PSNA
COLLEGE OF ENGINEERING & TECHNOLOGY
(An Autonomous Institution)
AICTE | Anna University | NBA | NAAC A++
Estd - 1984

CS2V161 - KNOWLEDGE ENGINEERING – ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

RULE-BASED EXPERT SYSTEM FOR BICYCLE FAULT DIAGNOSIS

PRESENTED BY,

JAGADEESWARAN A (2303921310421082)

Abstract

Diagnosing bicycle faults often requires expert knowledge, especially for beginners who are unfamiliar with mechanical systems. This project presents a Bicycle Maintenance Expert System that uses rule-based artificial intelligence to identify bicycle problems based on symptom selection. The system features a knowledge base containing symptoms, associated mechanical issues, and repair recommendations. Using set-theory matching and confidence-based reasoning, it calculates the likelihood of faults and suggests appropriate actions. Developed in Python using the Streamlit framework, the system provides an interactive web interface where users can choose symptoms and receive real-time diagnosis. The artificial intelligence logic mimics the reasoning process of a bicycle mechanic. Additionally, to make the system more engaging, a humorous Easter Egg has been implemented that appears when all symptoms are selected, suggesting that the bicycle is beyond repair and should instead be scrapped and replaced—or used to buy snacks. The project demonstrates the application of AI in real-world problem-solving, without the use of machine learning or data training.

Introduction

Expert systems are a fundamental branch of Artificial Intelligence designed to imitate the decision-making ability of human specialists. Unlike machine learning, which relies on data-driven statistical models, expert systems work on predefined rules derived from human knowledge. They are particularly useful in areas where domain expertise can be clearly structured into logical rules. The Bicycle Maintenance Expert System developed in this project aims to replicate the reasoning abilities of a bicycle mechanic. The system analyzes selected symptoms such as chain slippage, brake squealing, wheel wobbling, or saddle movement and uses rule-based inference to determine the most probable issue. It not only identifies causes but also explains how the diagnosis was made and suggests suitable corrective actions. The web interface, built using Streamlit, enhances accessibility, allowing even non-technical users to benefit from expert-level mechanical advice.

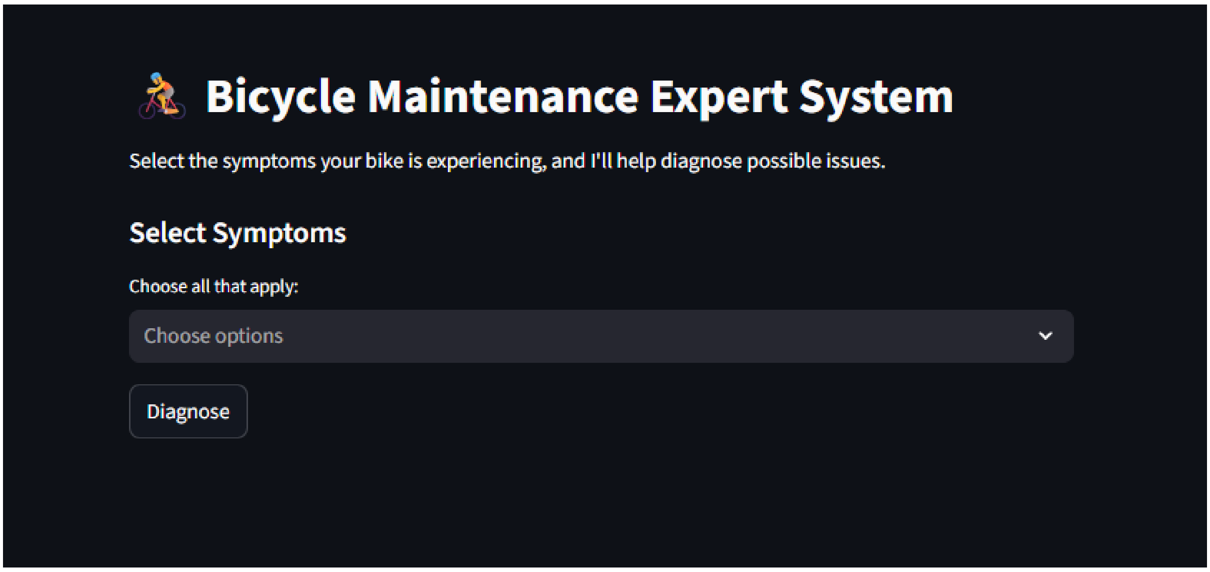
System Architecture

The architecture of this expert system includes five major components: the user interface, input module, inference engine, knowledge base, and explanation facility. The interface developed using Streamlit enables users to interact with the system by selecting symptoms from a list. These inputs are processed by the inference engine, which checks them against predefined rules in the knowledge base. The knowledge base stores information about bicycle issues, including symptoms, confidence scores, and recommended repairs. The inference engine uses set intersection to determine matching issues and then calculates confidence scores based on symptom match percentage and the issue's base reliability. Once diagnosis is completed, the explanation facility presents results in an understandable format, showing not only the likely causes but also the repair actions and severity levels.

Methodology

The methodology followed in building this expert system starts with creating a structured knowledge base consisting of commonly known bicycle faults and their associated symptoms. Each issue is mapped with a set of symptoms, a base confidence level, and repair suggestions. When a user selects symptoms in the interface, the inference engine compares these symptoms with each rule in the knowledge base using set theory logic. If one or more symptoms match, the system considers that issue as a potential diagnosis. Instead of providing fixed probabilities, a dynamic confidence calculation method is used. The more symptoms matched, the more confident the system becomes. The confidence value is calculated using the formula: Final Confidence = (Base Confidence + Match Percentage) / 2. Based on the confidence score, the system classifies the issue as high, medium, or low severity. This approach makes the system behave more realistically compared to simple rule matching. The final output includes the issue name, confidence percentage, severity level, and detailed repair guidance.

An additional feature implemented in this system is the Easter Egg, which activates when the user selects all available symptoms. This is treated as an extreme failure scenario where it is assumed that the bicycle has every possible mechanical problem. Instead of showing normal results, the system humorously suggests scrapping the bicycle and spending the money on snacks. This feature enhances user engagement and makes the system interactive and memorable.

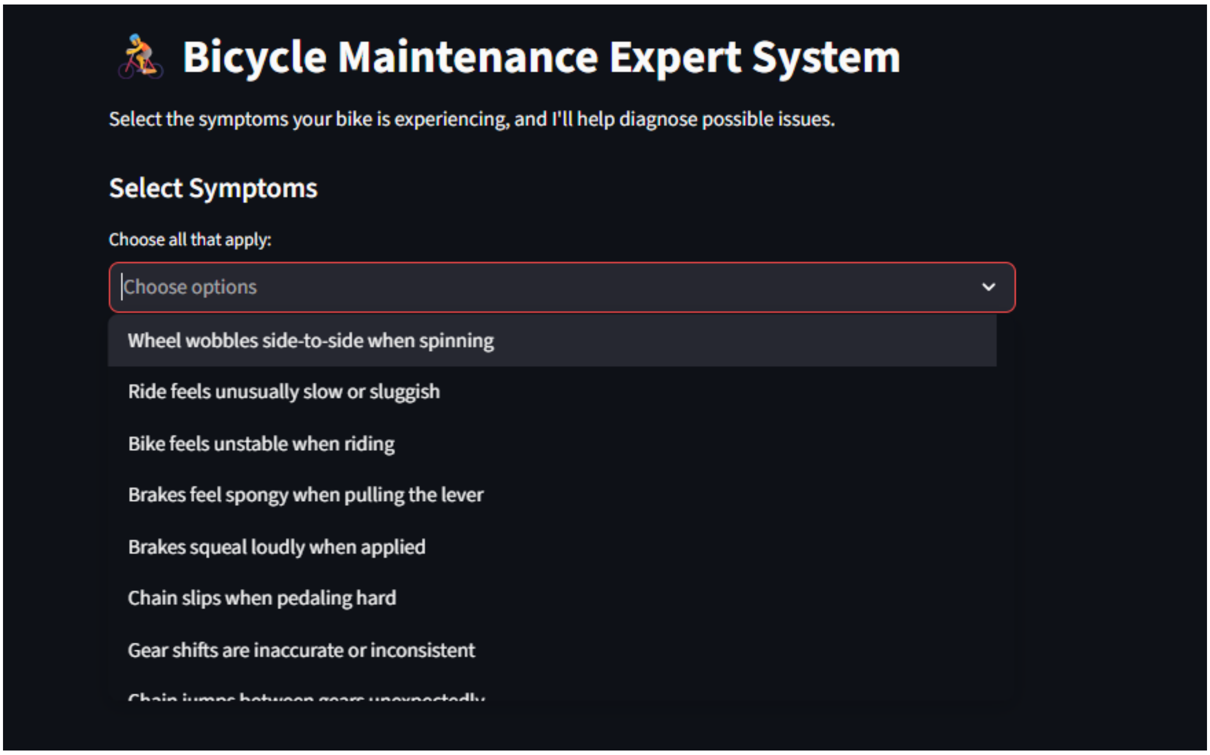


Technologies Used

This project is developed using Python as the core programming language because of its simplicity, readability, and strong support for AI applications. The Streamlit framework is used to build the user interface due to its ease of use and real-time interactivity. The system's reasoning mechanism relies on rule-based AI and set theory rather than machine learning, making it transparent and explainable. HTML and CSS were used to improve interface styling, while markdown formatting was used inside Streamlit to enhance presentation quality.

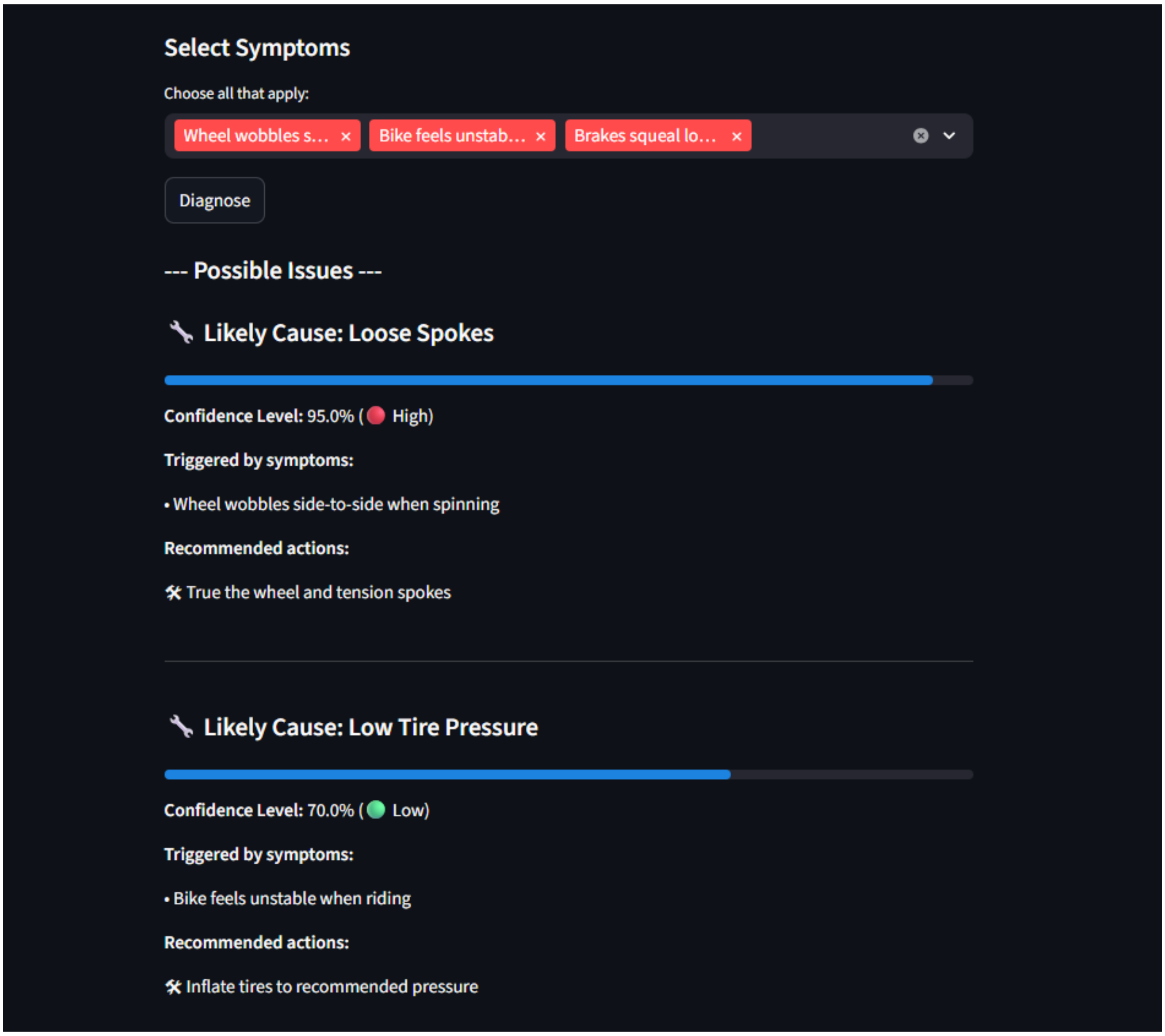
Results and Discussion

The system was tested with various combinations of symptoms to check accuracy and relevance. For instance, when users selected symptoms such as “Chain slips when pedaling hard” and “Gear shifts are inaccurate,” the system correctly diagnosed “Stretched Chain” with high confidence and suggested replacing the chain. In another case, when the symptoms included “Wheel wobbles” and “Brake rubs against rim,” the system suggested “Bent Rim” with high severity, explaining why it was triggered and how to fix it. The use of dynamic confidence calculation made the results more realistic by ranking the most probable faults higher. The system successfully demonstrated logical reasoning, clear explanations, and accurate diagnosis. When all symptoms were selected together, the system triggered the Easter Egg, which added a humorous touch and improved user satisfaction.



Future Scope

In the future, the system can be enhanced with image-based AI diagnosis, allowing users to upload photos of their bicycle and detect physical damage such as tire wear, broken spokes, or bent rims. Voice-based conversational diagnosis could also be added, allowing the system to act as a virtual mechanic chatbot. Another possible enhancement is the integration of IoT sensors, enabling real-time detection of tire pressure, chain tension, vibration, and brake performance. Additionally, exporting the diagnosis as a PDF report or converting the system into a mobile application would increase its usability and practicality.



Conclusion

This Bicycle Maintenance Expert System demonstrates the practical application of rule-based artificial intelligence in diagnosing real-world mechanical problems. It effectively mimics the reasoning approach of a bicycle mechanic using structured rules and dynamic confidence-based reasoning. The system provides reliable diagnosis output, repair recommendations, and clear explanations, making it suitable for beginners and intermediate users. The integration of an interactive interface, intelligent decision-making system, and a fun Easter Egg makes the application both educational and engaging. With further enhancements, it could develop into a powerful digital assistant for bicycle maintenance.

References

1. Giarratano, J. C., & Riley, G. D. (2005). *Expert Systems: Principles and Programming* (4th ed.). Cengage Learning.
   (Core concepts of rule-based expert systems, knowledge base design, and inference rules.)

2. Durkin, J. (1994). *Expert Systems: Design and Development.* Prentice Hall.
   (Explains expert system architecture, rule matching, reasoning, and confidence scoring.)

3. Rao, K., & Reddy, B. (2017). Rule-Based Expert Systems for Mechanical Fault Diagnosis. *International Journal of Computer Applications*, 162(6), 25–29.
   (Directly relates to diagnosing mechanical problems using symptom-based reasoning.)

4. Streamlit. (2024). *Streamlit Documentation: Building Intelligent Web Apps*. Retrieved from https://docs.streamlit.io/
   (Official Streamlit documentation used to build the web interface.)

5. Python Software Foundation. (2024). *Python Language Reference*. Available at https://www.python.org/doc/
   (Documentation for Python, used for system development and logic implementation.)

6. Shimano Technical Support. (2023). *Bicycle Component Troubleshooting and Maintenance Guide.* https://bike.shimano.com
   (Real-world reference for bicycle issues, symptoms, and repair suggestions.)

7. Parkin, J. (2015). *Designing and Maintaining Bicycles: A Mechanical Guide*. Routledge.
   (Explains bicycle mechanics, common faults, and maintenance procedures.)