ASHESI UNIVERSITY

IMPLEMENTING A PRIVACY BY DESIGN FEATURE IN AN IOT CONTROLLED INSURANCE SYSTEM.

THESIS PROJECT

B.Sc. Computer Science

HODO A. S.

2021

Table of Contents

CHAPTER 1: INTRODUCTION

Internet of Things (IoT) is a term used to describe the interconnectivity of smart objects(things) that collect data, process the data, communicate with each other, and actuate other objects in the environment. The integration of the Internet of Things into various sectors is improving lives daily. An example of such an industry is the insurance sector. Intelligent Transportation Systems (ITS), Usage-Based Insurance(UBI) etc., are a few examples of new trends in the car insurance business[1]. UBI calculates insurance premiums based on sensor data collected from OBD systems and is used for a more personalized policy. Its main benefits are reduced car accidents, lower premiums, and better risk calculation, saves cost for companies and a lot more.[2]

The major problem with the UBI and other IoT controlled insurance systems is the user privacy concerns[2], [3]. Most IoT systems that involve humans lack robust user privacy features that can ensure users' safety and privacy. Even though IoT is based on constrained networks and systems, it is fundamentally still information Technology(IT). All the risks and attack scenarios in the past concerning IT must be reconsidered in these systems[4]. The main challenge is the privacy aspect because IoT does not present the same interfaces or protocols to the typical IT systems. Users cannot precisely tell what data is being taken from them and have little or no say in it after the systems have been deployed, and there is currently no implementation of privacy by design in the IoT insurance systems.

Ethical issues about the IoT controlled environment might prevent the adoption of this technology in many countries. Users might also not patronize such systems if robust privacy profiles are not created for them because they would not want the general public, government agencies or insurance companies having their location and other personal data[2].
With the emergence of digital laws and ethical regulations such as the GDPR, which is the core of Europe's digital privacy legislation, such rights and other ethical considerations must be factored in when developing IoT systems. When we consider section 3 of the GDPR that talks about human Rights to Rectification and Erasure[5], most or all IoT systems do not have robust features to exercise such rights to their full capabilities, and this is significant. This research also is motivated by the Arm AI manifesto that provides guidelines for working with AI and user data and provides frameworks for engineers to use and take responsibility for user privacy and AI ethical issues[6]. Suppose IoT system designers don't begin to include such features. In that case, there is the risk of IoT systems not being adopted in most parts of the world as other governments are also coming up with similar Digital Rules.

This research aims to find out how we can implement privacy by design in an IoT controlled insurance system and find out the effects that privacy by design will have on such a system. This will be done by borrowing the website design concept that allows users to get access to certain features of websites even if they did not accept cookies and the users' data is not stored. This research will also find efficient machine learning algorithms that can be used on sensor data from the vehicle that does not directly relate to the user or can be used against the user. This will be done by determining driver dependent data and non-driver dependent data so as to provide users with different privacy and control options.

The **objectives** will be addressed by the following questions in the research;

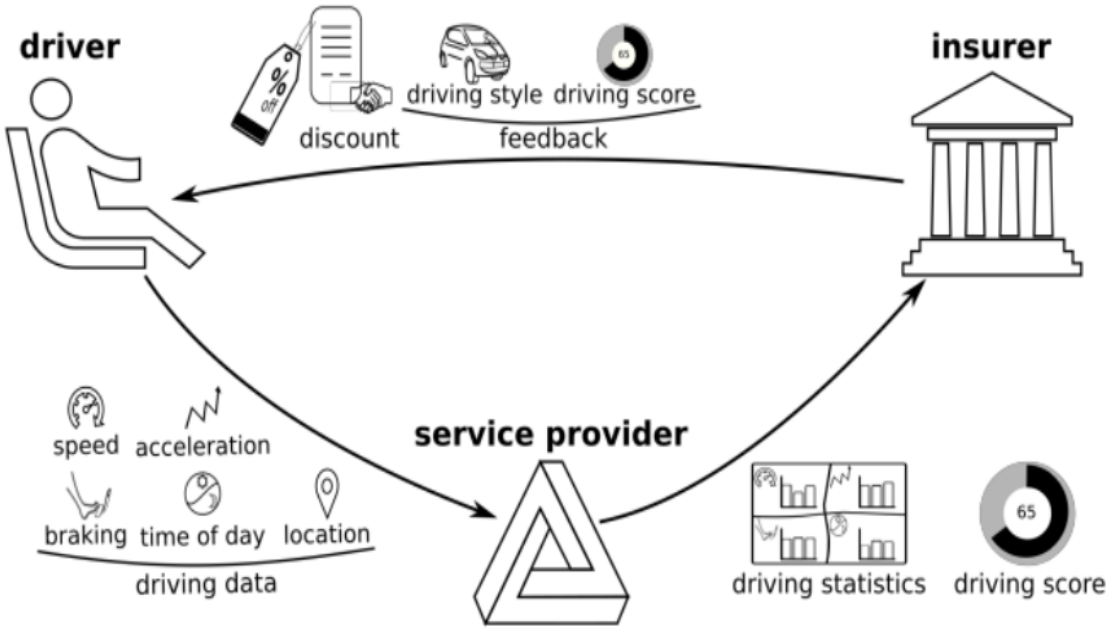
1. What are the mechanisms for implementing a privacy by design in an IoT controlled system?
2. What is the effect of implementing privacy by design on the accuracy of classifying a user as risky or not risky?

CHATPER 2: LITERATURE REVIEW

Research in the field of IoT controlled Insurance has been conducted over the years. It has mainly taken different forms, ranging from Usage-Based Insurance, Intelligent Transportation System, Driver Behavior Recognition etc. [2], [7]. One interesting aspect is that, aside from proving concepts and theories, no user privacy model has been implemented yet.

Advancement in UBI has provided different theories on how to mitigate the privacy issues concerned with the general model that collects data, transfers it to a service provider, which then calculates the driving score and sends it to the insurance companies. The general solution tried to mitigate the issues about privacy by allowing the service provider to handle all the data and only send statistics and a score to the insurance company. The general model has lots of benefits; it improves driving behaviour over time, allows the company to save money due to better risk calculation, etc. [2]. That is a good step, but now the GDPR, Ghana Data Protection Act(DPA) and others that are being developed by countries state that it is not enough and makes us responsible for the user's data protection rights. Users cannot control their data flow; they sometimes do not have access to their own data and cannot access their rights to be forgotten[5].

The diagram below shows the general model. Solutions have been proposed to the problem of the general model but have not been implemented yet. [2], [8] Researchers have proposed that the data be processed locally in the vehicle and aggregate data sent to the insurer for premium calculation.[8]proposed a privacy-enhancing architecture called PEAR's which highlights the



importance of design architecture in developing privacy by design solutions.

In terms of classifying the data gotten from the sensors, [7] In one research, car sensor data was collected to find which data best explained the driver behaviour based on the car's performance. Over 50 sensor data was collected and tested with various algorithms.

Popular machine learning models were tested with the data. From the testing done, the top 3 were Random Forests, Decision Tree and Gradient Boosting with over 95%, with Random Forests being the highest reaching 97.5%, the others performed poorly on the kind of vehicular data provided. The top 10 sensor data were chosen based on high correlation. More tests were done, and the top 3 sensor data that was representative of the drivers' performance and was difficult to manipulate by car owners were identified [7]. The research did not consider user

involvement and how the algorithm would act if users decided to stop the flow of data for some time or chose to hold on to some sensor data.

In this research, the main aim is to protect the users' data. From the literature review and much research, if this research implements a way to allow users control over the data, there would be a great tradeoff between privacy and efficiency, hence affecting the program's usefulness if users do not allow data to be sent. At the core, if a model is created such that specific data needs to be collected and the user chooses not to allow the flow of such data, then the system would not function. In that regard, the research solves this issue by borrowing the logic from websites that use cookies. This analogy was used; "If a user chooses to use the system and give out all data necessary, then the program works. However, if the user does not choose to give out sensitive data, then the program should still work based on general data that does not relate to the user directly or cannot be used to deduce the users' personal data." Hence the program would not compromise so much on efficiency since it should provide a fair idea of the users driving behaviour indirectly and still protects the user's privacy.

CHAPTER 3: METHODOLOGY

For the research, a prototype system was built to implement and validate the concept of privacy by design. Machine learning algorithms were tested with the data. The data was also paused at different times to simulate the users stopping data flow, and the effects were studied.

ARCHITECTURAL DESIGN

The standard three (3) layered IoT architecture was used in this study. Ie.

The IoT device later. This consists of the client-side. Sensors and actuators.

The IoT gateway layer. The operations on the server-side. Pre-processing and aggregation.

The IoT platform layer. Connecting clients and operations as well as further analysis.[9]

IoT architecture is also divided into four stages. These are; a. The sensor and actuator stage. b. Data acquisition stage, c. Edge IT and d. Data Center and Cloud. The first two stages will be done by the car On-Board Diagnostics (OBD II) system. The already existing car sensors will be used, and the OBD system will capture the data and present it to a module that is attached to the port. The module does some pre-processing on the data to aggregate and filter the data for security purposes. That forms the IoT device layer. In this research, a car OBD simulator will be used to manage cost and accessibility. The module then sends the aggregated data to the mobile phone of the user via Bluetooth. This is the second layer which consists of the edge IT stage. The App on the phone allows the user to visualize the data, know what data is being sent, and control the flow of the data. The information is then sent to the data centre stage, which falls in the IoT

platform layer. This is where the machine learning algorithms work on the data to calculate the driving score and behaviour category to be presented to the insurance company.
The research is done in 3 parts.

Simulating IoT system. This is the first stage of the study. This is where the OBD simulator was used to generate the data, which was aggregated and sent to the phone. The phone's App made it possible for the user to visualize the data, allow control, select what profile of data they wanted to be sent or stopped the App totally. The internet was used to transfer the data to the cloud for the machine learning algorithms and a final transfer of information to the insurance company.

Sensor data selection and machine learning testing. The second stage of the research. At this stage, the sensor data was divided into direct and indirect user-dependent data, respectively. Several machine learning models were tested, and the correlation threshold was set to 0.98. A supervised quantitative algorithm and a classification algorithm were chosen for the system based on performance. IF the user allowed all the necessary data, the first algorithm was used, but if the user did not allow the flow of some of the data, then the classification algorithm was used to give a categorical sense of what the user driving metrics is.

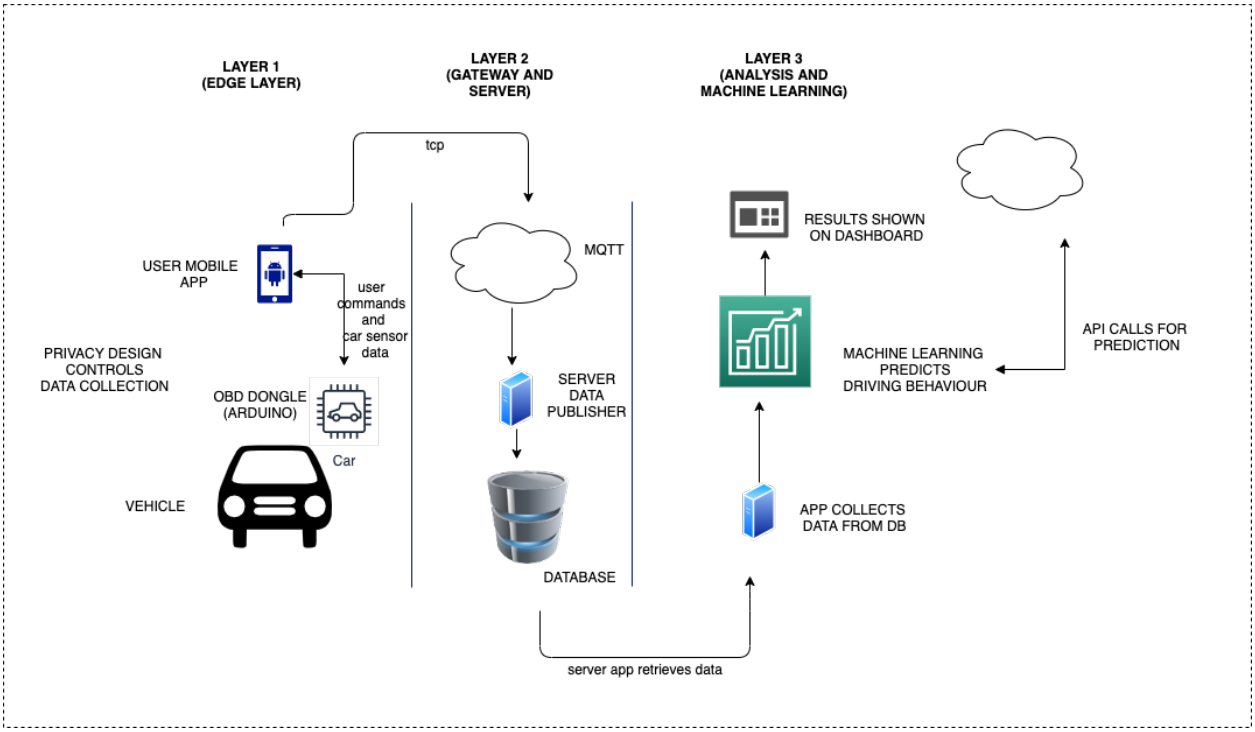


Figure 1: Overview of the proposed system

Studying the effects of privacy implementation on the system. This is the third stage of the research, where the efficiency of the system is tested and analyzed. At this stage, we checked the effects of the kind of sensor data that the system received to check its effects on the model. The second part of the security, which is allowing users to control the data, is also analyzed, and the effects were documented. It is at this stage that we tested how much control the user could have on the flow of data for the system to work.

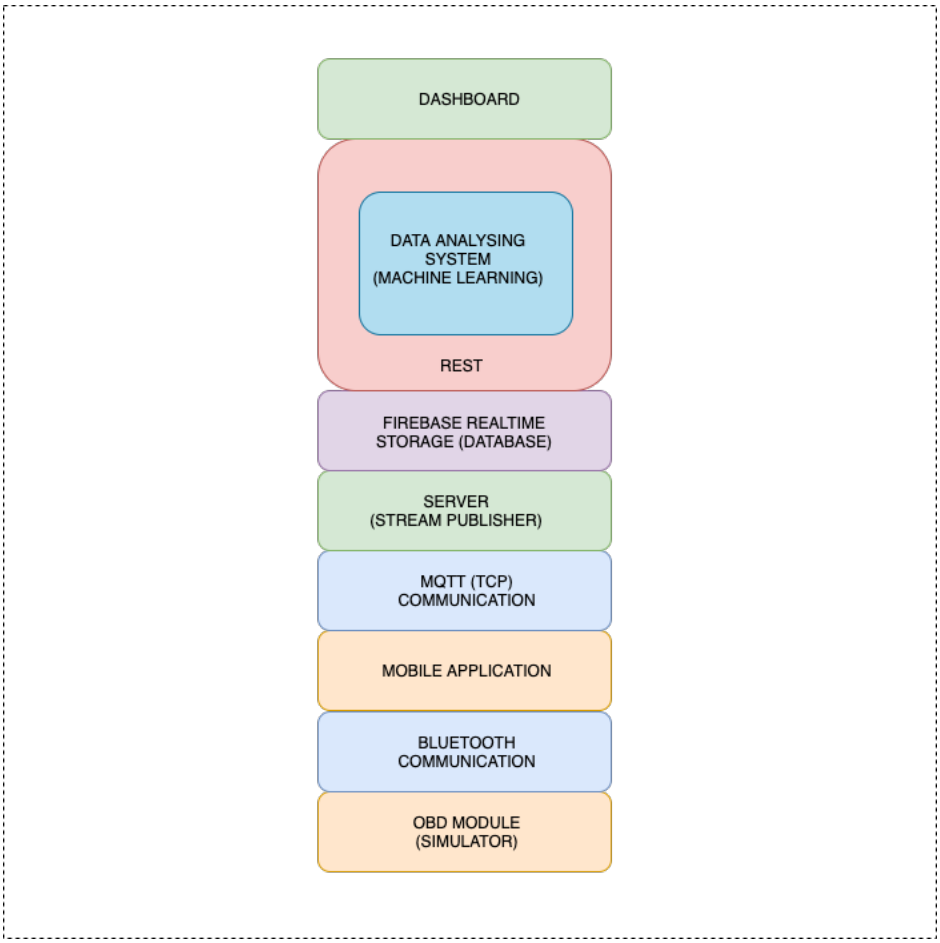


Figure 2: High-level architectural design of the system

SYSTEM DESIGN

MOBILE APP

The mobile application is in 2 versions. Version one is without the privacy design, and version 2 is with the privacy design alongside a stripped-down version of the functionality to basic requirements to serve as a proof of concept.

Version 1 has the following requirements, namely:

1. Retrieve vehicular data from the OBD simulator
2. Allow user to view the data being collected
3. Periodically send the data to a server

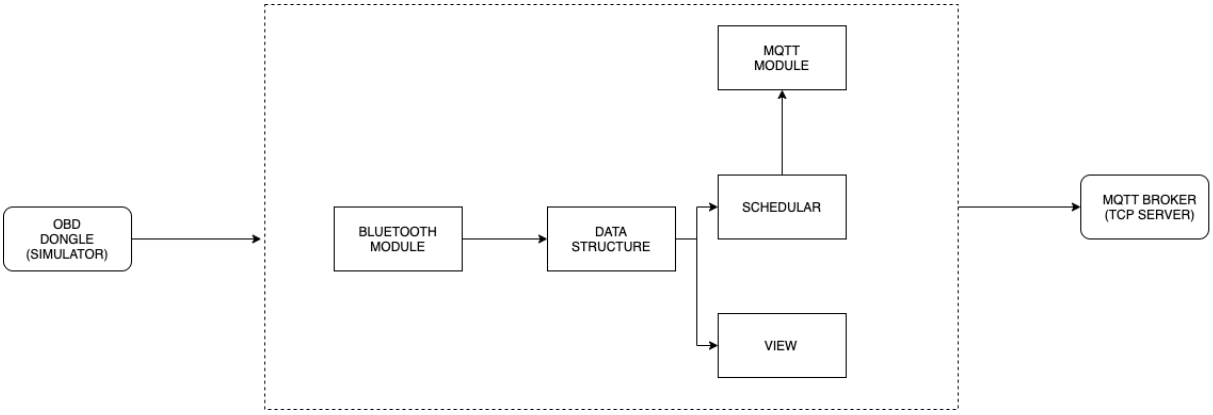


Figure 3: high-level architecture of mobile App (version 1)

Version 2 provides the same requirements as version 1 but also allows the privacy design that gives the user the opportunity to select and filter what data is collected and when it can be collected.

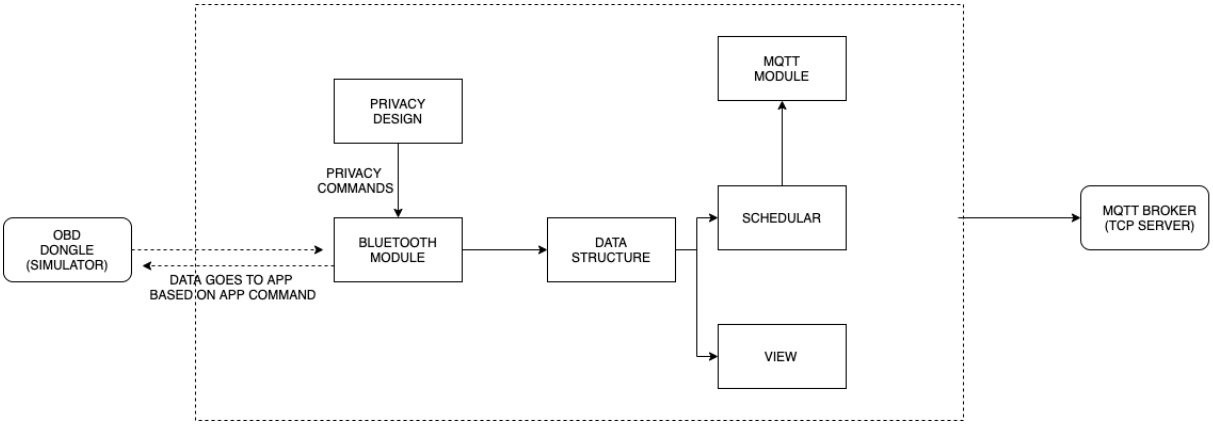


Figure 4: high-level of architecture of mobile App (version 2)

DRIVER PRIVACY DESIGN

The data privacy design aspect is the main section of version 2 of the App. Its main focus is to allow the user to filter and choose what data is sent to the cloud. To allow them to exercise their rights. Various methods were analyzed and….

VEHICLE DATA PUBLISHER & STREAM PUBLISHER

The Vehicle data publisher is the aspect of the system that passes data from all users to the cloud. It is designed to send the data to the MQTT server, which is a broker that allows clients to subscribe to the data to get them in realtime. The vehicle data publisher is part of the mobile application.

On the other hand, the stream publisher is a stand-alone server application that subscribes to the MQTT broker and collects the data upon arrival and takes it to the Firebase Realtime Database.

DRIVING BEHAVIOUR ANALYSIS

Write on this later……

CHAPTER 4: IMPLEMENTATION

LANGUAGE, TOOLS AND TECHNOLOGIES

The development of the system was separated into various tasks. The OBD module simulator was created in python, and the data was sent to an ATmega2560. From there, data sent to the phone via Bluetooth and to the cloud for analysis. A more detailed explanation of the technology used is presented here:

- OBD Module
  - Languages- Arduino, Python
  - IDE- Arduino, Visual Studio Code
  - Hardware - ATmega2560, one breadboard, 1 Bluetooth module, three male to female wires and two male to male wires
- Mobile App(v 1 & 2)
  - Languages – Arduino
  - IDE – Android studio
- Server
  - Languages – Python
  - IDE – Visual Studio Code
- Database
  - Languages – NoSQL
  - Engine – Firebase Realtime Database Engine
- Analysis
  - Languages- Python (Scikit learn library)
  - IDE – Jupyter Lab and Spyder (Anaconda environment)
- Dashboard
  - Languages – HTML, CSS, JS, PYTHON (bootstrap, jQuery, flask)
  - IDE – Visual Studio Code

OBD SIMULATOR (ARDUINO MODULE)

An OBD emulator was purchased from the Mac App store, but it lacked the necessary documentation and tutorials needed for it to be used; hence a simple simulator was created in python to send generated data to the Arduino module. The Arduino then sends the data every 10 seconds via Bluetooth to a paired mobile phone based on the command it receives from the phone.

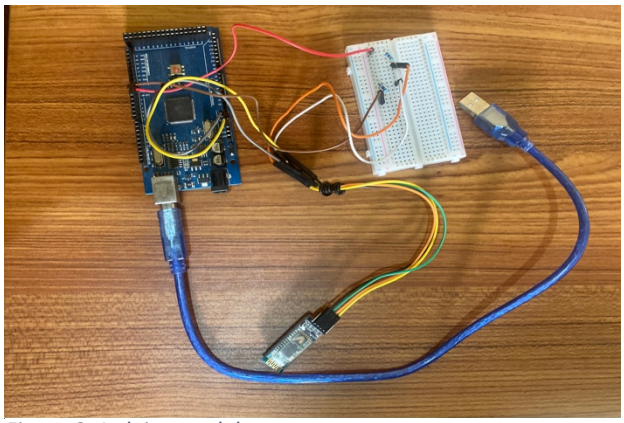


Figure 6: Arduino module setup

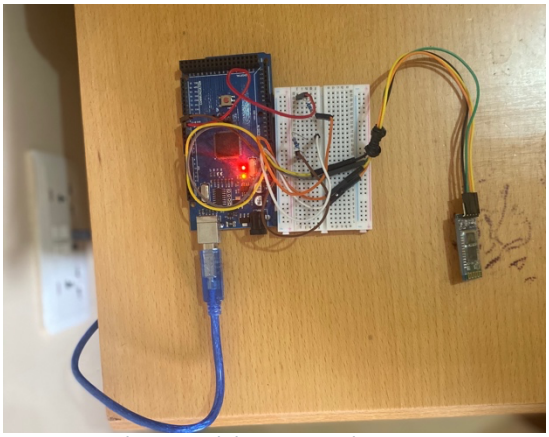


Figure 5: Arduino module connected to power

MOBILE APP

The mobile app is implemented using Android SDK; the minimum SDK-version to run the App is Android API-level 27. The mobile the runs the App must have Bluetooth and internet to execute the tasks. Version 1 of the app receives data from the OBD simulator and displays it. It also sends the data periodically to the MQTT broker via the internet. Version 2 of the application has a design that gives the user the ability to control the data that is collected and sent over the network.

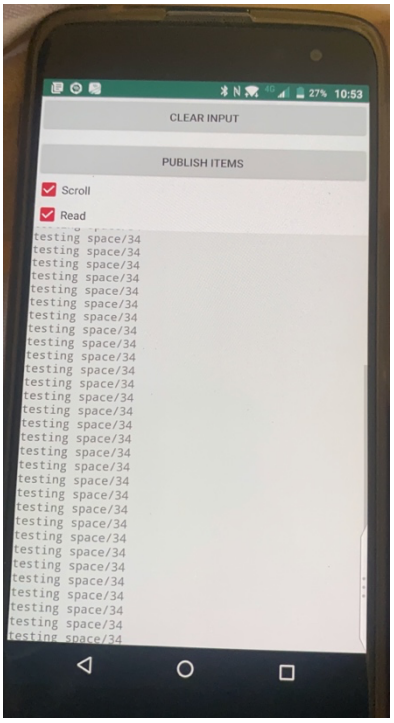


Figure 7: version 1 of mobile App in the testing phase

STREAM PUBLISHER AND DATABASE

The Stream publisher, in this case, is the server that is implemented. Data from the phone is sent to an MQTT broker, and the server subscribes to the necessary topic and listens for uploaded data. As soon as data is sent to the broker, it takes the data and sends it to the firebase Realtime Database. It was created in python using the Pyrabase library and MQTT library. Part of the code was gotten from GitHub [10] under the GNU General Public License by the Free Software Foundation. The database is the

online Realtime Database from Firebase.

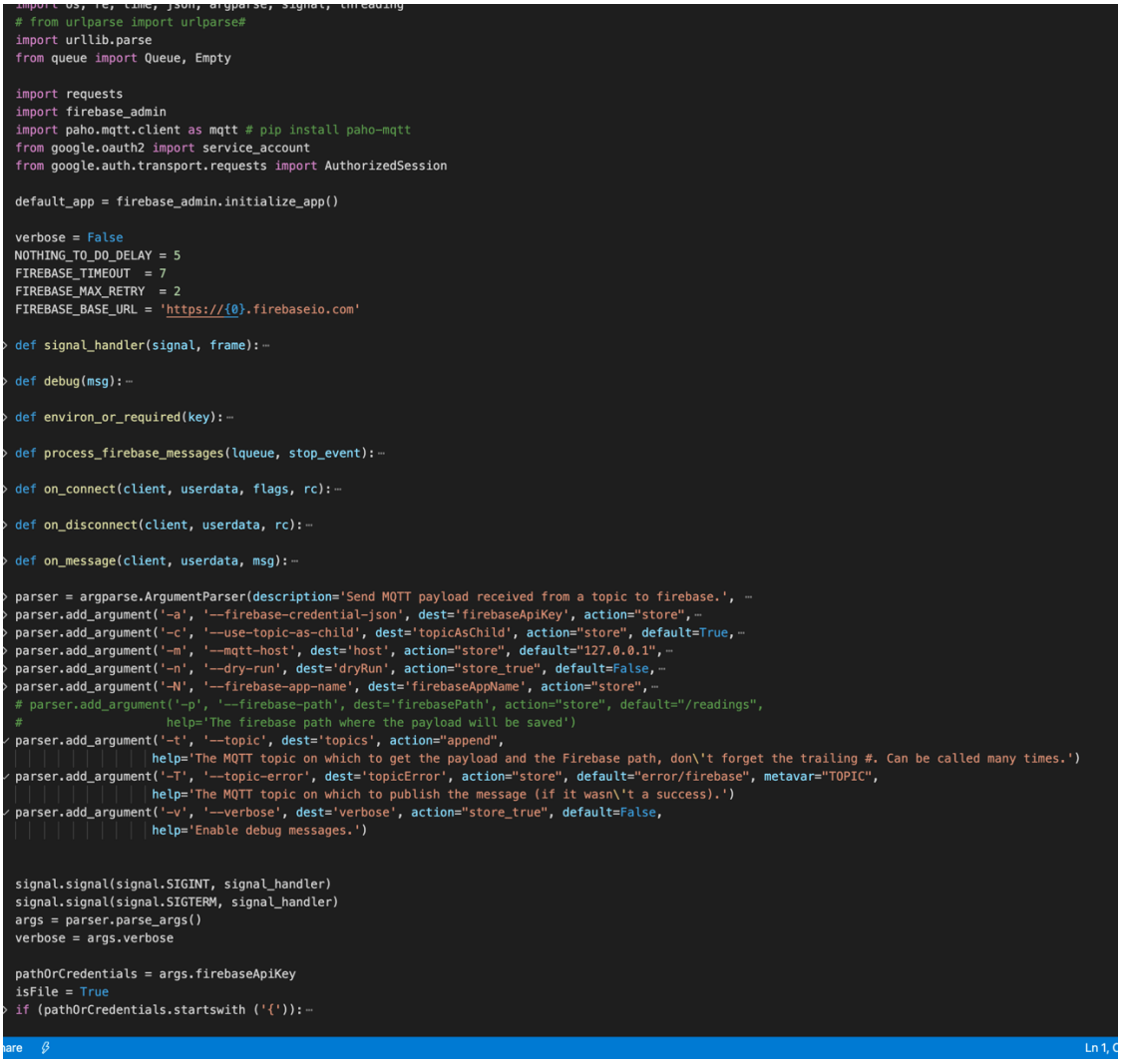


Figure 8: snippet from the stream publisher server code

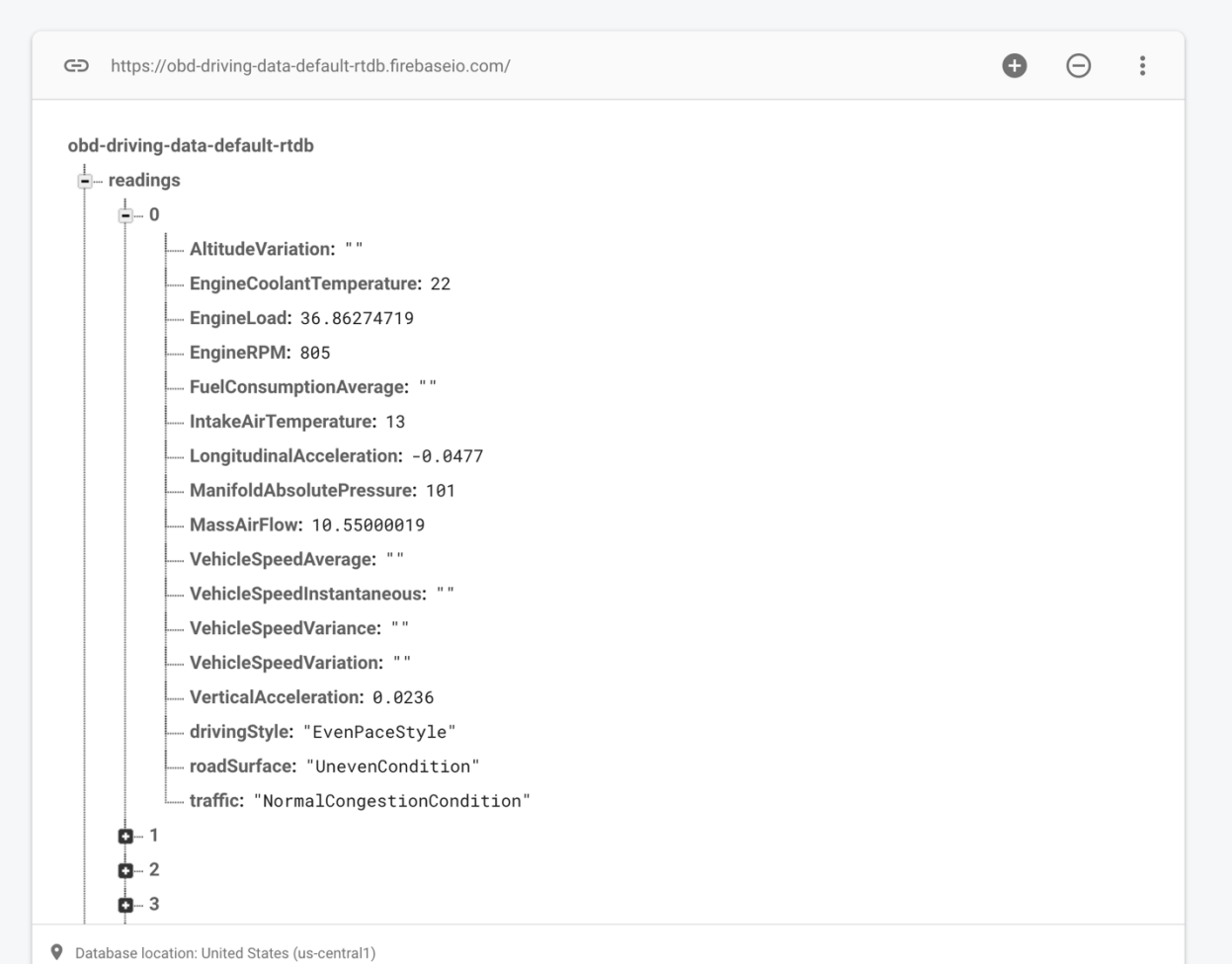


Figure 9: snippet from the firebase realtime database

DRIVER PROFILING PLATFORM

The profiling platform is made up of the machine learning model. It predicts a user to be a safe driver or an aggressive driver and the probability of accuracy of the prediction. It uses data such as the Engine Load, Vertical Acceleration etc., to perform the prediction.

IMPLEMENTATION OF DASHBOARD

The dashboard presents results for the Insurance companies as well as third-party companies that want to use the system. It shows the efficiency of the model, provides a test portal for people to test the system, shows data and driving predicted status from live cars, and API calls from the insurance companies for the results of the subscribed users in the system. The dashboard was implemented using python and the flask module. The machine learning module is serialized using the pickle library, so the model is built only once. Anytime there is an addition to the database, the data is retrieved from the firebase and goes through the prediction process and shows up on the dashboard. HTML, CSS, and bootstrap were used for the front end. ChartJS library was used in showing charts of the data on the dashboard.

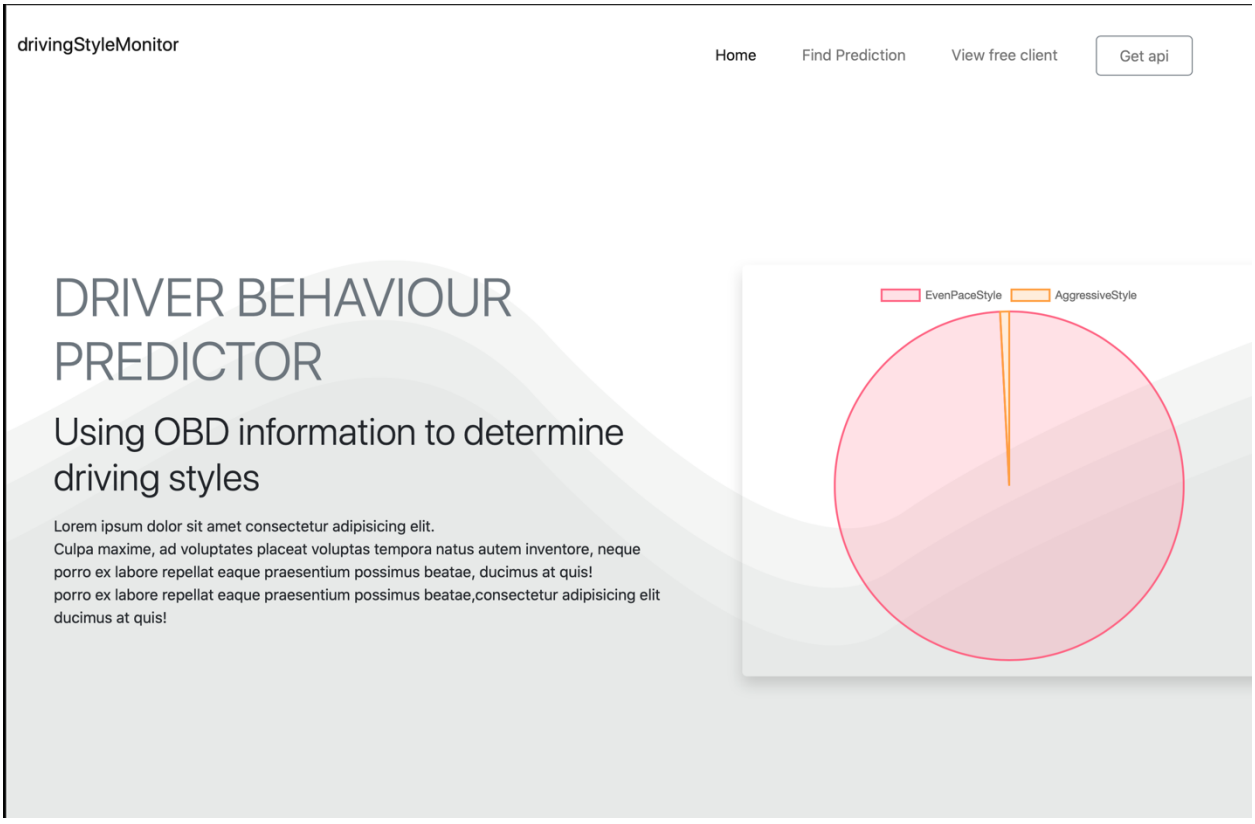


Figure 10: dashboard in testing phase

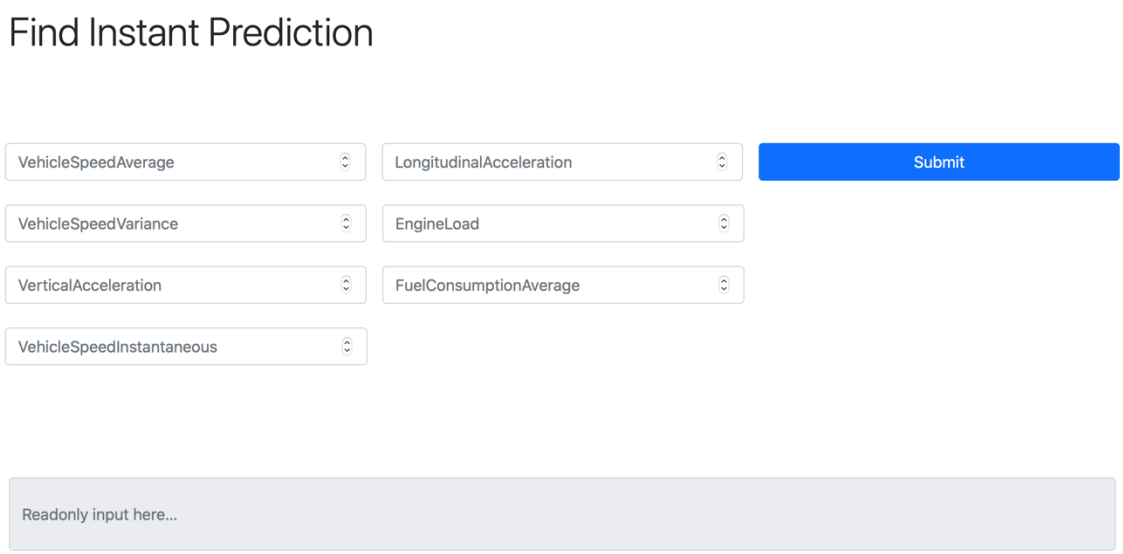


Figure 11:predictive test of the dashboard in testing phase

View Client Lsive Predictions

predictedProba	predictedLabels	actualLabels
0.68	EvenPaceStyle	EvenPaceStyle
0.7	EvenPaceStyle	EvenPaceStyle
0.7	EvenPaceStyle	EvenPaceStyle
0.7	EvenPaceStyle	EvenPaceStyle
0.7	EvenPaceStyle	EvenPaceStyle
0.7	EvenPaceStyle	EvenPaceStyle
0.68	EvenPaceStyle	EvenPaceStyle
0.7	EvenPaceStyle	EvenPaceStyle
0.69	EvenPaceStyle	EvenPaceStyle
0.7	EvenPaceStyle	EvenPaceStyle
0.67	EvenPaceStyle	EvenPaceStyle
0.95	EvenPaceStyle	EvenPaceStyle

Figure 12: free client predictions for evenly-paced style in testing phase.

CHAPTER 5: RESULTS

PRIVACY BY DESIGN

The figure above shows the implementation of privacy by design in the mobile app version 2. The App allows the user to turn off certain features he doesn't want to be collected as well as the ability to not send any personal data. Two options were considered. I.e. allow the data to come to the phone and, based on users preference, filter what goes to the cloud or the second option, to allow the user to to control what data comes to the phone via the Arduino module by controlling it from the phone. The second option is the most secure, and that gives the user the most security and privacy control over the data. The ……

PERFORMANCE AND ACCURACY

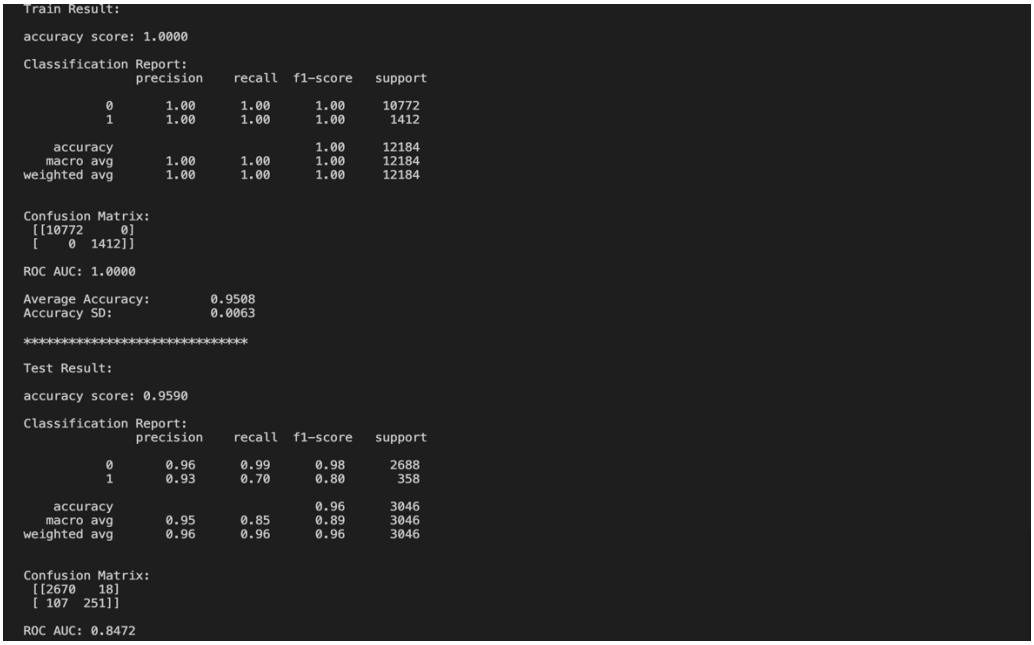


Figure 13: Machine learning report

To evaluate the accuracy of the system's predictive part, data[11] collected from the Kaggle datasets about low-level parameters collected by the cars OBD II system through a dongle since it would be expensive to drive around with cars to collect data and label it. The dataset had 17 features and three labels; we used the driving behaviour label and, from feature selection in the machine learning process, chose seven features to use in the prediction without changing the parameters of the model except for the random state of the random classifier which was set to 40. The scikit-learn random forest classifier was used in the model. It generated an accuracy of 0.96 in the test data and an ROC of 0.85. From the confusion matrix, it classified 2670 as safe drivers who were accurate but 18 as unsafe, which was wrong. It also classified 251 as risky driving patterns correctly and 107 incorrectly. This is a fairly good model, and because the data did not have as much data for risky driving patterns as safe patterns, it was more

accurate with safe drivers with an f1-score of 0.98 for safe drivers, which is better than the f1-score of 0.80 for risky drivers.

RESULT COMPARISON BETWEEN THE PRIVACY BY DESIGN AS AGAINST THE NORMAL MOBILE APPLICATION.

The results from the comparison show that…..

CHAPTER 6: SUMMARY

CONCLUSION

With the emergence of IoT in Insurance, developing systems to retrieve vehicular data and make informed decisions such as lower premiums and better driving styles are a few advantages. In Usage-Based Insurance's current state, it is important that privacy by design is introduced not only in Insurance IoT controlled Systems but in IoT in general. As the outcome features of this project, we came to the conclusion that privacy by design methods can be included in IoT controlled Insurance, and it does influence the outcomes of the machine learning models.

Talk about how exactly it is being influenced and the disadvantages and limitations it provides…….

CHALLENGES

Quite a number of challenges were faced during this project. First, the OBD II ELM Emulator purchased to be used with the Arduino did not come with documentation, and no information was on their website providing steps as to how it can be used. As a solution, a log file was created with test data to pass on data as an emulator.

Getting OBD labelled data for driving behaviour was another challenge. There are not a lot of OBD collected data that has been labelled for driving behaviours.

The schedular library for android that allows background tasks only works with a minimum schedule time of 15 minutes. Since the App was needed to send data more quickly, a thread was created to send data after every 5 minutes as a solution causing the mobile application to transmit data only when the App is opened.

FUTURE WORKS

Currently, the OBD Arduino dongle does not interface with an actual OBD or simulator, and test data is being sent via a log file. In future, a robust OBD Bluetooth dongle will be implemented to interface with an actual OBD II system. The mobile Apps only allow data to be transmitted when the App is opened, and it must be updated to allow it to transmit and receive data even when it is working in the background. Version two of the App does not allow the user to see graphs and performance indexes of how well they perform during trips, and this can be included to allow them to access themselves.

For the predictive module, a better solution can be generated to reduce errors in predictions when the user controls the flow of data. The current solution is highly dependent on data and does not work well with data interruptions.

Concerning the most important part of this project, the privacy by design, other design methods can be tested to make it robust enough such that the accuracy and performance of the system are not compromised.

CHAPTER 7: REFERENCES

[1] O. Andrisano, R. Verdone, and M. Nakagawa, "Intelligent transportation systems: the role of third-generation mobile radio networks," *IEEE Communications Magazine*, vol. 38, no. 9, pp. 144–151, Sep. 2000, doi: 10.1109/35.868154.

[2] J. Quintero and Z. Benenson, "Understanding Usability and User Acceptance of Usage-Based Insurance from Users' View," in *Proceedings of the 2019 2nd International Conference on Machine Learning and Machine Intelligence - MLMI 2019*, Jakarta, Indonesia, 2019, pp. 52–57, doi: 10.1145/3366750.3366759.

[3] S. Derikx, M. de Reuver, and M. Kroesen, "Can privacy concerns for insurance of connected cars be compensated?," *Electronic Markets*, vol. 26, Dec. 2015, doi: 10.1007/s12525-015-0211-0.

[4] Y. H. Hwang, "IoT Security & Privacy: Threats and Challenges," in *Proceedings of the 1st ACM Workshop on IoT Privacy, Trust, and Security - IoTPTS '15*, Singapore, Republic of Singapore, 2015, pp. 1–1, doi: 10.1145/2732209.2732216.

[5] "Chapter 3 – Rights of the data subject," *General Data Protection Regulation (GDPR)*. https://gdpr-info.eu/chapter-3/ (accessed Oct. 12, 2020).

[6] C. Herzog, E. G. Counsel, C. of A. E. W. Group, and Arm, "Engineering Ethics into AI," *Arm Blueprint*, Nov. 06, 2019. https://www.arm.com/blogs/blueprint/arm-ai-trust-manifesto (accessed Sep. 30, 2020).

[7] W.-H. Chen, Y.-C. Lin, and W.-H. Chen, "Comparisons of Machine Learning Algorithms for Driving Behavior Recognition Using In-Vehicle CAN Bus Data," in *2019 Joint 8th International Conference on Informatics, Electronics Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision Pattern Recognition (icIVPR)*, May 2019, pp. 268–273, doi: 10.1109/ICIEV.2019.8858531.

[8] P. Handel, J. Ohlsson, M. Ohlsson, I. Skog, and E. Nygren, "Smartphone-Based Measurement Systems for Road Vehicle Traffic Monitoring and Usage-Based Insurance," *IEEE Systems Journal*, vol. 8, no. 4, pp. 1238–1248, Dec. 2014, doi: 10.1109/JSYST.2013.2292721.

[9] V. Gandhi and J. Singh, "IoT: Architecture, Technology, Applications, and Quality of Services," 2019, pp. 79–92.

[10] S. Lucas, *seblucas/mqtt2firebase*. 2021.

[11] "Traffic, Driving Style and Road Surface Condition." https://kaggle.com/gloseto/traffic-driving-style-road-surface-condition (accessed Apr. 01, 2021).