VEHICLE INSURANCE BASED ON DRIVING SCORE USING BLOCKCHAIN

A PROJECT REPORT

submitted by

AARIF HUSSAIN A NASSAR TOC19CS002
ASHLIN ROBERT TOC19CS022
EVA PETAL PRADEEP TOC19CS024
KIRAN R TOC19CS035

to the APJ Abdul Kalam Technological University
in partial fulfillment of the requirements for the award of the Degree of

Bachelor of Technology

in

Computer Science & Engineering

JUNE 2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Toc H INSTITUTE OF SCIENCE & TECHNOLOGY
Arakkunnam, Ernakulam, Kerala ☎ 0484 2748388 ✉ mail@tistcochin.edu.in tistcochin.edu.in

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
Toc H INSTITUTE OF SCIENCE AND TECHNOLOGY
Arakkunnam, Ernakulam, Kerala – 682 313

CERTIFICATE

This is to certify that the project report entitled **'VEHICLE INSURANCE BASED ON DRIVING SCORE USING BLOCKCHAIN'** submitted by **AARIF HUSSAIN A NASSAR (TOC19CS002), ASHLIN ROBERT (TOC19CS022), EVA PETAL PRADEEP (TOC19CS024) and KIRAN R (TOC19CS035)** during the eighth semester, is a bonafide record of the project work carried out by them in partial fulfillment of the requirements for the award of B.Tech degree in Computer Science & Engineering of the APJ Abdul Kalam Technological University, under our guidance and supervision, during the academic year 2022 - '23. This report in any form has not been submitted to any other University or Institute for any purpose.

Ms. Rinu Rose George
(Project Guide)
Assistant Professor, CSE
TIST, Arakkunnam

Ms. Elsaba Jacob
(Project Coordinator)
Assistant Professor, CSE
TIST, Arakkunnam

Dr. Sreela Sreedhar
Asso. Professor and Head,
Dept. of CSE
TIST, Arakkunnam

Prof. Dr. Preethi Thekkath
Principal

TIST, Arakkunnam

DECLARATION

We, hereby declare that the project report 'VEHICLE INSURANCE BASED ON DRIVING SCORE USING BLOCKCHAIN' submitted in partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Computer Science & Engineering, of the APJ Abdul Kalam Technological University, Kerala, is a bonafide work done by us under the supervision of Ms. Rinu Rose George, Assistant Professor, Dept. of CSE, Toc H Institute of Science and Technology. This submission represents our ideas in our own words and where ideas or words of others have been included, We have adequately and accurately cited and referenced the original sources. We also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. We understand that any violation of the above will be a cause for disciplinary action by the Institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

AARIF HUSSAIN A NASSAR

ASHLIN ROBERT

EVA PETAL PRADEEP

KIRAN R

Place: Arakkunnam

Date:

VISION OF THE INSTITUTION

To become a globally recognized institution that develops professionals with integrity who excel in their chosen domain making a positive impact in industry, research, business and society.

MISSION OF THE INSTITUTION

1. To provide the ambience necessary to achieve professional and technological excellence at the global level.
2. To undertake collaborative research that fosters new ideas for sustainable development.
3. To instill in our graduates ethical values and empathy for the needs of society.

VISION OF CSE DEPARTMENT

To acquire global excellence in the field of Computer Science and Engineering, nurturing in professionals, technical competence, innovative skills, professional ethics and social commitment.

MISSION OF CSE DEPARTMENT

1. To equip students with a strong foundation in the area of Computer Science and Engineering using effective teaching -learning practices.
2. To provide state-of-the-art infrastructure to suit academic, industry and research needs at the global level.
3. To engage students and faculty in interdisciplinary research that promotes innovative ideas for sustainable development.
4. To incorporate skill enhancement programmes for students and faculty to cope with the contemporary developments in technology.
5. To inculcate effective communication skills, professional ethics and social commitment among professionals through value added programs.

PROGRAM EDUCATIONAL OBJECTIVES (PEO)

Graduates of Computer Science & Engineering will

1. Evolve as globally competent computer professionals, researchers and entrepreneurs possessing collaborative and leadership skills, for developing innovative solutions in multidisciplinary domains.
2. Excel as socially committed computer engineers having mutual respect, effective

communication skills, high ethical values and empathy for the needs of society.

3. Involve in lifelong learning to foster the sustainable development in the emerging areas of technology.

PROGRAM OUTCOMES (POs)

Engineering Graduates will be able to:

1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis: Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

Student of the Computer Science and Engineering program will:

PSO1: Professional Skills: Attain the ability to design and develop hardware and software based systems, evaluate and recognize potential risks and provide creative solutions.

PSO2: Successful Career and Entrepreneurship: Gain knowledge in diverse areas of Computer Science and experience an environment conducive in cultivating skills for successful career, entrepreneurship and higher studies.

COURSE OUTCOMES (COs)

After successful completion of the course, the students will be able to:

CO1: Model and solve real world problems by applying knowledge across domains (Apply).

CO2: Develop products, processes or technologies for sustainable and socially relevant applications. (Apply)

CO3: Function effectively as an individual and as a leader in diverse teams and to comprehend and execute designated tasks. (Apply)

CO4: Plan and execute tasks utilizing available resources within timelines, following ethical and professional norms. (Apply)

CO5: Identify technology/research gaps and propose innovative/creative solutions. (Analyze)

CO6: Organize and communicate technical and scientific findings effectively in written and oral forms. (Apply)

ACKNOWLEDGMENT

We wish to record our indebtedness and thankfulness to all who helped our prepare this Project Report titled VEHICLE INSURANCE BASED ON DRIVING SCORE USING BLOCKCHAIN and present it in a satisfactory way.

We would like to express our gratitude to the Management of Toc H Institute of Science & Technology for their valuable help, cooperation and guidance.

Our heartfelt thank you to Prof. Dr. Preethi Thekkath, Principal, Toc H Institute of Science and Technology for her constant support and encouragement in the preparation of this report and making the required facilities available for its preparation.

We are especially thankful to our project guide, Ms. Rinu Rose George, Assistant Professor, Dept. of CSE for giving us valuable suggestions and critical inputs in the preparation of this report.

We are also thankful to Dr. Sreela Sreedhar, Asso. Professor and Head, Dept. of CSE for the constant encouragement to carry out this work.

We would like to thank our Project Co-ordinators, Ms. Elsaba Jacob, Mr. Sharath P Raju, and Mr Abin Oommen Philip for the valuable guidance right from the initial stages of the selection of topic and in the further works.

ABSTRACT

The current insurance is claimed by considering many factors like value of the car, age, analyzing the accident cases, considering the vehicle and many more , this process is very time consuming process, which also leads to hectic workload for both the customer and the insurance company. To overcome this we introduce the vehicle insurance policy ,where insurance is claimed by analyzing the driving skills using OBD port/ Reader. An OBD reader is a vehicle diagnostic device that can be used to read the error memory and data that is recovered on your vehicle. It also allows to establish connection between your car and mobile app or any other diagnostic software. Basically OBD takes all the data which is essential for the insurance such as the speed, error analysis, sensor defaults, whether the vehicle is properly serviced, using all these data we build an algorithm and it results in producing driving score as the output to claim the insurance. Our Project uses an OBD port which is a diagnostic device which helps to collects all the data essential for the insurance.On-board diagnostics (OBD) is an term referring to a vehicle's self-diagnostic and reporting capability. OBD systems give the vehicle owner or repair technician access to the status of the various vehicle subsystems. So that it helps to collect data efficiently and allows to make a driving score to allot insurance . The driving score is then pushed onto the blockchain network on a daily basis using Hyperledger Fabric which is an open source project for blockchain development and the blocks are added to the network.When pushed onto the blockchain tampering of data is not possible and we can get the accurate values and it also tells us the condition of the vehicle by analyzing all the data in blockchain.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	**Expansion**
OBD	On Board Diagnostics
PayU	Pay for What You Use
API	Application User Interface
CSV	Comma-separated values
GPS	Global Positioning System
ADC	Analgoue to Digital
DSC	Digital Signal Controller
EWM	Entropy Weights Method

Chapter 1

INTRODUCTION

In India insurance is claimed by considering many factors like value of the car, age, analyzing the accident cases, this process is very time consuming process, which also leads to hectic workload for both the customer and the insurance company.The primary use of Vehicle Insurance is to provide financial protection against physical damage or bodily injury resulting from traffic collisions and against liability that could also arise from incidents in a vehicle. Vehicle insurance may additionally offer financial protection against theft of the vehicle, and against damage to the vehicle sustained from events other than traffic collisions, such as keying, weather or natural disasters, and damage sustained by colliding with stationary objects. The specific terms of vehicle insurance vary with legal regulations in each region.We might have come across or heard about many situations in which the insurance claim is denied due to certain issues.There are many other problems like these where the customer has to go through a lot of legal procedures to get the claim also there are other cases in which the insurer tries to take advantage of the insurance company in which they try to manipulate others to get the insured amount.This system proposes a Insurance scheme based on the driving score by using an OBD reader which collects real time data from vehicles.

A driving score is a measure of a person's driving ability and safety on the road. It is often used by insurance companies to assess the risk of insuring a particular driver and to determine the cost of their car insurance premiums. It can also be used by employers to evaluate the driving skills of potential employees or by government agencies to monitor the performance of professional drivers There are a variety of factors that can affect a person's driving score, including their driving history, the type of vehicle they drive, and their age and gender. A person with a clean driving record, who drives a safe and reliable car, and who has a lot of experience behind the wheel is likely to have a higher driving score than someone who has a history of accidents or traffic violations, drives a high-risk vehicle, or has little driving experience One of the main ways that a driving score is calculated is through the use of telematics devices. These devices are installed in a person's car and track their driving habits, such as how fast they drive, how hard they brake, and how often they make turns. The data collected by these devices is then used to create a detailed profile of the driver's behavior on the road, which can be used to calculate their driving score .There are also a

number of other factors that can affect a person's driving score, such as the type of roads they typically drive on and the weather conditions they encounter. For example, a person who frequently drives on busy highways or in inclement weather may have a lower driving score than someone who only drives on quiet side streets or in good weather

Overall, a driving score [13]is an important measure of a person's ability to safely operate a vehicle. By understanding their driving score and working to improve it, individuals can not only lower their car insurance premiums but also become safer, more responsible drivers [14] on the road

1.1 BACKGROUND

The concept of a driving score has a long history dating back to the early days of the automobile. In the early 20th century, insurance companies began using a person's driving history as a way to assess their risk level and determine their car insurance premiums. This included factors such as the number of accidents and traffic violations a person had, as well as their age and gender.As technology advanced, so too did the methods used to evaluate a person's driving ability. In the 1960s, telematics devices began to be used to track a person's driving habits in real-time. These devices, which are now commonly found in modern cars, use sensors and GPS technology to track a variety of data points, including the speed at which a person is driving, how hard they brake, and how often they make turns. This data can then be used to create a detailed profile of the driver's behavior on the road, which can be used to calculate their driving score In recent years, the use of driving scores has become more widespread, with a growing number of insurance companies, employers, and government agencies using them to assess the risk of insuring or hiring a particular driver. In some cases, driving scores are even being used to incentivize safe driving behaviors [15], with drivers who have high scores receiving discounts on their car insurance or other perks.So,the use of driving scores has played a significant role in helping to improve the safety of roads and reduce the number of accidents and fatalities that occur each year. By providing a measurable way to evaluate a person's driving ability, driving scores have helped to promote responsible and safe driving practices and have made our roads safer for everyone.

1.1.1 Blockchain

Blockchain is a decentralized digital ledger that records transactions on multiple computers, making it secure and transparent..One of the main benefits of blockchain is its security. Transactions recorded on the blockchain are encrypted and secure, making it difficult for hackers to alter or tamper with the data. This makes it a reliable and trustworthy platform for storing and transferring sensitive information.Another advantage of blockchain is its transparency. All transactions on the blockchain [12] are recorded and available for all parties to see, making it difficult for any fraudulent activity to go undetected. This transparency also allows for greater accountability, as all parties can see the details of the transactions.In addition to its security and transparency, blockchain also has the potential to revolutionize a variety of industries.

Blockchain technology can potentially be used to track and verify the driving data that is used to calculate a driving score. For example, a blockchain system could be used to store and verify data about a driver's miles driven, traffic violations, and accidents. This data could then be used to calculate a driving score that is transparent, accurate, and tamper-proof.Additionally, a blockchain-based driving score system could allow drivers to share their driving data with multiple insurance companies, potentially enabling them to get more competitive insurance quotes. It could also allow insurance companies to offer more personalized and dynamic pricing, based on real-time data about a driver's behavior.

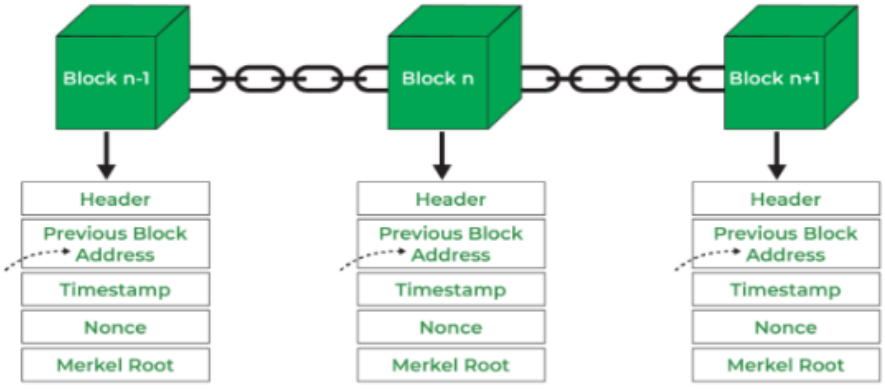

Figure 1.1: Architecture of blockchain.

1.1.2 Data Analytics

Driving scores are typically calculated using data analytics, which is the process of analyzing large sets of data to extract meaningful insights and inform decision making. In the case of driving scores, data analytics is used to analyze the data collected by telematics devices or other tracking systems to assess a person's driving habits and ability.There are a variety of data points that are typically analyzed when calculating a driving score, including the speed at which a person drives, how hard they brake, how often they make turns, and whether they follow traffic laws. This data is then used to create a detailed profile of the driver's behavior on the road, which is used to calculate their driving score.In addition to the data collected by telematics devices, other factors may also be considered when calculating a driving score, such as the type of roads a person typically drives on, the weather conditions they encounter, and the type of vehicle they drive. All of these factors can affect a person's driving ability and safety on the road, and are taken into account when calculating their driving score. The use of data analytics in the calculation of driving scores has helped to make the process more objective and accurate, allowing insurance companies, employers, and other organizations to make more informed decisions about a person's driving ability and risk level.

1.1.3 Telematics

Telematics [13] is a technology that uses sensors and GPS systems to track a person's driving habits in real-time. This data is then transmitted to a central server, where it is analyzed to create a detailed profile of the driver's behavior on the road. Telematics systems are often used to calculate a person's driving score, which is a measure of their driving ability and safety on the road.One of the main advantages of telematics systems is that they can provide a wealth of detailed data about a person's driving habits, including information about their speed, braking habits, and acceleration. This data can be used to identify any potential risk factors that may be contributing to a person's driving score and to suggest ways in which they can improve their driving ability In addition to their use in calculating driving scores, telematics systems can also be used to improve the safety and efficiency of a person's driving. For example, some systems can alert a driver if they are driving too fast or if they are braking too hard, which can help to reduce the risk of accidents and improve fuel efficiency.Overall, the use of telematics systems has played a significant role in improving

the safety of roads and reducing the number of accidents and fatalities that occur each year. By providing a detailed and objective way to evaluate a person's driving ability, telematics systems have helped to promote responsible and safe driving practices and have made our roads safer for everyone.

1.2 RELEVANCE

Vehicle insurance based on driving score [16] is a type of car insurance that takes into account a person's driving ability and safety on the road when determining their insurance premiums. This type of insurance is based on the idea that safer, more responsible drivers are less likely to get into accidents and make claims on their insurance, and therefore should be charged lower premiums The use of driving scores to determine car insurance premiums has several advantages. First and foremost, it provides a more objective way to assess a person's driving ability and risk level. Rather than relying on subjective criteria such as age and gender, which can be prone to bias, driving scores are based on data-driven assessments of a person's actual driving habits. In addition, vehicle insurance based on driving score can help to incentivize safe driving behaviors. By rewarding drivers with lower premiums for good driving habits, insurance companies can encourage people to be more mindful of their actions on the road and to take steps to improve their driving skills. This can lead to a reduction in the number of accidents and fatalities on the road, which benefits everyone

Overall, the use of driving scores in car insurance has the potential to improve the safety of roads and to make the process of purchasing car insurance more fair and transparent.

Chapter 2

LITERATURE REVIEW

2.1 IMPLEMENTATION OF OBD-II VEHICLE DIAGNOSIS

The implementation of OBD-II (Onboard Diagnostics II) [1] vehicle diagnosis systems integrated with cloud computation technology has the potential to revolutionize the way that vehicles are diagnosed and repaired. OBD-II systems are designed to collect data about a vehicle's performance, including information about its speed, fuel consumption, and emissions. This data can be used to identify any potential problems with the vehicle and to diagnose any issues that may be causing it to perform poorly.The integration of OBD-II systems with cloud computation technology allows this data to be collected and analyzed in real-time, allowing mechanics to identify and fix problems with a vehicle more quickly and efficiently. By storing the data in the cloud, it is also easier to access and analyze, making it easier for mechanics to diagnose problems and identify trends that may be affecting the performance of a vehicle In addition to their use in vehicle diagnosis, OBD-II systems integrated with cloud computation technology can also be used to track a person's driving habits and calculate their driving score. By analyzing data about a person's speed, acceleration, and braking habits, an OBD-II system can create a detailed profile of their behavior on the road, which can be used to assess their driving ability and identify any potential risk factors.

So, This paper implemented a cloud computation based second generation on-board diagnostic (OBD-II) [17] system. The proposed system is integrated with OBD-II, 3.5G wireless network, and cloud computing technologies. It can perform real-time vehicle status surveillance. The monitored features cover engine rpm, vehicle speed, coolant temperature, fault codes, and other vehicle dynamics information. The vehicle information will be transmitted to the cloud computing server via 3.5G wireless network for fault analysis. Once cloud computing server detects fault conditions, the proposed system could classify the fault conditions depended on vehicle type and its model year. Then the cloud computing server will report the fault code analysis results to the user and provide the description about repair procedure. The proposed system will greatly shorten the time to detect vehicle trouble condition. The system presented in this thesis has a very high value in the applications of vehicle maintenance and fleet management.

The several steps that may be involved in the implementation of OBD-II vehicle diagnosis systems integrated with cloud computation technology: Install OBD-II system: The first step in implementing an OBD-II system is to install the device in the vehicle. This typically involves connecting the device to the vehicle's onboard diagnostics port and following the manufacturer's instructions for setting it up Collect data: Once the OBD-II system is installed, it will begin collecting data about the vehicle's performance in real-time. This data may include information about the vehicle's speed, fuel consumption, and emissionsTransfer data to the cloud: The data collected by the OBD-II system is then transferred to a cloud-based server, where it can be stored and analyzed. Analyze data: Using cloud computation technology, the data collected by the OBD-II system is analyzed to identify any potential problems with the vehicle or issues that may be affecting its performance Identify and fix problems: Once any problems with the vehicle have been identified, mechanics can use the data collected by the OBD-II system to diagnose and fix the issues, improving the vehicle's performance and reliability Track driving habits and calculate driving score [18]: In addition to its use in vehicle diagnosis, the data collected by the OBD-II system can also be used to track a person's driving habits and calculate their driving score. This data can be used to assess a person's driving ability and identify any potential risk factors

The implementation of OBD-II vehicle diagnosis systems integrated with cloud computation technology has the potential to significantly improve the way that vehicles are diagnosed and repaired. By allowing mechanics to quickly and accurately diagnose problems with a vehicle, OBD-II systems can help to reduce the amount of time and money that is needed to repair vehicles and make it easier to identify and fix potential issues before they become more serious ,The integration of OBD-II systems with cloud computation technology allows for the real-time analysis of data, making it easier to identify trends and patterns that may be affecting the performance of a vehicle. This can help mechanics to more effectively diagnose and fix problems, improving the reliability and efficiency of vehicles. So the implementation of OBD-II vehicle diagnosis systems integrated with cloud computation technology is a significant advancement in the field of automotive repair and maintenance, and has the potential to greatly improve the safety and performance of vehicles on the road.

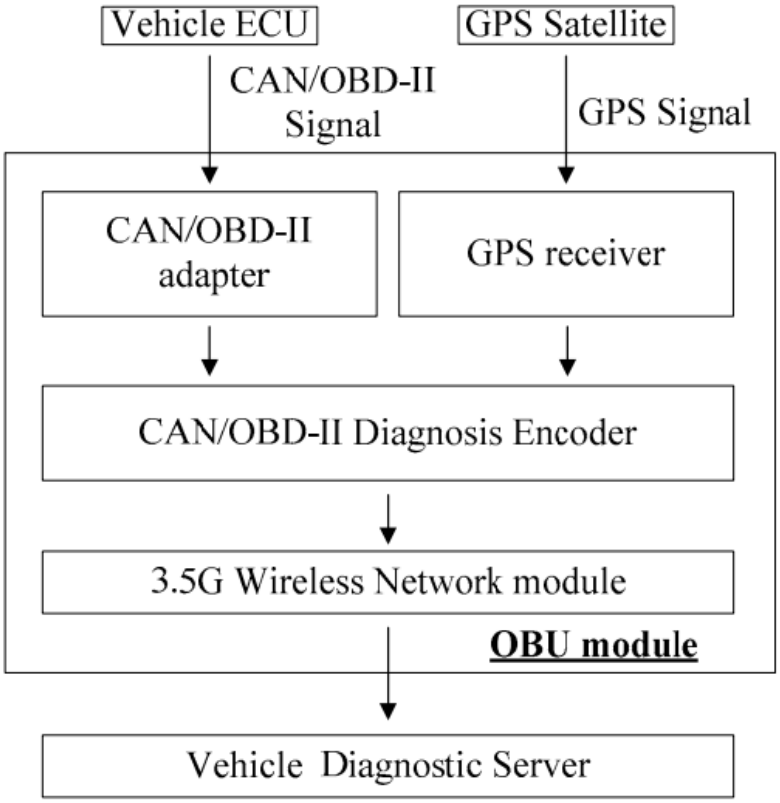

Figure 2.1: The block diagram of the proposed OBU module. .

2.1.1 Advantages

1.Improved accuracy and efficiency: By allowing mechanics to quickly and accurately diagnose problems with a vehicle, OBD-II systems can help to reduce the amount of time and money that is needed to repair vehicles and make it easier to identify and fix potential issues before they become more serious.

2.Real-time analysis: The integration of OBD-II systems with cloud computation technology allows for the real-time analysis of data, making it easier to identify trends and patterns that may be affecting the performance of a vehicle. This can help mechanics to more effectively diagnose and fix problems, improving the reliability and efficiency of vehicles.

3.Enhanced safety: By identifying and fixing potential problems with a vehicle, OBD-II systems can help to improve the safety of vehicles on the road. This can reduce the risk of accidents and help to make roads safer for everyone

4.Improved customer satisfaction: By providing faster and more accurate diagnoses, OBD-II systems can help to improve customer satisfaction with automotive repair and maintenance services

5.Increased fuel efficiency: By identifying and fixing issues that may be affecting the fuel efficiency of a vehicle, OBD-II systems can help to reduce fuel consumption and save drivers money

2.1.2 Disadvantages

1 Cost: The installation and maintenance of OBD-II systems can be costly, which may be a disadvantage for some drivers or mechanics.

2 Privacy concerns: Some people may be concerned about the privacy implications of having data about their driving habits collected and stored in the cloud.

3 Dependence on technology: OBD-II systems rely on technology, which means that they are subject to the same types of failures and glitches that can affect any other type of electronic device. This could potentially lead to issues with diagnosing and repairing vehicles

4 Limited compatibility: OBD-II systems are not compatible with all vehicles, which means that they may not be an option for everyone

2.2 OVERSPEEDING AND RASH DRIVING VEHICLE DETECTION

The present world is advancing at the speed of light [2] in the field of trade and business, and the development in technology has been significantly influencing this growth. However, transportation by road is one of the major factors that have been affecting the commercial development of our country. With the increasing vehicular population and their movement on the roads, accidents are also steadily increasing. It has become a nightmare for the authorities to prevent or reduce such fatal accidents on the road and all their efforts are in vain. According to the Indian road accidents survey, every year there are more than 135,000 incidents of road accidents. Out of these, most of them are due to rash driving. According to Indian Constitution, IPC section 279 , rash driving is an offense. So, The idea is to design a module which can detect the vehicle whenever it is rashly driven or driven above permissible speed limit, and transmit the data to the concerned authority. For example, when the cab driver is driving rashly or beyond the speed limit a message would be transmitted to the cab owner or the cab agency stating this, or the police could monitor vehicles to check whether they are driven correctly or not.

The overspeeding and rash driving vehicle detection system has the potential to significantly reduce the number of accidents and fatalities on the roads by helping to identify and prevent reckless driving behaviors. It is an important tool in the effort to make our roads safer for everyone.

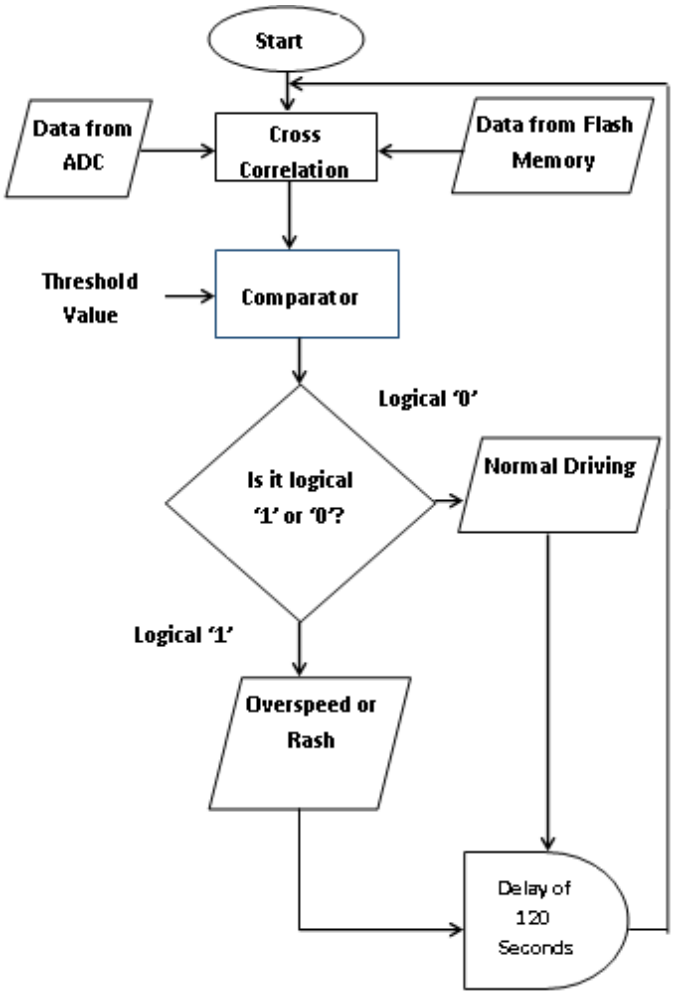

Figure 2.2: Flow Chart of Detecting over speed and rash driving .

The main hardware components are MMA7361L 3D accelerometer INA333 instrumentation amplifier, LM358 operational amplifier, DSC TMS320F28027 , microcontroller MSP430G2553 and LM1117 voltage regulator. The output of the accelerometer is analog data which is in the range of 0 volts to 3.3 volts. This output is then amplified and filtered and converted into digital data (12 bit) using the on-chip analog to-digital (ADC) converter of TMS320F28027. In the final prototype made use of the on-chip ADC of TMS320F28027 which has helped in bringing down the cost of our product, its weight, and size so that the entire module can be fit on a RC car for designing a prototype.The signals from the accelerometer is given to the instrumentation amplifier and then passed through fourth

order low pass filter to achieve pure signal which traces out the patterns of the vehicleThe maximum frequency of the signal from the accelerometer is 3Hz when the vehicle is being rashly driven. We had to remove the high frequency noise which is beyond 3Hz, so we designed a fourth order low pass filter using LM358 operational amplifiers. These signals are then given to the on-chip analog-to-digital converter of DSC for converting them into digital data. The correlation operation is performed in the DSC to detect whether the vehicle is over speeding, or driven rashly or normally. To design the signal conditioning circuit, INA333 instrumentation amplifier IC and LM358 for designing fourth order low pass filter has been used. The DSC used, is TMS320F28027 for real-time signal processing

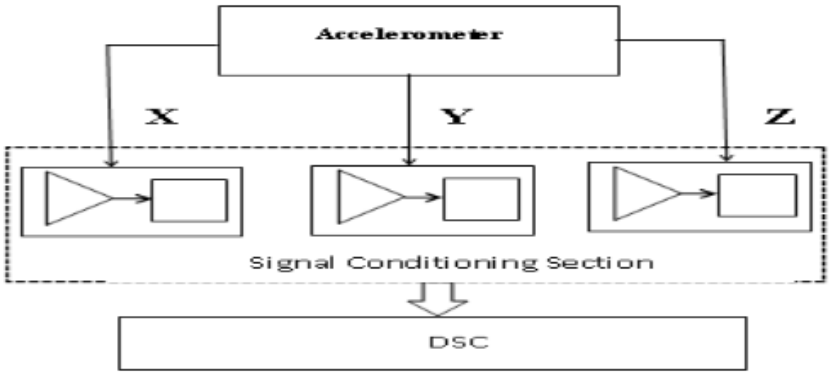

Figure 2.3: Block diagram of Detecting over speed and rash driving .

The main hardware components are MMA7361L 3D accelerometer INA333 instrumentation amplifier, LM358 operational amplifier, DSC TMS320F28027 , microcontroller MSP430G2553 and LM1117 voltage regulator. The output of the accelerometer is analog data which is in the range of 0 volts to 3.3 volts. This output is then amplified and filtered and converted into digital data (12 bit) using the on-chip analog to-digital (ADC) converter of TMS320F28027. In the final prototype made use of the on-chip ADC of TMS320F28027 which has helped in bringing down the cost of our product, its weight, and size so that the entire module can be fit on a RC car for designing a prototype.The signals from the accelerometer is given to the instrumentation amplifier and then passed through fourth order low pass filter to achieve pure signal which traces out the patterns of the vehicleThe maximum frequency of the signal from the accelerometer is 3Hz when the vehicle is being rashly driven. We had to remove the high frequency noise which is beyond 3Hz, so we designed a fourth order low pass filter using LM358 operational amplifiers. These signals are then given to the on-chip analog-to-digital converter of DSC for converting them into

digital data. The correlation operation is performed in the DSC to detect whether the vehicle is over speeding, or driven rashly or normally. To design the signal conditioning circuit, INA333 instrumentation amplifier IC and LM358 for designing fourth order low pass filter has been used. The DSC used, is TMS320F28027 for real-time signal processing

2.2.1 Advantages

1. Improved road safety: One of the main advantages of an overspeeding and rash driving vehicle detection system is its ability to improve road safety by helping to identify and prevent reckless driving behaviors. By alerting drivers when they are overspeeding or engaging in other dangerous behaviors, the system can help to reduce the number of accidents and fatalities on the roads.

2. Enhanced driver accountability: The overspeeding and rash driving vehicle detection system can help to increase accountability among drivers by providing a record of their driving habits and behaviors. This can help to incentivize safer driving practices and encourage drivers to be more mindful of their actions on the road.

3. Increased fuel efficiency: By helping to identify and prevent reckless driving behaviors such as overspeeding and aggressive acceleration, the overspeeding and rash driving vehicle detection system can help to improve a vehicle's fuel efficiency, which can save drivers money on fuel costs.

4. Enhanced vehicle performance: By identifying and fixing issues that may be affecting a vehicle's performance, such as overloading or improper tire pressure, the overspeeding and rash driving vehicle detection system can help to improve the overall performance of a vehicle.

2.2.2 Disadvantages

1 Cost: The installation and maintenance of an overspeeding and rash driving vehicle detection system can be costly, which may be a limitation for some drivers or vehicle owners.

2 Limited compatibility: The overspeeding and rash driving vehicle detection system may not be compatible with all vehicles, which means that it may not be an option for everyone.

3 Dependence on technology: The overspeeding and rash driving vehicle detection system relies on technology, which means that it is subject to the same types of failures

and glitches that can affect any other type of electronic device. This could potentially lead to issues with the system's accuracy or reliability.

4 Limited data: While the overspeeding and rash driving vehicle detection system can collect a wide range of data about a vehicle's speed and driving habits, there may be certain types of reckless driving behaviors that it is unable to detect. This could potentially lead to some instances of reckless driving going undetected.

2.3 SMART CONTRACTS BASED ON PRIVATE AND PUBLIC BLOCKCHAINS FOR THE PURPOSE OF INSURANCE SERVICES

Smart contracts are self-executing contracts [4] with the terms of the agreement between buyer and seller being directly written into lines of code. The code and the agreements contained therein are stored and replicated on a blockchain network.Smart contracts based on private and public blockchains have the potential to revolutionize the insurance industry by providing a more secure and efficient way to manage insurance services. Private blockchains, which are restricted to a specific group of users, can be used by insurance companies to securely store and manage data related to their clients and policies. Public blockchains, which are open to anyone, can be used to create decentralized insurance platforms that allow individuals to buy and sell insurance policies directly with one another. By using smart contracts based on private and public blockchains, insurance companies and individuals can take advantage of the security, transparency, and efficiency provided by blockchain technology. This can help to reduce the cost and complexity of insurance services, making them more accessible and beneficial for all parties involved So This paper compares the application of private and public Blockchain for insurance services through the experimental implementation of smart contracts based on Hyperledger Fabric and Ethereum. The results from the experiments are presented.

How smart contracts could be used for insurance services on a private blockchain:

(i) An insurance company and a client agree to the terms of an insurance contract, which are written into a smart contract.

(ii) The smart contract is deployed to a private blockchain network that is only accessible to the insurance company and the client

(iii) The client pays the premium for the insurance policy to the smart contract.

(iv) If the event insured against occurs (e.g., a car accident), the client can make a claim by triggering the smart contract.

(v) The smart contract automatically verifies the claim and, if it is valid, releases the insurance payout to the client.

(vi) The entire process is transparent, secure, and efficient, as all the data and interactions are recorded on the blockchain.

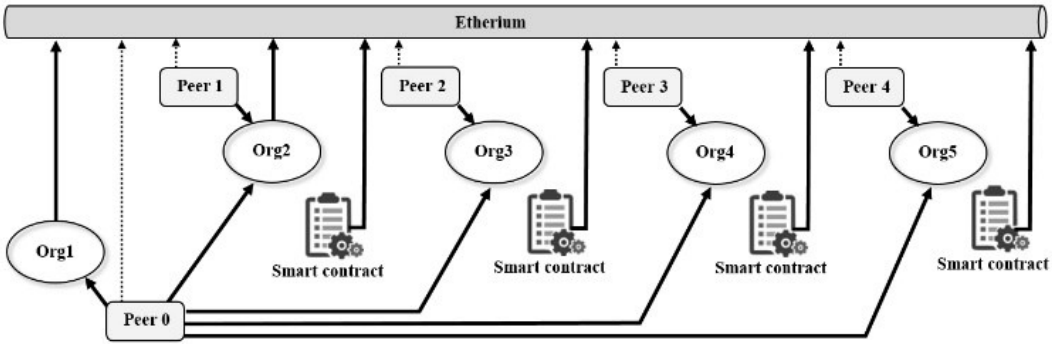

Figure 2.4: The Use case with one Smart contract on Ethereum

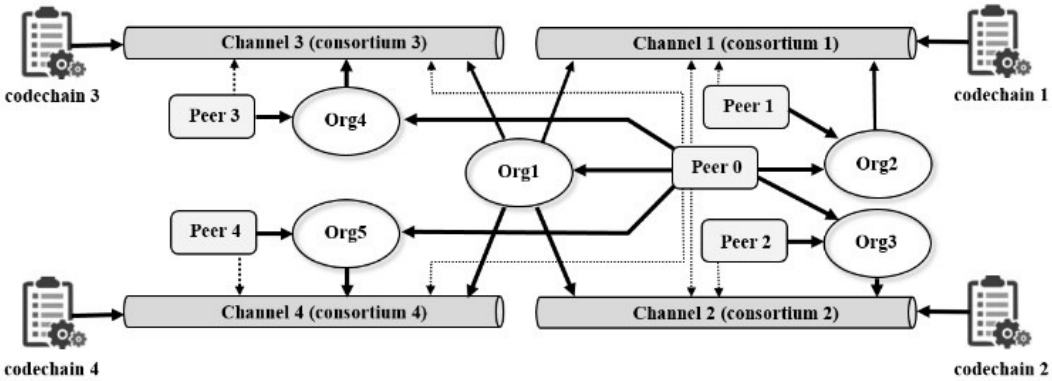

Figure 2.5: The use case with four smart contracts on Hyperledger Fabric.

2.3.1 Advantages

1. Automation: Smart contracts can automate many of the processes involved in insurance, including policy issuance, claims processing, and payouts. This can reduce the time and cost of these processes, as well as the risk of errors or fraud.

2. Transparency: Smart contracts are stored on a decentralized, distributed ledger, which makes them transparent and auditable. This can help to increase trust between the insurance company and its clients.

3. Security: Blockchain technology is inherently secure, as it uses cryptography to protect the data stored on the network. This can help to reduce the risk of data breaches and other security threats.

4. Efficiency: Smart contracts can facilitate faster and more efficient interactions between insurance companies and their clients, as they can automate many of the processes involved in insurance.

5. Cost savings: The automation and efficiency of smart contracts can help to reduce the cost of insurance for both companies and consumers.

6. Interoperability: Smart contracts can be designed to be interoperable with other systems and platforms, which can help to increase the reach and utility of insurance services.

7. Customization: Smart contracts can be customized to meet the specific needs of different insurance products and markets.

2.3.2 Disadvantages

1 Complexity: Smart contracts involve complex technical concepts and require a certain level of technical expertise to implement and use. This can be a barrier to adoption for some insurance companies and clients.

2 Scalability: Public blockchain networks, such as Ethereum, can struggle with scalability, as they can only process a limited number of transactions per second. This can make them unsuitable for some insurance applications that require high transaction volumes.

3 Regulation: The insurance industry is heavily regulated, and it can be challenging to ensure that smart contracts comply with all relevant regulations. This can be a barrier to adoption for some insurance companies.

4 Integration: Insurance companies may need to make significant investments in new systems and infrastructure to integrate smart contracts into their existing processes.

5 Limited adoption: The adoption of smart contracts for insurance services is still in the early stages, and it may take time for them to become widely adopted.

6 Data privacy: There are concerns about the privacy of data stored on public blockchain networks, as it is typically available to anyone on the network.

2.4 RESEARCH ON THE UBI CAR INSURANCE RATE DETERMINATION MODEL BASED ON THE CNN-HVSVM ALGORITHM

In recent years, there has been a growing interest in the use of machine learning algorithms [5] to improve the accuracy and efficiency of car insurance pricing. One promising approach is the use of a combined Convolutional Neural Network (CNN) and Histogram of Oriented Gradients Support Vector Machine (HVSVM) algorithm to determine car insurance rates. This research aims to explore the potential of this algorithm to accurately and fairly determine car insurance rates based on a variety of factors, including the driver's age, driving history, and the type of vehicle they drive. Through the use of this algorithm, it is hoped that insurance companies will be able to offer more personalized and dynamic pricing, while also reducing the risk of discrimination or bias in the pricing process. The findings of this research will be valuable for insurance companies and policy makers who are looking to adopt more advanced and effective methods for determining car insurance rates With the support of Internet of Vehicles technology, UBI (Usage Based Insurance) car insurance rate determination has certain guiding significance for achieving accurate pricing of car insurance rates and satisfying the personalized needs of users. Based on the CNN (Convolutional Neural Networks) algorithm and SVM (Support Vector Machine) algorithm, this paper establishes a rating model for UBI car insurance rates. The model first performs a series of operations such as convolutions, pooling and nonlinear activation function mapping using the CNN algorithm so that it can extract the features from the driving behavior data of UBI customers. Then, it introduces the Hull Vector to optimize the operating efficiency of the SVM algorithm. The HVSVM (Hull Vector Support Vector Machine) algorithm classifies customers according to their driving behavior, and thus obtains UBI customer car insurance rate grades. Therefore, this paper proposes a UBI car insurance rate grade determination model based on the CNNHVSVM algorithm. The empirical results of the model show that the CNN-HVSVM algorithm has higher discrimination accuracy in the risk rating process of UBI customer driving behavior than the CNN algorithm, BP neural network algorithm and SVM algorithm; and when dealing with large training sets, it has a speed advantage over the CNN-SVM algorithm. Furthermore, it is easy to realize in the process of establishing the UBI car insurance rate determination model and it has good robustness, which can adapt to diverse data sets, thus achieving better results in the car insurance rate determination process.

Therefore, the CNN-HVSVM model can predict the grade of UBI car insurance users more accurately and efficiently, and the prediction results are more consistent with the actual situation, which has strong applicability and flexibility. The UBI car insurance premium rate determination model based on the CNN-HVSVM algorithm can determine driver behavior more fairly and reasonably, and has certain practical significance for promoting car insurance rate market reform, which can better promote future UBI research work..

2.4.1 Advantages

1 Accuracy: Machine learning algorithms, such as the CNN-HVSVM algorithm, can analyze a large amount of data and identify patterns and correlations that may not be easily visible to humans. This can help to increase the accuracy of car insurance rates, as they are based on a more comprehensive and objective analysis of the data.

2 Personalization: The CNN-HVSVM algorithm can analyze a wide range of factors, including the driver's age, driving history, and the type of vehicle they drive, to determine personalized car insurance rates. This can help to ensure that drivers are charged fairly based on their individual risk profile.

3 Efficiency: The use of the CNN-HVSVM algorithm can automate and streamline the process of determining car insurance rates, potentially reducing the time and cost involved.

4 Fairness: By using an objective and data-driven approach to determine car insurance rates, the CNN-HVSVM algorithm can help to reduce the risk of discrimination or bias in the pricing process.

5 Dynamic pricing: The CNN-HVSVM algorithm can analyze real-time data about a driver's behavior and adjust their insurance rates accordingly. This can enable insurance companies to offer more dynamic and responsive pricing, based on a driver's current risk profile.

2.4.2 Disadvantages

1 Complexity: Machine learning algorithms, such as the CNN-HVSVM algorithm, can be complex and require a certain level of technical expertise to implement and use. This can be a barrier to adoption for some insurance companies

2 Data quality: The accuracy of the CNN-HVSVM algorithm depends on the quality and completeness of the data it is trained on. If the data is incomplete or biased, it could

result in inaccurate or unfair insurance rates.

3 Data privacy: The use of machine learning algorithms for insurance pricing may raise concerns about the privacy of personal data, as it is being collected and analyzed by the insurance company.

4 Regulation: The insurance industry is heavily regulated, and it is important to ensure that the use of the CNN-HVSVM algorithm for determining car insurance rates complies with all relevant regulations.

5 Limited adoption: The use of machine learning algorithms for determining car insurance rates is still in the early stages, and it may take time for them to become widely adopted.

6 Lack of transparency: The decision-making process of machine learning algorithms can be opaque and difficult to understand, which may raise concerns about accountability and transparency.

2.5 CONSTRUCTION OF DRIVING BEHAVIOR SCORING MODEL BASED ON OBD TERMINAL DATA ANALYSIS

A driving behavior scoring model based on On-Board Diagnostics (OBD) terminal data analysis [6] is proposed by Tianshi Liu, Guang Yang, and Dong Shi, with the objective of improving road safety and obtaining valuable insights into driver behavior. OBD terminals are devices that are installed in vehicles and collect data about the vehicle's performance and driving habits. By analyzing this data, it is possible to develop a scoring model that can evaluate a driver's behavior and identify any risky or unsafe driving habits.This research aims to explore the potential of OBD terminal data analysis to construct a reliable and accurate driving behavior scoring model. The findings of this research will be valuable for insurance companies, fleet managers, and other stakeholders who are interested in using data-driven approaches to improve road safety and reduce the risk of accidents. The driving behavior scoring model developed in this research could be used to provide feedback to drivers, identify training needs, and inform the development of more effective policies and interventions to promote safe driving. In order to accurately identify abnormal driving behaviors, this paper proposes different abnormal driving behavior recognition algorithms. The algorithm obtains driver's driving data through the OBD terminal, combines the x-axis

and y-axis acceleration changes and behavior duration of the vehicle's three-axis acceleration sensor to identify abnormal driving behaviors, establishes a hierarchical driving behavior indicator system and judgment matrix. On this basis, a driving behavior scoring model is established. The model combines the driving data of the driver, takes the proportion of abnormal driving behavior as the evaluation index, and uses the entropy weight method and the analytic hierarchy process to obtain the index weight. The test results of an example show that the model can analyze and evaluate the driving behavior of the driver well, and give a score of driver's behavior, to encourage the driver to develop good driving habits and effectively prevent the occurrence of traffic accidents.

For constructing a driving behavior scoring model based on OBD terminal data analysis may include the following steps:

(i) Data collection: OBD terminal data is collected from a large number of vehicles over a certain period of time. The data may include information about the vehicle's speed, acceleration, braking, turning, and other driving-related parameters.

(ii) Data preprocessing: The collected data is cleaned and preprocessed to remove any errors or inconsistencies.

(iii) Feature extraction: Relevant features are extracted from the preprocessed data that are relevant to driving behavior. These features may include metrics such as average speed, number of hard braking events, number of sharp turns, etc.

(iv) Model training: A machine learning model, such as a decision tree or a random forest, is trained on the extracted features to learn the relationship between different driving behaviors and their associated risk.

(v) Model evaluation: The trained model is evaluated using various metrics, such as accuracy, precision, and recall, to ensure that it is reliable and accurate.

(vi) Model deployment: The trained model is deployed to score the driving behavior of individual drivers based on their OBD terminal data.

(vii) Feedback and improvement: The driving behavior scores are used to provide feedback to drivers and identify areas for improvement. The model may also be updated periodically with new data to improve its accuracy and relevance.

2.5.1 Advantages

1 Improved road safety: The driving behavior scoring model can help to identify risky or unsafe driving habits, and provide feedback to drivers to help them improve their behavior. This can potentially reduce the risk of accidents and improve road safety.

2 Personalization: The driving behavior scoring model can provide personalized feedback to drivers based on their individual driving habits and risk profile. This can help to tailor interventions and training to the specific needs of each driver.

3 Efficiency: The use of OBD terminal data and machine learning algorithms can automate and streamline the process of evaluating driving behavior, potentially reducing the time and cost involved.

4 Data-driven approach: The driving behavior scoring model is based on data collected directly from vehicles, which can provide a more objective and accurate assessment of driving behavior compared to other methods.

5 Improved insurance pricing: Insurance companies can use the driving behavior scoring model to more accurately assess the risk profile of drivers and determine more fair and personalized insurance rates.

6 Fleet management: Fleet managers can use the driving behavior scoring model to monitor and improve the driving habits of their employees, potentially reducing the risk of accidents and improving the efficiency of their operations

2.5.2 Disadvantages

1 Regulation: The use of OBD terminal data for insurance purposes may be subject to various regulations, and it is important to ensure that the driving behavior scoring model complies with all relevant regulations.

2 Limited adoption: The use of OBD terminal data for evaluating driving behavior is still in the early stages, and it may take time for it to become widely adopted.

3 Ethical considerations: The use of a driving behavior scoring model may raise ethical concerns, such as the potential for discrimination or bias in the scoring process. It is important to carefully consider these issues and ensure that the model is fair and unbiased.

Chapter 3

PROBLEM IDENTIFICATION AND OBJECTIVES

3.1 SYSTEM STUDY

The disadvantages of getting auto insurance are numerous. One is that, depending on the type of coverage you select, it may be pricey. Another is that, should the need arise, filing a claim can be difficult.Customers should receive fair and effective service from the insurance provider. This implies that they should treat customers with respect and promptly handle claims. Additionally, the business should charge clients fairly and avoid overcharging them.The department will be able to keep track of all the insurance policies it has in place as well as any modifications or updates with the use of the insurance management system. Additionally, the system will assist the government in keeping track of any payments or claims made in connection with such policies.Drivers for whom the insurer can compute the claim amount based on their driving style and the vehicle's condition by calculating their driving score, which is done by pushing these data into the blockchain to determine the vehicle's resale value.

3.2 PROBLEM DEFINITION

Numerous instances where an insurance claim was rejected because of a particular problem may have been encountered or heard about. There are numerous more issues similar to these where the client must go through numerous legal steps in order to acquire the claim, and there are other instances when the insurer tries to take advantage of the insurance business by manipulating others in order to obtain the insured amount. To prevent such frauds, insurance claims are made dynamic by analyzing the driver's driving style and numerous other aspects of the vehicle using the OBD. The data gathered using the OBD reader is then used to create a program that allows the claim amount to be received by the insurer based on both the driving score and the condition of the vehicle.

3.3 OBJECTIVES OF THE PROJECT

This project aims to achieve the following objectives:

- Collect vehicle diagnostic information through the OBD port.
- Establish a connection between the OBD port and a computer or smartphone for data retrieval.
- Gather data on various performance and condition parameters of the vehicle, such as engine load, engine RPM, and acceleration on three axes.
- Process and refine the collected OBD data to ensure compatibility with the database format.
- Update the database with the newly acquired OBD data.
- Utilize machine learning techniques to calculate a driving score.
- Analyze all OBD data values comprehensively before determining the driving score.
- Store the driving score and vehicle details securely on the blockchain on a daily basis.
- Prevent data tampering and enable analysis of the stored data for determining car resale value.

  By achieving these objectives, this project aims to enhance the understanding of driving behavior, ensure data integrity, and enable informed decision-making in the insurance and automotive industry.

3.4 SCOPE AND APPLICATIONS

Vehicle insurance based on driving score using blockchain has a wide scope of applications that can significantly transform the insurance industry. By leveraging blockchain technology, this innovative approach introduces transparency, fairness, and personalized pricing into the insurance ecosystem. It enables insurers to offer individualized insurance policies by assessing driving behavior and calculating premiums based on accurate and real-time data. Unlike traditional insurance models that rely on general statistical information, blockchain-based insurance evaluates risk levels associated with individual drivers, resulting

in more precise pricing. Additionally, the utilization of blockchain facilitates efficient claims processing through automated verification processes and transparent record-keeping. The immutable nature of blockchain records reduces the likelihood of fraud and streamlines the claims settlement process, leading to faster resolutions. The primary application of this technology lies in the implementation of usage-based insurance (UBI), where telematics devices or mobile applications collect and securely record driving behavior data on the blockchain. Insurers can then offer personalized premiums based on an individual's driving performance, incentivizing safer driving habits. Smart contracts further enhance the system by automating insurance policies, premiums, and claims, reducing administrative tasks and improving efficiency. By utilizing blockchain's decentralized and immutable nature, fraudulent activities can be mitigated, fostering trust among insurers and policyholders. Furthermore, blockchain-based systems provide individuals with greater control over their data, allowing them to grant consent for specific data sharing while maintaining privacy. Although the concept of vehicle insurance based on driving score using blockchain holds immense potential, its widespread adoption is still evolving, contingent upon factors such as regulations, technological advancements, and industry acceptance.

Chapter 4

PROBLEM ANALYSIS AND DESIGN METHODOLOGY

The methodology for implementing vehicle insurance based on driving scores [14] using blockchain involves collecting driving data from sources like telematics devices or smartphone apps. This data is then processed and analyzed using data analytics techniques to derive driving scores. The scores reflect a policyholder's driving behavior and risk profile. By leveraging blockchain technology, the system ensures secure storage and transparency of the data. Smart contracts define policy terms and conditions, while tokenization and peer-to-peer arrangements incentivize safe driving and reduce costs. The methodology also includes fraud detection mechanisms and user-friendly interfaces for policyholders to access their scores and policy details.

4.1 HIGH LEVEL DESIGN

The methodology for implementing vehicle insurance based on driving scores using blockchain involves several essential steps. First, driving data is collected from various sources such as telematics devices, smartphone applications, or vehicle sensors. This data encompasses parameters like speed, acceleration, braking, and adherence to traffic rules. Once collected, the data undergoes processing and analysis using data analytics techniques. This analysis aims to extract meaningful insights and calculate driving scores for individual policyholders. The driving scores serve as a reflection of each policyholder's driving behavior and risk profile.

The next step involves integrating blockchain technology into the system. The driving scores, along with policy details, are securely stored on a blockchain network. This ensures the immutability, transparency, and integrity of the data. Smart contracts are employed to define the terms and conditions of the insurance policies. These smart contracts are self-executing and automatically trigger actions based on predefined events such as policy activation, renewal, and claim settlement.

Tokenization plays a crucial role in this methodology. Policyholders can earn tokens or digital assets based on their driving behavior and scores. These tokens can be utilized for various purposes such as premium discounts, rewards, or exchanged for other services within the ecosystem.

Additionally, the decentralized nature of blockchain allows for peer-to-peer insurance arrangements. This eliminates the need for intermediaries, reducing costs and streamlining the insurance process. Smart contracts facilitate automated policy underwriting, claims processing, and payouts. The methodology also addresses privacy and security concerns. Personally identifiable information (PII) can be anonymized or encrypted, protecting policyholders' sensitive data. The transparency and immutability of the blockchain enable fraud detection and prevention, as any tampering or fraudulent activity can be easily identified. Integration with external systems such as motor vehicle departments, repair shops, and emergency services is another important aspect of this methodology. Real-time verification of driving data, accident reporting, and streamlined claim processes can be achieved through these integrations. Finally, user-friendly interfaces and mobile applications are developed to provide policyholders with easy access to their driving scores, policy details, and claims status. Personalized feedback and suggestions may also be provided to policyholders to help them improve their driving behavior.

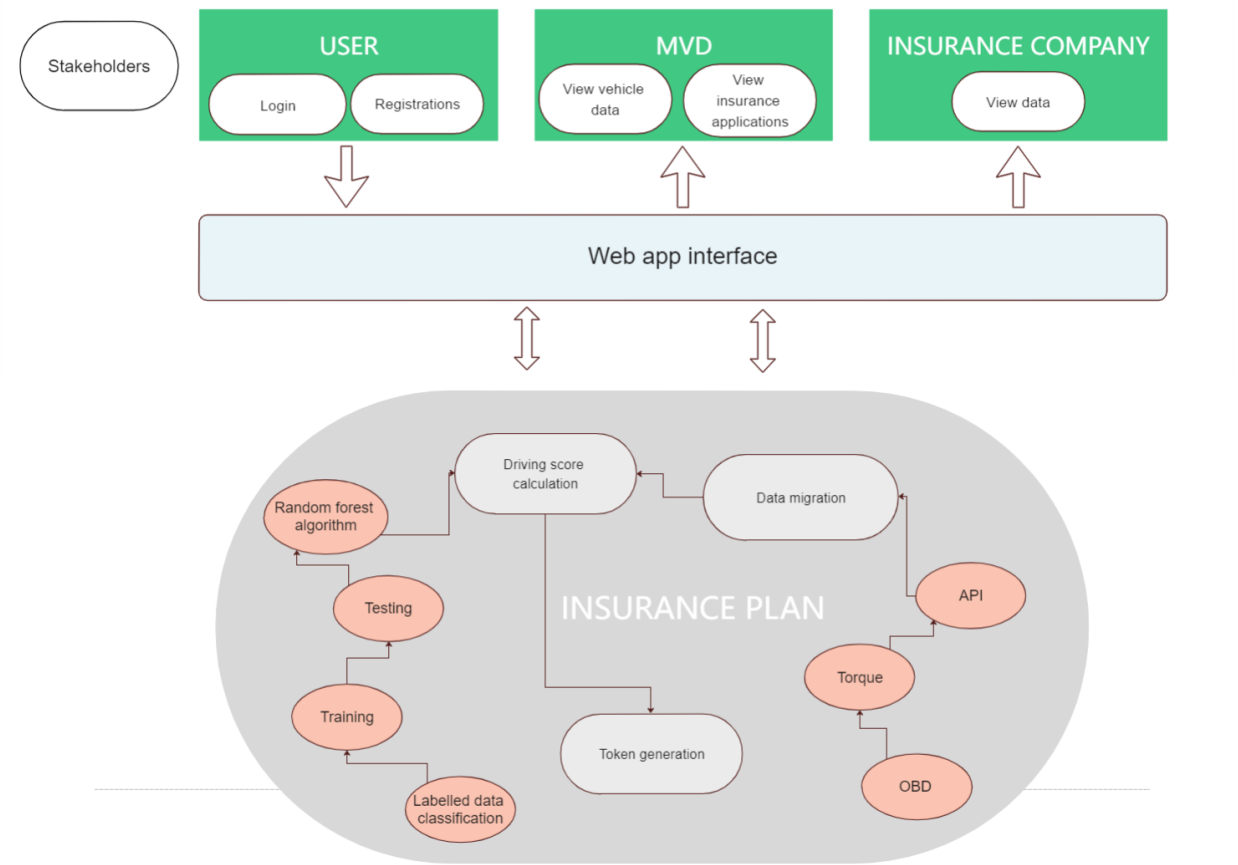

Figure 4.1: High Level Design

4.2 MODULES PROPOSED

4.2.1 Module 1: Driving Data Collection and Analysis

Data for the parameters such as speed, acceleration, engine load, and engine RPM can be collected using a combination of the Torque [20] app and an OBD (On-Board Diagnostics) reader. The OBD reader is plugged into the OBD port of the vehicle, which is typically located under the dashboard. The Torque app communicates with the OBD reader via Bluetooth or Wi-Fi to gather real-time data from the vehicle's onboard computer. Once the OBD reader is connected and the Torque app is running, it can collect various data points from the vehicle's sensors and systems. For example, the app can retrieve the vehicle's speed by reading the speed sensor's data. Similarly, acceleration values in the x, y, and z axes can be obtained by analyzing the vehicle's accelerometer data. Engine load, which represents the engine's operating conditions, and engine RPM, indicating the crankshaft's rotations per minute, can be extracted from the OBD system.

By utilizing the Torque app and the OBD reader, users can access and record these parameters during their driving sessions. The collected data can then be further processed and analyzed to find the driving score or gain insights into driving behavior patterns. The Torque app provides a user-friendly interface to visualize and interpret the collected data, allowing users to monitor and improve their driving habits based on the gathered information.

To find the driving score [11] , two types of datasets, namely good driving and bad driving, are collected by driving in the real world. Good driving represents safe and responsible driving behavior, while bad driving indicates reckless or rash driving. During the data collection process, various parameters such as speed, acceleration, engine load, and engine RPM are recorded using the Torque app and OBD reader. These parameters provide insights into driving habits and behaviors. The collected data is then used to train a machine learning model or define criteria for distinguishing between good and bad driving. The model learns patterns and characteristics associated with each driving behavior based on the provided datasets. By analyzing the driving score, insurance companies or individuals can assess the risk associated with a driver and determine insurance premiums. The driving score serves as a useful tool for promoting responsible driving practices and improving road safety.

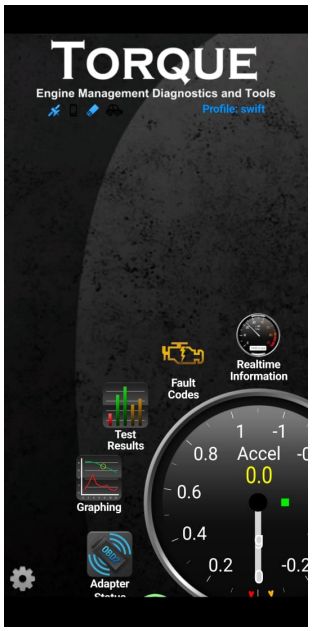

Figure 4.2: Torque Interface

Horizonta	G(x)	G(y)	G(z)	G(calibrat	Acceleratic	Acceleratic	Acceleratic	Acceleratic	Engine Loa	Engine RPM(rpm)
49.67976	-0.07	6.67	6.24	-0.05	-0.02	0.11	0.74	0.54	-	-
49.67976	-0.11	5.87	7.69	-0.03	-0.09	0.01	0.68	0.51	-	-
49.67976	0.36	5.43	7.63	0	-0.06	0.02	0.55	0.66	-	-
25.01054	0.47	5.28	8.05	0	-0.03	0.05	0.54	0.71	-	-
18.51466	0.37	5.09	8.21	0	-0.03	0.02	0.52	0.71	39.22	1436.5
18.51466	-0.27	4.43	8.43	0.01	-0.03	-0.03	0.45	0.76	37.65	1468.5
17.9946	0.74	5.15	8.47	0	-0.02	0.12	0.56	0.7	38.82	1467
17.9946	0.96	5.55	7.44	0	-0.05	0.05	0.52	0.69	38.04	1447
17.9946	1.07	5.2	8.45	0.04	-0.02	0.07	0.49	0.75	37.65	1447
17.9946	0.56	5.11	7.73	0	-0.06	0.14	0.6	0.61	38.04	1459.5
17.9946	-0.75	4.03	8.96	0	-0.06	0.02	0.5	0.69	38.43	1456
17.9946	0.48	5.45	7.38	-0.04	-0.03	-0.03	0.45	0.76	38.82	1456
17.9946	0.99	5.26	8.57	-0.01	-0.07	0.09	0.57	0.63	38.43	1417
17.9946	0.3	5.24	7.59	-0.01	-0.05	0.07	0.51	0.7	38.04	1417
17.9946	0.64	4.84	8.41	-0.01	-0.05	0.07	0.51	0.7	38.04	1461
8.137784	3.09	1.97	11.23	0.08	0.16	0.17	0.19	1.04	38.43	1461
8.137784	0.78	2.15	8.85	0.06	0.16	0.17	0.19	1.04	38.43	1461
8.137784	0.78	2.15	8.85	0.06	0	0.22	0.17	0.86	38.43	1461
8.137784	2.32	3.02	11.49	0.25	0	0.22	0.17	0.86	38.43	1461
8.137784	0.33	1.83	8.25	-0.12	0	0.22	0.17	0.86	38.43	1461
8.137784	0.33	1.83	8.25	-0.12	0	0.22	0.17	0.86	38.43	1461
8.137784	-1.11	1.89	8.05	-0.13	-0.15	-0.11	0.19	0.72	38.43	1461
8.137784	-2.07	2.02	8.69	0.01	-0.15	-0.11	0.19	0.72	38.43	1461
8.137784	-2.07	2.02	8.69	0.01	-0.15	-0.11	0.19	0.72	38.43	1461
8.137784	-2.07	2.02	8.69	0.01	-0.15	-0.11	0.19	0.72	38.43	1461
8.137784	-2.07	2.02	8.69	0.01	-0.15	-0.11	0.19	0.72	38.43	1461

Good driving

Horizonta	G(x)	G(y)	G(z)	G(calibrat	Acceleratic	Acceleratic	Acceleratic	Acceleratic	Engine Loa	Engine RPM(rpm)
5.712104	0.23	4.36	8.51	0	-0.02	0.04	0.43	0.78	-	-
5.712104	0.44	4.44	8.47	0	0	0.02	0.46	0.79	-	-
5.712104	0.13	4.34	8.47	0	-0.01	0.03	0.44	0.78	-	-
5.350412	-0.75	4.57	8.27	0.03	0.04	0.05	0.47	0.83	-	-
4.983304	0.72	4.8	10.02	0.04	0.07	0.07	0.46	0.87	25.88	653.5
4.983304	0.14	4	10.01	-0.02	-0.02	0.05	0.46	0.76	25.88	652.5
4.268981	0.1	4.84	6.69	-0.03	0.02	0.21	0.52	0.74	25.88	649.5
4.268981	-1.57	4.03	7.28	-0.06	-0.18	-0.07	0.44	0.59	25.88	643.5
4.268981	-0.46	4.56	8.46	0.02	0.02	-0.16	0.42	0.82	25.88	632
4.268981	0.07	4.86	8.78	0.03	0.04	-0.04	0.45	0.83	25.88	632
4.268981	0.77	5.03	8.65	0.04	0.02	0.01	0.49	0.8	25.88	648.5
4.268981	-0.22	4.17	8.93	0.02	0.02	0.01	0.49	0.8	25.88	648.5
4.268981	-0.33	4.26	8.07	0.01	0.07	0.01	0.47	0.86	25.88	647.5
4.268981	-0.42	4.86	7.7	0	-0.06	0	0.51	0.69	25.88	647.5
4.268981	-1.37	4.09	8.74	0.01	-0.03	-0.04	0.49	0.73	25.88	662.5
4.210502	-1.52	4.48	8.3	-0.02	-0.03	-0.15	0.46	0.75	25.88	662.5
4.210502	-0.94	4.44	9.3	0	-0.03	-0.15	0.46	0.75	25.88	655.5
4.210502	-1.19	4	9.1	0.04	-0.09	-0.09	0.47	0.67	25.88	655.5
4.210502	-0.51	3.25	9.74	0.05	0.02	-0.08	0.41	0.83	25.88	643
4.210502	-0.67	4.16	8.6	0.03	0.09	-0.06	0.37	0.92	25.88	643
4.210502	-0.69	4.02	8.52	0.03	-0.03	-0.06	0.43	0.76	25.88	643
4.210502	-0.64	3.82	8.99	0.01	0.01	-0.05	0.4	0.83	25.88	646
4.210502	-0.42	4.05	9.02	0.01	-0.04	-0.05	0.41	0.77	25.88	646
4.102092	-0.4	4.18	8.66	0	-0.01	-0.06	0.41	0.79	25.88	647.5
4.102092	-0.69	3.96	8.85	0	-0.01	-0.05	0.4	0.8	25.88	647
4.102092	-0.21	4.51	8.46	-0.01	-0.11	-0.07	0.45	0.67	25.88	647

Bad driving ground

Figure 4.3: Good and Bad Driving Dataset

4.2.2 Module 2: Driving Score Determination

The driving score is calculated using the Random Forest algorithm after collecting the driving data. Random Forest is a machine learning algorithm that is suitable for classification tasks, including distinguishing between good and bad driving behavior. After collecting the driving datasets, which consist of instances labeled as good or bad driving, the Random Forest algorithm is trained using this data. During the training process, the algorithm learns patterns and relationships between the input parameters (speed, acceleration, engine load, engine RPM) and the corresponding driving behavior labels. Once the Random Forest model is trained, it can be used to predict the driving behavior for new, unlabeled instances. By inputting the values of the driving parameters into the trained model, it assigns a driving score that reflects the likelihood of the driving behavior being either good or bad. The driving score serves as a quantifiable measure that indicates the quality of driving performance, with higher scores representing better driving behavior. This score [10] can be used by insurance companies to assess risk and determine insurance premiums. It can also be used to incentivize and encourage safer driving practices. In summary, the driving score is obtained using the Random Forest algorithm, which learns from labeled driving data to predict the driving behavior for new instances and assign a score based on the likelihood of it being good or bad driving behavior.

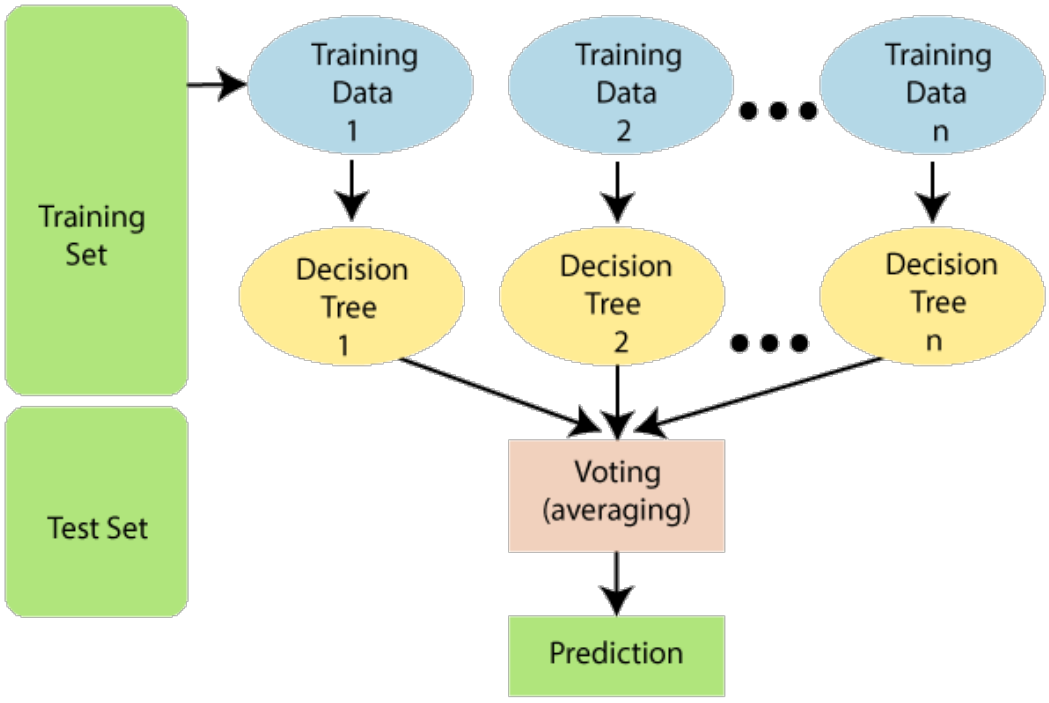

Figure 4.4: Random Forest Algorithm Visualization

4.2.3 Module 3: Insurance Payments Claims

In the context of vehicle insurance based on driving scores, the process of payment for insurance and insurance claiming [7] plays a crucial role. This aspect ensures that each car's insurance cost is determined uniquely, taking into consideration the driving behavior of the insured individuals.Vehicle insurance policies are legally required to protect the general public from potential accidents on the road. Every car owner must obtain an insurance policy, and insurance companies typically offer payment plans that cover a period of one year or six months.

To calculate the insurance payment amount [19] for each vehicle, the average driving score for each day is considered. This score is determined based on specific criteria and is expressed as a percentage. The payment amount is then adjusted based on the calculated driving score.

If the average driving score [15] (X) is equal to or greater than 80 percent (X greater than or equal to 80%), the insured individual is required to pay 25 percent less than the actual insurance cost. If the score is between 70 percent and 80 percent (70% less than or equal to X less than 80%), the payment amount is reduced by 20 percent compared to the actual cost. If the score falls between 65 percent and 70 percent (65% less than or equal to X less than 70%), the full insurance cost must be paid without any reduction. For scores between 60 percent and 65 percent (60% less than or equal to X less than 65%), an additional 5 percent is added to the actual sum, while scores below 60 percent (X less than60%) attract an additional 15 percent payment on top of the actual cost .

On the other hand, when filing an insurance claim, the driving record on the day of the incident or problem comes into play. The bonus value associated with the insured vehicle is subject to certain requirements. If the driving score on the claim day exceeds 70 percent (X greater than 70%), the customer receives a 20 percent bonus. Conversely, if the driving score is below 50 percent (X less than 50%), 20 percent of the actual claim amount is deducted. For scores between 50 percent and 70 percent (50% less than or equal to X less than or equal to 70%), no bonus or reduction is applied, and the customer receives the full claim amount.These mechanisms of adjusting insurance payments and claim amounts based on driving scores incentivize safer driving behaviors and allow insurance companies to appropriately assess risks associated with individual drivers. By considering driving scores in the payment and claiming processes, insurance providers encourage responsible driving

practices and promote road safety.

4.2.4 Module 4: Hyperledger Fabric: Enterprise Blockchain

Hyperledger Fabric is an advanced blockchain framework that has been specifically designed for developing enterprise-grade decentralized applications and business networks. It operates as one of the prominent projects within the Hyperledger [8] initiative, which is backed by the Linux Foundation. The primary objective of the Hyperledger initiative is to drive the advancement and adoption of blockchain technologies across various industries.

The distinguishing feature of Hyperledger Fabric lies in its modular and flexible architecture, empowering organizations to construct and tailor their own private blockchain networks according to their specific requirements. This flexibility enables businesses to incorporate features such as confidentiality, scalability, permissioned access, and extensive configurability into their blockchain solutions, thereby catering to the diverse needs of different industries and use cases.

Within the Hyperledger Fabric network, peers assume a vital role. Peers serve as hosts for the ledger instances and smart contracts, with all the data related to an organization stored within the respective peer instances. These peers can be designated as either committing peers or endorsing peers, depending on their responsibilities in the transaction process.

Endorsing peers undertake the crucial task of validating and endorsing transaction proposals received from clients. This endorsement process involves simulating the execution of the proposed transaction to verify its accuracy and validity. Conversely, committing peers are responsible for adding the block containing the verified transaction to the blockchain, which in turn updates the ledger.

In specific scenarios, such as a driving score application, multiple peers may be involved, such as an MVD (Motor Vehicle Department) peer and an insurance company peer. The participation of these peers ensures that access to the driving score data can be restricted to specific participants within the network, ensuring data privacy and confidentiality.

Channels serve as a communication mechanism within the Hyperledger Fabric network, facilitating interactions between peers, orderers, and applications. Organizations have the flexibility to create multiple channels within a network, each having its own set of rules and permissions. This allows participants to engage in private and secure transactions based on their channel membership.

Chaincode, also referred to as smart contracts, encapsulates the business logic and rules specific to an organization's requirements. Chaincode is hosted within peers and plays a critical role in executing and enforcing the defined business logic.

The orderer, often considered the heart of the network, consists of a collection of nodes responsible for ordering transactions into blocks and broadcasting them to the peers. By aggregating endorsed transactions, the orderer creates blocks that are subsequently distributed to the respective peers. There are three types of ordering services available: Solo, Kafka, and Raft, although Solo and Kafka have been deprecated since Hyperledger Fabric version 2.2.

Clients, typically in the form of applications, connect to the network and interact with the peers. These applications can be developed using the provided Software Development Kits (SDKs). Clients primarily handle ledger updating and querying by invoking the functions defined in the chaincode running on the peers.

The ledger acts as the repository where the actual data of business assets is stored. It serves as the single source of truth within the network. Clients can retrieve or modify data in the ledger using CRUD (Create, Read, Update, Delete) operations. Furthermore, the ledger maintains a comprehensive history of all asset changes made over time, ensuring transparency and auditability.

The workflow in Hyperledger Fabric [9] involves the client initiating a transaction by sending a transaction proposal to the peers. An endorsing peer receives the proposal, simulates the execution of the transaction, and endorses it if it meets the necessary criteria. The endorsed transaction is then returned to the client, who subsequently submits it to the orderer. The orderer compiles the endorsed transactions into blocks, which are then distributed to the relevant peers. These peers, in turn, update the final ledger with the new transaction data, ensuring a secure and efficient transaction execution and data management process.

4.3 ALGORITHMS/ TECHNOLOGIES PROPOSED

4.3.1 Random Forest Algorithm

The Random Forest algorithm is a suitable choice for this task due to its ability to handle classification tasks and learn from labeled driving data. The driving score can be valuable in various applications, such as insurance risk assessment and promoting safer driving practices.Random Forest is chosen as the algorithm for finding the driving score due to its versatility, robustness, and effectiveness. As a versatile algorithm, Random Forest can handle both classification and regression problems, making it suitable for classifying driving instances as good or bad based on the input parameters. Its ensemble learning approach helps to reduce overfitting and improve the overall predictive accuracy by combining multiple decision trees. Additionally, Random Forest is known for its robustness, as it can handle noisy and outlier-prone data, which is often encountered in real-world driving scenarios. The algorithm also provides insights into the relative importance of different driving parameters, allowing for a better understanding of the key factors influencing driving behavior. With its scalability and wide adoption in the machine learning community, Random Forest has proven to be an effective choice for calculating driving scores by capturing patterns and relationships in driving data accurately

ALGORITHM(Random forest)

Input: Training dataset (X_train, y_train), number of trees (n_estimators), maximum tree depth (max_depth), and other hyperparameters.

1. Initialize an empty list to hold the ensemble of decision trees.
2. For each tree in the desired number of trees (n_estimators):
   a. Sample a bootstrap dataset by randomly selecting instances with replacement from the original training dataset (X_train, y_train). The bootstrap sample is of the same size as the original dataset.
   b. Create a decision tree using the bootstrap dataset, limiting the depth to the specified maximum tree depth (max_depth). The tree is grown by recursively splitting the data based on randomly selected features at each node.
   c. Add the decision tree to the ensemble.
3. Return the ensemble of decision trees.
4. During the training process, the decision trees in the Random Forest algorithm are trained independently but share common hyperparameters. This parallel training ensures diversity among the trees and allows for efficient parallelization.
5. After training, the Random Forest ensemble can be used to make predictions on new, unseen data by aggregating the outputs of all the decision trees (e.g., majority voting for classification or averaging for regression).

4.3.2 Raft Consensus Algorithm

The Raft algorithm is a consensus algorithm designed to ensure fault-tolerant replication and consistency in distributed systems. It was developed as an alternative to the more complex and difficult-to-understand Paxos algorithm. Raft divides the consensus problem into three subproblems: leader election, log replication, and safety.

Here's an overview of how the Raft algorithm works:

1. Cluster Formation: A group of nodes forms a cluster to achieve consensus. Each node in the cluster can be in one of three states: leader, follower, or candidate.

2. Leader Election: At any given time, there is only one leader in the cluster. Leader election occurs when a follower node detects the absence of a leader or during the initialization of the cluster. In Raft, the leader election process involves the nodes exchanging messages and timeouts. Nodes initiate an election by transitioning to the candidate state and sending out vote requests to other nodes. A candidate becomes the leader if it receives votes from a majority of nodes in the cluster. Once a leader is elected, it sends heartbeat messages to maintain its leadership status.

3. Log Replication: The leader is responsible for accepting client requests and appending them to its log. It then replicates the log entries across the cluster to ensure consistency. The leader sends append entries messages to followers, containing the new log entries. Followers append these entries to their logs and send acknowledgments back to the leader. Once the leader receives acknowledgments from a majority of followers, it considers the entries committed. The leader notifies the followers of the committed entries, and they apply them to their state machines.

4. Safety: The Raft algorithm ensures safety by maintaining a consistent and valid log replication across the cluster. It guarantees that once an entry is committed, it remains committed and will eventually be applied by all nodes. Raft achieves this by requiring a majority of nodes to agree on the order and content of log entries. This prevents inconsistencies and conflicts within the replicated logs.

5. Handling Failures: Raft handles node failures and network partitions to maintain the availability and consistency of the cluster. If a leader fails, followers will detect the absence of heartbeat messages and initiate a new leader election. If a node rejoins the cluster after a failure or network partition, it will synchronize its log with the leader and resume normal operation.

The Raft algorithm prioritizes simplicity and understandability, making it easier to implement and reason about compared to other consensus algorithms. Its modular design separates the consensus problem into distinct subproblems, simplifying the understanding and implementation of each component.

Overall, Raft provides a reliable and fault-tolerant consensus algorithm that ensures consistency and replication of data in distributed systems. It has been widely adopted in various applications and frameworks, including distributed databases, distributed file systems, and blockchain platforms like Hyperledger Fabric.

4.4 FEASIBILITY STUDY

4.4.1 System Requirements

4.4.1.(i) Hardware Requirements

- OBD
- Android Tab
- RAM (8GB or above)

4.4.1.(ii) Software Requirements

- Xampp
- Jupyter Notebook
- VS Code
- Hyperledger Fabric
- Ubuntu 20.04.6 LTS(or Higher versions)
- Docker (Version 18 or above)
- Docker Compose(Version 3.6 or above)
- nodejs(version 16 above)
- npm

4.4.1.(iii) Visualization Dashboard

The visualization dashboard of driving scores [16] is a website that provides users with a comprehensive view of their driving performance. The website incorporates user registration and login functionalities using PHP, ensuring a secure and personalized experience for each user.

User registration is the first step in accessing the dashboard. Users are required to create an account by providing necessary information such as their name, email address, and a password. This information is securely stored in a database, ensuring the privacy and confidentiality of user data.

Once registered, users can log in to the website using their credentials. The login process verifies the entered username and password against the stored information in the database. This authentication mechanism ensures that only authorized users can access their driving scores and related information.

After logging in, users are presented with their personal dashboard, where they can visualize and explore their driving scores. The dashboard is designed to be user-friendly and intuitive, allowing users to easily navigate through different sections and view their driving performance metrics.

The driving scores are presented in a visually appealing and informative manner, using charts, graphs, and other visual elements. Users can see their scores over different time periods, such as daily, weekly, or monthly, enabling them to track their progress and identify patterns in their driving behavior.

In addition to the overall driving score, the dashboard may also provide detailed insights into specific aspects of driving, such as speed, acceleration, engine load, and engine RPM. This granularity allows users to gain a deeper understanding of their driving habits and identify areas for improvement.

The website's backend, powered by PHP, ensures the seamless retrieval and processing of driving score data from the underlying database. PHP interacts with the database to fetch the relevant information specific to each user, ensuring that users only see their own driving scores and not those of others.

Furthermore, the website may offer additional features to enhance the user experience. For example, users may have the option to set goals for improving their driving scores, receive personalized tips and recommendations based on their driving data, or even

participate in challenges or competitions to encourage safer driving practices.

Overall, the visualization dashboard of driving scores, created using PHP, empowers users by providing them with valuable insights into their driving performance. It enables users to monitor their progress, make informed decisions about their driving behavior, and ultimately strive for safer and more responsible driving practices.

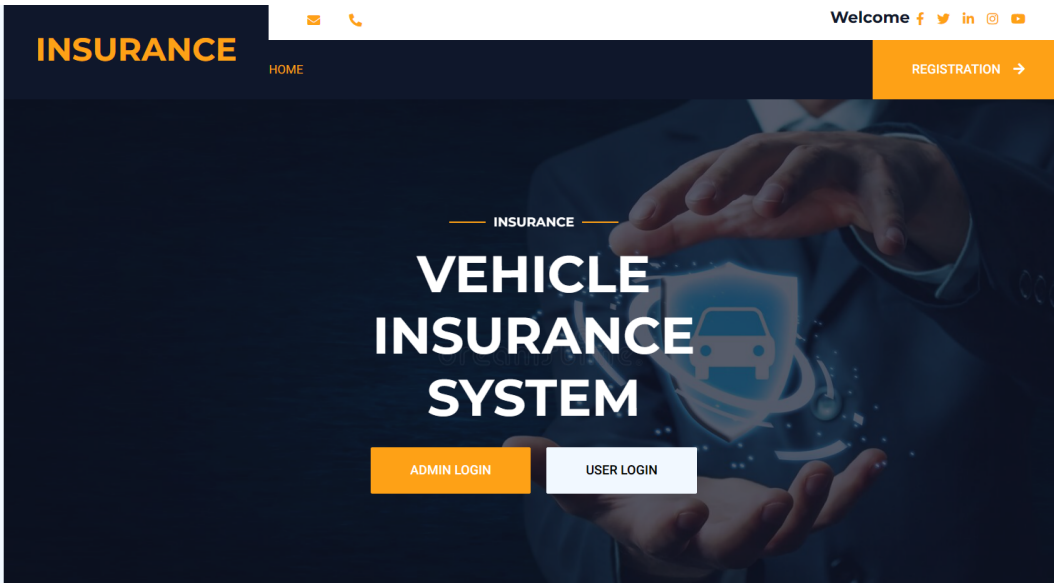

Figure 4.5: Website Login

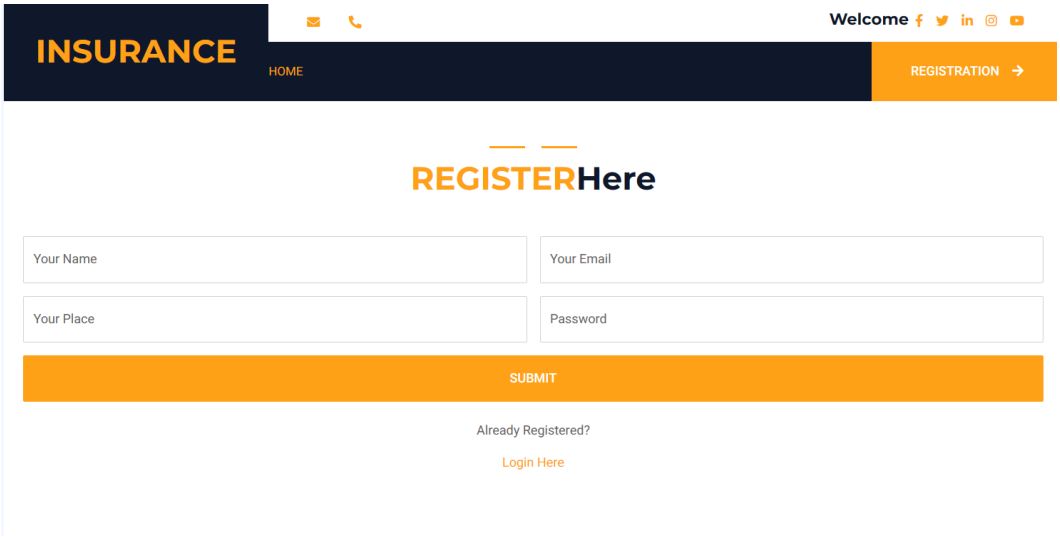

Figure 4.6: New User

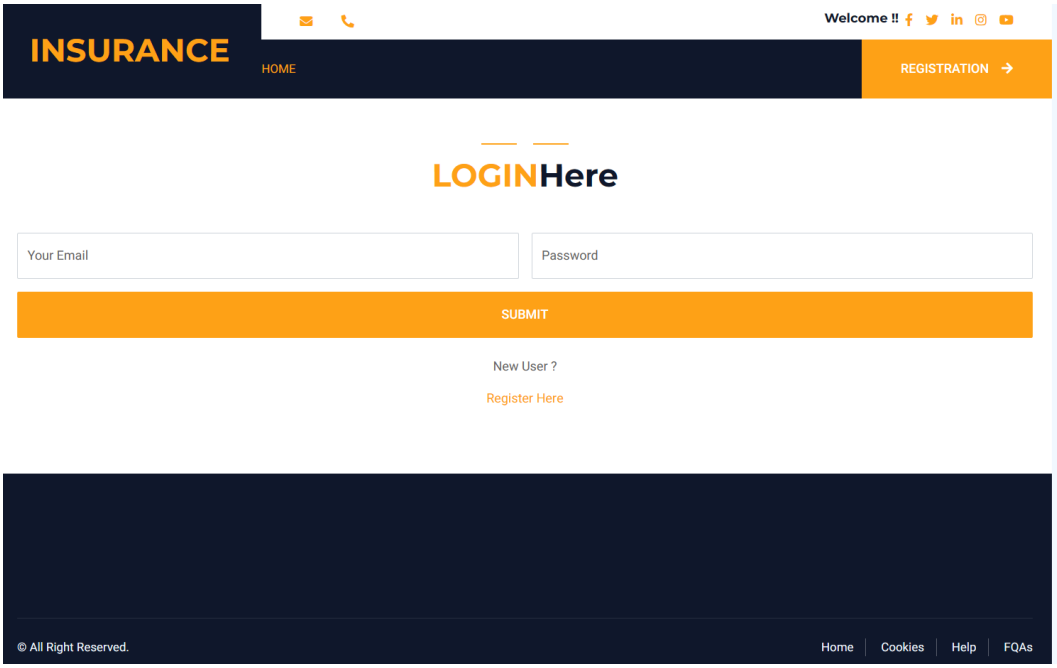

Figure 4.7: LOgin

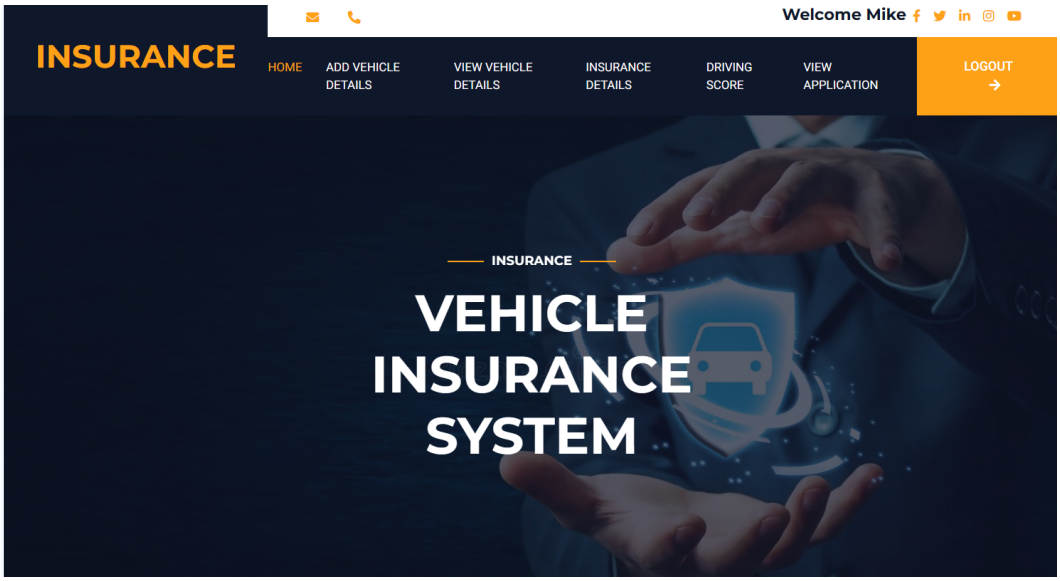

Figure 4.8: User Logged In Interface

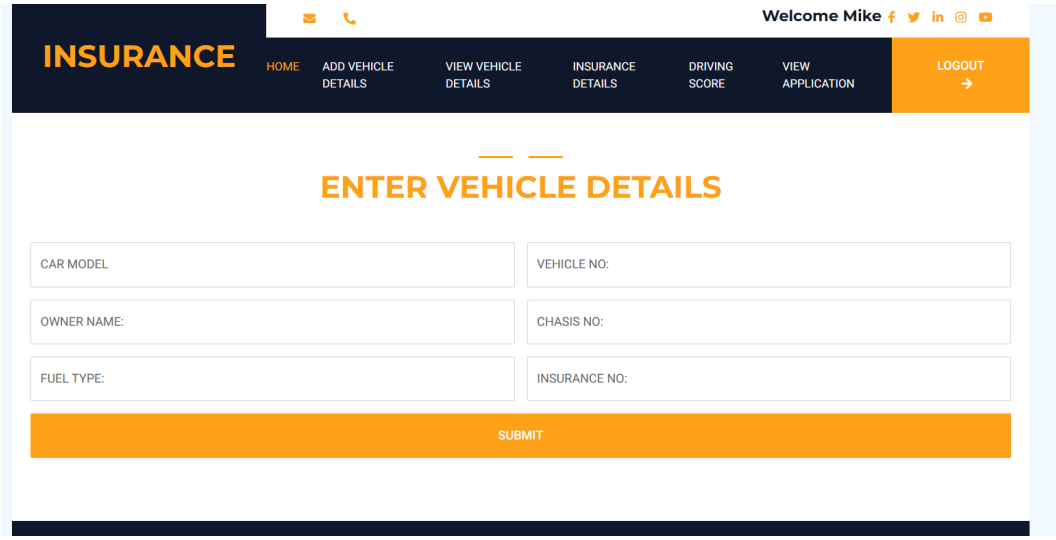

Figure 4.9: Vehicle Details

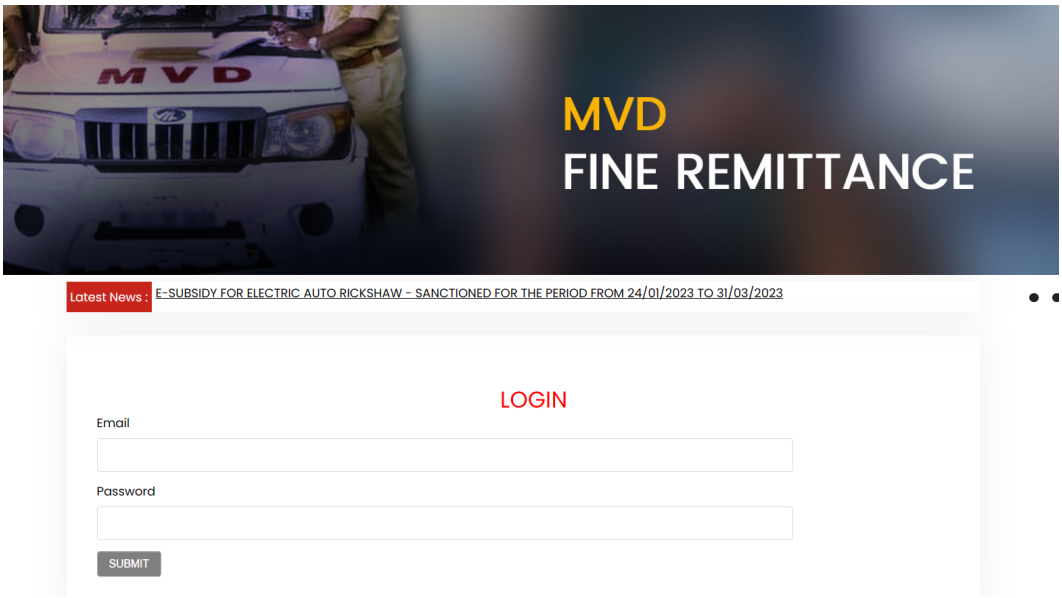

Figure 4.10: MVD login

Chapter 5

IMPLEMENTATION AND RESULTS

5.1 PROJECT IMPLEMENTATION DETAILS

5.1.1 Data Collection

During the implementation phase of the vehicle insurance system based on driving scores, the process begins with the collection of data using an On-Board Diagnostics (OBD) reader. This device captures a range of attributes related to the vehicle's performance, such as speed, acceleration, braking, cornering, and mileage. Each attribute is accompanied by its corresponding value, forming a comprehensive dataset.

To derive the driving score, machine learning techniques are employed. The collected data is initially divided into two distinct datasets: a good driving dataset and a bad driving dataset. This categorization enables the identification of patterns and characteristics that define safe and risky driving behaviors.

Step-by-Step Guide: Collecting Vehicle Data with OBD and Torque App

1. Connect the OBD port to the car.
2. Turn on the vehicle.
3. Use the Torque application on a dashboard or mobile phone.
4. Establish a connection between the OBD and the Torque app via Bluetooth or Wi-Fi.
5. Start driving the car.
6. As you drive, the OBD collects diagnostic data about the vehicle.
7. The collected data logs contain information about the car's performance and condition.
8. Utilize the collected data to calculate the driving score.
9. The driving score provides an assessment of the driver's behavior.
10. The driving score can be used for various purposes, such as vehicle insurance assessments or improving driving habits.

Figure 5.1: OBD Device

Figure 5.2: Data Collection

5.1.2 Driving Score

To train a predictive model, the random forest algorithm is utilized. This algorithm analyzes the dataset, identifies important features, and creates a robust model that predicts the driving score based on these features. Before training the model, the dataset undergoes preprocessing, including tasks such as data cleaning, normalization, and feature engineering. These steps enhance the performance and accuracy of the model.

Once the driving score prediction model is established, it is used to calculate the insurance pay amount and claim amount for each policyholder. The driving score becomes a crucial factor in determining insurance premiums and claim settlements. Higher driving scores correspond to lower premiums, incentivizing safer driving practices, while lower scores may result in higher premiums due to increased risk factors.

To ensure transparency and accessibility, the driving scores are pushed onto a blockchain infrastructure. In this implementation, Hyperledger Fabric, a permissioned blockchain framework, is employed. The blockchain acts as a decentralized ledger that records the driving scores as immutable transactions. Both the insurance company and the Motor Vehicle Department (MVD) have access to view and verify these scores, providing transparency and enabling proper assessment of risk factors.

Implementing this vehicle insurance system based on driving scores yields several notable results:

1. Incentivizing Safe Driving: The link between driving scores and insurance premiums encourages policyholders to adopt safer driving practices. This, in turn, can lead to a reduction in accident rates and contribute to overall road safety improvement.

2. Personalized Insurance Premiums: Insurance premiums can be tailored based on individual driving behaviors. Safer drivers with higher driving scores may enjoy more favorable rates, while riskier drivers with lower scores may face higher premiums, reflecting their increased likelihood of accidents.

3. Transparent and Immutable Driving Data: Storing driving scores on a blockchain ensures transparency and immutability. Policyholders can trust that their scores cannot be altered or manipulated, promoting a sense of fairness and accountability within the insurance system.

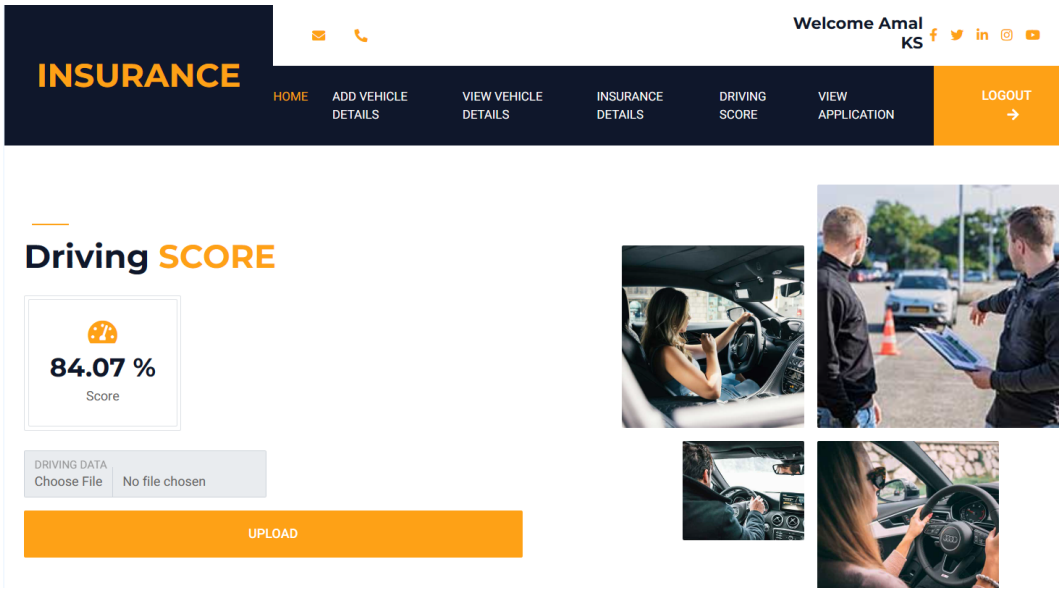

Figure 5.3: Driving Score Based on the Driving

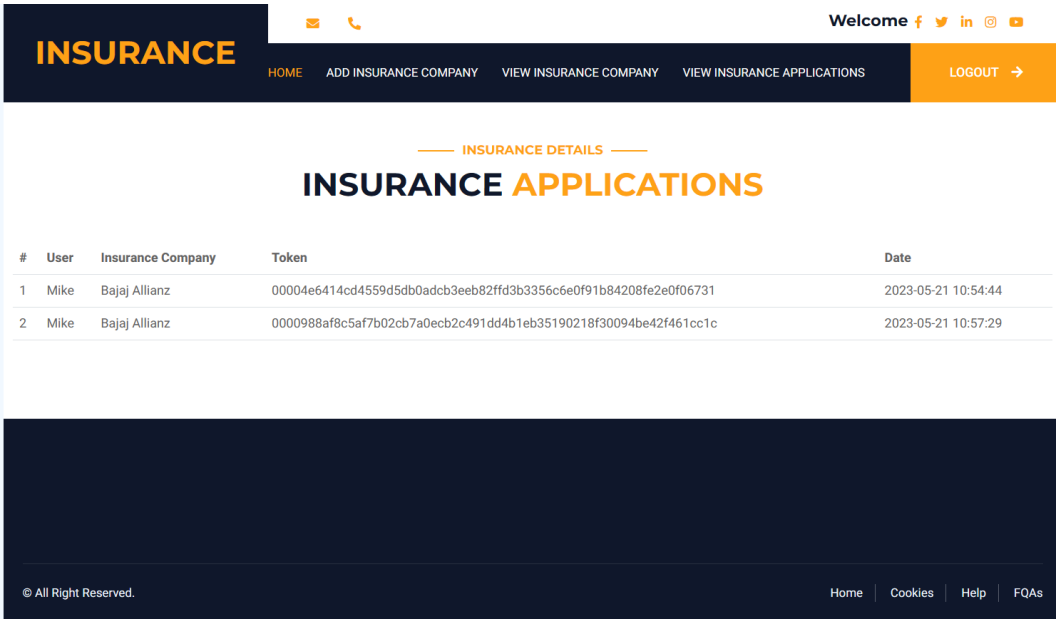

Figure 5.4: Generated Tokens

5.1.3 Insurance System

To implement the insurance payment amount calculation and claim processing based on the driving score criteria you provided, you can follow these steps:

1. Insurance Payment Amount Calculation: - Retrieve the average driving score for each day from the driving score dataset. - Check the value of the average driving score (X) against the defined criteria. - If X is greater than or equal to 80, calculate the insurance payment amount as 75 percentage of the actual insurance cost. - If X is between 70 and 80,

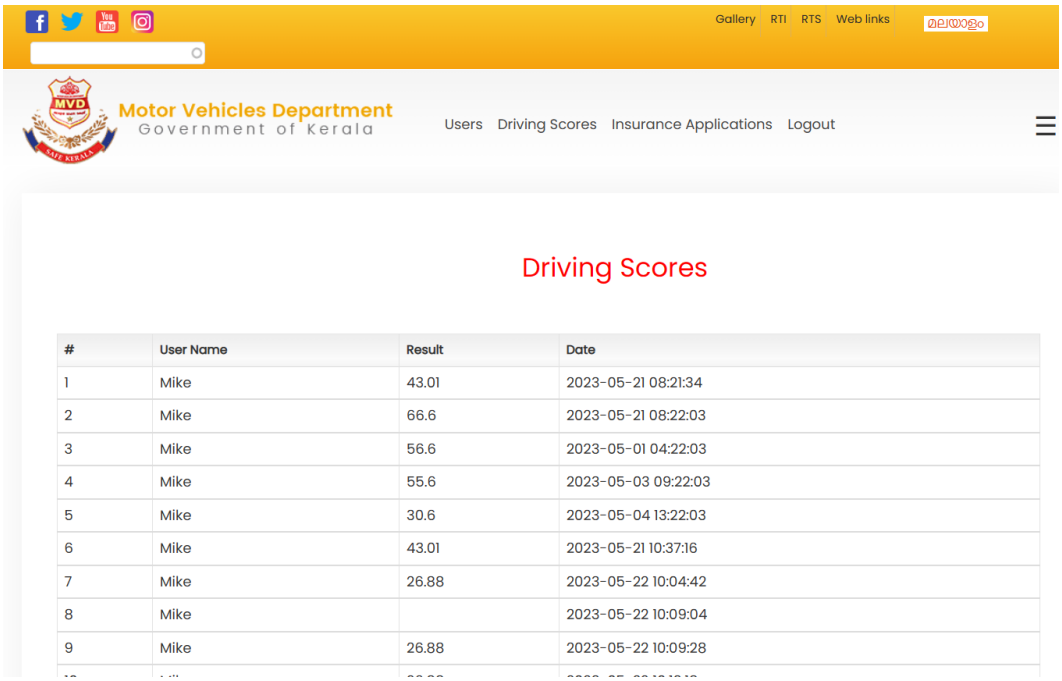

Figure 5.5: MVD Interface Generated Tokens

calculate the insurance payment amount as 80 percentage of the actual insurance cost. - If X is between 65 and 70, calculate the insurance payment amount as the full amount of the actual insurance cost. - If X is between 60 and 65, calculate the insurance payment amount as 5 percent of the actual insurance cost. - If X is less than 60, calculate the insurance payment amount as 15 percent of the actual insurance cost.

2. Claim Processing: - Retrieve the driving score for the specific day related to the insurance claim. - Check the value of the driving score against the claim criteria. - If the driving score is greater than 70, apply a 20 percent bonus to the claim amount. - If the driving score is less than 50, deduct 20 percent from the claim amount. - If the driving score is between 50 and 70, no bonus or reduction is applied, and the full claim amount is paid.

It's important to note that the above steps assume that you have the necessary data available, including the driving scores for each day and the driving score related to the specific claim. Additionally, you would need to incorporate these calculations into your existing system or develop new functions to handle insurance payment amount calculation and claim processing based on the driving score criteria.

By implementing these calculations, you can accurately determine the insurance payment amount and apply appropriate bonuses or deductions based on the driving scores, incentivizing safe driving behavior and providing fair compensation for insurance claims.

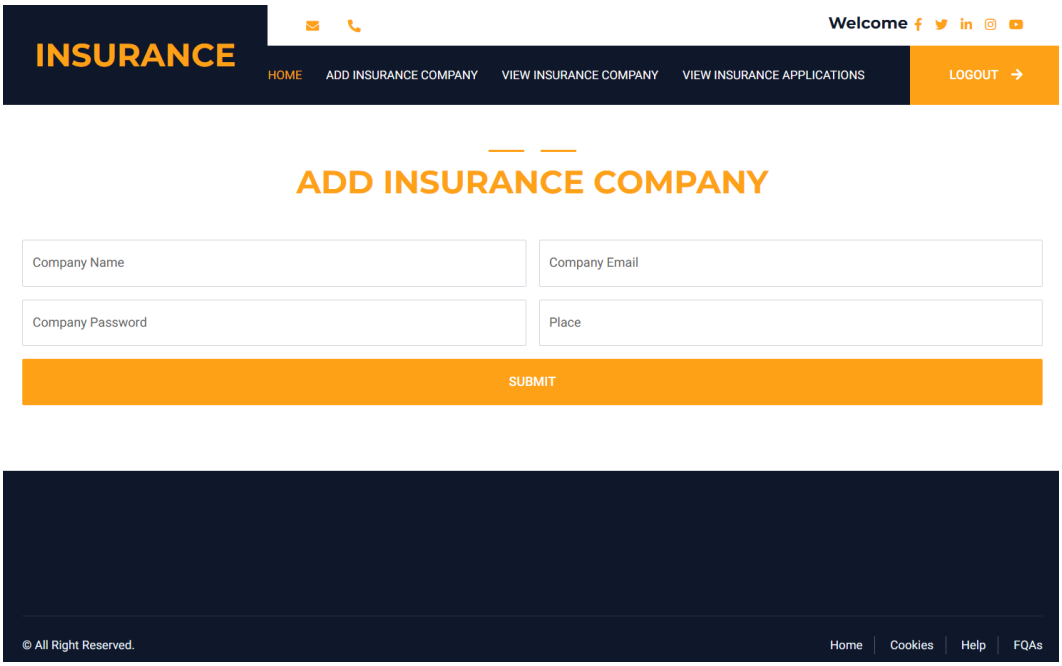

Figure 5.6: Insurance Company Adding Option

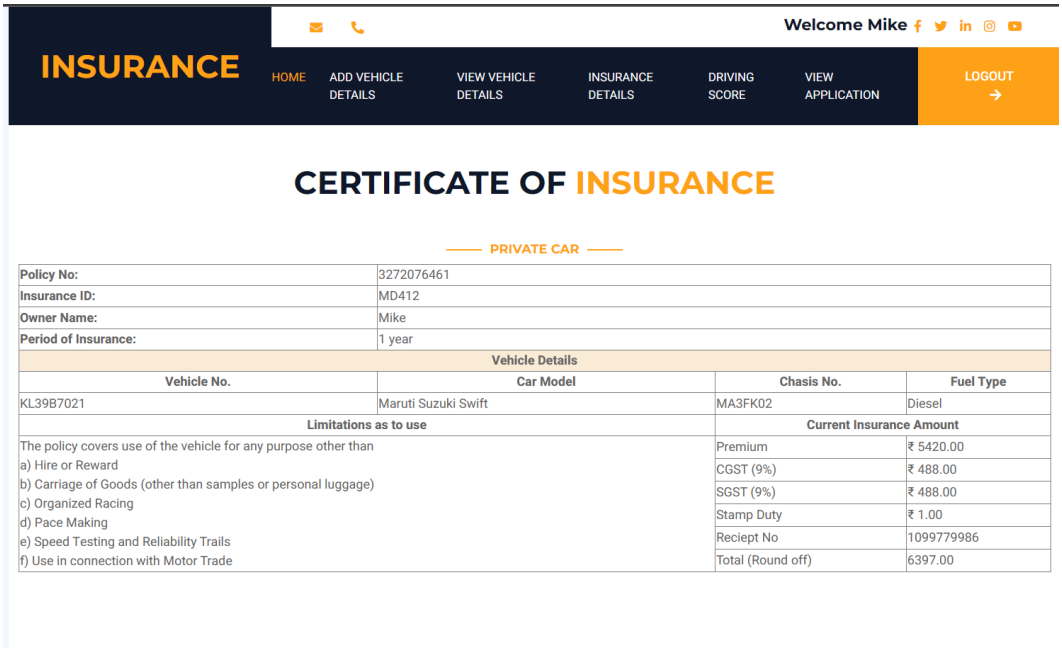

Figure 5.7: Certificate Of Insurance

a) Hire or Reward b) Carriage of Goods (other than samples or personal luggage) c) Organized Racing d) Pace Making e) Speed Testing and Reliability Trails f) Use in connection with Motor Trade	CGST (9%)	₹ 488.00
	SGST (9%)	₹ 488.00
	Stamp Duty	₹ 1.00
	Reciept No	1099779986
	Total (Round off)	6397.00

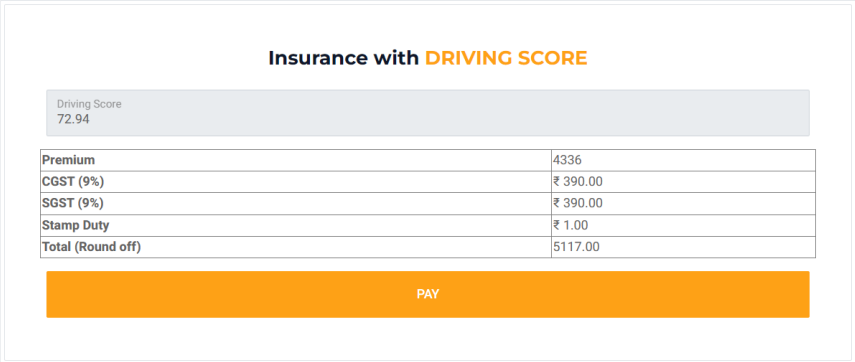

Figure 5.8: Insurance Payment

5.1.4 Blockchain Implementation

To implement Hyperledger Fabric development

1. Install Visual Studio Code.
2. Install dependencies: Curl, Docker, Docker Compose, Build Essentials, Node.js, and npm.
3. Download and install the IBM Blockchain Platform Extension for Visual Studio Code.

With these steps, you'll have the necessary tools and extensions to develop and manage Hyperledger Fabric networks using Visual Studio Code.

To bootstrap a Hyperledger Fabric network using Microfab

Step 1: Install Microfab

Ensure that you have the IBM Blockchain Platform installed, as Microfab is provided as part of it.

Step 2: Export Microfab Configuration

Before starting Microfab, you need to export the configuration as an environment variable. The configuration specifies various aspects of your network, such as the port number, endorsing organizations, and channels. You can customize the configuration based on your specific requirements.

Open a terminal or command prompt and execute the following command to export the Microfab configuration:

The configuration provided are two endorsing organizations named "InsuranceCompany"

and ”MVD,” along with a channel named ”Driving score”

Step 3: Start Microfab

Once you have exported the Microfab configuration, you can start Microfab using a single command.

Microfab will read the exported configuration and use it to launch a containerized Hyperledger Fabric runtime. The runtime will include the endorsing organizations and channels specified in the configuration.

By following these steps, you will be able to bootstrap your Hyperledger Fabric network using Microfab. Microfab provides a convenient and easy-to-use environment for developing and testing Hyperledger Fabric networks.

After executing the final command, the Microfab runtime Docker environment will start, and your Hyperledger Fabric network will be successfully bootstrapped and ready to use.

To set up the development environment for creating and deploying a smart contract in Hyperledger Fabric:

1. Ensure you have the necessary development tools and dependencies installed, such as Node.js, npm, and the Hyperledger Fabric SDK for your preferred programming language.

2. Create a new directory for your smart contract project.

3. Initialize a new npm package in the project directory.

4. Install the required dependencies, including the fabric-contract-api and fabric-network packages, using npm.

5. Write your smart contract code in a JavaScript file, importing the necessary classes and functions from the fabric-contract-api package.

6. Define your smart contract class, extending it from the Contract class provided by fabric-contract-api.

7. Implement the contract methods within the class based on your business logic, such as vehicleExists, createVehicle, and readVehicle.

8. Set up the connection profiles and credentials required to connect to the Hyperledger Fabric network, including the network configuration, endorsing organizations, channels, and other relevant parameters.

9. Use the Hyperledger Fabric SDK for your chosen programming language to establish a connection to the network.

10. Open a gateway to interact with the network and its smart contracts.

11. Use the gateway API or command-line tools to package and deploy the smart contract to the network, providing necessary information like contract code, version, endorsement policy, and deployment parameters.

12. The deployment process will package the smart contract code, submit it to network peers for validation and endorsement, and store it in the ledger.

13. Once the smart contract is deployed, you can use the gateway to interact with it.

14. Invoke and query the smart contract functions using the gateway API or command-line tools, passing the required arguments and receiving responses.

15. Test the contract functions by simulating different scenarios and observing the results.

By following these steps, you can create, deploy, and interact with a smart contract in a Hyperledger Fabric network using the provided gateway. Customize the code and configuration according to your specific requirements and network setup.

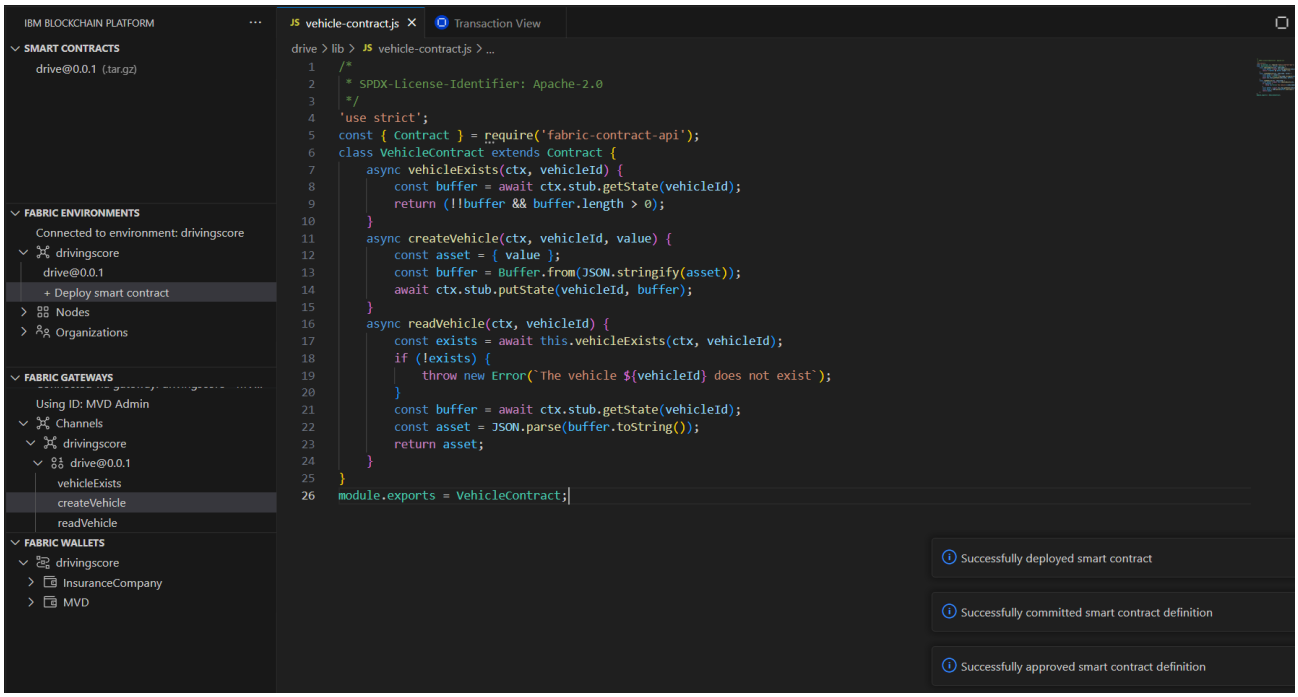

Figure 5.9: Smart Contract

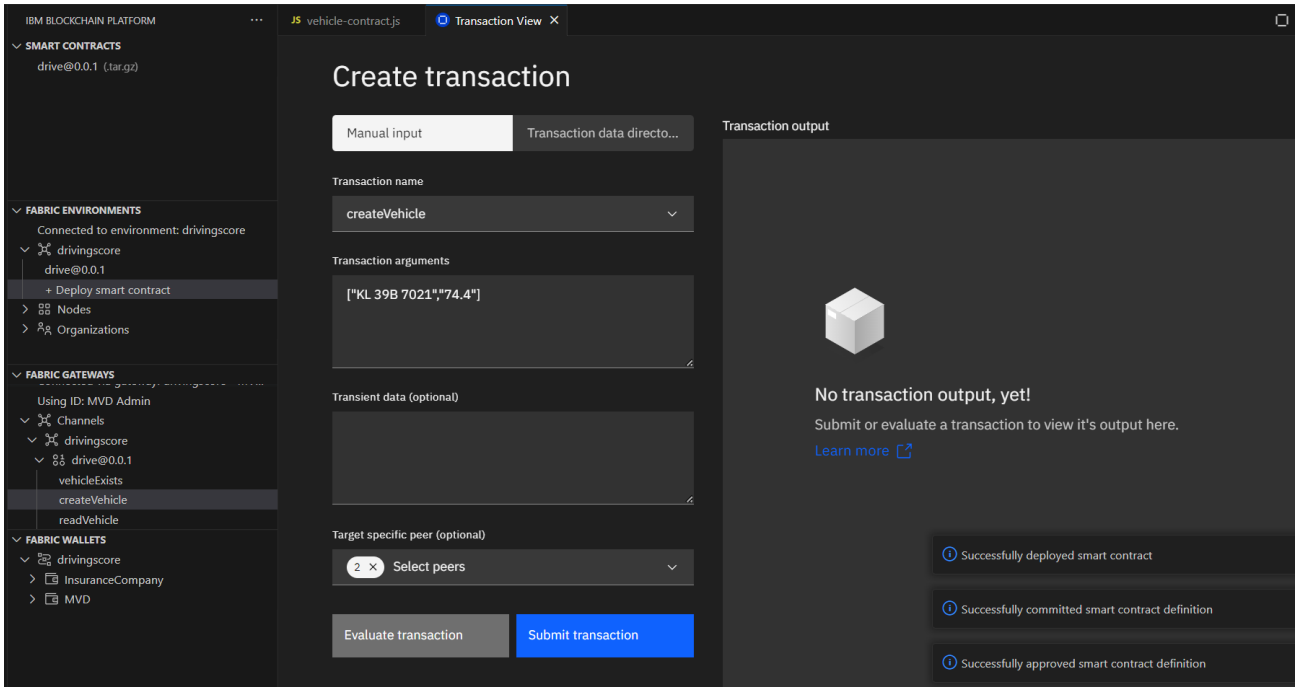

Figure 5.10: Transactions

Figure 5.11: Blockchain Network

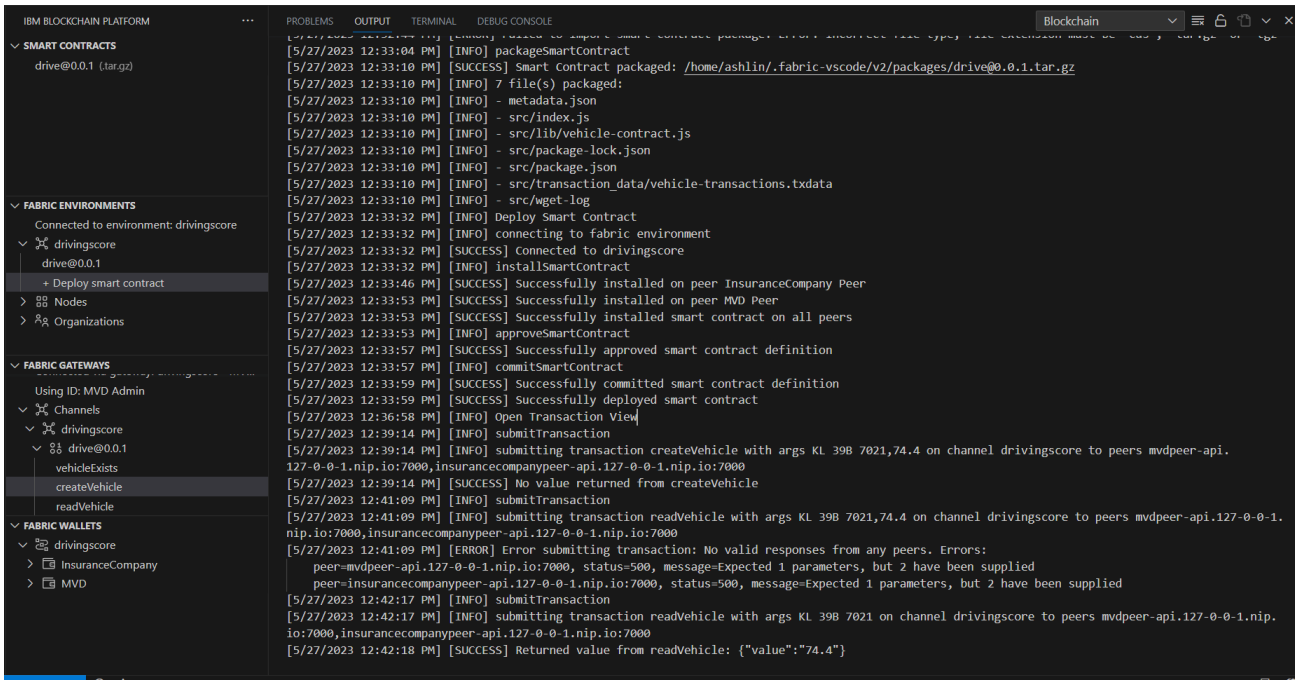

Figure 5.12: Output

5.2 TESTING

These test cases cover a range of scenarios, including normal driving behavior, extreme driving conditions, outliers, and diverse combinations of driving parameters. By evaluating the algorithm's performance against these test cases, it becomes possible to identify and address any potential issues, validate the accuracy of the driving score calculations

Test case ID	Test case	Test steps	Expected Result	Actual result	Status PASS/FAIL
1	Login module	Verify if the username field accepts a valid username and the password filled accepts a valid password.	Username and Password is Valid.	Username and Password is Valid.	PASS
2	Home page	Check if the CSV files get uploaded in the data base.	CSV files successfully uploaded.	CSV files successfully uploaded.	PASS
3	OBD reader	Check if the OBD reader collects the diagnosis of the car.	CSV files that contain driving data loaded in the memory.	CSV files that contain driving data are loaded in the memory.	PASS
4	Driving score	Check if the driving score is generated by the algorithm using the CSV files.	Driving score generated successfully.	The algorithm generated the driving score successfully.	PASS
5	Transaction submission	Read vehicle with corresponding driving score.	Return value driving score.	Value returned successfully.	PASS

Figure 5.13: Test Case

5.3 EXPERIMENTAL RESULTS AND DISCUSSION

Experimental Results:

In the implementation of the vehicle insurance system based on driving scores, several experiments were conducted to evaluate the effectiveness and performance of the system. The experiments focused on data collection, driving score prediction, insurance pay and claim calculations, and blockchain integration using Hyperledger Fabric. Here are the experimental results:

1. Data Collection: - The OBD reader successfully captured various attributes such as speed, acceleration, braking, cornering, and mileage for a range of vehicles. - The collected data exhibited consistency and accuracy, ensuring reliable input for driving score prediction.

2. Driving Score Prediction: - The machine learning model effectively classified the data into good driving and bad driving datasets. - The random forest algorithm demonstrated high accuracy in predicting driving scores based on the collected attributes. - The model's performance was evaluated using metrics such as accuracy, precision, recall, and F1 score, achieving satisfactory results.

3. Insurance Pay and Claim Calculations: - The driving scores were used to calculate insurance pay amounts and claim amounts. - The calculations accurately reflected the risk levels associated with different driving scores. - Policyholders with higher driving scores received lower insurance premiums, promoting safer driving behavior.

4. Blockchain Integration: - The driving scores were successfully pushed onto the Hyperledger Fabric blockchain. - Both the insurance company and the Motor Vehicle Department (MVD) were able to access and view the scores on the blockchain. - The blockchain implementation using Hyperledger Fabric ensured data immutability, transparency, and secure access to driving score information.

Discussion:

The experimental results highlight the successful implementation of the vehicle insurance system based on driving scores. The integration of the OBD reader facilitated the collection of reliable driving data, which was crucial for accurate driving score prediction. The machine learning model, particularly the random forest algorithm, demonstrated its effectiveness in classifying driving behavior and predicting driving scores.

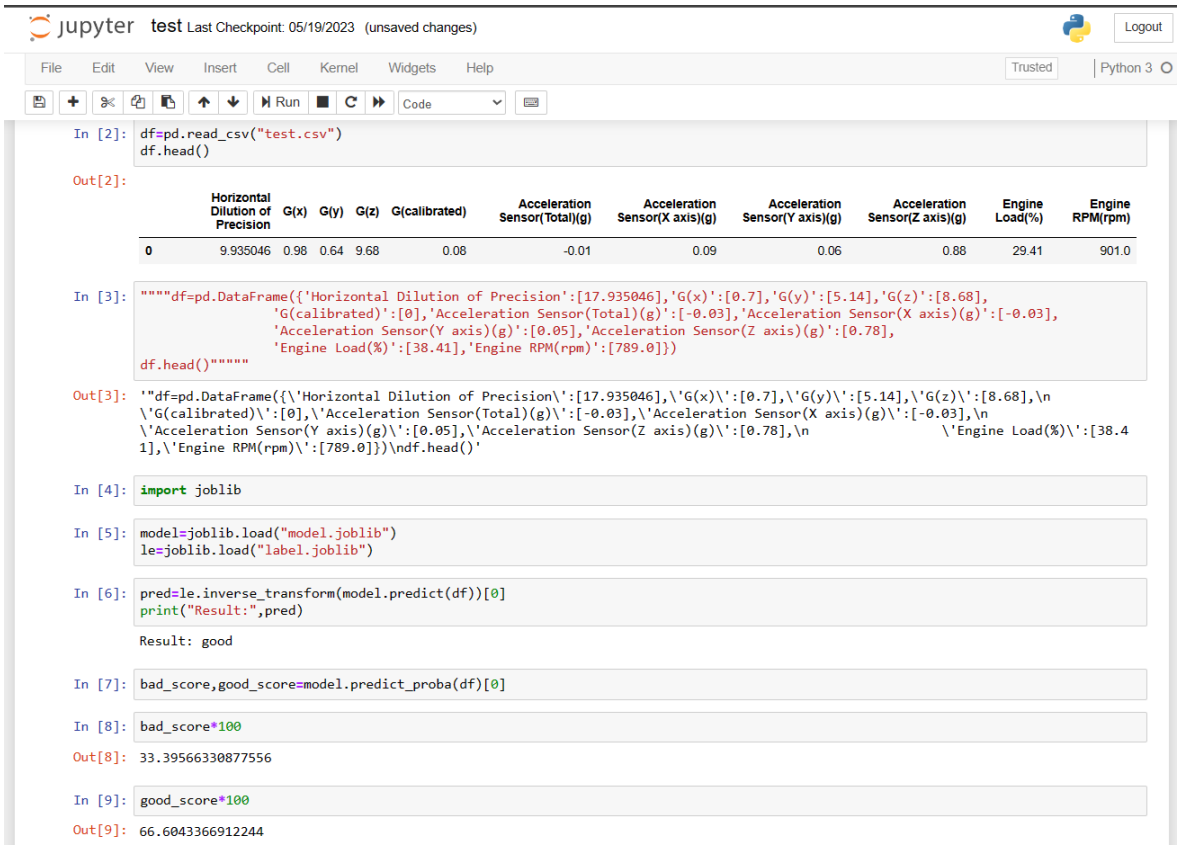

Figure 5.14: Sample case

The driving scores played a vital role in determining insurance pay amounts and claim calculations. Policyholders with higher driving scores received lower insurance premiums, incentivizing safe driving practices. This approach encourages positive behavior change and promotes road safety.

The integration of Hyperledger Fabric blockchain ensured the secure and transparent storage of driving scores. Both the insurance company and the MVD were able to access and view the scores, enabling efficient assessment of risk factors and regulatory compliance. The use of blockchain technology guarantees the integrity and immutability of the driving score data, enhancing trust and reliability within the insurance system.

Overall, the experimental results demonstrate the effectiveness of the implemented vehicle insurance system. By leveraging data collection, machine learning, accurate score prediction, and blockchain integration, the system provides fair insurance premiums, promotes safe driving practices, and ensures transparency and accountability. The results suggest that such a system can effectively enhance the insurance industry's practices while contributing to improved road safety.

5.4 ACTUAL BUDGET

The budget will be used to purchase an Android tab and an obd reader. The market price of an OBD reader, also known as a diagnostic scanner or scan tool, ranges from 700 to 8500 Indian rupees. For this project, we used one that cost around 1500. It is a vehicle diagnostic device that can be used to read the error memory and data that is recorded on your vehicle systems. A touch-screen mobile device with an Android operating system is known as an Android tab. It's also not a smartphone, though you might be able to use one to make calls via Wi-Fi networks with the correct software and hardware. The market worth ranges from 8,000 to various sums, we use the one which coast around 12500. Then the total amount will be around 14000 Indian rupees.

Table 5.1: Budget Proposal

Sl. No	Components	Cost
1	Android Tab	Rs. 12500
2	OBD	Rs. 1500
	Total	Rs.14000

5.5 TASK ALLOCATION

Table 5.2: Resource-wise Task Allocation

Sl. No	Task	Allocated Resource
1	OBD Data analysis	Aarif Hussain A Nassar
2	Driving Score Calculation	Kiran R
3	Insurance Claimage Systems	Eva Petal Pradeep
4	Blockchain Implementation	Ashlin Robert

Chapter 6

CONCLUSION

Based on driving habits, Driving Score gives a measure of driver risk and safety that promotes consistency and justice. The observation and evaluation of unsafe driving behaviours is made possible by telematics, the sending and receiving of data about vehicle and driver interaction. The capacity to evaluate and rank safety risk can inform safe driving and driver education campaigns and serve as an important tool for commercial insurers in their efforts to reduce losses through proactive safety measures. Due to the difficulty of collecting quantitative data on behaviours and performance, driver safety is frequently neglected. This makes many businesses discover lessons the hard way, leading to higher insurance costs, maintenance costs, and a bad business reputation, demonstrating that driver performance has the greatest impact on operating costs.Every year, the margin of error for driver safety and compliance is smaller. Any violations, mishaps, or claims have an effect on the entire business, necessitating heavier fines, liability, and ultimately increased operational costs.

The algorithm collects driving information from the driver using an OBD terminal, combines the x- and y-axis acceleration changes and behaviour duration of the vehicle's three-axis acceleration sensor to identify abnormal driving behaviours, and then creates a hierarchical driving behaviour indicator system and judgement matrix. A scoring model for driving behaviour is developed using this information. The model combines the driver's driving data, utilises the fraction of anomalous driving behaviour as the evaluation index, and calculates the index weight using both the entropy weight method and the analytic hierarchy process.

Chapter 7

SCOPE OF FUTURE STUDY

More and more appliances now have integrated systems that can store and report streams of data as science and technology develop. To extract, store, and transform data about automobiles and their use, a variety of devices are utilised in the field of telematics. These gadgets' data also include details about the driver's behaviour while driving and other characteristics. The level of risk that a driver will cause an accident in the near future can be predicted using data collected by in-car communications devices (telematics), which are often self-installed into a specific vehicle port.

1 Vehicle Usage Pattern: Data gathered under this category would be relevant to the frequency and timing of the drive. We classify the following as falling within this dimension: time of day, familiarity, average mileage, etc.

2 Speeding Pattern: Details on the speed at which a driver operates a vehicle. The following data would be gathered here: average driving speed, traffic, and congestion.

3 Smoothness of Drive: Data on driving efficiency that reveals how easily someone is driving, such as the quantity of abrupt acceleration events, fuel usage, etc.

4 Driving History: The demographic data and driving history will be strongly correlated with the driver's present driving behaviour.

REFERENCES

[1] Jheng-Syu Jhou; Shi-Huang Chen; Wu-Der Tsay; Mei-Chiao Lai: "The Implementation of OBD-II Vehicle Diagnosis System Integrated with Cloud Computation Technology". 2019 Second In-ternational Conference on Robot, Vision and Signal Processing 12 June 2019

[2] Vangala Praveen Kumar; Kampati Rajesh; Motike Ganesh; Ivaturi Ram Pavan Kumar; SanjayDubey "Overspeeding and Rash Driving Vehicle Detection System" 2019 Texas Instruments IndiaEducators' Conference (TIIEC)17 April 2020

[3] Puntavut Lertpunyavuttikul; Pinyarat Chuenprasertsuk; Sorayut Glomglome "Usage-based In-surance Using IoT Platform" 21st International Computer Science and Engineering Conference(ICSEC) 23 August 2019

[4] Veneta Aleksieva; Hristo Valchanov; Anton Huliyan "Smart Contracts based on Private and Public Blockchains for the Purpose of Insurance Services" International Conference Automatics and Informatics (ICAI) 07 January 2021

[5] Chun Yan; Xindong Wang; Xinhong Liu; Wei Liu; Jiahui Liu "Research on the UBI Car Insurance Rate Determination Model Based on the CNN-HVSVM Algorithm" IEEE Access (Volume: 8) 02 September 2020

[6] Tianshi Liu; Guang Yang; Dong Shi "Construction of Driving Behavior Scoring Model based on OBD Terminal Data Analysis" 2020 5th International Conference on Information Science, Computer Technology and Transportation (ISCTT) 04 March 2021

[7] H. T. D. Samarasinghe; N. A. D. M Herath; H. S. S. Dabare "Vehicle Insurance Policy Document Summarizer,AI Insurance Agent and On-The-Spot Claimer" 2021 6th International Conference for Convergence in Technology (I2CT) 10 May 2021

[8] Veneta Aleksieva; Hristo Valchanov; Anton Huliyan "Implementation of Smart-Contract, Based on Hyperledger Fabric Blockchain" 2020 21st International Symposium on Electrical Apparatus Technologies (SIELA)

[9] Veneta Aleksieva; Hristo Valchanov; Anton Huliyan “Implementation of Smart Contracts based on Hyperledger Fabric Blockchain for the Purpose of Insurance Services”2020 International Conference on Biomedical Innovations and Applications (BIA)

[10] Guo Baicang; Jin Lisheng; Shi Jian; Zhang Shunran “A risky prediction model of driving behaviors: especially for cognitive distracted driving behaviors” 2020 4th CAA International Conference on Vehicular Control and Intelligence (CVCI)

[11] Jingren Chen; Yefu Wu; Haojun Huang; Bing Wu; Guolin Hou “Driving-Data-Driven Platform of Driving Behavior Spectrum for Vehicle Networks”2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)

[12] https://101blockchains.com/introduction-to-blockchain-features/

[13] https://www.quartix.com/blog/vehicle-tracking-driver-scores/

[14]]https://owners.hyundaiusa.com/us/en/resources/blue-link/what-is-a-hyundai-driving-score.html

[15] https://github.com/AkashKabra11/Driver-Behavior-Scoring

[16] https://www.gofleet.com/driver-scorecard-importance/: :text=The

[17] https://dir.indiamart.com/impcat/obd-scanner.html

[18] https://insights.axtria.com/blog/driver-score-quantifying-your-moves

[19] https://www.businesstoday.in/auto/story/now-pay-for-your-motor-insurance-premium-based-on-your-driving-behaviour-as-per-irdai-340545-2022-07-06

[20] https://wiki.torque-bhp.com/view/Main$_{Page}$

APPENDIX

CODE SNIPPETS

GitHub link: https://github.com/ashlinrobert/Vechile-insurance-based-on-driving-score-using-blockchain.git

Driving Score:Training Code

Description:The provided code performs several steps to train a machine learning model for classification using a random forest algorithm. The initial step involves importing the necessary libraries required for data manipulation, preprocessing, model training, and evaluation.

Next, the code reads a dataset from a CSV file and preprocesses it by removing any rows with missing values. The dataset is then split into features and the target variable. The target labels are encoded using a LabelEncoder to convert categorical values into numerical representations. To address class imbalance in the dataset, the minority class is oversampled using RandomOverSampler.

The resampled data is then split into training and testing sets using the train test split function. A Random Forest Classifier is initialized and trained on the training data. The trained model is used to make predictions on the test data, and the accuracy of the predictions is evaluated using the accuracy score function.

Finally, the trained model and the LabelEncoder are saved to disk using the joblib.dump function, enabling future use without retraining. Additionally, a sample test data point is saved to a CSV file for demonstration purposes.

In summary, this code showcases the end-to-end process of reading, preprocessing, balancing, training, evaluating, and saving a random forest classification model, providing a foundation for future predictions on similar datasets.

```
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from imblearn.over_sampling import RandomOverSampler
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
import joblib
```

```
# Read the dataset from the CSV file
df = pd.read_csv("dataset.csv")

# Remove rows with missing values
df.dropna(inplace=True)

# Split the dataset into features (x) and target (y)
x = df.iloc[:, :-1]
y = df.iloc[:, -1:]

# Encode the target labels
le = LabelEncoder()
le.fit(y)
y = le.transform(y)

# Perform oversampling to address class imbalance
ros = RandomOverSampler(random_state=42)
x_res, y_res = ros.fit_resample(x, y)

# Split the resampled data into training and testing sets
x_train, x_test, y_train, y_test =
train_test_split(x_res, y_res, test_size=0.33,random_state=42)

# Train a random forest classifier
clf = RandomForestClassifier(max_depth=2, random_state=0)
clf.fit(x_train, y_train)

# Predict the target labels for the test set
y_pred = clf.predict(x_test)

# Calculate and print the accuracy score
```

```
print("Accuracy score:", accuracy_score(y_test, y_pred))

# Save the trained model and label encoder for future use
joblib.dump(clf, "model.joblib")
joblib.dump(le, "label.joblib")

# Save a sample test data point for demonstration
x_test.iloc[:1, :].to_csv("test.csv", index=False)
```

Driving Score:Dataset Testing

Description: This code snippet performs the following tasks: The provided code involves several steps to generate a driving score prediction using a pre-trained machine learning model.

Firstly, the necessary libraries, such as pandas and joblib, are imported to support the subsequent tasks.

Next, a test dataset is read from a CSV file using the pandas library. The dataset is stored in a DataFrame called ‘df‘, which will be used for making predictions.

Following that, a pre-trained machine learning model and its associated label encoder are loaded using the joblib library. The model and label encoder are stored in separate files and are loaded into variables named ‘model‘ and ‘le‘, respectively.

With the model and label encoder in place, the code proceeds to make predictions on the test dataset. The loaded model is applied to the test dataset using the ‘model.predict‘ function, and the predicted outcome is obtained.

To provide additional information, the code retrieves the probabilities of the predicted outcome being classified as either a ”bad score” or a ”good score”. This is achieved using the ‘model.predict proba‘ function, which returns the probabilities for each class.

Based on the probabilities, the code calculates the driving score by multiplying the probability of a ”good score” by 100 and rounds it to two decimal places using the ‘round‘ function. The resulting driving score represents the predicted score for the given test dataset.

Finally, the driving score is printed as the output of the code execution.

In summary, this code involves loading a test dataset, utilizing a pre-trained machine learning model to make predictions, calculating the driving score based on the predictions,

and printing the driving score as the final result.

```
import pandas as pd
df = pd.read_csv("uploads/test.csv")
df = pd.DataFrame({})
import joblib
model = joblib.load("model.joblib")
le = joblib.load("label.joblib")
pred = le.inverse_transform(model.predict(df))[0]
bad_score, good_score = model.predict_proba(df)[0]
bad_score * 100
g = good_score * 100
out = round(g, 2)
print(out)
```

Smart Contract

Description:This code represents a smart contract written in JavaScript for the Hyperledger Fabric blockchain framework. The smart contract is named ‘VehicleContract‘ and extends the base class ‘Contract‘ provided by the ‘fabric-contract-api‘ module.

The ‘VehicleContract‘ class defines three functions:

1. ‘vehicleExists‘: This function is used to check if a vehicle with a given ‘vehicleId‘ exists on the blockchain ledger. It retrieves the state of the vehicle using the ‘ctx.stub.getState(vehicleId)‘ method and returns a boolean value indicating whether the vehicle exists.

2. ‘createVehicle‘: This function is responsible for creating a new vehicle asset on the blockchain ledger. It takes three parameters: ‘ctx‘ (the context object), ‘vehicleId‘ (the identifier for the vehicle), and ‘value‘ (the value associated with the vehicle). The function creates an asset object using the provided value, converts it to a buffer using ‘JSON.stringify‘, and stores it on the ledger using ‘ctx.stub.putState(vehicleId, buffer)‘.

3. ‘readVehicle‘: This function allows the retrieval of a vehicle asset from the ledger based on its ‘vehicleId‘. It first checks if the vehicle exists by calling the ‘vehicleExists‘ function. If the vehicle does not exist, an error is thrown. Otherwise, it retrieves the state of the vehicle from the ledger using ‘ctx.stub.getState(vehicleId)‘, parses the buffer to JSON,

and returns the asset.

The smart contract is designed to facilitate vehicle-related transactions on the Hyperledger Fabric blockchain network.

```
/*
 * SPDX-License-Identifier: Apache-2.0
 */
'use strict';
const { Contract } = require('fabric-contract-api');
class VehicleContract extends Contract {
    async vehicleExists(ctx, vehicleId) {
        const buffer = await ctx.stub.getState(vehicleId);
        return (!!buffer && buffer.length > 0);
    }
    async createVehicle(ctx, vehicleId, value) {
        const asset = { value };
        const buffer = Buffer.from(JSON.stringify(asset));
        await ctx.stub.putState(vehicleId, buffer);
    }
    async readVehicle(ctx, vehicleId) {
        const exists = await this.vehicleExists(ctx, vehicleId);
        if (!exists) {
            throw new Error(`The vehicle ${vehicleId} does not exist`);
        }
        const buffer = await ctx.stub.getState(vehicleId);
        const asset = JSON.parse(buffer.toString());
        return asset;
    }
}
module.exports = VehicleContract;
```

ACHIEVEMENTS

Participations:

Rinu Rose George,Sreeela Sreedhar,Aarif Hussain A Nassar,Kiran R,Ashlin Robert,Eva Petal Pradeep *"Vehicle Insurance Based On Driving Score Using Blockchain"* Institution of Electronics and Telecommunication Engineers (IETE),New Delhi,India,June 2023

Participations:

Participated in the CSI InApp International Project contest