International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-2, July 2019

A Novel CBR-Decision Tree Based Intelligent Car Fault Diagnosis System (CFDS)

Sumana De, Baisakhi Chakraborty

Abstract: ***Fault diagnostic systems find numerous applications in almost all service domains. Now a days, interest lies on intelligent fault diagnosis. For this, a huge database of cases called Case Base (CB) comprising of fault descriptions and their solutions needs to be maintained. Case Based Reasoning (CBR) is a popular Artificial Intelligence technique that supports huge databases that find popular applications in fault diagnosis systems. CBR is a very useful process for solving problems, detecting and diagnosing faults, learning, reasoning and supporting decisions. As the size of CB increases, the accuracy of the system too increases leading to an increase in computational time complexity. So CBR techniques are coupled with machine learning approaches to reduce the same. This paper proposes a CBR methodology based Intelligent Car Fault Diagnosis System (CFDS) that integrates decision tree as a machine learning technique and jaccard similarity method to diagnose faults of cars accurately in a minimum time. A car fault diagnosis and detection system requires individual expertise gathered from personal experience and technical skills. Many times, not only the fault, but also the car part from where the fault originates or the cause of the fault needs to be known to handle or repair the problem. So, to help car mechanics as an assistant tool, CFDS is proposed; so that they can deal with various types of car faults very easily. Here, the proposed methodology integrates decision trees and jaccard similarity method to diagnose faults where the usage of decision trees is to store cases and jaccard similarity method is used to calculate the similarity percentages between user new query case and stored cases in the CB. User can post a new query about his car problem to the user interface of the CFDS. The CFDS uses the proposed methodology to find the solutions of that problem, and finally ~~then~~ ~~at last~~ these solutions are displayed to the user. To obtain better performance of the CFD system, this paper introduces a novel model of CBR cycle called CR4 model that is slightly modified version of traditional CBR cycle of R4 model, proposed by Aamodt and Plaza in 1994.***

Index Terms**: *CBR, CR4 Model, CFDS, Case Base(CB),Car Description Decision tree(CDDT), Car Fault Description Decision tree(CFDDT),Case Clustering, Jaccard Similarity Method, User Query, Feedback.***

I. INTRODUCTION

The automobile industry is one of the world's biggest economic sectors that design, develop, manufacture and sell vehicles. The section of car production of the automobile industry is a major section throughout the years. The usage of cars has increased considerably. That is why car breakdowns often take place during regular use. On such an event, usually, the matter is referred to car repair experts who, from their knowledge and experience, determine the fault type and the cause of the fault. But sometimes the faults are so complicated that it becomes very difficult to identify the main cause of faults.

To help car mechanics to deal with such cases, a CBR methodology based intelligent Car Fault Diagnosis System (CFDS) has been proposed in this paper. CFDS is designed with Decision Tree and Jaccard Similarity Method to identify the actual reason of car fault and also to figure out the solution by computational intelligence application. This paper is an extension version of original work presented in 3rd IEEE International Conference for Convergence of Technology (I2CT) [1].

CBR is mainly dependent on cases, case base and reasoning technique. A case means an experience of a solved problem. A Case is an encapsulation of description of problem (e.g., features or symptoms) and a solution (e.g., a diagnosis result) .A memory that stores such cases is called Case Base (CB). A CB is a huge database that always stores previous cases. The term based means that the reasoning is based on cases. Reasoning is an approach to solve a given problem intended to draw conclusions using cases. Case Based Reasoning (CBR) methodology is a technique that solves a new problem using old problems, stored in the CB. That is why CBR methodology is also called an experience based problem-solving method. As many cases with similar problems contain similar solutions as referred to in [2], a new case is to be solved by searching similar cases in CB. If there exists a similar case with same problems in the CB, then the solutions of that similar case will be adapted as the solution of the new case. Whenever a new case finds a new solution for its problem, that new case is to be stored in the CB to enhance its efficiency as referred to in [2]. CBR methodology is a technique which helps the computer to develop an intelligent diagnosis system.

The CB of CFDS contains cases with old car problems and solutions. Here, each case contains proper description of problem that includes description of car and description of car fault, and a solution for each fault. Case base stores such cases with the help of decision trees. CBR methodology solves a new car problem using stored old car problems. In 1994, Aamodt and Plaza proposed a cyclic model of CBR called R4 model. As referred to in [2], in R4 model, there are 4 activities-

Revised Manuscript Received on 30 July 2019.
* Correspondence Author
Sumana De*, is with the Department of Computer Science & Engineering, National Institute of Technology, Durgapur, West Bengal, India
Baisakhi Chakraborty is with the Department of Computer Science & Engineering, National Institute of Technology, Durgapur, West Bengal, India

- Retrieve: Retrieve the most relevant cases from CB to solve a new case.
- Reuse: reuse the solution of retrieved case(s) to the new case.
- Revise: After reusing the old solution to the new case, test the applicability of the new solution in the real world.
- Retain: After the solution is successfully adapted to the new case, store that new case to the CB.

This paper introduces a novel model of CBR cycle called CR4 model that is slightly modified version of traditional CBR cycle of R4 model. There are 5 activities-

- Clustering: To make an efficient system, it is very important to arrange the cases well within the CB. So, in the CB of CFDS, cases are stored with the help of decision trees. Different cases with different car description will be stored into different clusters of car description decision tree (CDDT). And each case's different car fault description with their solutions will be stored into different clusters of car fault description decision tree (CFDDT). Detail illustrated in Section 4.
- Retrieve: CFDS retrieves the most relevant cases from CB to solve a new user query case. Details have been illustrated in Section 5.
- Reuse: CFDS reuses the solution of retrieved case(s) to the new case. Details have been provided in Section 5.
- Revise: After reusing the old solution to the new case, new solution is tested and revised in the real world by the Administrator or Knowledge Workers (KWs).
- Retain: After the solution is successfully adapted to the new case, Administrator or Knowledge Workers (KWs) update the CB with that new case. Fig. 1 show the proposed CBR cycle of CR4 Model.

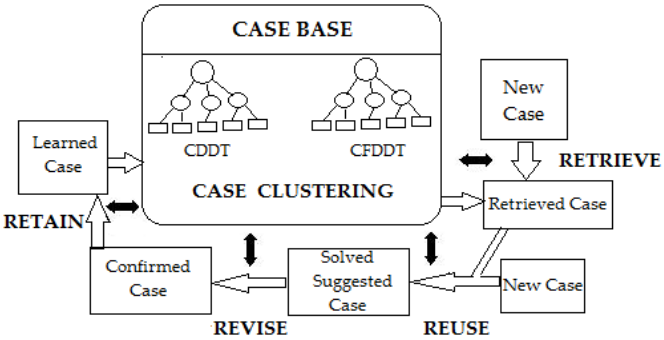

Figure 1: Proposed CBR cycle of CR4 Model

To make the system more efficient and fast, CR4 model is proposed that uses the technique of clustering to store cases in a particular way that helps the system to expedite the search process. R5 model of CBR cycle is proposed in [3], to show the advantages of repartitioning cases within the CB.

Car fault diagnosis is a critical task. To be an efficient and effective, CFDS should diagnose cases fast and accurately. In the proposed system, accuracy is dependent on the amount of knowledge in the CB. As car brands, models, features are updated regularly in the commercial market; administrator needs to update CFDS CB frequently. As the CB of CFDS is repartitioned and decision trees are used to store cases, so, whenever a new case arrives in the system, that new case shall be compared with only most appropriate case(s) stored in the CB. For this reason, system is able to diagnose faults in a minimum time.

The remaining part of the paper has following sections. Section 2 illustrates the previous related works; Section 3 expands the proposed model; Section 4 shows detail about case base of CFDS and the case representations within the CB; Section 5 gives a detail about the proposed methodology; Section 6 depicts the algorithm; Section 7 analyses the performance of CFDS; Section 8 shows comparative and descriptive analysis of previous works with the proposed CFDS model; and Section 9 focuses on the simulation results.

II. RELATED PREVIOUS WORKS

In Artificial Intelligence, an expert system is designed to detect and diagnose complex problems by reasoning. Nowadays, expert systems have a crucial role to detect and diagnose the automobile faults. In many previous works, CBR methodology is used to help the expert system in finding the faults in the automobile.

An expert system that is based on analysis of fault tree has been used to analyze the reasons of vehicle fault, solve the complexity and difficulty of fault detection has been discussed in [4]. With an example of vehicle brake lights fault diagnosis, the expert system was implemented.

As Expert System (ES) becomes one of the famous Artificial Intelligence (AI) techniques to solve difficult task. In the research paper [5], a knowledge-based system has been proposed that is capable of detecting and diagnosing car failure and car malfunction. This proposed system helps to train the mechanics to detect car problems.

DDAS means distributed diagnosis agent system that is based on signal analysis and machine learning, is proposed in [6]. DDAS has been used to diagnose vehicle problems. In this paper, two CBR techniques have been presented, CBR techniques are responsible to find the main cause of vehicle fault using the information provided by the signal agents in DDAS.

The researchers have developed a system for car failure diagnosis using fault tree analysis in [7]. This paper shows the usage of fuzzy set logic by the proposed diagnostic system, to calculate the indeterminacy and incertitude in data and information at the time of usage of fault tree analysis.

Ensemble with neural networks with generalization capability, the paper [8] proposed a procedure to diagnose vehicle faults. There are two-step ensemble approach is presented in this paper, one of the approaches is BFES that is an ensemble selection algorithm and another is A-Bayesian-Entropy that is analog Bayesian ensemble decision function.

As application of NC machine is widely spread so it is very important to diagnose and repair NC machine. CBR method is used to identify reference case and diagnose to find a reasonable solution of NC machine failure. CBR method for fault diagnosis of NC machine is one of the excellent research work is done in [9].

To diagnose car faults, an integrated reasoning method has been proposed in [10]. The integrated reasoning method includes both case based and rule based reasoning method, to assist decision making. Recalling from a previous similar situation or from a matching rule(s) the model identifies the process to solve a car fault.

Expert system is treated as the temporary assistance for the people who want instance help. Development of Knowledge-Based Systems is shown in [11] for Car Failure Detection using Expert System. This system stored 19 rules in the knowledge- base of the system that helps the mechanics to take correct decisions.

As the continuous development and use of high-speed trains brings different types of locomotive faults so the research paper [12] presents a research work on vehicle-equipment fault diagnosis for high-speed railway using Case-Based Reasoning. In this paper, to increase the accuracy of retrieval process, nearest neighbor matching strategy is used by Case Based Reasoning method.

Chemical plant demands highly efficient and consistent products so fault diagnosis is very important and also challenging work in chemical plant. So, to predict the status of Tennessee Eastman process (TE process), an improved case-based reasoning method has been used in [13].

The proposed case based fault diagnosis system in [14] uses neural network to diagnose the new faults. To manage the size of the cases and to control the operations like addition, deletion of cases in the case library, this research work uses another neural network. For an experiment, this diagnosis system is applied to motor rolling bearing.

Researchers find that there is a lack of semantic understanding in the existing CBR systems. As the semantic understanding is very important for retrieval of knowledge in decision support systems, so, in the paper [15], the researchers integrate ontology technology into CBR system to propose combine semantic retrieval method and numerical measurement in case retrieval.

For supporting the experts to manage the increasing complexity, this paper [16] proposes an assembled fault diagnosis and test case selection assistance system. This proposed system used CBR method to find actual cause of fault and to give a recommendation for further handling.

Mechanical faults can be detected through abnormal acoustic signals. So in this paper [17], to diagnose robot, case-based reasoning approach collects recorded sounds from normal robots and also from faulty robots, then stores them together with their diagnosis results in the case library. This system tests the industrial robots.

In paper [18], case base reasoning is used in Wireless Sensor Network (WSN) to build a fault diagnostic model application for Wildlife Preservation. In this paper, animal monitoring application running on the proposed fault tolerant WSN, is designed to check the health, location of the animals and the environment conditions of the sanctuary.

Kidney failure may occur gradually so Case-Based Reasoning is used for kidney failure diagnosis in [19]. For similarity calculation Simple matching coefficient is used. In this paper, waterfall methodology is used that needs to analyze, design, code and test.

As the diagnosis of psychiatric abnormality is a very difficult work for the physicians and researchers. So, in the research work [20] the case based reasoning model has been designed and developed to diagnose psychiatric disorder.

As in Jakarta, the congestion problem is increasing so in the research paper [21] Cased-Based Reasoning (CBR) approach is proposed to find the solutions for congestion problem in Jakarta. From CCTV cameras via website congestion data is obtained. CBR's integrated model uses that data to optimize traffic time and traffic flow so that congestion can decompose.

Nowadays, depression becomes one of the serious diseases. Normally, depression is diagnosed through questionnaire-based interviews taken by experienced doctors. To minimize the doctor's labor using Three-electrode EEG Data Case-based Reasoning Model has been proposed in [22] to identify depression.

The paper [23] proposes a case-based medical diagnostic system. The system is a web based application that supports queries in Thai language. Users can query to the system by inputting their symptoms in Thai. To store previous cases MySQL database has been used here.

To conclude and classify different failure, case-based reasoning (CBR) methodology is used to present an intelligent fault diagnosis system in the paper [24]. In this system, according to eighteen diesel engine fault features, CBR methodology diagnoses 4 faults of 4135 diesel engine. This paper presents a novel approach for case retrieve.

As the complexity of the launching vehicle increases and it has many subsystems, so in this paper [25], based on Java and MySQL database a case-based reasoning system is designed to apply case-based reasoning methodology to these several subsystems.

Diagnosis of human problems based on case-based reasoning (CBR) method and signal processing, is proposed in [26]. In this paper, case base maintenance (CBM) technique is used and wavelet packet transform (WPT) is presented for feature extraction of feature.

There are some research papers that use rough set theory to diagnose faults. For example, vehicular transmission fault diagnosis system has been discussed in [27], diesel engine valve clearance fault diagnosis in [28], power transformer fault diagnosis is in [29] and application of Jaccard similarity method for semantic analysis in [30].

III. THE PROPOSED MODEL

A. Architecture of CFDS

This paper proposes decision tree based Car Fault Diagnosis System (CFDS). This proposed system follows the request- response model where the user submits his query as a form of request and the CFDS provides the solution in form of responses. Fig. 2 displays the architectural elements of CFDS model. It has 4 principal modules which are the CBR, case extractor, computational and administrator modules. In this model the core module is that of the case base (CB).

There are two components in the user interface, such as, the Knowledge Workers (KWs) and the User.

- User: Whenever a user posts a problem related to his car in the user interface, then
  that problem is regarded as a new case. The new case is combined with car description and car fault description features.
- Knowledge Workers (KWs): Knowledge Workers (KWs) or Administrators have proper knowledge about several car-faults and repairing techniques. KWs should have personal login id and password to access the whole case base and also cases. KWs have the right to fill the CB with cases where each case is in the form of Car description, Car Fault description, and Solution. Each case represents some reputed car faults corresponding to respective car models along with the solution(s) to handle those problems.
- Administrator module assists the Knowledge Workers or Administrators for administering CFDS including the cases in the CB. Only KWs have the permission to add, delete, update cases and also modify the CB.
- Case extractor module is responsible to extract the features of the new user query case and provide it to the Computational Module.
- Computational Module calculates the similarity percentages between new user query case and stored cases in the CB, after calculation this module identifies the most similar case(s) stored in the CB, then, refers the identified case(s) with its fault part to the CBR module.
- CBR module receives car problems about referred identified case(s)'s fault part, retrieves the exact solutions for those problems and then displays those solutions to the user. In such way, user can get the solutions according to his queried car problems.
- Case Base is the database of the system. It stores previous cases. Each case contains proper description of problem that includes description of car and the car fault, and a solution for each fault.

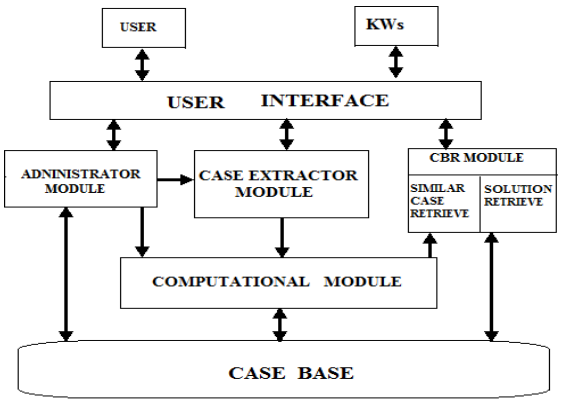

Figure 2: Architecture of the CFD System

B. Context Diagram of the CFDS

Fig. 3 depicts the context diagram corresponding to the architecture of CFDS, which simplifies the process flow of CFDS. A user enters into the CFD system and then the system provides him a query form related to car description. The user posts his car description query directly to the user interface of the system. System accepts the query form only if the user fills it up properly. After acceptance, the system considers the query as new query case (that comprises of features represented as attributes-values) and the system is ready to process that query case. At the beginning of processing, it is the responsibility of the Case Extractor Module to extract the features of the new user query case and provide it to the Computational Module. Then similarity percentages is calculated by the Computational Module, between new user query case's car description features and stored cases' car description features and the most similar case is identified from the CB. After this step, system provides another query form related to car fault description. This time, the user posts his car fault description query directly to the user interface of the system. System accepts the query form only if the user fills it up properly. After acceptance, car fault description features of that identified new case are again extracted by the Case Extractor Module and are provided to the Computational Module. Then Computational Module calculates the similarity percentages between new user query case's car fault description features and stored cases' car fault description features. Finally, after all calculations, system infers as to which car part and car fault the user has been referring to. Now, the CFDS provides all the problems related to that referred car fault part. The user will select problems one by one and system will provide solutions according to the problems.

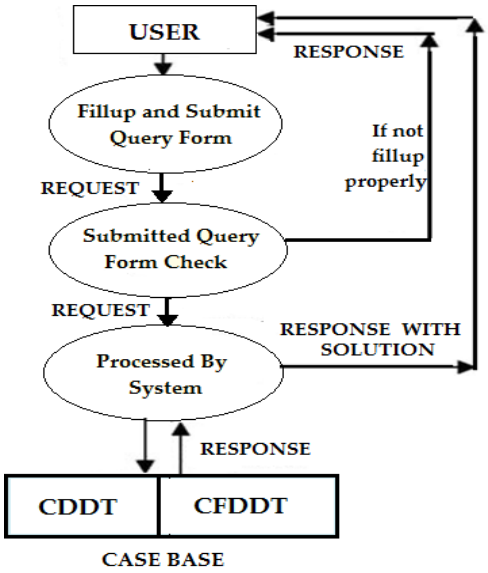

Figure 3: Context diagram of the CFDS

C. Use Case Diagram of the CFDS

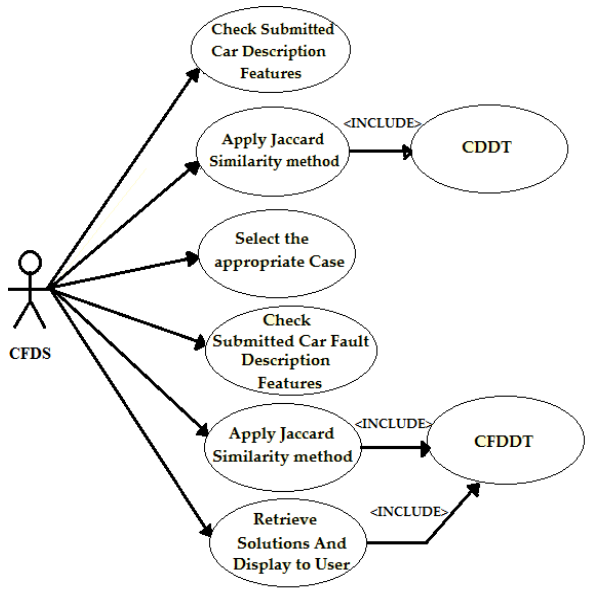

Figure 4: Use Case Diagram of the CFDS

The use case diagram defining several actions performed by CFDS is shown in Fig. 4. The actor of the use case diagram in Figure 4 is the CFDS and associated actions have been shown. The stepwise description is as follows.

- Check Submitted Car Description features – After the user posts a car description query form to the system, CFDS reads the submitted query form from the interface and starts checking each submitted Car Description features. CFDS accepts the submitted form if all fields of form are properly filled-up.
- Apply Jaccard similarity method – CFDS checks similarity of each submitted car description features with the stored car description features with the help of Car Description Decision Tree (CDDT). CFDS uses Jaccard similarity method for similarity calculation.
- Select Appropriate Case – After calculating the similarities, for further calculation, CFDS selects the most appropriate case that holds most similar car description features according to the submitted car description features.
- Check Submitted Car Fault Description features – A car fault description query form is posted by the user to the system; CFDS reads the submitted query form from the interface and starts checking each submitted Car Fault Description features. CFDS accepts the submitted form if all fields of form are properly filled-up.
- Apply Jaccard similarity method – CFDS checks similarity of each submitted car fault description features with the stored car fault description features of the selected Case with the help of Car Fault Description Decision Tree (CFDDT). CFDS uses Jaccard similarity method for similarity calculation.
- Retrieve Solutions and Display to the user – After all similarity calculations, using the Car Fault Description Decision Tree, CFDS displays various problems to the user according to the description of car and car fault submitted to the system by the user. When user chooses one problem and submits it to CFDS for solution, the CFDS retrieves solutions for the problem and displays it to the user.

IV. CASE BASE AND CASE REPRESENTATION

CBR system performs better if the cases in the CB are well organized. In CFDS, primarily CB does not contain any data. KWs have the responsibility to fill the CB with different cases. To fill up the CB, at first, CFDS provides two formats to the KWs. In the first format, KWs fill up with car description data and in the second one, KWs fill up with car fault description data with solutions. After that, with the help of Case extractor and Computation Module of CFDS, KWs are able to store several cases of mostly reputed car problems with their corresponding solutions to the CB. CB uses decision trees to store cases. CFDS provides the format of car description, shown in Table I and the format of car fault description with solution, shown in Table II. Table III and Table IV show a small example of data provided by the KWs.

Table I: Format of Car Description

Car Brand	Car Model	Manufacture Year	Case Number

Table II: Format of Car Fault Description

Case Number	Car Fault Part	Problem	Solution

Table III: Example of Car Description Data provided by KWs

Car Brand	Car Model	Year	Case Number
Toyota	Camry	1985	Case1
Toyota	Camry	1995	Case2
Toyota	Camry	2017	Case3
Toyota	Highlander	2002	Case4
Toyota	Highlander	2018	Case5
Toyota	Sienna	1999	Case6
Toyota	Sienna	2008	Case7
Toyota	Sienna	2017	Case8
Ford	Taurus	1990	Case9
Ford	Taurus	2018	Case10
Ford	Edge	2009	Case11
Ford	Edge	2018	Case12
Hyundai	Accent	1995	Case13
Hyundai	Accent	2018	Case14
Hyundai	Sonata	1990	Case15
Hyundai	Sonata	2018	Case16
Hyundai	Elantra	1995	Case17
Hyundai	Elantra	2018	Case18

Table IV: Example of Car Fault Description Data along with solution provided by KWs

Case Number	Car Fault Part	Problem	Solution
Case1	Engine	Engine: rough idle	Replace egr valve
		In cold, Engine stalls while idling	had to replace the cold star valve
	Light	Battery Light always on	If battery not dying,replace alternator
	Brake	When brakes are applied, there is vibration from front end	There is Alignment issue with the front brake pads and discs.
		While braking, Brakes make an odd sort of rattling sound	This is likely due to loose ani-rattle springs or loose caliper bolts.
	Win-dows	Driver window working	Problem in window regulator
	Clutch	Clutch slippage	Clutch linkage needs adjustment
	Suspen-sion	Excessive bounciness in the suspension	Failed shock absorber
		Worn suspension bushes	Bushes will need replacement

Table IV Shows Only One Example Of KW's Provided Car Fault Description Data With Solution Corresponding To Case1.

Likewise, KWs provide data for all 18 cases listed in Table III. Whenever these data are submitted to the system, each feature of cases is extracted by case extractor module. Then Computation Module then derives a Car Description Decision Tree (CDDT) where Years will be stored in an incremental way as per Fig. 5.It derives another decision tree termed Car Fault Description Decision Tree (CFDDT) as per Fig. 6. Both of the trees are stored into the CB.

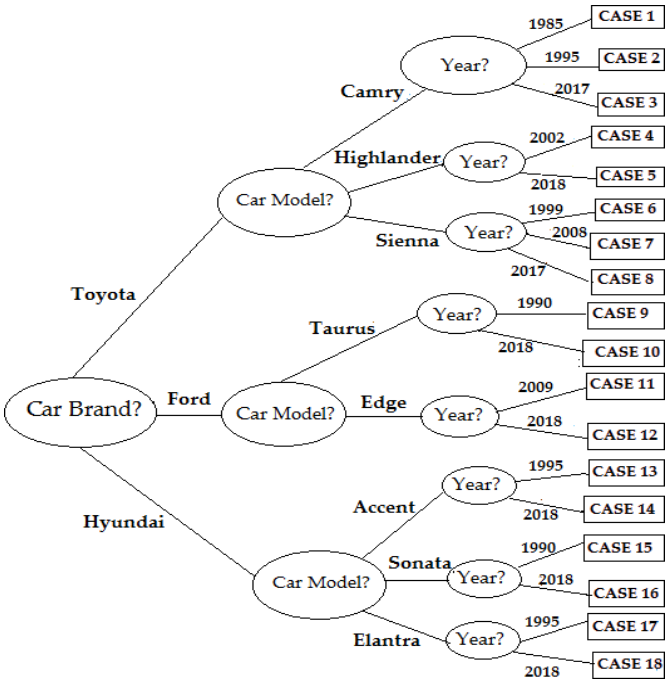

Figure 5: Car Description Decision tree (CDDT) in CB

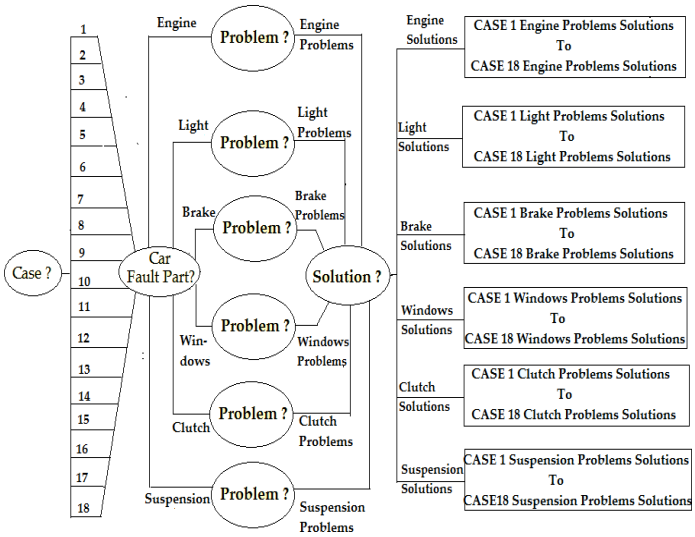

Figure 6: Car Fault Description Decision Tree (CFDDT) In CB

For experiment, some examples of different cases have been adopted from websites: www.toyotaproblems.com and Cases may be enhanced further.

V. THE PROPOSED METHODOLOGY

The proposed CFDS extracts solutions for the car problems using decision tree in conjunction with Jaccard similarity method.

A. Decision Tree

An example of supervised machine learning technique is a decision tree. It is a tree-shaped model that is a decision support tool that uses

branching method to illustrate every possible outcome of a decision. Decision tree represents possible solutions for a decision based problem. It consists of nodes and branches. Internal nodes of the decision tree represents events, the branches represent possible decisions and leaf nodes represent the solutions. Some advantages of decision tree model are:

- Decision tree model is very simple and easy to follow.
- The lucid nature of Decision Tree model makes the model unique from other decision-making models. It clearly shows all the possible choices for a decision and traces each choice to its conclusion. It becomes very easy to compare different choices in this tree model.
- Reaching to the absolute conclusion becomes easy as different nodes of the tree indicate different decisions and different branches indicate different choices.
- Decision tree model is very efficient model for logical diagnosis of a mechanical failure in equipment.

This paper makes use of two decision trees as indicated in Fig. 5 and 6. Fig. 5 shows the car description decision tree (CDDT) which derives case number as output according to different car descriptions and Fig. 6 shows car fault description decision tree (CFDDT) which infers possible solutions as output for the different car fault descriptions.

Jaccard Similarity Method

Jaccard similarity calculation is performed with two sets. For example, S1= {12, 35, 14} and S2= {12, 57, 14}

Then, Jaccard similarity = [(S1∩ S2) / (S1 ∪ S2)] = 2/4 =0.5 To get percentage =2/4 * 100 = 50%

The higher similarity between two sets means greater percentage of match between two sets as mentioned in the website: www.statisticshowto.com/jaccard-index/.

B. How decision tree along with Jaccard similarity are applied in CFDS.

There are two phases of calculations in CFDS.

Phase I: At the initial phase, a car description format is provided by the system to the user containing three car features which are Brand Name, Model Name and Manufacturing year of the corresponding car. The user must have to fill up and submit all of the three features of his faulty car. After that, system gets a new user query case's car description sets.

Let there be a new Query whose description is:

Brand = Hyundai,

Model = Elantra and Manufacturing Year= 2012

After getting Query Case's car description sets, CFDS follows the following steps:

Step 1: These car description sets are extracted by the Case Extractor Module as,

Brand = {Hyundai}, Model = {Elantra},

Year = {2} {0}{1}{2}

It then provides it to the Computational Module which calculates match between new query case and stored cases using Jaccard Similarity technique.

Step 2: Computational module extracts three different car brand names as Toyota, Ford and Hyundai from the CDDT (mentioned in Figure 5) of CB and calculates similarity between new query case's brand and stored case's brand names.

Step 2.1: 1st Jaccard similarity calculation: From the CDDT, stored cases' brand = {Toyota}

new case's car brand = {Hyundai}

Jaccard similarity percentage = 0/2 *100 = 0 %

Step 2.2: 2nd Jaccard similarity calculation: From the CDDT, stored cases' brand = {Ford}

new case's brand = {Hyundai}

Jaccard similarity percentage = 0/2 *100 = 0 %

Step 2.3: 3rd Jaccard similarity calculation: From the CDDT, stored cases' car brand name = {Hyundai}

new case's car brand name = {Hyundai} Jaccard Similarity percentage = 1/1 *100 = 100 %

After all computations, from the CDDT, Brand 'Hyundai' is selected for further calculation since Car Brand name 'Hyundai' gets highest similarity. In CDDT under car brand 'Hyundai' there are three different Car Models.

Step 3: Next, from the same CDDT, under 'Hyundai' Brand Computational Module extracts three other car model names as Accent, Sonata and Elantra; and calculates similarity between new query case's car model name and stored case's car model names;

Step 3.1: 1st Jaccard similarity calculation: From the CDDT, stored case's model = {Accent}

new case's model = {Elantra}

Jaccard Similarity percentage = 0/2 *100 = 0 %

Step 3.2: 2nd Jaccard similarity calculation: From the CDDT, stored cases' car model name = {Sonata}

new case's car model name = {Elantra} Jaccard Similarity percentage = 0/2 *100 = 0 %

Step 3.3: 3rd Jaccard similarity calculation: From the CDDT, stored cases' car model name = {Elantra}

new case's car model name = {Elantra}

Jaccard Similarity percentage = 1/1 *100 = 100 %

After all computations, from CDDT, stored Car Model name Elantra is selected as it exhibits greatest similarity.

Step 4: From the same CDDT, under 'Hyundai' Brand and Elantra model, Computational Module extracts two different stored Car Manufacture Years as 1995 and 2018; and calculates similarity between new query case's year and stored cases' years.

Step 4.1: 1st Jaccard similarity calculation: From the CDDT, stored cases' year = {1}{9}{9}{5}

new case's year = {2}{0}{1}{2}

Jaccard similarity percentage = 1/5 *100 = 20 %

Step 4.2: 2nd Jaccard similarity calculation: From the CDDT, stored cases' year = {2}{0}{1}{8}

new case's year = {2}{0}{1}{2}

Jaccard Similarity percentage= 3/4 *100 = 75 %

Step 5: Now, From Step 4 calculations, computational module finds 75 % is greater than 20 %. So, it selects Case18 from the Figure 5 CDDT, for further processing to derive the solution. For any case, if calculation results are same then only lastly calculated similar Year will be considered as the highest similar Year.Step 6: From the CFDDT,

computational model finds Case18 using Jaccard similarity method.

Phase II: At the second phase, a car fault description format is provided by the system to the user. This format includes one feature of a Car fault part. The user must have to fill up and submit that feature of his faulty car. After that, system gets a new user query case's car fault description sets.

New Query Case's car fault description:
Car fault part = Suspension
Steps followed by CFDS:
Step1: Case extractor extracts the query case's car fault description set as Car fault part= {Suspension}
and provides it to the Computational Module.
Step2: Computational module extracts six different car fault parts under Case18 as Engine, Light, Brake, Windows, Clutch and Suspension from the CFDDT (mentioned in Figure 6) of CB and calculates similarity between new query case's fault part and stored case18's fault parts.
Step2.1: 1st Jaccard similarity calculation: From the CFDDT, stored case18's fault part = {Engine} and
new case's fault part = {Suspension} Jaccard similarity percentage= 0/2 *100 = 0 %
Step2.2: 2nd Jaccard similarity calculation: From the CFDDT, stored case18's fault part = {Light} and
new case's fault part = {Suspension} Jaccard Similarity percentage = 0/2 *100 = 0 %
Step2.3: 3rd Jaccard similarity calculation: From the CFDDT, stored case18's fault part = {Brake} and
new case's fault part = {Suspension} Jaccard Similarity percentage = 0/2 *100 = 0 %
Step2.4: 4th Jaccard similarity calculation: From the CFDDT, stored case18's fault part = {Windows} and
new case's fault part = {Suspension} Jaccard Similarity percentage = 0/2 *100 = 0 %
Step2.5: 5th Jaccard similarity calculation: From the CFDDT, stored case18's fault part = {Clutch} and
new case's fault part = {Suspension} Jaccard Similarity percentage = 0/2 *100 = 0 %
Step2.6: 6th Jaccard similarity calculation: From the CFDDT, stored case18's fault part = {Suspension} and
new case's fault part = {Suspension}
Jaccard Similarity percentage = 1/1 *100 = 100 %
After calculation, from CFDDT, the fault part 'Suspension' gets highest match. Now, computational model transfers the control to the CBR module.
Step 3: CBR Module displays the user different common problems of 'Suspension' under Case18. Whatever problem user selects, this module helps the user to display the solutions of that particular 'Suspension' problem.
Computational module always follows one rule when it calculates similarity. From the CDDT as in Figure 5, computational module calculates similarity between user posted 'Car Brand Name' with the stored 'Car Brand Names'; whenever it gets '100%' similarity it stops the calculation. That means whenever new query 'Car Brand Name' matches 100% with one stored 'Car Brand Name', Computation model does not calculate similarity with that new 'Car Brand Name' with the rest of other stored 'Car Brand Names'. This rule is also applied for 'Car Model Name', 'Year', 'Case', and 'Car fault part'. Detail algorithm is illustrated in section 6.

VI. THE PROPOSED ALGORITHM

```
Step 1:   Start
Step 2:   Input:Car_Brand = {Toyota},
          Car_Model={Camry},
          Year= {2,0,1,2};
Step 3:   Declare CBrand[ N]; // Declare an array of string
          // N is the total number of Car Brands in CDDT
Step 4:   Initialization:
          CBrand[N]={Toyota,Ford,……
                     NthCarBrandName}
Step 5:   For i=1 to N do
          J = JaccardSimilarity{ (Car_Brand) and
          (CBrand[i])}*100
          If J=100
          CBrand_Index= i ;
          Break ;
          End If
          End For
Step 6:   Declare CModel[N]; // Declare an array of string
          // N is the total number of Car Models under the
          selected car brand in CDDT
Step 7:   Initialization:
          CBrand[N]={Camry, Highlander,……..
                     NthCarModelName}
Step 8:   For i=1 to N do
          J = JaccardSimilarity{ (Car_Model) and
          (CModel[i])}*100
          If J=100
          CBrand_Index= i ;
          Break ;
          End If
          End For
Step 9.:  Declare CYear1[ ] , CYear2[ ], CYearN[ ]
          // Declare arrays of integers
          // N is the total number of Year under the selected
          car Model in CDDT
Step 10:  Initialization:
          CYear1[ ] ={1,9,8,5}
          CYear2[ ] ={1,9,9,5}
          .
          .
          .
          CYearN[ ] ={2,0,1,9}
Step 11:  Declare Max=0; Index=0;
          For i=1 to N do
          J = JaccardSimilarity{ (Year) and (CYeari)}*100

          If J>=Max
          Max = J;
          Index = i;
          End If
          End For
```

Step 12: Declare Case_Number; // Case_Number is the number of most appropriate Case, selected under the highest similar Year.
Step 13: Declare CNumber[N];
// Declare an array of integers
Step 14: Initialization:
CNumber[N] ={1,2,3, N}
Step 15: For i=1 to N do
J = JaccardSimilarity{ (Case_Number) and (CNumber[i])}*100
If J=100
CNumber_Index= i ;
Break;
End If
End For
Step 16: Input: Car_FaultPart = {Engine};
Step 17: Declare CFault[N];
// Declare an array of string
// N is the total number of Car Fault Part in CFDDT
Step 18: For i=1 to N do
J = JaccardSimilarity{ (Car_FaultPart) and (CFault[i])}*100
If J=100
CFault_Index= i ;
Break ;
End If
End For
Step 19: Display //System shows different problems under Stored Toyota-Camry-2017-Case3-Engine
Step 20: Input // User selects problem
Step 21: Output // System replies with solution
Step 22: Stop

VII. CFDS PERFORMANCE ANALYSIS

The performance of CFDS depends on following two conditions

1. The amount of knowledge in the case base (CB) and
2. The Time Complexity of the proposed algorithm.

A. The amount of knowledge in case base

The more knowledge CFDS has the more accurate diagnosis is done. That means, CFD system diagnosis accuracy depends on the storage of data within the trees of CB. Accuracy will increase after each Revise and Retain process of CBR cycle. The necessary work for the administrator is to update the system case base time to time because with each passing year, a car gets new features. For example, at the earliest in 2007, embedded Bluetooth hands-free car kits were seen in market. Maruti Suzuki launched Auto Gear Shift with the Celerio in 2014. Following example shows the accuracy rates according to updating data in CB.

Scenario-I: Let the CFDS Case Base is updated till 2005 and Let system diagnoses for five different inputs, shown in Table V.

Table V: Five Random Inputs with Diagnosis

Car Brand	Car Model	Year	Fault Part	Problem	Accurate Diagnosis
Toyota	Sienna	2001	Engine	Engine Stall	Correct
Ford	Escape	2010	brake	Car shaking while braking	Correct
BMW	M2	2017	Auto gear	Gear wont shift in 3rd gear	Can't predict Correctly (show result for manual gear)
Audi	A8	2006	Air Filter	Air filter dirty	Correct
Audi	A4	2019	Bluetooth technology	Not working	Can't predict

Accuracy=(total correctly diagnosed solution / total cases to diagnose)*100
= 3/5*100
= 60%
Precision= (total correctly diagnosed solution / total correctly diagnosed solution + diagnose result but incorrectly)*100
= ¾*100
=75%
Recall= (total correctly diagnosed solution / total correctly diagnosed solution + can't predict diagnose result)*100
= ¾*100
=75%
Scenario-II: Let CFDS Case base is updated till 2010 and Let system diagnoses for five different inputs, shown in Table VI:

Table VI: Five Random Inputs with Diagnosis

Car Brand	Car Model	Year	Fault Part	Problem	Accurate Diagnosis
Toyota	Sienna	2001	Engine	Engine Stall	Correct
Ford	Escape	2010	brake	Car shaking while braking	Correct
BMW	M2	2017	Auto gear	Gear wont shift in 3rd gear	Can't predict Correctly (show result for manual gear)
Audi	A8	2006	Air Filter	Air filter dirty	Correct

Audi	A4	2019	Bluetooth technology	Not Working	Correct

Accuracy= (total correctly diagnosed solution / total cases to diagnose)*100 = 4/5*100 = 80%

Precision= (total correctly diagnosed solution / total correctly diagnosed solution + incorrectly diagnosed solution)*100= 4/5*100 = 80%

Recall= (total correctly diagnosed solution / total correctly diagnosed solution + the cases whose solutions cannot be predicted)*100 = 4/4*100 = 100%

That means with the same input, different Accuracy, Precision and Recall rate, shown in following Fig. 7 graph according to 'till the year of data update'.

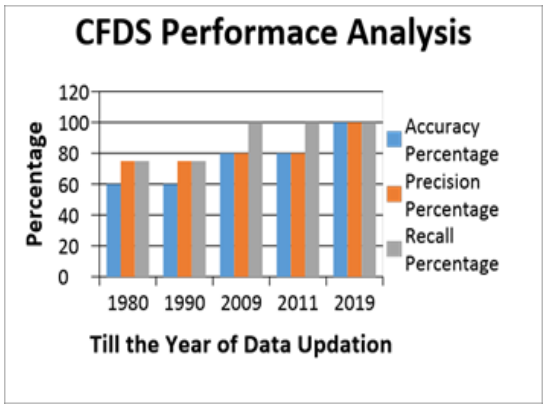

Figure 7: Accuracy, Precision and Recall Result

B. Time Complexity of the proposed algorithm

Following analysis calculates the time complexity by analyzing each step of the proposed algorithm in Section 6.
Step 1: Start the algorithm
Step2: It consists of a sequence of statements involving basic assignment operations. Total Time is: O(1).
Step3: For declaration of an array, total Time is: O(1).
Step4: For initialization statement total time is: O(N).
Step5: This step calculates Jaccard Similarity between user provided query case's Car Brand name with stored Car Brands names using for loop.
In worst case: the total time is O(N) In best case: the total time is O(1)
Step6: For declaration of an array, total Time is: O(1).
Step7: For initialization statement total time: O(N).
Step8: This step calculates Jaccard Similarity between user provided query case's Car Model name with stored Car Models names using for loop.
In worst case: the total time is O(N) In best case: the total time is O(1)
Step9: total time is constant: O(1).
Step10: For initialization statement total time: O(N).
Step11: This step calculates Jaccard Similarity between user provided query case's Year with stored Years using for loop.
In worst case: the total time is O(N) In best case: the total time is O(1)
Step12: For declaration of an integer, total Time is: O(1).
Step13: For declaration of an array, total Time is: O(1).
Step14: For initialization statement total time: O(N).
Step15: This step calculates Jaccard Similarity between query case's number with stored case numbers using for loop.
In worst case: the total time is O(N) In best case: the total time is O(1)
Step16: For initialization statement total time: O(1).
Step17: For declaration of an array, total Time is: O(1).
Step 18 :This step calculates Jaccard Similarity between user provided query case's fault part with stored case fault parts using for loop.
In worst case: the total time is O(N) In best case: the total time is O(1)
Step19: total time is: O(1).
Step 20: total time is: O(1).
Step 21: total time is: O(1).
Step 22: Stop the algorithm
So, The worst case, time complexity of the proposed algorithm in the CFD system to compute a solution is, O(N).
And The best case, time complexity of the proposed algorithm in the CFD system to compute a solution is also, O(N).

VIII. COMPARATIVE AND DESCRIPTIVE ANALYSIS OF PREVIOUS WORKS WITH THE PROPOSED CFDS MODEL

Case Base Reasoning Methodology has been used in many different fields. The survey, presented in Table VII is done to focus which type of CBR models and algorithms are used to implement CBR based expert system. From Table VII, it is now clear that most of the research work uses CBR R4 Model and nearest neighbor algorithm for the retrieval of cases.

A. Comparative Analysis between Aamodt and Plaza R4 based system and CR4 based CFDS

1) R4 Model includes Case retrieval, reuse, revise and retain. The Case Base of traditional CBR R4 model is not repartitioned. So, when a new user query case arrives in the system that new case shall be compared with all cases stored in the CB. Referring to Table III, if a new case comes that contains Car Brand Name as Toyota, then similarity calculation will be done between the new case and all cases in the CB even if some stored cases may contain totally different Car Brand Name in CBR R4 model of Aamodt and Plaza; whereas the CB of proposed CR4 model of CFDS in this paper is clustered. So, when a new case arrives in the system, that new case shall be compared with only the most appropriate case stored in the CB. In Table III, if a new case contains Car Brand Name as Toyota, then at first similarity calculation will be done between the new case's Car Brand Name and stored case's Car Brand Names for

CR4 model of CFDS system. For this reason, only some cases in CR4
model of CFDS are compared to fetch the solution which is less than that of CBR R4 model of Aamodt and Plaza.

2) To find solutions, traditional CBR R4 model does extra unnecessary calculations. To find solutions, proposed CR4 model uses decision trees to avoid unnecessary calculation overhead.

3) As in R4 model, the CB is not repartitioned so when cases increase in the CB, the complexity of case retrieval also increases. In CR4 model the complexity of CB is reduced so it reduces the complexity of case retrieval.

B. Comparative Analysis between R5 based system and CR4 based CFDS

R5 Model includes case representation, retrieval, reuse, revise and retain. As the analysis of paper [13], case presentation in R5 model faces some problems such as what information is to be stored in a case and how to organize the information. In CFDS, CR4 model cases are represented in a well-structured manner. In this paper decision tree is used to store different cases.

C. Comparative Analysis between K-Nearest Neighbour (kNN) algorithm and Jaccard Similarity Algorithm in CFDS

From Table VII it is seen that in the previous works, similarity calculations have been done by different similarity algorithms. But the most commonly used similarity algorithm is K-Nearest Neighbour (kNN) Classifier. The problem with KNN classifier is it does not work well with different text data type of attributes in the CB. All the values of attributes have to be converted into numerical values and then they have to be stored into the CB.

The proposed model uses Jaccard Similarity Algorithm that works well with text data type. So, data type conversion overhead is avoided in this model.

Table VII: Descriptive Analysis of Some Previous CBR-based Works

SL No.	Work	Year	Domain	Work Abstract	CBR Model	Algorithm
1.	in [6] vehicle faults are diagnosed using Case base reasoning	2003	Automobile	To detect faults of a device that is based on signal analysis and machine learning, a Distributed Diagnostic Agent System (DDAS) is developed.	R5 Model	signal analysis and machine learning
2.	in [17] Industrial Robot faults are diagnosed using Case base reasoning	2004	Industrial Robots	Case-Based Reasoning is used for robot diagnosis.	R4 Model	K-nearest neighbour
3.	in [9] NC machine faults are diagnosed and repair using Case base reasoning	2013	NC Machine	Fault diagnosis of NC machine is done using CBR technology.	R5 Model	fuzzy comprehensive evaluation method
4.	Case-Based Fault Diagnostic System in [14]	2014	Motor Rolling Bearing	Case based fault diagnosis system is applied for the motor rolling bearing. It uses neural network to diagnose the new faults.	R4 Model	Neural Network

5.	in [19] Kidney Failure is diagnosed using Case base reasoning	2016	Medical	Case-Based Reasoning is used for kidney failure diagnosis. For similarity calculation Simple matching coefficient is used.	R4 Model	simple matching coefficient
6.	in [20] Psychiatric Disorder is diagnosed using Case base reasoning	2016	Medical	Diagnosis of psychiatric disorder is done by designing and developing case based reasoning model.	R4 Model	K-nearest neighbor
7.	in [13] Tennessee Eastman process faults are diagnosed using Case base reasoning	2017	Tennessee Eastman	to predict the status of Tennessee Eastman process, an improved Case-based reasoning method is proposed	R4 Model	K-nearest neighbor
8.	In [21] congestion problem is solved using case based reasoning	2018	Traffic	Cased-Based Reasoning (CBR) approach is proposed to find alternative solution for congestion problem in Jakarta.	R4 Model	Binary Pattern comparation
9.	in [22] Depression is diagnosed using Case base reasoning	2018	Medical	case-based reasoning model is used to identify depression.	R4 Model	K-nearest neighbor

IX. SIMULATION FOR RESULT

Simulation has been done to establish the proposed methodology in .NET environment with Excel as backend tool. User interfaces of CFDS are presented in Fig. 8 to Fig. 14. User should fill up all the fields of forms, provided by system. After clicking the submit button, the system will feedback the appropriate solutions. System provides Fig. 8 and 9 forms to the administrator or KWs to fill up and fill the CB with cases;

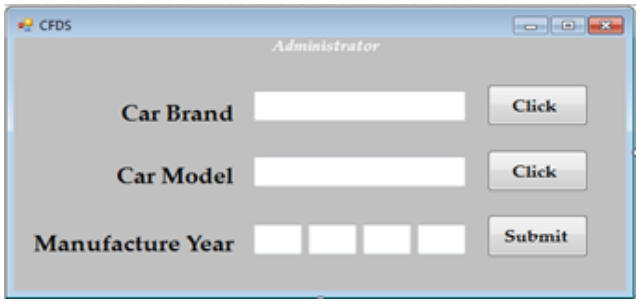

Figure 8: CFDS Provides Car Description Form To The Administrator (Kws)

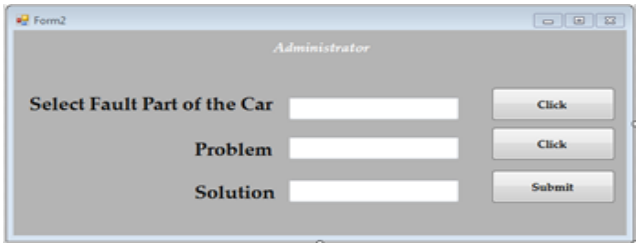

Figure 9: CFDS Provides Car Fault Description Form To The Administrator (Kws)

Fig. 10 And Fig. 11 Show The Diagnosis Result According To The User Query About His Car Problem.

Figure 10: CFDS Provides Feedback Result For The User Query About The Car Description

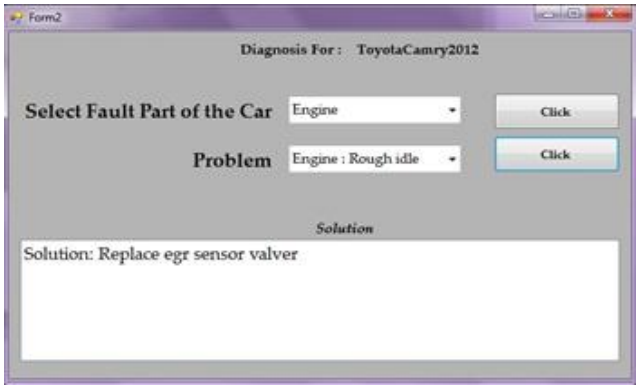

Figure 11: CFDS Provides Feedback Solution For The User Query About The Car Fault Description

Fig. 12 shows the diagnosis result of another car problem. Fig. 13 and Fig. 14 show the diagnosis result according to the user query about his car problem.

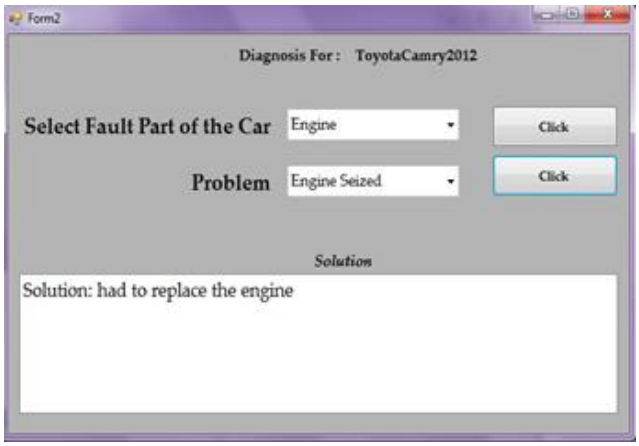

Figure 12: CFDS Provides Feedback Solution For The User Query About The Car Fault Description

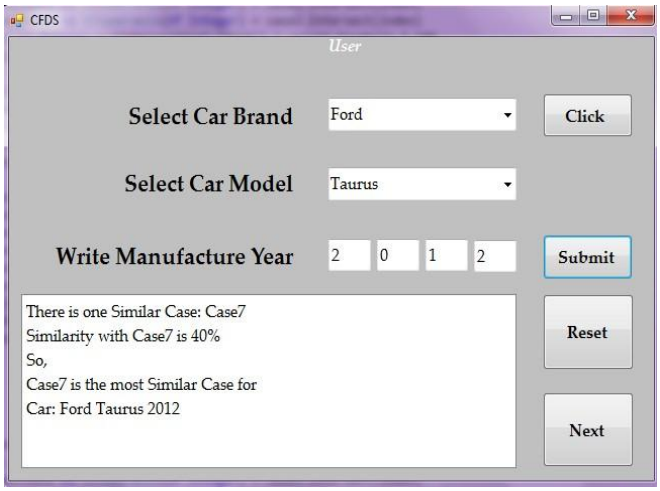

Figure 13: CFDS Provides Feedback Result For The User Query About The Car Description

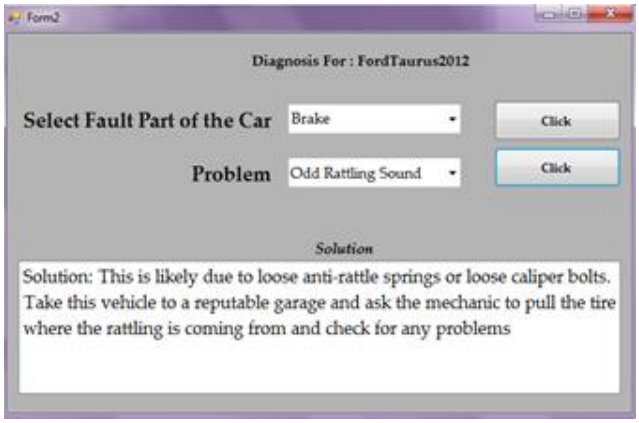

Figure 14: CFDS Provides Feedback Solution For The User Query About The Car Fault Description

CONCLUSION AND FUTURE

In today's world, as automobile industries produce huge number of cars, use of cars has increased in society and so car mechanics may need an application to help them solve complicated car problems. CBR methodology is one of the popular techniques in AI that may be used to effectively solve the problems related to cars.

To help car mechanics as an assistant tool, this paper proposes CBR-Decision Tree based intelligent Car Fault Diagnosis System (CFDS) that uses decision tree and Jaccard similarity methodology to diagnose car faults. The proposed methodology in CFDS integrates decision trees and jaccard similarity method to diagnose faults where the usage of decision trees are to store cases and jaccard similarity method is used to calculate the similarity percentages between user new query case and stored cases in the CB. User can post a new query about his car problem to the user interface of CFDS, CFDS using the proposed methodology finds the solutions of that problem, and then at last these solutions are displayed to the user. To obtain better performance of the CFD system, this paper introduces a novel model of CBR cycle called CR4 model that is slightly modified version of traditional CBR cycle of R4 model, proposed by Aamodt and Plaza in 1994. As the CR4 model uses decision trees for repartitioning or clustering the cases so whenever user posts a new query about his car problem to the system, CFDS returns appropriate solution in a minimum time using CR4 model. CFDS will give 100% accuracy, precision and recall rate if the CB of the system is updated till the current year. This CFDS is a fault diagnostic system that gives the user an idea to solve his car faults.

This paper has discussed about the architectural structure, implementation, Simulation result and background theory of CFDS, and the comparative and descriptive analysis of previous works with the proposed CFDS model. To get better efficiency of the CFDS, future work will be to enhance cases in CB and update the cases till the current year of operation.

ACKNOWLEDGMENT

The authors are thankful to the Department of Computer Science and Engineering in NIT, Durgapur (WB, INDIA), for the academic support for this research work.

REFERENCES

1. S. De, B. Chakraborty, "Case Based Reasoning(CBR) Methodology for Car Fault Diagnosis System (CFDS) Using Decision Tree and Jaccard Similarity Method" in 3rd IEEE International Conference for Convergence of Technology (I2CT), Pune, India, 2018, pp. 1–6.
2. R. Bergmann, K.-D. Altho, M. Minor, M. Reichle, K.. Bach, "Case-Base Reasoning-Introduction and Recent Developments" in Künstliche Intelligenz, Böttcher IT, Bremen, 2008.
3. A. Baghel, T. Sharma, "A Quick Tour to Understand and Improve CBR Performance" *International Journal of Application or Innovation in Engineering & Management (IJAIEM)*, vol. 3, 2014, pp. 335–341.
4. P. Chen, Y. Min, "Research and Implementation of Vehicle Fault Diagnosis Expert System" *Advances in information Sciences and Service Sciences (AISS)*, vol. 5, 2013, pp. 28–36.
5. S. A. Mostafa, Md. S. Ahmad, M. A. Mohammed, O. I. Obaid, "Implementing an Expert Diagnostic Assistance System for Car Failure and Malfunction" *IJCSI International Journal of Computer Science Issues*, vol. 9, 2012, pp. 1–7.
6. Z. Wen, J. Crossman, J. Cardillo, Y.L. Murphey, "Case Base Reasoning in Vehicle Fault Diagnostics" in Proceedings of the International Joint Conference on Neural Networks, Portland, 2003.
7. M. Sharma, R. Bhardwaj, R. Shrivastava, "Car failure fuzzy fault diagnostic system based on fault tree analysis" *International Journal of Theoretical & Applied Sciences*, vol. 2, 2010, pp. 18–21.
8. Y.L. Murphey, Z. Chen, M. Abou-Nasr, R. Baker, T. Feldkamp, I. Kolmanovsky, "ensembles of neural networks with generalization capabilities for vehicle fault diagnosis" in International Joint Conference on Neural Networks, Atlanta, GA, 2009.
9. J. Zhao, "Research on the NC machine fault diagnosis and repair system with Case-Based Reasoning" *Applied Mechanics and Materials*, vol. 401, 2013, pp. 1243-1246,.
10. D. M.L. Wong, "Integrated reasoning approach for car faulty diagnosis" *International Journal of Computer and Information Engineering*, vol. 5, 2011, pp. 232-237.
11. S. Adsavakulchai, "Developing Knowledge-Based Systems: Car Failure Detection using Expert System" *Information and Knowledge Management*, vol. 4, 2014, pp. 133-140.
12. L. Chen, D. Ou, H. Yao, "Research on Fault Diagnosis of Vehicle Equipment for High-speed Railway Based on Case-Based Reasoning" in Proceedings of the 3rd International Conference on Electrical and Information Technologies for Rail Transportation (EITRT), Springer.
13. H. Zhao, J. Liu, W. Dong, X. Sun, Y. Ji, "An improved case-based reasoning method and its application on fault diagnosis of Tennessee Eastman process" *Information and Knowledge Management*, vol. 249, 2017, pp. 266-276.
14. A.H. Mohamed, "Case-Based Fault Diagnostic System" *Arab Journal of Nuclear Science and Applications*, vol. 47, 2014, pp. 1-6.
15. A. Adla, A. B. Bella, "Intelligent Semantic Case Based Reasoning System for Fault Diagnosis" *Journal of Digital Information Management*, vol. 16, 2018, pp. 53-63.
16. S. Abele, M. Weyrich, "A Combined Fault Diagnosis and Test Case Selection Assistant for Automotive End-of-Line Test Systems," in 2016 IEEE 14th International Conference on Industrial Informatics (INDIN), Poitiers, France, 2016.
17. E. Olsson, P. Funk, M. Bengtsson, "Fault Diagnosis of Industrial Robots Using Acoustic Signals and Case-Based Reasoning" *Advances in Case-Based Reasoning. ECCBR 2004*, vol. 3155, 2004, pp. 686-701.
18. V. Latha, C. Subramaniam, S. Shanmugavel, "Fault Tolerant Wireless Sensor Network Using Case Based Reasoning with Semantic Tracking" in Proceedings of the 5th WSEAS international conference on Communications and information technology, 2011.
19. A. Anggrawan, K. Hidjah, Q. S. Jihadil, "Kidney failure diagnosis based on case-based reasoning (CBR) method and statistical analysis" in 2016 International Conference on Informatics and Computing (ICIC), Mataram, 2016.
20. P. Singh, A. P. Singh, S. Ahmad, "Case based reasoning model in the diagnosis of psychiatric disorder" in 2016 International Conference on Communication and Electronics Systems (ICCES), Coimbatore, 2016.
21. Surjandy, F. Anindra, H. Soeparno, T. A. Napitupulu, "CCTV traffic congestion analysis at Pejompongan using case based reasoning" in 2018 International Conference on Information and Communications Technology (ICOIACT), Yogyakarta, 2018.
22. H. Cai, X. Zhang, Y. Zhang, Z. Wang, B. Hu, "A Case-based Reasoning Model for Depression based on Three-electrode EEG Data" in IEEE Transactions on Affective Computing,2018.
23. J. Muangprathub, V. Boonjing, "Online Thai medical diagnostic system using case-based reasoning" in 2014 International Computer Science and Engineering Conference (ICSEC), KhonKaen, 2014.
24. C. Zhang, X. Wang, Y. Yu, L. Liu , Y. Zhou, "Fault diagnosis method for complex equipment using cased-based reasoning" in The 26th Chinese Control and Decision Conference (2014 CCDC), Changsha, 2014.
25. Z. Lan , H. Yuan, "Development of CBR system for embedded purpose" in 2017 Second International Conference on Reliability Systems Engineering (ICRSE), Beijing, 2017.
26. C. Vong , H. Huang , P. K. Wong, "Engine spark ignition diagnosis with wavelet packet transform and case-based reasoning" in The 2010 IEEE International Conference on Information and Automation, Harbin, 2010.
27. L. Xiaolei, W. Xiaobing,"The application of rough set theory in vehicle transmission system fault diagnosis" in Proceedings of the IEEE International Vehicle Electronics Conference (IVEC'99), Changchun, China, 1999.
28. C.S.Wang, "Research on extracting vehicle fault symptoms based on rough set theory in mechanical engineering" *Advanced Materials Research*, vol. 2013, pp. 644,110–114,.
29. Z. Ming, W. Taiyong, "fault diagnosis of power transformer based on association rules gained by rough set" in 2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE), Singapore, 2010.
30. A. J. Agrawal, Dr. O. G. Kakde, "Semantic Analysis of Natural Language Queries Using Domain Ontology for Information Access from Database" in I.J. Intelligent Systems and Applications, 2013.

AUTHORS PROFILE

Sumana De is a parttime research scholar in Computer Science and Engineering in the National Institute of Technology (NIT), Durgapur, West Bengal, India. She received the BTech and MTech degree in Computer Science and Engineering from West Bengal University of Technology (WBUT), Kolkata, West Bengal India, in 2009 and 2012 respectively. She completed her project of Library Management System in JSP in Btech and Topology Construction Algorithms in Wireless Sensor Network in Mtech. Her recent publication is "Case Based Reasoning(CBR) Methodology for Car Fault Diagnosis System (CFDS) Using Decision Tree and Jaccard Similarity Method" in 3rd IEEE International Conference for Convergence of Technology (I2CT), Pune, India. Her research area includes Knowledge Management System (KMS): System for Problem detection and diagnosis.

Baisakhi Chakraborty is an Associate Professor in the department of Computer Science and Engineering in National Institute of Technology, Durgapur. She received her PhD. degree in 2011 from the same institute in Computer Science and Engineering. She has a decade long experience in the industry and about two decades of experience in the academic field. Her research interests includes knowledge systems, knowledge engineering and management, database systems, data mining, case based reasoning, natural language processing and software engineering. She has several research scholars under her guidance. She has more than 50 international publications. Her present work in on Knowledge Management Systems whose applications are on e-government services which employ natural language processing techniques, machine learning techniques, case based reasoning and several related several techniques.