www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 15(2), May-Aug 2024, 185-196. ISSN: 2007-1558. https://doi.org/10.61467/2007.1558.2024.v15i2.450

Vehicle Engine Fault Diagnosis Approach Based on a Decision Tree and Knowledge Base

Antonio Pérez-Vázquez, Mario Anzures-García, Luz A. Sánchez-Gálvez
Benemérita Universidad Autónoma de Puebla.
E-mails: antonio.perezv@alumno.buap.mx, mario.anzures@correo.buap.mx, sanchez.galvez@correo.buap.mx

Abstract. This paper proposes an approach to vehicle engine fault diagnosis utilizing a decision tree, knowledge bases, databases, and fuzzy logic. It focuses on identifying patterns that allow us to determine the cause of irregular behaviors in engines, implementing a practical system in mechanic workshops. The study is based on the use of machine learning techniques, especially decision trees, which allow representing fault diagnosis in a branching manner, depending on the presented characteristics or driving conditions. These characteristics are stored in a knowledge base that establishes conditions for making the appropriate and timely diagnosis, initially fed by a database that contains OBDII codes from the vehicle's computer with the cause and the elements embedded. In order to improve the precision and accuracy of the proposal, the decision tree is pruned. Finally, the proposal is validated to probe its feasibility with various evaluation metrics to obtain an accurate diagnosis in the automotive field.

Keywords: Decision tree, Knowledge base, Fuzzy Logic, Vehicle engine fault, Diagnosis approach.

Article Info
Received May 11, 2024
Accepted Jun 1, 2024

1 Introduction

Automobiles for about 138 years have been presented, when on July 3, 1886, Karl Benz made his first public journey in the Patent-Motorwagen through the streets of Mannheim, Germany; establishing a milestone in automotive history. Since then, the need to provide service to these automobiles emerged. These services have changed over time and evolved alongside engines. From the implementation of additives to improve the performance of the car, to the introduction of electronics to enhance efficiency and facilitate diagnosis.

The introduction of electronics in automobiles was a gradual process that began in the 1960s and 1970s. In the latter decade, electronic emissions control systems were introduced to comply with environmental regulations. These systems use sensors and electronic control units to monitor and adjust the engine's operation, enabling precise control of the fuel and air mixture, thereby reducing pollutant emissions (Pérez Darquea, 2018).

On-Board Diagnostic (OBD) codes were developed to interpret faults in the sensors and actuators that the vehicle now had. They were first introduced in the 1980s with the OBD-I system. The gradual implementation of OBD-II began in 1994 for vehicles manufactured in the United States, becoming mandatory for all new cars and light trucks from 1996 onwards (Rouhiainen, 2018). It became the standard diagnostic system used by most car manufacturers, allowing the detection and recording of faults in different vehicle systems, thereby facilitating the diagnostic and repair process.

However, a fault code is not enough to determine an accurate diagnosis. Therefore, this paper proposes motor fault diagnosis based on decision trees, knowledge bases, and fuzzy logic. Primarily supervised learning and Decision Trees (Sempere, 2014) because this diagnosis is represented in a branching manner depending on the set of presented characteristics, where leaf or terminal nodes symbolize the specific fault. The decision tree uses the Base Knowledge (Guarino & Giaretta, 1995) - containing driving characteristics or conditions, a set of rules and facts that feed into the decision tree algorithm. So that the model learns

the relationships and patterns by itself, and Fuzzy Logic (Trillas & Eciolaza, 2015) - allows an optimal classification of the engine block coolant temperature range to determine a possible heating problem— in order to generate a diagnosis depending on the characteristics or conditions observed in the vehicle engine. On the other hand, the interpretability of complex decision tree models can be a limitation, therefore this research employs pruning methods as a solution. Pruning techniques aim to simplify decision trees removing unnecessary branches or nodes, thus reducing the model's complexity while preserving its predictive accuracy.

Implementing and maintaining the diagnostic system isn't a challenge because it is based on diagnostic tools and manuals of automotive maintenance, which are known to the technicians. Furthermore, this system has been implemented in a mechanic workshop web application that supplies a Chatbot (Adamopoulou & Moussiades, 2020), facilitating and simplifying its use for any user (technicians and customers).

The document is organized as follows: Section 2 briefly presents the state of the art related to vehicle failure diagnostic and/or decision tree. Section 3 describes the development of the proposed approach to make an accurate fault diagnosis. Section 4 explains the results and discussion. Finally, Section 5 outlines the conclusions and future work.

2 State of the art

Research works related to decision trees applied to classification problems, fault diagnosis through expert systems and agents, websites including user experiences sharing and symptoms of failure, and desktop diagnostic applications providing fault codes and vehicle symptoms are analyzed.

De Sumana & Chakraborty (2018) introduces a Case-Based Reasoning (CBR) methodology for the Car Fault Diagnosis System (CFDS). This system employs a decision tree to store cases and the Jaccard similarity method to calculate the similarity between new and stored cases. However, the article does not indicate the exact type of decision trees used. Neither, it does not provide specific numerical results derived from these metrics in terms of accuracy, recall, F1-score, or other typical evaluation metrics for car fault diagnosis systems.

Murphey et al. (2009) discusses an ensemble approach of neural networks for vehicle fault detection. It introduces a two-step ensemble approach utilizing an ensemble selection algorithm (BFES) and an analogical ensemble decision function based on Bayes (A-Bayesian-Entropy). Additionally, it presents experimental results demonstrating the performance of various proposed methods in detecting faults in vehicle engines. However, specific performance values may vary depending on the dataset, algorithmic approach used, and other study considerations.

Crossman et al. (2003) focuses on diagnosing faults in automotive signals, particularly within the context of vehicle signal analysis. It discusses key tasks such as analyzing the behavior of faulty signals, automatically segmenting signals, extracting features, and selecting important features. Although it does not mention the evaluation metrics used to measure the performance of automotive signal fault diagnosis, it describes evaluating the performance of proposed algorithms and methods without specifying any values.

Djurdjanovic et al. (2007) discusses the application of an anomaly detection system based on the Growing Structure Multiple Models System (GSMMS) to monitor crankshaft dynamics in an engine. It mentions that the performance of the anomaly detection system improves as more samples are used, and nearly perfect detection is achieved when a data length of around 1000 revolutions is reached. It also notes that the average detection time is substantially shorter to achieve a similar detection accuracy when increases of 20% and 30% in viscous damping coefficients are detected.

Charkaoui et al. (2005) focuses on the application of a decision tree classifier to detect and isolate faults in vehicles in the automotive aftermarket industry. As the automotive industry has evolved with the introduction of onboard electronic systems and the growth of electronics-based functions, repairability for aftermarket technicians has become more challenging. The article uses metrics such as error rate and novelty rejection rate to evaluate the classifier's performance. It was found that the classifier had an average error rate on the order of 3%.

Covarrubias et al. (2013) uses an inference engine and Prolog to create the DIMv3 system as an aid in the diagnosis process. DIMv3 is a desktop application where the compatibility of the operating system, manual updates, and familiarity with the software should be considered. It evaluates the expert system from various perspectives, including its performance, ease of use, reliability, ability for continuous knowledge enrichment, and capability for updating to remain relevant in an ever-evolving automotive environment.

Xiaolei & Xiaobing (1999) discusses the use of rough set theory in diagnosing faults in vehicle transmission systems. It explains how the fault feature vector is minimized by calculating the importance of each attribute and its dependency, simplifying data and enhancing real-time diagnostic efficiency. However, the article does not provide specific details on the metrics used or the accuracy of the proposed method.

Gutiérrez (2008) addresses the use of rule-based systems in deterministic situations in our daily lives, such as traffic control, security, and banking transactions. In this context, deterministic rules are a simple methodology used in expert systems. It focuses on discussing concepts and principles related to rule consistency, fact consistency, and handling uncertainty in rule-based expert systems. These metrics are more qualitative and conceptual, and their evaluation is generally done through logical and conceptual analysis rather than numerical values.

Tapia Barrientos (2009) presents a prototype rule-based expert system for automated diagnosis of the car's hydraulic brake system. However, maintaining and updating the knowledge base of the expert system presents challenges, requiring time and resources to incorporate new rules, remove obsolete rules, and ensure that information is accurate and up-to-date. Additionally, it does not involve evaluation metrics.

Soria Mamani (2013) focuses on developing a probabilistic expert system for diagnosing failures in electronic injection engines of vehicles, which can help users reliably and educationally diagnose problems in engines of this type of vehicles. Additionally, the accuracy of the results provided by the expert system is assessed on a scale from "Poor" to "Excellent," with a 90% acceptance rate. The evaluations are conducted by experts in the field of automotive mechanics, specifically by final-year students in the Automotive Mechanics technical program at the Technical Institute (ITAB).

Ayala Ramos (2000) an agent model based on expert knowledge of Automotive Mechanics is designed for the diagnosis of mechanical faults in automobile engines, using fuzzy logic and rule-based. The amount of information provided is evaluated with an 85% acceptance rate for the quantity of information provided. The results of these evaluations were conducted by final-year students in the Automotive Mechanics technical program at the Technical Institute (ITAB).

Dietterich (2000) explores and compares the effectiveness of three methods to enhance the performance of the C4.5 decision tree algorithm: Bagging, Boosting, and Randomization. The study focuses on constructing a diverse and accurate ensemble of classifiers. Additionally, the article describes the differences in error rates between the different ensemble methods (Randomized C4.5, Bagged C4.5, and Adaboosted C4.5) and the base classifier C4.5 across various application domains, but it does not include specific numerical values for these differences.

Song & Ying (2015) provides an overview of decision tree applications in the context of biostatistics and psychiatry, highlighting their advantages, disadvantages, and practical applications in medical research. It presents commonly used algorithms for developing decision trees, such as CART, C4.5, among others. Furthermore, it does not specify the exact accuracy metrics or evaluation methods used in the examples presented.

Patel & Prajapati (2018) explores the use of decision tree algorithms in the context of machine learning, focusing on three specific decision algorithms: ID3, C4.5, and CART. The main goal of the study is to compare these algorithms in terms of accuracy, runtime, and other metrics using a dataset of automobiles. It employs the confusion matrix, accuracy, and precision to evaluate the proposed algorithms, achieving an accuracy of 97.11% and a precision of 97.2% The results show that, although CART has the slowest execution time, it has the highest accuracy compared to ID3 and C4.5.

In addition, opinion platforms (Opinautos, 2023), (Carcomplaints, 2023) and diagnostic tools (Scanator,2023) have been analyzed. Unfortunately, the information provided by users is not always thoroughly verified, so situations may arise where problem reports are based on misunderstandings or incorrect diagnoses as well as limitations in coverage and technical support hours provided by

3 Development

In this paper, an approach for the detection of engine failures based on decision trees, databases, knowledge bases, and fuzzy logic; is proposed (see Figure 1). This aims to provide accurate diagnosis to customers in an automotive workshop, improving on previously addressed proposals in the state of the art. Our proposal is integrated into Coq's Tuning Performance web application, which will enable more efficient and precise management of engine faults, optimizing the time and resources dedicated to their repair. Additionally, it is expected that this approach will contribute to increasing the reliability and accuracy of the diagnosis, which in turn could enhance customer satisfaction as it is available to clear customers' doubts.

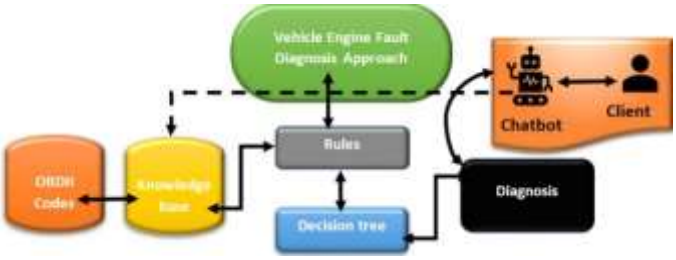

Figure 1. Proposal outline.

Database with OBDII Codes

To manage the set of OBDII fault codes, a database is created (Figure 2), including the cause of the present problem as well as possible solutions, as well as the element(s) generating the alert.

id	ask	answer	solution
1	P0001	Coil volume regulation control circuit	NULL
2	P0002	Fuel Volume Regulation Control Circuit...	Wiring, governor control solenoid
3	P0003	Fuel Volume Regulation Control Circuit...	Short circuit to ground in the wiring, solenoid ...
4	P0004	Fuel Volume Regulation Control Circuit...	Wiring open circuit/short circuit to pos...
5	P0005	Control circuit of Valve "A" for shutting off fuel...	Wiring open circuit, fuel shut-off valve...
6	P0006	Control circuit of Valve "A" for shutting off fuel...	Short circuit to ground in the wiring, control valve...
7	P0007	Control circuit of Valve "A" for shutting off fuel...	Short circuit to positive in the wiring, valve...

Figure 2. OBDII Codes Database.

Engine block fault diagnosis interface

This database was created from the control panel developed with AJAX, which works as follows:

- The codes are queried through the Chatbot, displaying the cause and solution respectively, as depicted in Figure 3.

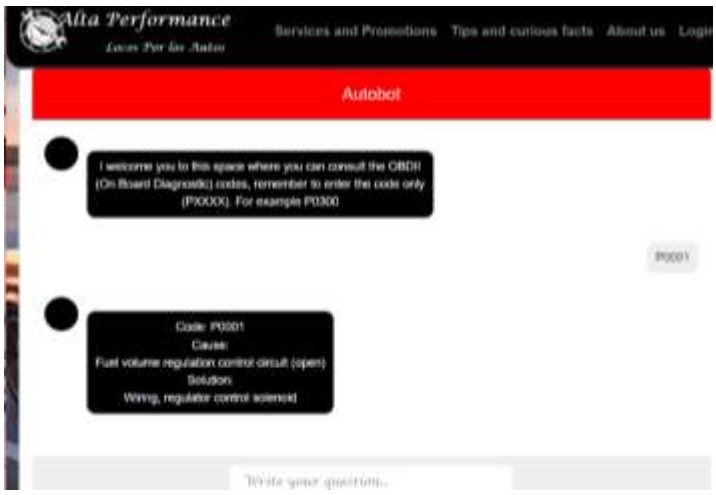

Figure 3. Querying Fault Codes from the Chatbot

- If the code is not found, it is stored in a temporary table in the database. So that a specialized technician can address it, as well as the ability to update existing information. As it is well known, new models of automotive vehicles are released every year, incorporating a broader set of sensors and actuators to improve their performance.

Therefore, it is extremely important to keep the dataset updated. Thus, an unknown codes panel was developed, displaying those faults that are not yet in the database, and a fault update panel to provide updates or corrections to previously entered records, as shown in Figure 4 and Figure 5.

Figure 4. Unknown Codes Panel

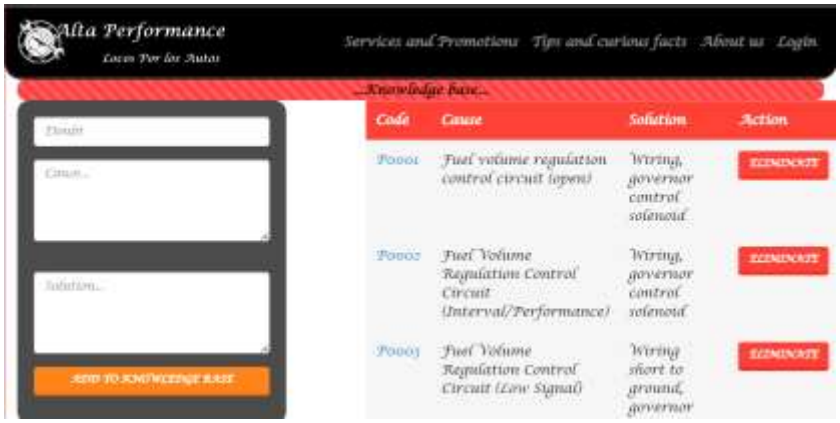

Figure 5. Fault Codes Update Panel

While OBD-II fault codes are not updated annually in the same way as software versions of diagnostic tools, they do undergo changes and additions over time as the electronics embedded in cars advance, thus it is very important to update information about them. Therefore, an update panel is created to simplify that task (Figure 5).

Development of a Knowledge Base

For the diagnosis of engine block failures, a knowledge base is created, considering certain characteristics, or driving conditions, along with a set of rules and facts that feed into the decision tree algorithm.

These fault conditions are divided into 3 groups:

- Failures upon engine start.
- Performance failure
- Heating failure

Each group encompasses various elements of the engine block (sensors, actuators, and mechanical parts) that may be operating out of its range, and depending on the values obtained, so the responsible for the failure is identified.

The knowledge base is stored in a CSV file (see Figure 6), which is used as a training set for the model.

faulty_system	Voltage_on	Voltage_off	Cranking	Spark	Injection	At_time	Compression	OBDII_codes
Bad_alternator	V<12	V>12	Cranks	Spark_present	Injection_present	Timely	Compression>120	no
Starter_or_power_supply_issue	V>12	V>12	No_crank	Spark_present	Injection_present	Timely	Compression>120	no
Bad_battery	V>12	V<12	Cranks	Spark_present	Injection_present	Timely	Compression>120	no
Disuse	V>12	V>12	Cranks	Spark_present	Injection_present	Timely	Compression>120	no
Ignition_system_issue	V>12	V>12	Cranks	No_spark_present	Injection_present	Timely	Compression>120	no
Fuel_pump_or_power_supply_issu	V>12	V>12	Cranks	Spark_present	No_injection	Timely	Compression>120	no
Timing_issue	V>12	V>12	Cranks	Spark_present	Injection_present	Out_of_time	Compression>120	no
Engine_opening	V>12	V>12	Cranks	Spark_present	Injection_present	Timely	Compression<130	no
Check_codes	V>12	V>12	Cranks	Spark_present	Injection_present	Timely	Compression>120	yes
Inspect_intake_system	V>12	V>12	Cranks	Spark_present	Injection_present	Timely	Compression>120	no
Fuel_pump_and_lines	V>12	V>12	Cranks	Spark_present	Injection_present	Timely	Compression>120	no
Spark_plugs,_wires,_coils	V>12	V>12	Cranks	No_proper_spark_present	Injection_present	Timely	Compression>120	no
CKP_or_CMP_sensor	V>12	V>12	Cranks	Spark_present	Injection_present	Timely	Compression>120	no
Defective_injector	V>12	V>12	Cranks	Spark_present	Injection_present	Timely	Compression>120	no
Top_up	V>12	V>12	Cranks	Spark_present	Injection_present	Timely	Compression>120	no
Change_gasket_or_hose	V>12	V>12	Cranks	Spark_present	Injection_present	Timely	Compression>120	no
Change_thermostat	V>12	V>12	Cranks	Spark_present	Injection_present	Timely	Compression>120	no
Change_fan_or_power_supply	V>12	V>12	Cranks	Spark_present	Injection_present	Timely	Compression>120	no

Figure 6. Knowledge Base

Decision Tree Development

A decision tree is implemented using CART (Classification and Regression Trees), which works recursively by dividing the dataset into smaller subsets. Each split is made, selecting the feature and threshold that best separates the data, such as information gain in the case of classification (McTavish et al., 2022).

The dataset provided in CSV format is used as the training set. This training set is essentially a collection of examples used to teach the decision tree model how to make predictions based on certain observed features or driving conditions.

The separation of attributes (X) and labels (Y) is done to facilitate the model training process, where:

- X refers to all columns in the data except 'faulty_system'. These are the features the model uses to make predictions.
- Y refers to the 'faulty_system' column. These are the values the model attempts to predict.

Many machine learning algorithms, including decision trees, work better with numerical data. However, real-world datasets often encounter categorical features, i.e., variables that can take a limited set of categories or labels.

One-Hot Encoding is a technique that converts categorical variables into a format that can be provided to machine learning algorithms to improve their performance.

This process ensures that all features, even categorical ones, are represented in a numerical way that can be interpreted by the algorithm during training and prediction.

Creation and Pruning of Decision Trees

Once One-Hot Encoding is performed, four decision tree models are created and trained using different pruning methods: CCP (cost complexity pruning), Max Depth (pruning by height), Min Samples Leaf (pruning by minimum number of samples in leaves), and Min Samples Split (pruning by minimum number of samples to split a node). Pruning methods are essential for reducing the complexity of decision trees (Zhong et al., 2008). These methods selectively remove branches or nodes from the tree that do not contribute significantly to its predictive accuracy. Doing so, pruning prevents the trees from becoming overly complex and helps prevent overfitting, where the model performs well on the training data but poorly on unseen data (Tong et al., 2022).

Each model is trained using the encoded data X_encoded and the labels Y as illustrated in algorithm showed below.

```
# Cost complexity pruning method (CCP)
clf_ccp = DecisionTreeClassifier(ccp_alpha=0.05)
clf_ccp = clf_ccp.fit(X_encoded, y)

# Height pruning method
clf_max_depth = DecisionTreeClassifier(max_depth=7)
clf_max_depth = clf_max_depth.fit(X_encoded, y)

# pruning method by minimum number of samples in leaves
```

```
clf_min_samples_leaf = DecisionTreeClassifier(min_samples_leaf=3)
clf_min_samples_leaf = clf_min_samples_leaf.fit(X_encoded, y)

# pruning method by minimum number of samples to split a node
clf_min_samples_split = DecisionTreeClassifier(min_samples_split=10)
clf_min_samples_split = clf_min_samples_split.fit(X_encoded, y)
```

An instance of the DecisionTreeClassifier class provided by the scikit-learn library is created. This instance is the decision tree classifier that will be used to train the model.

Afterward, a new instance is encoded using the same encoding process used for the training data. It ensures that the columns of the new instance match the columns of the training set. The new instance is generated with the data captured by the user in the form. Predictions are made with each of the trained models for the new instance using the prediction method.

Subsequently, the models are evaluated using a confusion matrix, obtaining true positives, true negatives, and calculating the precision, accuracy, error rate, novelty rejection percentage, and mean cross-validation score of each model.

This is similar to the learning process, but instead of using the entire data set to train the model, the knowledge acquired by the model is applied to a specific instance to predict that instance, but in a more specific context and practical.

Classification using fuzzy logic

Controlling and knowing the engine coolant temperature of a car is crucial for several reasons. Primarily, to prevent engine overheating, and keep it running within a safe range. If the coolant temperature rises too much, it can cause the coolant liquid to evaporate or even boil. This can lead to damage in the cooling system, such as the formation of air bubbles and loss of pressure. As well as causing severe damage such as internal component melting, deformation of metal parts, and even complete engine failure.

Furthermore, optimizing performance and efficiency, as the engine operates optimally within certain temperature ranges. If the coolant temperature is too low, the engine may operate less efficiently, resulting in lower performance and higher fuel consumption. On the other hand, if the temperature is too high, it can cause power loss and an increase in polluting emissions.

The importance of using fuzzy logic lies in the fact that not all vehicles have temperature indicators on the dashboard, and its correct classification generates a broader overview of the set of sensors and actuators that could present abnormal behavior. For the user who does not know the optimal temperature of his unit, he can identify if there is any latent risk or if it is in the optimal range with the help of these three categories

The fuzzy logic plays a crucial role here; since it allows us to classify a vehicle's temperature not only as low or high but with more values to classify temperatures as categories using fuzzy membership functions.

The first step is to define the temperature universe (temperature_universe = np.arange(0, 150, 1)) as a range of values from 0 to 150 degrees Celsius with increments of 1 degree.

This is done because current internal combustion engines are mostly made with aluminum heads and do not support values above that temperature. Formerly they were made of cast iron or cast iron, so they could withstand operating temperatures of up to approximately 600-700°C without experiencing permanent deformation.

In the next step, Fuzzy membership functions are defined for the categories of low, medium, and high temperature using fuzz.trimf. Which is a function provided by the scikit-fuzzy library, used to assign degrees of fuzzy membership to each temperature in the universe as depicted below.

- low = fuzz.trimf(temperatura_universe, [0, 40, 80])
- mean = fuzz.trimf(temperatura_universe, [75, 90, 110])
- hig = fuzz.trimf(temperatura_universe, [105, 130, 150])

Then, the temperature is classified, and its degree of membership to the categories of low, medium, and high temperature is calculated using fuzz.interp_membership as depicted in the following.

- membership_low=fuzz.interp_membership(temperatura_universe,low,temp)
- membership_mean=fuzz.interp_membership(temperatura_universe,mean,temp)

- membership_hig=fuzz.interp_membership(temperatura_universe,hig,temp)

Finally, the membership degrees are printed alongside the temperature, providing a fuzzy classification of temperature into the categories of low, medium, and high.

4 Results

The results obtained, from this research work, are mentioned below.

The confusion matrix in Figure 7 illustrates that all predicted values match the actual values, indicating an excellent ability of the model to accurately predict classes.

Results show that the pruning method using CCP achieved 100% precision and accuracy with 0% error rate and novelty rejection as illustrated in Figure 8.

In contrast, Max Depth, Min Samples Leaf, and Min Samples Split pruning methods showed varied results in terms of precision and accuracy.

Max Depth pruning method reached a precision of 39.39%, an accuracy of 44.44% an error rate of 50%, and a novelty rejection of 0% as depicted in Figure 9.

Min Samples Leaf pruning method yielded less favorable results with a precision of 0.31% an accuracy of 5.56% an error rate of 94% and a novelty rejection of 0%. These results suggest that adjusting the minimum number of samples per leaf may not have been effective in improving the tree model's performance as illustrated in Figure 10.

The Min Samples Split pruning method showed slightly more favorable results, with a precision of 50.62% an accuracy of 55.56% an error rate of 44%, and a novelty rejection of 0% as depicted in Figure 11.

Cross-validation means score for each pruning method a score of 100% for CPP is shown, 36% for Max Depth, 09% for Min Samples Leaf and 54% for Min Samples Split as shown in Figure 12.

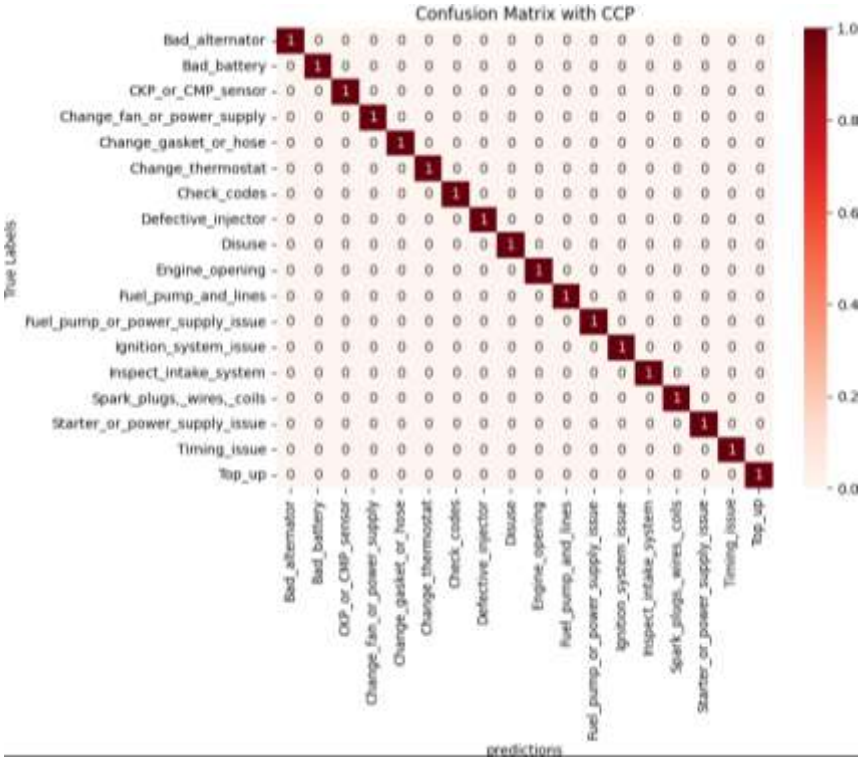

Figure 7. Confusion matrix with CPP.

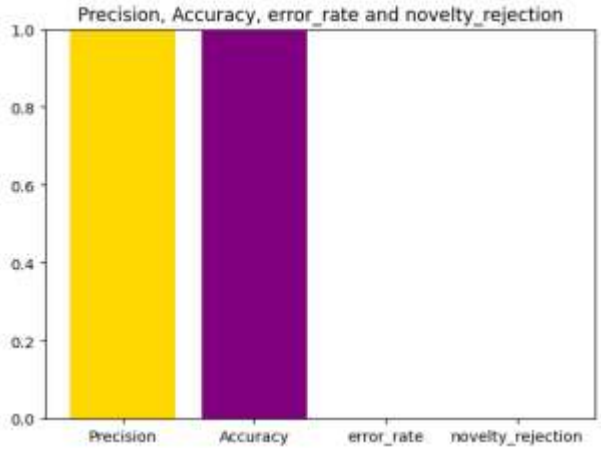

Figure 8. Precision, accuracy, error rate, and novelty rejection with CPP.

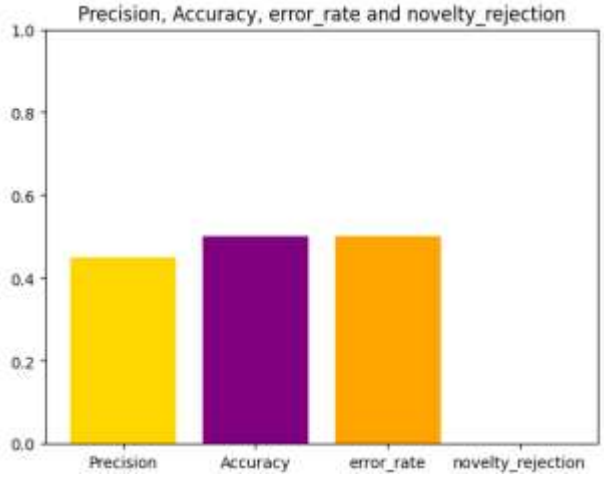

Figure 9. Precision, accuracy, error rate, and novelty rejection with Max Depth.

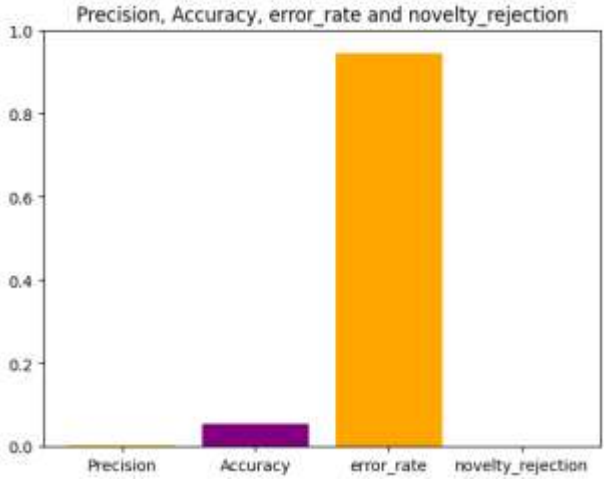

Figure 10. Precision, accuracy, error rate, and novelty rejection with Min Samples Leaf.

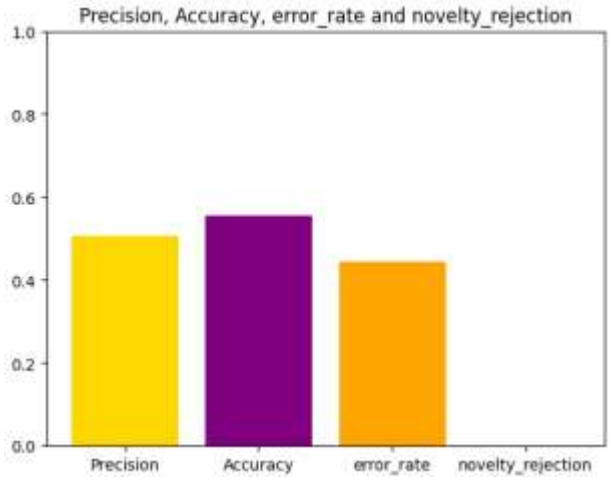

Figure 11. Precision, accuracy, error rate, and novelty rejection with Min Samples Split.

In summary, the results of different pruning methods in the CART decision tree model emphasize the importance of carefully selecting appropriate pruning techniques to optimize model performance and ensure accurate prediction of output classes.

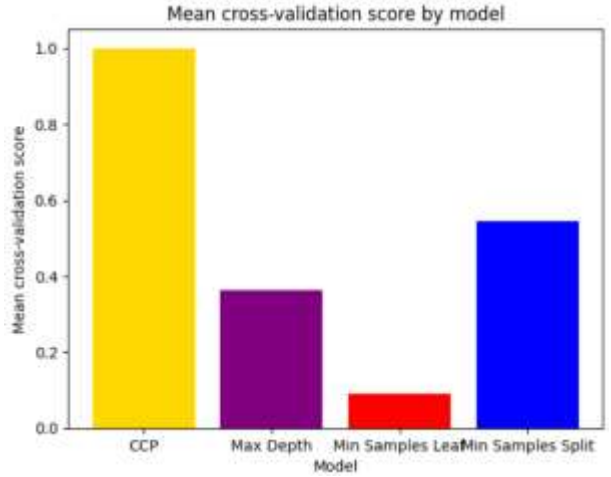

Fig12. mean cross-validation score

Finally, the best pruning method for this case study is CCP, used to generate a diagnosis and identify the cause of the mechanical failure in the unit. CPP reduces response time generating a decision tree without overfitting.

This precision is achieved because the rules or expert knowledge are specific and focus on a particular area of the problem. Since the rules are designed specifically for the engine block, and based on diagnostic procedures established in maintenance manuals.

To gather information about the symptoms or conditions present in the unit, the Chatbot requests information using predefined options such as checkboxes, buttons, or range sliders to be more effective than simply asking the user to type their responses in an open text field as illustrated in Figure 13.

Figure 13. Predefined options shown by the Chatbot.

For temperature classification using fuzzy logic, the membership values for a set of temperatures are shown as depicted in Figure 14. Each line represents a temperature value followed by its membership values to the low, medium, and high categories, respectively. These values represent the degree of membership of each temperature to each category based on the defined fuzzy membership functions.

```
Temperature: 50 °C - Low: 0.75, Medium: 0.0, High: 0.0
Temperature: 60 °C - Low: 0.5, Medium: 0.0, High : 0.0
Temperature: 70 °C - Low: 0.25, Medium: 0.0, High: 0.0
Temperature: 80 °C - Low: 0.0, Medium: 0.3333333333333333, High: 0.0
Temperature: 90 °C - Low: 0.0, Medium: 1.0, High: 0.0
Temperature: 100 °C - Low: 0.0, Medium: 0.5, High: 0.0
Temperature: 110 °C - Low: 0.0, Medium: 0.0, High: 0.2
Temperature: 120 °C - Low: 0.0, Medium: 0.0 , High: 0.6
Temperature: 130 °C - Low: 0.0, Medium: 0.0, High: 1.0
```

Figure 14. Membership values.

4 Conclusions

In this paper, a comprehensive approach to engine failure diagnosis that integrates machine learning techniques, knowledge bases, and fuzzy logic, is proposed. The approach improves the accuracy and efficiency of diagnosing irregular behavior in automobile engines. Developing a database for OBD-II fault codes, a knowledge base for fault diagnosis, and implementing decision trees with pruning techniques; a high accuracy in this approach is achieved for fault identification. Additionally, applying fuzzy logic for temperature classification improves the system's ability to accurately assess engine temperature. The research also emphasizes the importance of structured input elements for collecting diagnostic-relevant information, thereby improving the overall effectiveness of fault identification and reparation processes in automotive workshops. Overall, the proposed approach offers a practical solution to increase engine fault diagnosis, contributing to the efficiency of automotive services. The future work focuses on the application of fuzzy logic evaluating other vehicle operating parameters. These improvements could help refine the proposed approach and make it even more effective in detecting faults in automotive engines.

References

Pérez-Darquea, D. G. (2018). Evolución de los dispositivos electrónicos en un automóvil.

Rouhiainen, L. (2018). Inteligencia artificial. *Madrid: Alienta Editorial*, 20-21.

Sempere, J. (2014). Aprendizaje de árboles de decisión. *Universidad Politécnica de Valencia, Valencia*.

Guarino, N., & Giaretta, P. (1995). Ontologies and knowledge bases. *Towards very large knowledge bases*, 1-2.

Trillas, E., & Eciolaza, L. (2015). Fuzzy logic. *Springer International Publishing. DOI*, *10*, 978-3.

Adamopoulou, E. y Moussiades, L. (2020). *Una descripción general de la tecnología de Chatbot. En conferencia internacional IFIP sobre aplicaciones e innovaciones de inteligencia artificial*. Springer, Cham.

De Sumana, S., & Chakraborty, B. (2018, April). Case based reasoning (CBR) methodology for car fault diagnosis system (cfds) using decision tree and jaccard similarity method. In *2018 3rd International Conference for Convergence in Technology*

(I2CT) (pp. 1-6). IEEE.

Murphey, Y. L., Chen, Z., Abou-Nasr, M., Baker, R., Feldkamp, T., & Kolmanovsky, I. (2009, June). Ensembles of neural networks with generalization capabilities for vehicle fault diagnostics. In *2009 International Joint Conference on Neural Networks* (pp. 2188-2194). IEEE.

Crossman, J. A., Guo, H., Murphey, Y. L., & Cardillo, J. (2003). Automotive signal fault diagnostics-part I: signal fault analysis, signal segmentation, feature extraction and quasi-optimal feature selection. *IEEE Transactions on Vehicular Technology*, *52*(4), 1063-1075.

Djurdjanovic, D., Liu, J., Marko, K. A., & Ni, J. (2007, August). Immune systems inspired approach to anomaly detection and fault diagnosis for engines. In *2007 International Joint Conference on Neural Networks* (pp. 1375-1382). IEEE.

Charkaoui, N., Dubuisson, B., Ambroise, C., & Millemann, S. (2005). A decision tree classifier for vehicle failure isolation. *WIT Transactions on Information and Communication Technologies*, *35*.

Covarrubias, R. F., & Covarrubias, A. G. F. (2013). Desarrollo de un sistema experto para el diagnóstico de fallas automotrices. *Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología*, (11), 83-91.

Xiaolei, L., & Xiaobing, W. (1999, September). The application of rough set theory in vehicle transmission system fault diagnosis. In *Proceedings of the IEEE International Vehicle Electronics Conference (IVEC'99)(Cat. No. 99EX257)* (pp. 240-242). IEEE.

Gutiérrez, J. M. (2008). Sistemas expertos basados en reglas. *Recurso personal. Dpto. de Matemática Aplicada. Universidad de Cantabria*, 1-12.

Tapia Barrientos, J. C. (2009). *Sistema experto para el diagnóstico automotriz* (Doctoral dissertation).

Soria Mamani, J. L. (2013). *Sistema experto para el diagnóstico de fallas en motores a inyección electrónica de vehículos* (Doctoral dissertation).

Ayala Ramos, A. R. (2000). *Agente para el diagnóstico de motores de automóviles. Universidad Mayor de San Andrés. Facultad De Ciencias Puras Y Naturales Carrera De Informática.*

Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine learning*, *40*, 139-157.

Song, Y. Y., & Ying, L. U. (2015). Decision tree methods: applications for classification and prediction. *Shanghai archives of psychiatry*, *27*(2), 130.

Patel, H. H., & Prajapati, P. (2018). Study and analysis of decision tree based onclassification algorithms. *International Journal of Computer Sciences and Engineering*, *6*(10), 74-78.

Opinautos. (s/f). Opinautos.com. Retrieved on June 27, 2023, from https://www.opinautos.com/mx

Carcomplaints.com. (s/f). Carcomplaints.com. Retrieved on June 27, 2023, from https://www.carcomplaints.com/

Scanator PC. (s/f). Com.mx. Retrieved on June 27, 2023, from https://scanator.com.mx/

McTavish, H., Zhong, C., Achermann, R., Karimalis, I., Chen, J., Rudin, C., & Seltzer, M. (2022, June). Fast sparse decision tree optimization via reference ensembles. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 36, No. 9, pp. 9604-9613).

Zhong, M., Georgiopoulos, M., & Anagnostopoulos, G. C. (2008). A k-norm pruning algorithm for decision tree classifiers based on error rate estimation. *Machine learning*, *71*, 55-88.

Tong, L., Liu, Z., Jiang, Z., Zhou, F., Chen, L., Lyu, J., ... & Zhou, H. (2022). Cost-sensitive Boosting Pruning Trees for depression detection on Twitter. *IEEE transactions on affective computing*.