Securing car data and analytics using blockchain

1st Gokay Saldamli
Computer Engineering Department
San Jose State University
San Jose, CA 95192, USA
gokay.saldamli@sjsu.edu

2nd Kavitha Karunakaran
Computer Engineering Department
San Jose State University
San Jose, CA 95192, USA

3rd Vidya K. Vijaykumar
Computer Engineering Department
San Jose State University
San Jose, CA 95192, USA

4th Weiyang Pan
Computer Engineering Department
San Jose State University
San Jose, CA 95192, USA

5th Siddesh Puttarevaiah
Computer Engineering Department
San Jose State University
San Jose, CA 95192, USA

6th Levent Ertaul
Department of Computer Science
California State University East Bay
Hayward, CA 94542, USA

***Abstract*—Automakers in collaboration with technology industries are swiftly innovating and transforming the automobile industry. The current trend of connected cars relies on retrieving various kinds of vehicular data since there is a huge demand from associated entities included insurance companies, vehicle buyers/sellers and government authorities. Currently, the data collection is done either manual or unsupervised that poses trust, legitimacy and accuracy issues such as duplicated or falsified vehicular data records, tampered safety checks and meddled driving history, etc. Hence, a strong tool that can protect the vehicular data; log the changes for audit purposes and eventually build the trust in the system is necessary. We propose the use of blockchain technology for these needs. The proposed solution involves connecting an IoT module to a car data port to collect rich telemetric data; analyze the driving behaviors on various fronts and storing the outcomes to a blockchain. For various good reasons we use the Ethereum blockchain in this study. However, other blockchain deployments can also be utilized. If adopted by the stakeholders, the proposed solution can provide a trusted, transparent and easily accessible platform to auto buyers/sellers, insurance agencies, vehicle dealers, law enforcements and vehicle history providers.**

***Index Terms*—Car data, Blockchain, Ethereum, Smart contracts, OpenXC device**

I. Introduction

In the modern era, as web content is getting more and more accessible, open and available to everybody, security concerns are rising. Every application should be built securely along with transparency. Agencies, government organizations and public demand for systems that can be trusted to make their data integrity process easier and hassle-free. As data security concern arises, the demand for platforms which can provide high-end security along with full accessibility and scalability increases. The rising demand for such secure systems shifts the focus towards Blockchain implementation [1]–[3]. There are other systems which try to provide security for data but they are either owned by a single authority neither scalable nor auditable. The Blockchain environment overcomes all these drawbacks. Blockchain is distributed, decentralized, scalable and secure. It provides access to all users and tracks any change in data.

Vehicular data [4] is in demand from auto industry players. Preliminary telematics data set produces a plethora of significant information for these entities in determining the value of a vehicle, safety check and maintenance conditions. The auto industry is currently relying on data that is manually entered into a centralized database, managed by either automobile manufacturers or dealerships. The vehicle reports generated from a centralized database authority lacks transparency and trust for users. Two significant areas that create trust issues are (i) Origin/Source of the preliminary dataset. (ii) Management of data in a centralized database. The automotive data tampering such as Odometer rollback is common and therefore is a major concern in auto resale [5].

In this study, we have collected various vehicular data statistics from an OpenXC IoT device mounted in a car module [6]. The device uses ODB-II port in the car to extract rich dataset comprising of vehicle speed, engine speed, brake status, GPS location, etc., which is communicated to a phone/tablet using Bluetooth technology. This data is analyzed for safety checks and driving patterns which present information to buyers and various auto industry sectors that assist in decision making. Logic for deriving these patterns is determined by historic data and therefore provides solid foundation and better data visualization. Our work thus justifies authenticity of the source data through data collection from IoT device and integrity of the data by performing chained hash using SHA 256 algorithm on distributed data nodes. This constructs an immutable, transparent database and cater to decentralized way for accessing the data from blocks. This also renders a means for audit and validation in case of falsification and manipulation of reports by any associated entity. As the data is directly extracted from the car module in real-time, we can ensure that preliminary dataset obtained is from the verified source and not been tampered by any factor.

II. Related Work

A. Driving and Maintenance data References

This work starts with car data as an important aspect, making Car Data Secure, transparent and trustworthy. We make use of this car data to derive and analyze, get to a conclusion about the car driver and status of the car. Data such as coolant temperature, brake pedal position, Odometer readings, and many such parameters play a vital role in data analysis. The important stepping stone to start the work was to get data from automobiles in large and accurate sets. After going through several papers mentioned in the references we came across an OBD-II (On-Board Diagnostics) device called OpenXC [6]. OpenXC with OBD version two helped us capture the car data and related parameters using the CAN buses in the cars. The OpenXC device does not provide all parameters to get Car Maintenance Rating, parameters such as Engine Coolant Temperature, Engine Oil Temperature and many more. Hence, for deriving maintenance rating, we had to get test data from the online resource. We found it in Kaggle [4]. We have derived the maintenance rating using the test data set.

B. Data Analysis Methodology

There is a substantial work discussing how different risky driving behavior, such as speeding, hard accelerating, and drowsy driving, increase the possibility of car accident and injury (see [7]–[11]). According to these connections, we generated formulas that take all these aspects into consideration and finally get a score for a driving behavior. This offers a quantitative measuring to represent a risk level in driving and helps the industry of car insurance and car resale.

C. Blockchain and Automotive data

There have been plans proposed to deal with the security and privacy threats in the automotive industry [12], [13]. The approach is to use the blockchain smart contracts to secure the data and communication between automotive vehicles. There are not enough successful use cases that deal with automotive data tampering beyond some proof of concepts. Several other papers have been published which focus on Blockchain technology but in distributed ledger [14], [15].

The latest article came up is about the rising popularity of the blockchains in the leading automobile market [15]. Car companies like Renault have launched a system using the blockchain of piloting digitized car maintenance programs which has all the car and maintenance history securely written. This is for the auto indication to the drivers for maintenance. Tesla also trying to come up with something similar where the tolls have been paid automatically as the car passes through the toll booths. So these are some of the announcements made by leading automobile companies to explore the Blockchain platform to make their system more secure and private, though there are no concrete results yet. The data tampering like Odometer, maintenance issues are also only in consideration but nothing has been published or modeled yet.

Vinchain [16] provides a similar idea discussed in this work, where blockchain passport and reports are issued to users by checking across car VIN (Vehicle Identification Number). Vinchain uses graphene as its Blockchain platform. It aims to provide a single Blockchain database for all automobile information. It also uses data sourcing from members and VIN tokens for client payments. Carblock [17] also provides a similar idea but emphasizes on users controlling the data sharing process and are rewarded with tokens for valid data. It is a mutually beneficial system.

D. Odometer Fraud

Current statistics shows that many vehicles in market undergo Odometer fraud and as a result people lose trust in used car market. This mileage fraud especially is hard to detect but there are efforts being made to detect them. A vehicle with a history of damage or and mileage rollback represents an increased risk for the buyer which directly affects the residual value, price, functionality, durability of the vehicle and even the public road safety. CARFAX [5] provides list of tips to find out the Odometer fraud. But the insecure and centralized nature of these systems run by a single authority increases risks of data misuse. In European countries like Germany, the statistics says every third car in the market is subjected to Odometer fraud which eventually leads to a huge capital loss every year. Investigating mileage rollback involves verifying previous maintenance records, which is expensive.

Many other countries are trying to come up with some diagnosis tool to detect the Odometer clocking or spinning but no product is 100% accurate yet. Automobile companies came up with Digital Odometer to overcome this risk but even they are not safe anymore. The digital odometer correction tools can be easily used to clock them. This kind of fraud not only costs more money than the actual worth during purchase but can also indicate the poor car condition which can lead to various other repairs and maintenance issues in the near future. Therefore, it is most important requirement to come up with something which resolves this concerning issue. Securing odometer reading in Blockchain can be a solution, as it records all the changes in reading and any tampering can easily be caught.

III. ARCHITECTURE

The system architecture consists of several operational modules including:

- Application Users
- Car Maintenance Authority
- Driver behavior rating Authority
- End Users.

All entities interact with Blockchain application through a Web User Interface (UI). The front-end application is built using React JS with Ethereum Blockchain as backend.

A. Application Users

The role of application users is to register their vehicle data through Web UI. The vehicle data consists of Car VIN as key and other details as value(Car model, License Plate, Car Owner, OpenXC device registered with car). The Car Owner field is automatically registered through metamask address of logged-in individual. Maintenance, Driver Behavior and odometer records are preset to zero.

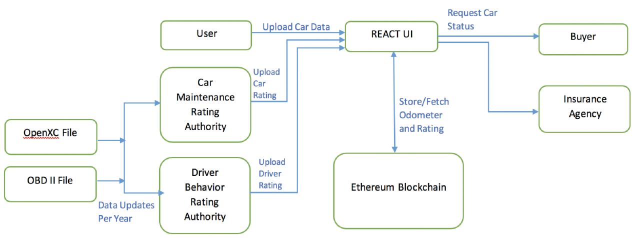

Fig. 1. System Architecture.

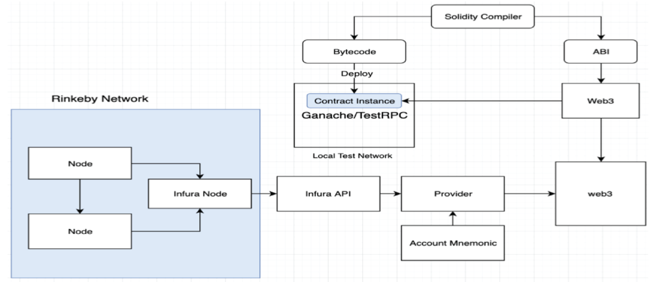

Fig. 2. Blockchain Backend design.

B. Car Maintenance Authority / Driving Behavior rating Authority

The role of these authorities is to derive maintenance rating and driving behavior rating through the OBD II data and OpenXC trace files from OpenXC device. Latest Odometer value is also obtained from the OpenXC device and updated through Web UI. Only authorized accounts are permitted to update maintenance rating/behavior rating/Odometer reading respectively using car VIN.

C. End Users

Application end users can retrieve a particular car records by keying-in VIN. Anyone can view the record if VIN is known. End users can be Insurance Agency, Car Buyer, Government agency etc. This design helps used-car buyers to find out a car's condition and its value. The ratings and latest Odometer are updated by respective authorities from time to time, which is an input to the blockchain through the web application. This data is then written to the nodes of a blockchain which ensures decentralized security. Whenever an end user requests for the ratings and the latest Odometer reading it can be fetched from the blockchain through the front end application and provided to the requester.

IV. System Design

A. Blockchain Backend Design

The backend design mainly consists of designing and writing smart contract. Once contract is written we can deploy our contract into any of the Ethereum networks: Main network, Test networks (Ropsten, Kovan, Rinkeby), local-host 8245 and custom RPC. Our choice of Ethereum network is Rinkeby Test Network. This network is used for hosting test applications. Any transactions with the network such as storing records and retrieving data is associated with spending gas and ethers. As transactions costs ether, we have ensured data correctness through rigid testing in remix and individual test units before deploying the contract. The choice of programming language to write a smart contract is Solidity. Remix is an in-browser IDE which is used for developing, testing and deploying a smart contract.

The Car contract consists of car related parameters is the form of data structures and various functions to update these parameters. The environment setup involves installing below dependencies

Solidity Compiler - Compiles the .sol file into ABI (Application Binary Interface) and byte code.

- **ABI:** To interact with rest of the JavaScript code.
- **Web3 Library:** A library which is used to establish communication between JavaScript and Ethereum network.
- **Ganache CLI and Mocha:** Libraries for testing Ethereum projects.
- **Infura API:** Infura provides the API to access node that is required for deploying a smart contract into Blockchain network.
- **Metamask:** A Google chrome plugin to create accounts(associated with private keys, 12-word mnemonic). Accounts are used in connecting to Ethereum Network.

B. React JS Front End Design

React JS is the technology used to create the Front-end web UI for our Ethereum car application. In traditional web services, Server is responsible for all data interaction between the user and the backend database. But in Ethereum architecture the role of server is minimal. React JS provides extensive libraries for simpler and easier Web UI development. The application is built using JavaScript code to interact with Ethereum network. It focuses on various functionalities starting with the status of the contract and then calls other backend functions from contract ABI such as, register car, update rating and retrieve data from Blockchain.

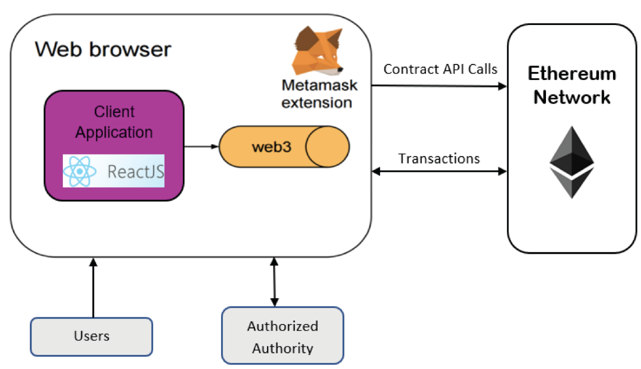

Fig. 3. Front End for Ethereum Architecture.

Web3 is the library which make these interactions possible. Metamask plugin for google chrome browser injects the web3 library into any active page. Metamask uses the web3 v0.20

which is older version and bit complicated. This version is injected on to the page and points to the test network. In this work, we are using web3 v1.0 by manually setting up configuration. Once Metamask, web3 dependencies are installed successfully, React front end application has been created and functionalities are added to write/fetch to/from Ethereum Blockchain. The ReactJS code is written to send and call the functions to the blockchain backend with passing appropriate parameters retrieved from UI web forms.

V. Implementation

A. OpenXC Device

Primarily, we need to collect data from several sensors from the vehicles. We are using a device called OpenXC which performs this activity. OpenXC is an open source platform that provides car sensor data using On-Board Diagnostics(OBDII) [6]. OBD is mandatory to be installed on all cars that are sold in the US after 2008. OpenXC vendor provides a device that can be plugged in the car OBD II port, usually located underneath car dashboard. This device will connect to a mobile phone or tablet via Bluetooth or USB and we have an option of storing the data locally on any connected device or on cloud [6]. The standard data parameters that are provided by OpenXC device are listed below. We have used most of these parameters to derive analysis ratings. Angle on the Steering Wheel, Transmissions Torque, Engine Speed, Vehicle Speed, Accelerator - Position on the pedal, Parking Brake Status, Status on Brake pedal, Transmission - Position of the gear, Odometer, Latitude, Longitude, Button Event, Wiper Status on the Windshield, High Beam Status, Headlamp Status, Fuel Level, Ignition Status.

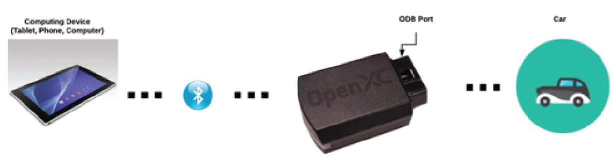

Fig. 4. Infrastructure for Data Collection.

Initially, we connect the OpenXC device to the car OBD-II interface. Once the device is connected we pair the device with our phone Bluetooth. Our phone should have an application from OpenXC available in Play Store and for Apple Devices is a tricky procedure. We do get assistance in connecting our phone to the OpenXC device using the application on the phone. After we get our phones connected, we can store the data on the phone locally or upload it to the cloud using URL, option is known as Target URL on the application, we do have option to send data to Dweet.io. We transferred the data locally from our phones to the blockchain. All the data we get is in JSON format. Fig. V-A shows us the sample data format and we have highlighted some of the important parameters that we get of the car using the OpenXC device.


{"value":1548.25,"name":"engine_speed","mKey":{"mParts":{"name":"engine_speed"},"mHashCode":1191599776},"timestamp":1446861361.191}
{"value":9.019608,"name":"engine_load","mKey":{"mParts":{"name":"engine_load"},"mHashCode":1699931751},"timestamp":1446861361.192}
{"value":70.0,"name":"vehicle_speed","mKey":{"mParts":{"name":"vehicle_speed"},"mHashCode":1497406494},"timestamp":1446861361.194}
{"value":1549.25,"name":"engine_speed","mKey":{"mParts":{"name":"engine_speed"},"mHashCode":1191599776},"timestamp":1446861361.338}
{"value":9.019608,"name":"engine_load","mKey":{"mParts":{"name":"engine_load"},"mHashCode":1699931751},"timestamp":1446861361.372}
{"value":70.0,"name":"vehicle_speed","mKey":{"mParts":{"name":"vehicle_speed"},"mHashCode":1497406494},"timestamp":1446861361.374}
{"value":18.0,"name":"intake_manifold_pressure","mKey":{"mParts":{"name":"intake_manifold_pressure"},"mHashCode":1159743017},"timestamp":1446861361.552}
{"value":1548.25,"name":"engine_speed","mKey":{"mParts":{"name":"engine_speed"},"mHashCode":1191599776},"timestamp":1446861361.554}
{"value":9.411765,"name":"engine_load","mKey":{"mParts":{"name":"engine_load"},"mHashCode":1699931751},"timestamp":1446861361.557}
{"value":70.0,"name":"vehicle_speed","mKey":{"mParts":{"name":"vehicle_speed"},"mHashCode":1497406494},"timestamp":1446861361.558}
{"value":84.0,"name":"engine_coolant_temperature","mKey":{"mParts":{"name":"engine_coolant_temperature"},"mHashCode":396491228},"timestamp":1446861361.732}
{"value":1545.0,"name":"engine_speed","mKey":{"mParts":{"name":"engine_speed"},"mHashCode":1191599776},"timestamp":1446861361.736}
{"value":9.803922,"name":"engine_load","mKey":{"mParts":{"name":"engine_load"},"mHashCode":1699931751},"timestamp":1446861361.737}
{"value":70.0,"name":"vehicle_speed","mKey":{"mParts":{"name":"vehicle_speed"},"mHashCode":1497406494},"timestamp":1446861361.792}
{"value":1540.0,"name":"engine_speed","mKey":{"mParts":{"name":"engine_speed"},"mHashCode":1191599776},"timestamp":1446861361.938}
{"value":12.941176,"name":"engine_load","mKey":{"mParts":{"name":"engine_load"},"mHashCode":1699931751},"timestamp":1446861361.973}
{"value":70.0,"name":"vehicle_speed","mKey":{"mParts":{"name":"vehicle_speed"},"mHashCode":1497406494},"timestamp":1446861361.974}
{"value":1541.25,"name":"engine_speed","mKey":{"mParts":{"name":"engine_speed"},"mHashCode":1191599776},"timestamp":1446861362.153}
{"value":14.117647,"name":"engine_load","mKey":{"mParts":{"name":"engine_load"},"mHashCode":1699931751},"timestamp":1446861362.155}

Fig. 5. Sample data set from OpenXC device.

B. Data Analysis

Data collected by the OpenXC devices will be used to generate two ratings: driving behavior rating and maintenance rating.

We have worked on data analysis to generate a rating for driving behavior and maintenance performance of the car. For driving behavior rating, speeding and hard reversing is easy to define. However, to define and monitor hard braking and hard accelerating, and generate a numerical rating, we referred to some papers discussing hard braking and hard accelerating behaviors. After researching and comparing with these papers, we decided to calculate the hard speeding rating as the ratio between accelerator pedal percentage to the vehicle speed. That means, step on the accelerator pedal hard when the vehicle speed is low will be considered as a hard speeding behavior. For hard braking, we will just calculate the speed change over time.

1) Driving Behavior: For driving behavior rating, it contains four parts: Speeding, Hard Braking, Hard Accelerating, and Driving Time. For speeding, we collect the vehicle speed and the speed limit at different highways, and deduct one rating score for every ten miles per hour above the speed limit. For Hard Accelerating, we monitor if the pedal is stepped hard when vehicle speed is relatively low. The Hard Accelerating score will be deducted one point for each 5% difference between pedal percentage and vehicle speed. For hard braking, we calculated the acceleration during each braking. If its larger than 2 mile/h*s, we will deduct one point for each 1 mile/h*s larger. For driving time, we consider continuously driving for more than 4 hour as drowsy driving, and will deduct one point for each one hour extended. Finally, we calculate the mean average of these four fields to get the overall driving behavior score for data collected during one driving session.

We implemented the data analytics tool using python. The steps in the implementation flow chart are:

1. Collect data from OpenXC device.
2. Parse the data as json using the python json library.
3. Take out only data fields that are needed.
4. Calculate the four driving behavior ratings: speedingRating calculated from vehicle speed data, acceleratingRating from acceleratorPedalPercentage, brakeRating from vehicle acceleration, drivingTimeRating from drivingTimeHour.

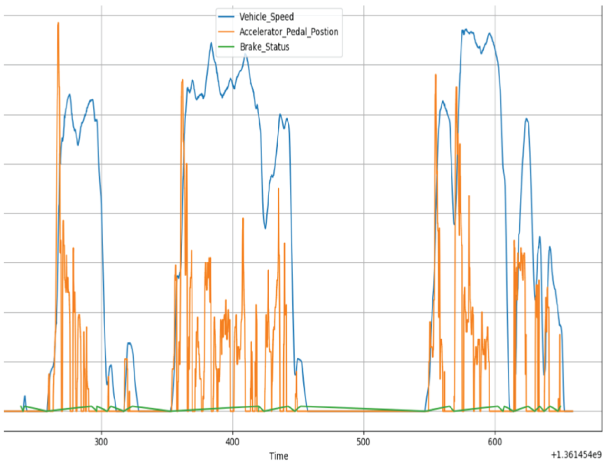

Fig. 6. Data Analysis.

5. The average of the four ratings is the overall driving behavior rating score.

2) Car Maintenance: Car Maintenance is one of the important parameters to be considered while buying/selling a used car. Users buying the used car need to know how the previous owners have maintained the car to trust it and put money on it. Similarly, a seller selling his car can get a better price or can sell it for a better price if the car is genuinely maintained well. Hence, Car Maintenance data or rating plays a crucial role in deciding for a buyer to trust the car and seller to get more out of it than a poorly maintained car. We have derived such a rating from the car Data. We did get some parameters from the OpenXC device data set and most of them were from online test data sites such as Kaggle [4] etc. These ratings can be updated in real-time and provide current car owner an indication to check-in the car for maintenance.

We have considered many parameters and calculated rating for maintenance for a Car, the first parameter is Engine Coolant Temperature. A well-maintained car Engine Coolant temperature should always be between 195 to 220 degrees Celsius. If a car is poorly maintained and if the temperature is higher than normal during the car operation, it damages the engine and lowers the efficiency of a car. This could easily affect the life of the cars engine. The second parameter is Engine Oil Temperature, this parameter is also crucial in deciding the life of an engine and car efficiency. A car owner should make sure that engine oil temperature should be at least 220-degree Fahrenheit, to burn of all deposits and accumulated water in the chambers during combustion. Every pound of fuel-burning will create a pound of water, hence all water should be evaporated or will accumulate in the chambers mixing with sulfur to create Sulfuric acid that will damage the bearings. The third parameter that we have considered is known as the Manifold Pressure Sensor parameter, cars are driven in different atmospheric conditions. Like in higher altitudes, hot and cold conditions. When the car is driven in higher altitudes atmosphere is thin and has less oxygen and hence to provide the same power to the wheels we need more gas to be burnt in the chambers. Hence, more gas needs to be provided for every stroke. Manifold Pressure Sensor works with Barometric Sensor to open up the gas valve accordingly to provide the same power to car wheels during all conditions. No wonder why more gas is utilized at higher altitudes. If the driver experiences less power or throughput from the engine during different conditions driver can notice or can be notified to get a maintenance check on the car. Several other parameters can be considered Fuel Pressure, Absolute Barometric Pressure to derive a much more inclusive Maintenance rating of a car.

3) Odometer fraud check: [8] Odometer rolling is a serious concern but can be tackled easily by extracting odometer reading from the OpenXC device in real-time. We have acquired the latest odometer rating by running a python script by giving OpenXC file as input. This reading is then pushed to Blockchain backend. The smart contract in Ethereum ensures to verify the current odometer reading entered be greater than previous reading else the transaction would fail. All changes to odometer reading can be audited at any time. The login address making changes to the odometer reading is also maintained. This provides security and prevents tampering of data.

C. Blockchain Backend Development

The application backend is implemented as a smart contract and is deployed on the Rinkeby Ethereum Network. The Car smart contract consists of a key-value mapping for car VIN (Vehicle Identification Number) to its parameters such as car owner, car model, odometer reading, maintenance and driving behavior records and functions to update the parameters Solidity compiler produces ABI which is used to interact with UI through Web3 library.

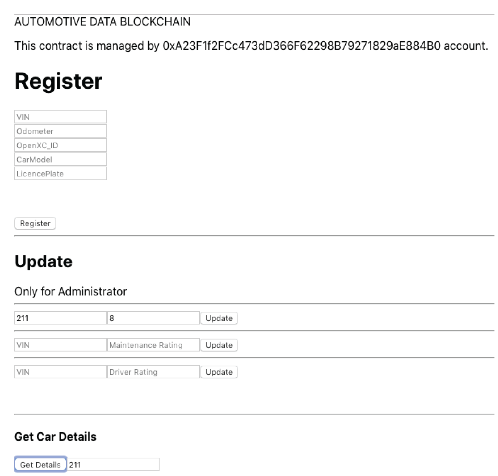

Fig. 7. Web UI for Application.

Testing and Deployment: Remix and Ganache CLI are used for testing and verification of the car smart contract. Ganache provides a simulated Ethereum network for performing test cases. Testing involves registering a car, verifying if car owners and related records are consistent and properly updated.

Finally, deploying the car contract connects to the Infura node through a meta mask account using web3 and truffle library.

D. Frontend Development

The application user is facilitated by interactive Web UI as shown in Fig. V-C to register his/her car into the Blockchain. Various functionalities such as updating the records are performed through the UI, however, only authorized accounts are allowed to update driving behavior, odometer and maintenance records. Anyone can view car records stored in Ethereum backend by keying in-car VIN.

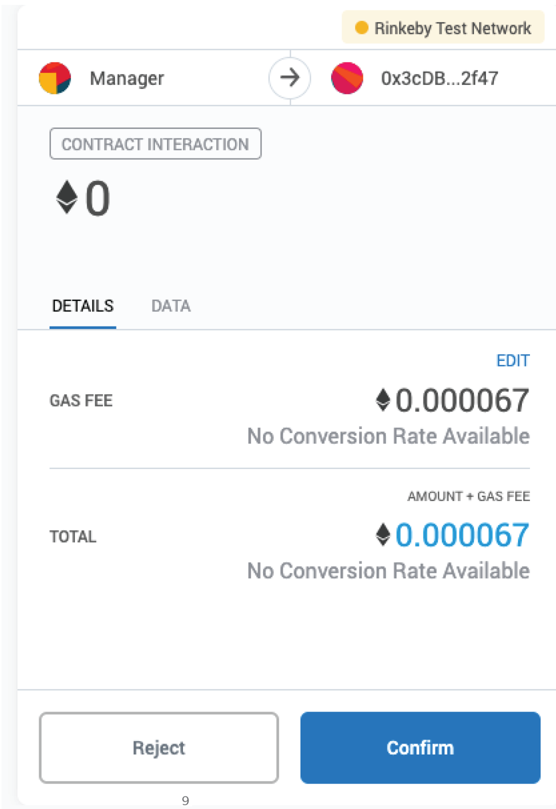

Fig. 8. Blockchain Metamask Transaction.

Frontend web application is implemented using the React JS framework, which renders the web UI and runs javascript code. Actions in the form triggers event which connects to the metamask account through injected web3 library. The web application front end is built by rendering 3 HTML forms for users and management authority to register and update important data. The blockchain smart contract ensures if the authorized authority is updating the records otherwise, the transaction is reverted back. Fig. V-C shows the Web UI and input fields.

VI. Results and Evaluation

Compared to the other implementations [14], our work incorporates ideas that are more secure, trustworthy and is beneficial in many ways for end users of the application. Fig.

TABLE I
Features and Advantages

Features	Advantages
OpenXC device data	Real-time data for relevant driving behavior and maintenance ratings ensure secure data source
Data analytics analytics	Rather than just providing car reports from historic data source provides extra realistic analysis
Blockchain	Any changes to data can be audited and tracked in case of inconsistencies
De-centralize	Records are not controlled by a centralized authority case of inconsistencies
Odometer fraud check	Real-time data from OpenXC device and blockchain backend can detect Odometer frauds

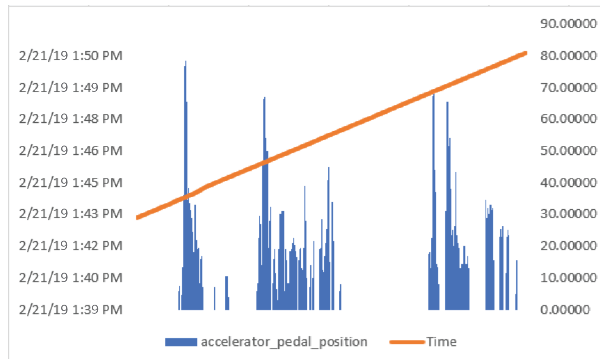

Fig. 9. Accelerator Pedal Position.

VI provides the features and advantages of using the devices and applications in this study.

The automotive data have been critically analyzed with different sets of real-time data and testing data available. This includes aggressive driving data, speeding data, and some live data from real-time drivers. The data has been analyzed with respect to various safety standards of road and the vehicle to derive the driver behavior and car maintenance. Figs. VI and VI are graphs which indicate some of the parameters considered for deriving the ratings.

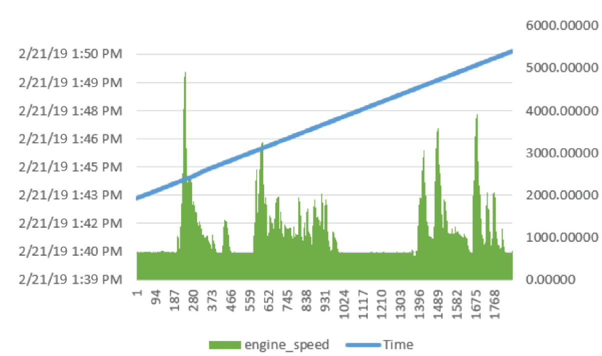

Fig. 10. Engine Speed.

Fig. VI shows the graph of the accelerator pedal position to time, the spikes in the graph indicates hard accelerating driving behaviors. We can also see similar spikes in Fig. VI, as engine speed increases sharply when the driver hard accelerated. The sharper the spikes are considering both the graphs, the more driving behavior rating score we deduct.

VII. Future work

There is plenty of room for future work and optimizations. During the data collection from the OpenXC device, we have collected all data locally from the application. We stored the data on the phone and transferred it to the local system. We can make this process much secure by uploading the data directly to the cloud. A microservice can be run to parse data collected on the cloud and derive analytics. This makes the process tamper-free. If we get enough data sets to train a Machine Learning Model, these computations can predict the car maintenance driving behavior in real-time. We should be able to predict factors responsible for accidents based on the vehicles real-time trace files. For example, the Machine learning model can consider parameters like duration of continuous driving hours by a driver in a particular day, road conditions, vehicle maintenance ratings, traffic and time of the day to predict drivers rating. This kind of prediction can be further used by insurance agencies to improve and strategize on insurance claims. Depending on the Engine oil Temperature and Manifold pressure readings models can predict the life of the vehicle and time at which vehicle may need maintenance in the future.

VIII. Conclusion

The latest statistics show the importance of automotive data in designing applications for various needs of automobile market. One significant demand is bringing trust in vehicle history records and avoiding inconsistencies in records maintained by various parties. The Blockchain approach in this work tries to overcome these limitations along with analyzing data and computing ratings. This approach not only provides security but also allows an easy accessibility and maintenance of records. The property of immutability brings accountability and authenticity to the system. The implementation of the proposed solution, helps automobile resale industry along with agencies like Insurance and road safety regulators. If ratings, odometer or any records are falsified, it can be easily noticed and the source of invalid data can be tracked. However, extreme verification steps are necessary as the database is immutable. In conclusion, the approach will be beneficial to customers, who lack trust in present centralized authority.

References

[1] T. Salman and M. Zolanvari and A. Erbad and R. Jain and M. Samaka, “Security services using blockchains: A state of the art survey,” *IEEE Communications Surveys Tutorials*, vol. 21, no. 1, pp. 858–880, 2019.

[2] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, “On blockchain and its integration with iot. challenges and opportunities,” *Future Generation Computer Systems*, vol. 88, pp. 173 – 190, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X17329205

[3] T. M. Fernández-Caramés and P. Fraga-Lamas, “A Review on the Use of Blockchain for the Internet of Things,” *IEEE Access*, vol. 6, pp. 32 979–33 001, 2018.

[4] Kaggle.com, “OBD-II datasets,” 2019. [Online]. Available: https://www.kaggle.com/cephasax/obdii-ds3

[5] Carfax, “How to detect mileage rollback?” 2019. [Online]. Available: https://www.carfax.eu/article/mileage-rollback

[6] “Vehicle trace files - openXC,” OpenXC, 2019. [Online]. Available: http://openxcplatform.com/resources/traces.html

[7] G. Morrison and D. Cebon, “Combined emergency braking and turning of articulated heavy vehicles,” *Vehicle System Dynamics*, vol. 55, no. 5, pp. 725–749, 2017. [Online]. Available: https://doi.org/10.1080/00423114.2016.1278077

[8] One.nhtsa.gov, “Research on drowsy driving — national highway traffic safety administration (nhtsa),” 2019. [Online]. Available: https://one.nhtsa.gov/Driving-Safety/Drowsy-Driving/Research-on-Drowsy-Driving

[9] B. Simons-Morton, M. Ouimet, J. Wang, S. Klauer, S. Lee, and T. Dingus, “Hard braking events among novice teenage drivers by passenger characteristics,” *Proceedings of the ... International Driving Symposium on Human Factors in Driver Assessment, Training, and Vehicle Design*, vol. 2009, pp. 236–242, 06 2009.

[10] J. Guo, Y. Liu, L. Zhang, and Y. Wang, “Driving behaviour style study with a hybrid deep learning framework based on gps data,” *Sustainability*, vol. 10, p. 2351, July 2018.

[11] S. Ontan, Y.-C. Lee, S. Snodgrass, F. K. Winston, and A. J. Gonzalez, “Learning to predict driver behavior from observation,” The AAAI 2017 Spring Symposium on Learning from Observation of Humans, Tech. Rep., 2017.

[12] Y. Rahulamathavan, R. C.-W. Phan, M. Rajarajan, S. Misra, and A. Kondoz, “Privacy-preserving blockchain based iot ecosystem using attribute-based encryption,” in *2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*. IEEE, Dec 2017. [Online]. Available: http://dx.doi.org/10.1109/ANTS.2017.8384164

[13] Q. He, Y. Xu, Z. Liu, J. He, Y. Sun, and R. Zhang, “A privacy-preserving internet of things device management scheme based on blockchain,” *International Journal of Distributed Sensor Networks*, vol. 14, no. 11, 2018.

[14] Carvertical, “First global decentralized car history registry. built on blockchain technology,” 2019. [Online]. Available: carvertical.com

[15] Hackernoon.com, “Blockchain writing a new chapter for automotive industry,” 2019. [Online]. Available: https://hackernoon.com/blockchain-writing-a-new-chapter-for-automotive-industry-48a8151eec99

[16] Vinchain, “Vinchain decentralized vehicle data,” White paper, 2019. [Online]. Available: https://vinchain.io/files/VinChainWhitePaperEn.pdf

[17] Shopify, “A global transportation data protocol with decentralized applications,” White paper, 2019. [Online]. Available: https://cdn.shopify.com/s/files/1/0703/1149/files/whitepaper_en_v4.0.pdf