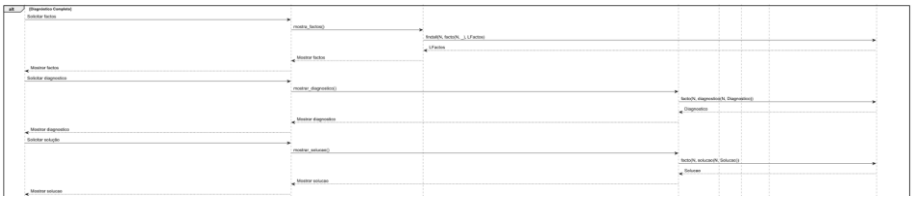

Fig. 3. Diagrama de sequência para mostrar diferentes resultados (factos, justificações e o diagnóstico)

Fig. 4. Diagrama de sequência para o módulo de explicação “como”.

Fig. 5. Diagrama de sequência para o módulo de explicação “porque”.

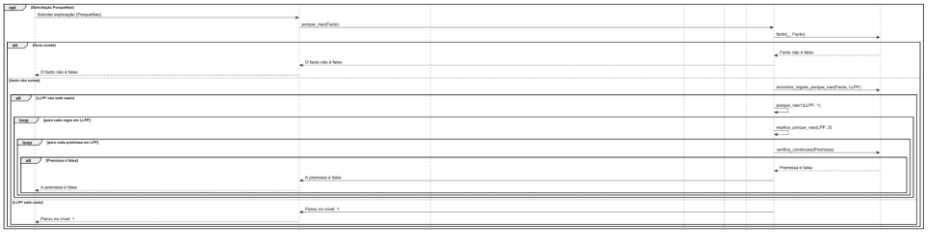

Fig. 6. Diagrama de sequência para o módulo de explicação “porque não”.

5 Resultados e Discussão

Nesta secção apresentamos os resultados obtidos pela utilização desta aplicação.

5.1 Processo de Desenvolvimento e Decisões Tomadas

O objetivo inicial era desenvolver um sistema pericial abrangente para diagnosticar avarias em várias marcas e modelos de veículos. Contudo, devido à complexidade do projeto e ao tempo limitado, foi necessário redefinir o escopo. Decidiu-se focar em problemas mecânicos específicos e restringir o sistema a duas marcas: Opel e Peugeot, com modelos de 2010-2020. Essa escolha baseou-se na experiência e conhecimento aprofundado dos peritos envolvidos no projeto.

5.2 Implementação das Funcionalidades

O sistema pericial desenvolvido inclui funcionalidades importantes para um diagnóstico eficaz e fácil. Possui um *frontend* com um *layout* simples e atrativo, com imagens que orientam o utilizador pelo processo de diagnóstico. A seleção sequencial do veículo permite que o utilizador escolha a marca, modelo e motor, facilitando a identificação e personalizando o diagnóstico com base nos parâmetros específicos de cada modelo.

Fig. 7. Escolha do carro realizada pelo utilizador no PROLOG e na interface amigável.

O processo de diagnóstico guiado do sistema faz perguntas sequenciais com opções de resposta pré-definidas, validadas por predicados específicos. Isso evita sobrecarregar o utilizador com informações excessivas e direciona-o para a verificação dos componentes ou estado do veículo, tornando o diagnóstico mais eficiente.

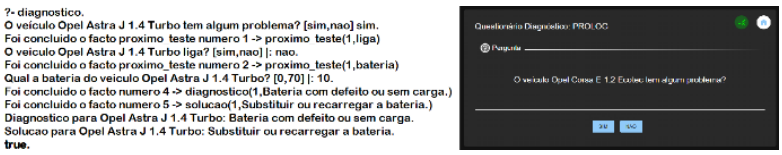

Fig. 8. Diagnostico no sistema PROLOG e na interface.

Fig. 9. Diagnostico após resposta a todas as perguntas na interface amigável.

5.3 Módulos de Explicação no Sistema Pericial Prolog

A implementação dos módulos de explicação no sistema pericial em Prolog foi fundamental para aumentar a transparência e a confiança do utilizador nos diagnósticos. Esses módulos foram criados para ajudar o utilizador a entender o raciocínio por trás das conclusões do sistema, respondendo detalhadamente às questões "como", "porquê" e "porquê não".

Predicado "Como": Detalha o processo lógico que conduziu ao diagnóstico, listando os fatos e regras consideradas.

Predicado "Porquê": Explica as razões pelas quais um determinado diagnóstico foi apresentado.

Predicado "Porquê Não": Justifica por que certos diagnósticos ou hipóteses não foram considerados.

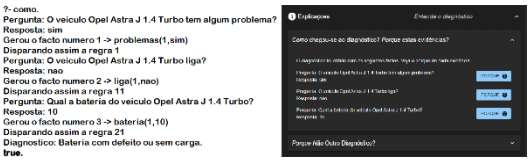

Fig. 10. Modulo de explicação “como” tanto no PROLOG como na interface

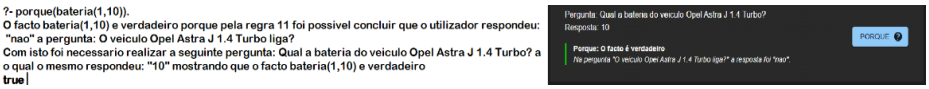

Fig. 11. Modulo de explicação “porque” no PROLOG e na interface.

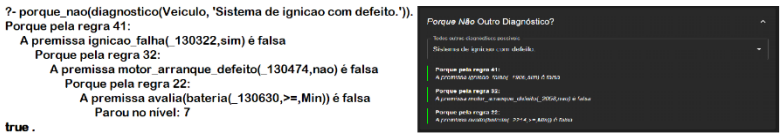

Fig. 12. Modulo de explicação “porquê não” tanto no PROLOG como na interface

5.4 Análise dos Resultados, Desafios e Considerações

A implementação dessas funcionalidades resultou em um sistema capaz de diagnosticar avarias mecânicas em 19 modelos de veículos, fornecendo respostas precisas e justificadas. A inclusão do módulo de explicação, da interface amigável e da opção de

converter o diagnóstico final em PDF melhorou significativamente a usabilidade e a confiança do utilizador nos diagnósticos.

O desenvolvimento enfrentou desafios na gestão do conhecimento e na implementação das funcionalidades dentro do prazo. Foi necessário equilibrar a profundidade dos diagnósticos com a usabilidade, exigindo planeamento cuidadoso e adaptações constantes. A colaboração dos peritos foi crucial para validar as regras e fatos na BC, assegurando diagnósticos precisos. Porém, as limitações de tempo impediram a expansão para outras marcas, modelos ou tipos de avarias, como problemas elétricos ou eletrónicos.

6 Conclusão

O projeto resultou no desenvolvimento de um sistema pericial em PROLOG para diagnóstico de avarias mecânicas em automóveis ligeiros das marcas Opel e Peugeot (2010-2020). A limitação do escopo permitiu focar no aprimoramento das funcionalidades essenciais, criando um sistema eficiente e de alta qualidade. Funcionalidades como a seleção sequencial de veículos, diagnóstico guiado, interface amigável e módulos de explicação proporcionaram uma experiência positiva e aumentaram a confiança nos diagnósticos. As restrições de tempo direcionaram o projeto para um foco mais preciso, mostrando que soluções eficazes podem ser desenvolvidas mesmo com recursos limitados.

6.1 Perspetivas Futuras

Para desenvolvimentos futuros, pretende-se expandir o sistema para incluir mais marcas e modelos, além de diagnósticos de avarias elétricas e eletrónicas. No *frontend*, planeia-se adicionar imagens das peças necessárias e vídeos com instruções da reparação. Também é planeamos a inclusão de informações sobre preços e locais de compra das peças. A integração com o sistema OBD e o uso de técnicas de inteligência artificial, como aprendizagem automática e diagnóstico preditivo, são objetivos futuros. Essas melhorias ajudarão o utilizador a reparar e manter o veículo em bom estado, antecipando avarias.

6.2 Considerações Finais

Este projeto comprova a viabilidade e eficácia do uso de PROLOG no desenvolvimento de sistemas periciais para diagnóstico de avarias mecânicas em automóveis. A combinação de programação lógica, uma BC bem estruturada e uma interface amigável mostrou ser uma abordagem valiosa para a engenharia automotiva.

A experiência reforça a importância de um planeamento estratégico e de uma definição clara do escopo, especialmente sob restrições de tempo. Focar em objetivos específicos permitiu alcançar resultados relevantes e estabelecer uma base sólida para futuras melhorias e expansões.

7 Referências

1. M. Huth and M. Ryan, *Logic in Computer Science: Modelling and Reasoning about Systems*, updated ed. Cambridge University Press, 2019.
2. S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Pearson, 2020.
3. J. W. Lloyd, *Foundations of Logic Programming*, 2nd ed. Springer, 2020.
4. W. F. Clocksin and C. S. Mellish, *Programming in Prolog: Using the ISO Standard*, 5th ed. Springer, 2021.
5. L. Sterling and E. Shapiro, *The Art of Prolog: Advanced Programming Techniques*, 2nd ed. MIT Press, 2019.
6. B. G. Buchanan and E. H. Shortliffe, *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*. Addison-Wesley, 2019.
7. J. Durkin, *Expert Systems: Design and Development*. Macmillan, 2020.
8. E. T. Brown, *Logic-based Automotive Diagnostics: Techniques and Applications*, 2nd ed. Springer, 2022.
9. H. W. Jones and M. Ford, "Knowledge Representation in OBD Systems: From Basics to Advanced Implementations," *International Journal of Automotive Engineering*, vol. 33, no. 4, pp. 152–160, 2021.
10. J. V. Lee, "Application of Horn Clauses in Automotive Logic Programming," *Computer Logic in Engineering*, vol. 19, no. 2, pp. 102–113, 2020.
11. R. Smith and G. A. Robinson, "Logical Inference and Variable Unification for Car Diagnostics," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 9256–9265, 2020.
12. L. Stevens and D. Harper, *Metaknowledge Applications in PROLOG for Complex Diagnostics*, McGraw-Hill, 2021.
13. M. Richards and S. Burman, "Enhanced Explanation Mechanisms for Automotive Systems," *Journal of Expert Systems in Engineering*, vol. 15, no. 3, pp. 417–425, 2022.
14. P. Singh and K. Lin, "Symptom Variability and Diagnostic Ambiguity in OBD Systems," *Journal of Automotive Diagnostics*, vol. 25, no. 4, pp. 342–351, 2021.
15. C. J. White, "Complexity Analysis in Rule-Based Automotive Systems," *Engineering Intelligence Review*, vol. 14, no. 5, pp. 274–281, 2020.
16. S. Tanaka, *Uncertainty Management in Automotive Diagnostic Systems*, Academic Press, 2022.
17. H. C. Davies and J. Lin, "User Interfaces for Expert Diagnostic Systems: Challenges and Developments," *Journal of Human-Machine Systems*, vol. 34, no. 1, pp. 102–110, 2021.