Driver Usage Risk Profiling by Analyzing Vehicle Driving Behavior using Machine Learning Model Based on Vehicular Cloud Telematics Data

MSc Research Project
Cloud Computing

Anuj Kumar
Student ID: X17157641

School of Computing
National College of Ireland

Supervisor: Victor Del Rosal
<table>
<thead>
<tr>
<th>Student Name:</th>
<th>Anuj Kumar</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student ID:</td>
<td>X17157641</td>
</tr>
<tr>
<td>Programme:</td>
<td>Cloud Computing</td>
</tr>
<tr>
<td>Year:</td>
<td>2018</td>
</tr>
<tr>
<td>Module:</td>
<td>MSc Research Project</td>
</tr>
<tr>
<td>Supervisor:</td>
<td>Victor Del Rosal</td>
</tr>
<tr>
<td>Submission Due Date:</td>
<td>20/12/2018</td>
</tr>
<tr>
<td>Project Title:</td>
<td>Driver Usage Risk Profiling by Analyzing Vehicle Driving Behavior using Machine Learning Model Based on Vehicular Cloud Telematics Data</td>
</tr>
<tr>
<td>Word Count:</td>
<td>5921</td>
</tr>
<tr>
<td>Page Count:</td>
<td>21</td>
</tr>
</tbody>
</table>

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

**ALL** internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author’s written or electronic work is illegal (plagiarism) and may result in disciplinary action.

**Signature:**

**Date:** 19th December 2018

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:**

- Attach a completed copy of this sheet to each project (including multiple copies).
- Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).
- You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

**Office Use Only**

<table>
<thead>
<tr>
<th>Signature:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Date:</td>
<td>19th December 2018</td>
</tr>
<tr>
<td>Penalty Applied (if applicable):</td>
<td></td>
</tr>
</tbody>
</table>
Driver Usage Risk Profiling by Analyzing Vehicle Driving Behavior using Machine Learning Model Based on Vehicular Cloud Telematics Data

Anuj Kumar
X17157641

Abstract

This paper is a novel approach to assist the vehicle insurance companies for calculating the annual insurance premium cost for the vehicle owner by developing Vehicle Prediction System (VPS) using Vehicular Cloud and Machine learning Models. By the development of new technologies in recent years, the potential to handle the Vehicle sensing data is increased. This Proposed system is using real-time operational information from a vehicle including engine Revolutions per minute (RPM), vehicle speed, throttle position, Mass Air Flow (MAF), engine load, time advance through On-Board Diagnostic (OBD) interface. By utilising Cloud infrastructure and Machine learning algorithms, the proposed system will make use of vehicle data to create a drive behaviour prediction model to calculate the risk profile of insurance policyholder. Using Cloud Computing technology, a massive amount of Telematics data can be handled generated by these vehicular sensors and Machine learning algorithms can be used to create a risk profile of the vehicle owner by measuring the driving behaviour. This proposed model is using Unsupervised and Supervised machine learning methods to measure the driving behaviour. This platform can be beneficial for the policy subscriber as this prediction model will be used by the insurance companies to improve the User-based insurance (UBI) and Pay-as-you-drive (PAYD) pricing model by offering lucrative price to their insurance subscribers.

1 Introduction

Handling a Massive amount of data generated by the Internet of Things (IoT) devices requires an efficient platform. We require an adequate platform to analyse and process this amount of data. Cloud Infrastructure is an efficient platform to handle this significant volume of data by utilising its availability and scalability featuresSimmhan et al. (2013).

Traditional insurance premium calculation models, the insurance companies are using information related to the drivers instead of vehicle condition and driving behaviour to charge their customersTroncoso et al. (2011). In this proposed paper, we are discussing numerous sensors installed in the vehicles to get the real-time data of vehicle health to analyse and use it for driving behaviour prediction model.

This paper aims to develop a driving prediction model for the insurance companies to charge their customers by their driving behaviour instead of being charged with a fixed amount. Currently, there are very few companies which are charging their customers using
User Based Insurance (UBI) and Pay-How-you-Drive (PHYD) pricing model. Insurance companies should conspire with information technology (IT) to apply digital marketing approach through which they can attract more consumers by providing benefits like lowering the premium cost to their existing and future customers along with earning profits.

All vehicle drivers have a different style of driving such as how to accelerate, decelerate, braking and different ways to use vehicle control devices. We are developing the prediction model to use these vehicle sensors data for driving behaviour analysis using cloud technology as its platform Amsalu et al. (2015) for the insurance companies. The Proposed Machine learning model consists of three major components 1) On-Board-diagnostic (OBD) device: To collect the vehicle sensor telematics data from the vehicle. OBD port is available in all the vehicles to fetch all the sensors data Takefuji (2018). 2) Cloud infrastructure: Using cloud infrastructure to analyse and process massive amount because of its high scalability and availability. 3) Vehicle prediction server (VPS): The proposed machine learning model is running on a server build on cloud infrastructure. For Driving behaviour analysis, the Proposed model is using six different attributes to measure the driving behaviour like Engine Throttle position, Engine load, MAF, Speed, RPM, and Time Advancing. These values are selected after calculating the correlation between RPM and all other attributes.

There are some driving prediction models, where the author is using Supervised machine learning techniques with the combination of mobile and vehicle sensor data Zhang et al. (2016) Meseguer et al. (2013). Additionally, there are some models, which are using Unsupervised machine learning techniques Van Ly et al. (2013). This proposed model, is using both supervised and unsupervised machine learning techniques mentioned in Figure 1 to analyse the driving behaviour for the insurance companies.

![Figure 1: Machine Learning methods used](image)
Unsupervised Machine learning: EM: This Technique is used to find the maximum similarity of function, AH: This technique forms a treelike structure by creating partition sequence where a single group is sub-divided into multiple groups, K-means: This technique uses centroid, which is the average value of an instance in a group. It uses Euclidean distance to measure the similarity between the instances\textsuperscript{[Hastie et al. (2009)]}.

Supervised Machine learning: AdaBoost (AB): It works on the base algorithm, and it iteratively improves the classification in training set, Decision tree (J48). A decision tree is a technique which forms a tree-like structure to sort new instances, K-Nearest Neighbors (KNN); this technique provides a standard class in training set for classification, Naive Bayes (NB). This technique is based on Bayes theorem and doesn’t provide the same result for the same base, Multi-Layer Perceptron (MLP). It works on the basis of the nervous system, having layers of neurons with a strong connection, Random Forest (RF): This technique forms a large number of decision trees to find the best possible way for result, Support Vector Machine (SVM): This Technique is base on Statical methods\textsuperscript{[Kumari and Godara (2011)]} [\textsuperscript{[Rätsch et al. (2001)]}]

2 Related Work

This Section outlines previous work happened in the area of Telematics, the Insurance industry and Machine Learning models. The main focus of this purposed prediction system is to improve the Insurance premium pricing model for the insurance business by combining different technologies. Additionally, to reduce the flaws in the existing inefficient and unfair traditional insurance premium criteria such as charging the individual user with a fixed or lump sum amount by covering total kilometres driven by the vehicle owner irrespective of Driving behaviour\textsuperscript{[Butler et al. (1988)]} [\textsuperscript{Parry (2004)}]. In below mentioned subsection 2.1 we are discussing a few previous insurance risk profiling models using telematics data.

2.1 Telematics Data usage for insurance Risk profiling.

The Major focus of Insurance market is to modify the Pay-As-You-Drive (PAYD), Pay-How-You-Drive (PHYD) and Usage-Based Insurance (UBI) pricing model to utilise the driving behaviour pattern information of an individual policy owner to decide their premium cost\textsuperscript{[Boquete et al. (2010)]}. There are multiple mobile and tablets applications available in the market like Aviva RateMyDrive\textsuperscript{[1]} State Farm driver feedback\textsuperscript{[2]} to score an individual driving behaviour by using different technologies like GPS, Magnetometers accelerometers. These applications are using the driving score to provide 20% of discount for the insurance premium\textsuperscript{[Castignani et al. (2015)]}.

\textsuperscript{[Eren et al. (2012)]} developed an algorithm to distinguish between safe and risky drivers having an accuracy of 93.3% by evaluating data from 15 iPhone User with the fixed point of the journey. The Author is using empirical thresholds and moving algorithm to detect

\textsuperscript{[1]}Aviva PLC.(2013) Aviva RateMyDrive\textsuperscript{http://www.aviva.co.uk/drive/}
\textsuperscript{[2]}Statefarm DriverfeedbackState Farm Mutual Automobile Insurance Company.(2013): \textsuperscript{http://www.statefarm.ca/about/mobile/}
driving events like aggressive steering, acceleration, braking etc via gyroscope, magnetometer and smooth acceleration data from smartphones. This paper is using Bayesian classification and Dynamic Time Warping (DTW) to differentiate between safe and risky drivers by using template data (Similar patterns collected for safe and risky events).

The future work is to [Paeggen et al. (2012)] utilise the Smartphone data mainly for insurance companies to predict the driving behaviour by manually using user setting and the direction of the vehicle after the calibration process and start collecting data such as braking, Steering events and acceleration. By defining, a threshold value for sensing data (like 0.2g for steering and 0.1g for acceleration), the events are trigger and compare the event detection data by a smartphone with the values of telematics boxed fixed in the vehicle on the Inertial Measurement Unit (IMU). According to the observation of the author due to the variation in smartphone position in the vehicle, the event count is match by distinct analytical distribution. However, the paper, the author provided some error sources and correlation between that IMU and smartphone-based events.

Later on, many insurance premium calculation models are introduced like [Baecke and Bocca (2017)] propose PAYD pricing model using the Internet of Things (IoT) Device to create risk profile of insurance policyholder by analysing the driving behaviour. [Tselentis et al. (2016)] propose combination of PAYD and PHYD. In 2016 [Nai et al. (2016)] propose UBI pricing model for risk analysis of insured vehicle by Fuzzy Risk Mode and FRAME method (i.e. effect Analysis) where the telematics data is provided to the Expert team to analyse and provide a grade to the vehicle according to the basic risk analysis [Bian et al. (2018)].

2.2 Existing Cloud Implementation to process OBD telematics Data.

In this subsection, discussion is about the existing approach to process OBD telematics data to measure the driving behavior for the insurance policy subscriber using cloud technology as their platform. Initially OBD device were locally install in vehicles to measure the health of vehicle. By the advent of technology ,current OBD system is improve by the on-line feature to fetch the real-time vehicular data for analysis. [Iqbal and Lim (2006)] propose a Global Positioning System (GPS) based insurance calculation system ;this proposed insurance model was base on mobility of vehicle that includes total kilometers driven by vehicle, Zone of traveling, time of traveling and average speed to measure the risk profile of vehicle owner.

Different attributes used by [Jhou et al. (2013)] to reduce the time to detect the fault in vehicle in-case of any breakdown. In this proposed, cloud-based detection model, all the dynamic real-time information like Engine RPM, Coolant temperature, fault codes, vehicle speed were used. The information is received on cloud server called Vehicle Diagnostic Server (VDS) over a 3.5 wireless network for the analysis. The cloud server is an online expert system providing a solution in-case of any breakdown by analysing all the parameters using statistical algorithms.

The early fault detection system was lately improved by [Amarasinghe et al. (2015)] the author is using the Android mobile application to connect the OBD device with a
server running on Cloud infrastructure. This application was utilising mobile data for its communication.

2.3 Different Technique used for measuring Driving Behavior

Araújo et al. (2012) developed a smart-phone application to analyse the Driving behaviour and help the drivers by suggesting them the better way to drive. The main aim of this application was to reduce fuel consumption by improving the driving style. This application was using different vehicle sensors like vehicle Speed, Engine RPM, Throttle signal etc to measure the driving behaviour.

Similarly, Meseguer et al. (2013) also used Speed, RPM, Acceleration data for analysing the driving behaviour. The author of this paper used Multilayer perceptron (MLP) algorithm for the classification of data, and the approach of this paper is considering adding the road conditions as well to analyse the driving behaviour. Zhang et al. (2016) proposed a platform using Support Vector Machine (SVM) algorithm that uses both smart-phone and vehicle sensors to analyse the driving behaviour by Support-Vector-Machine (SVM) classifier.

2.4 Discussion

In the above literature review section 2, we have discussed different studies using only Mobile sensor and Only vehicle sensors to analyse the driving behaviour also we have discussed some papers, the author is using both mobile and vehicle sensor to analyse the driving behaviour. Some of the studies are utilising cloud infrastructure as their platform, and the outcomes of most of the studies are using vehicle sensors provides better accuracy.

We have discuss some of the studies using Unsupervised and supervised machine learning techniques to analyse the data. The purpose of those studies was different for this proposed model. Most of them analysed the driving behaviour to reduce the CO2 emission, to analyse risky and safe driver, Reducing the fault analysis time in case of breakdown and reducing the fuel consumption by proving better suggestion based on the vehicle driving behaviour.

This proposed paper is using both machine learning techniques, i.e. Unsupervised and Supervised. Unsupervised techniques are use for clustering the data without a class label and supervised techniques for the classification of data Michalski et al. (2013).

3 Methodology

The chief objective of this research paper is to develop a prediction system/model for analysing the driving behavior to calculate the cost of insurance subscription. In this section, we are discussing about all the steps involved to develop the proposed vehicle driving prediction model. Below are the steps mentioned involved:

1. Data Procurement
2. Data Preprocessing
3. Apply Machine Learning Methods and Post Processing

(a) Apply Clustering Methods over Dataset
(b) Evaluate Partitions
(c) Apply Classification Methods over Best Partitions
(d) Evaluate Results from Classification Methods
(e) Relabel Best Partition

4. Build Model According to Step 3

5. Develop cloud application

These steps are based on methodology defined by Barreto (2018) for building machine learning models with clustering and classification methods and some steps for validation at the end of using ML methods. The Figure 2 shows the methodology used and steps defined to it.

![Methodology Diagram]

Figure 2: Methodology used - based on Barreto et al. (2018)

3.1 Data Procurement

The proposed model is using Dataset from Kaggle.com3 with the permission of dataset owner. This dataset is holding vehicle telematics raw data from 19 different drivers on a single vehicle on a single route. Dataset is having total 28 different columns and 8261 rows.

3https://www.kaggle.com/cephasax/obdii-ds3
3.2 Data Preprocessing

This subsection is carried all the steps involved to fetch out the required information in a specific format from the raw information. The Dataset includes different 28 vehicle sensor data like Longitude, Latitude, Altitude, Fuel Level, Vehicle Id, Engine Coolant Temperature, Barometric Pressure etc. Using all the attributes is not required for the proposed model, selection of few attributes for better accuracy is necessary. below are the steps involved in clearing the data and selection of different attributes.

3.2.1 Fixing Values.

After we have Data for vehicle sensors, we need to check and verify all the values if there are any missing, noisy or outliers values in the dataset. In the proposed model is using Mean strategy to fill all the missing values for all the attributes used.

1. Using MS Excel, we are calculating the mean values by using Minimum and maximum values for all the columns.

2. These Mean values are used to replace the missing values.

3.2.2 Normalization of Data.

After fixing, the missing values for all the attributes used, we need to normalise the data for better accuracy and avoiding the redundancy.

\[
\text{Normalize value} = \frac{(DataToNormalize - Minimum)}{(Maximum - Minimum)} \tag{1}
\]

3.2.3 Calculate the Correlation

The proposed model is using the Pearson method for fetching the correlation between RPM and other different attributes to check the relationship between them; how the values are changing concerning the RPM.

3.2.4 Attributes Selection

Based on the correlation values, six different attributes to be processed by a proposed model including RPM values using MS Excel Filter option.

3.2.5 Building a new Dataset

After performing all the steps mentioned above, new data is formed to be used by the prediction model. That dataset is name as “anuj_norm_6” for this research proposals.

3.3 Apply Machine Learning Methods and Post Processing

This sub-section includes all the steps involved while building the machine learning model like clustering of data, classification of clusters and re-labeling the data.
3.3.1 Apply Clustering Methods over Dataset

No class label available in the dataset, Unsupervised Machine learning technique is used for clustering the data. An open-source data mining tool Weka\footnote{https://www.cs.waikato.ac.nz/ml/weka/} is used for this process. The propose model is using three different clustering techniques: Expectation Maximization (EM), Agglomerative Hierarchical (AH) and K-Means. All of these clustering techniques are necessary to perform for clustering the data with different configurations such as a number of clusters varies from 2 to 11 (K=2,311). Beyond this configuration EM, and K-means are probabilistic techniques. It is necessary to use different numerical seeds(proposed model is using five different seeds). As a result of this part of work, we had 110 partitions: EM (50), AH(10) and K-means(50).

3.4 Evaluate Partitions

After using clustering techniques, we are using Silhouette and Davies Boudin (DB) indexes to evaluate the quality of the cluster. To measure, the better quality of the cluster silhouette index value should be near to 1 but inverse for Davies Boudin, the index value should be near to 0\footnote{Wiwie et al. (2015)}. It is observable in Figure 3, the best partitions index results(for every clustering methods)are for cluster value equals to 2. With this value for K, the Silhouette index reaches the value 0.468 for EM method (average), and the DB reached 0.379 for the same method (average).

Based on the results supported by the depreciation of the index values that with the increase of k, EM (with k = 2 and seed = 11) is chosen as the best partition. Similarly, the other two partitions are select as input for the next part of the research are based on the same index result values. These other two partitions are K-means (with k = 2 and seed = 11) and AH (with k = 2). These three partitions are used based on the result in Figure 3 for the next phase of work.

3.4.1 Apply Classification Methods over Best Partitions

Based on Silhouette and Davies Boudin Index Values, the best three partitions are chose for further classification. Seven different methods made the classification: Ada Boost (AB), Decision Tree (J48), K-Nearest Neighbors (KNN), Naive Bayes (NB), Multi-Layer Perceptron (MLP), Random Forest (RF) and Support Vector Machine (SVM).

For all classification methods used, five different approaches are applied for subdividing the dataset into two partitions (training Set and Test Set). Approach for subdividing the dataset are : 90%/10%, 75%/25%, 50%/50%, 66%/34% and 10 Cross-fold Validation. These percentages are the division of dataset into training and test set for example that 90%/10% is a divide in 90% of the dataset are use for training and 10% for the testing. The approach 10 Cross fold means that the dataset is divided in 10 parts and 1/10 is used for test and 9/10 for training.

3.5 Evaluate Results from Classification Methods

As shown in Figure 4, the average results for the accuracy of all methods is above 94.3%. The results obtained by MLP are better over the K-means partitions, with more than
Figure 3: Silhouette and Davies Bouldin Index Values

99.8% of accuracy. These results are taken as the best result from classification methods and are supported by the best average for all the values (for MLP against all methods).

Figure 4: Classification values
3.5.1 Friedman Chi-square

In order to give robustness to the decision for the best partition, we used Friedman statistical test for check if there is any relevant difference (under statistical terms) between results from classification methods. The Friedman test is performed over the data presented in Figure 4 returned p-value value equals to 0.001259, which means that there is the significant difference between data distribution for this data. As MLP is selected as the best method for this model, the post-hoc test Nemenyi is perform over the data that returned values less than 0.05 when MLP is compare with other methods (see Figure 5) J48 and NB as the accuracy of this method is above 99%.

<table>
<thead>
<tr>
<th>Method</th>
<th>AB</th>
<th>J48</th>
<th>KNN</th>
<th>MLP</th>
<th>NB</th>
</tr>
</thead>
<tbody>
<tr>
<td>J48</td>
<td>0,0524</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KNN</td>
<td>0,8960</td>
<td>0,5917</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MLP</td>
<td>1,0000</td>
<td>0,0469</td>
<td>0,8808</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NB</td>
<td>0,0419</td>
<td>1,0000</td>
<td>0,5391</td>
<td>0,0373</td>
<td></td>
</tr>
<tr>
<td>RF</td>
<td>0,8280</td>
<td>0,6942</td>
<td>1,0000</td>
<td>0,8081</td>
<td>0,6437</td>
</tr>
<tr>
<td>SMO</td>
<td>0,0585</td>
<td>1,0000</td>
<td>0,6179</td>
<td>0,0524</td>
<td>1,0000</td>
</tr>
</tbody>
</table>

Figure 5: Safe Unsafe

3.5.2 Relabel Best Partition

With the best clustering and Classification methods and configurations selected, it is necessary to make changes in the name of cluster for a better understanding. To do this change, we used the average values of each attribute, shown in the Figure 6. It is observed that all values for the “cluster1” is are more than the values for the “cluster2” (average values) for all attributes. It means that data clustered as “cluster1” is more “risky” then values clustered in “cluster2”. For this reason, we decided to relabel the “cluster1” as “unsafe” and “cluster2” as “safe”.

3.6 Build Model According to Step 3

Using the weka data mining tool with User Application Interface, it is possible to make the entire process selected as best methods for clustering and classification. As mentioned, K-means with k=2 and seed=11 was chosen as the best clustering method for the dataset. After opening the dataset “anuj_kmeans_k2.arff” was applied the best classification method, Multi-Layer Perceptron with ten cross-fold validation configuration. Then, the file with extension “.model” was made and was used inside the application as the core of it.

3.7 Developing Cloud Application

Development of Cloud platform was made as a prototype and is made with Eclipse IDE and Spring Boot framework to provide RESTfull endpoints to use the business model proposed here. In the next section, platform details and its use are used.
4 Design Specification

The designed platform is used as a cloud-based service for insurance companies. Multiple insurance companies can utilise the proposed model to utilise Machine learning model for the identifying of the driving behaviour (safe/unsafe) of insurance policyholder and modify their pricing approach. Beyond this, companies can use their pricing model and use the information about driver behaviour as a factor to do its price more flexible and robust.

The platform receives data from insurance companies containing telematics information of its insured vehicles and starts the internal process to predict driving behaviour using the data. The overview or architecture designed to achieve this goal is shown in Figure 7. For this work, we will name the Platform as Driver Behavior Cloud Application (DBCA).

4.1 Basic Functional Architecture of proposed model

The platform works using simple steps covered by Figure 7 mentioned below:

1. Insurance Companies send data to Drive Behavior Cloud Application;
2. Driver Behavior Cloud Application (DBCA) receives the data;
3. DBCA process data;
4. DBCA classifies data according trained ML model;
5. DBCA stores the data classified;
6. DBCA sends the data to Insurance company, when it is requested.
4.1.1 DBCA receives the data

The model receives the telematics data (with vehicle telematics inside) from insurance companies through the REST API endpoint.

4.1.2 DBCA process the Data

The received data is pre-processed to be analyse by the machine learning model. The data processing module will fix the missing values and normalise the data to turning into the specific pattern (ranges and value types) and allowing to use it by machine learning model.

4.1.3 DBCA classifies data according trained ML model

Data from the previous step is accepts as the input of ML model, that classifies it under “safe” or “unsafe”, and sends it to the next step.

4.1.4 DBCA stores the data classified

Data already classified is stored respecting the driver that produces it and the insurance company sends it to the DBCA service.

4.1.5 DBCA sends the data to Insurance company

When the insurance company needs some price decision or information, it sends a request over REST API to the DBCA service with driver identification and receives back all the data already classified for the driver specified. Insurance companies can use the data sent by DBCA for their pricing model which includes their company policy and possibly the

Figure 7: Driver Behavior Application General Architecture
policyholder related data (Previous claims and other information) or to allow or facilitate any decision they require.

These all information is used by insurance company system for offering their customer with a beneficial policy price i.e the price can be more related to the real condition or the driving behaviour of a specific driver not with any average values for some person (average profile).

4.2 Data Pattern

Data needed by DBCA is required a specific pattern delimited for proper function of the system. Description of all the attributes is defined as described in Figure 8. This types and names are defined arbitrarily and use in the same pattern as the input of the ML model.

![Figure 8: Data pattern defined](image)

- **Insurance Company Identification**: name: “insuranceCompanyIdentification” and type: string
- **Driver Identification**: name: “driverIdentification” and type: string
- **Engine Load**: name: “engineLoad” and type: float point number;
- **Engine RPM**: name: “engineRpm” and type: float point number;
- **MAF**: name: “maf” and type: float point number;
- **Speed**: name: “speed” and type: integer;
- **Throttle Position**: name: “throttlePosition” and type: float point number;
- **Timing Advance**: name: “timingAdvance” and type: float point number;
- **Drive Behavior**: name: “driverBehavior” and type: string;
- **Date**: name: “date” and type: string.
4.3 Other Possible Clients

Beyond the Insurance Companies, other types of business can also have some benefits with the use of DBCA. From the best of our knowledge, it is possible that business like fleet managers and assemblers of vehicles that offer services such as scheduled revisions with pre-fixed price can use DBCA. They can offer their services to their clients with attractive/discounted prices based on their driving behaviour.

5 Implementation

The development of DBCA is made with Java Language using Eclipse IDE. The model is using a straightforward Rest API approach added by class for Machine learning model (exported from Weka -already mentioned in section 3.6 ). This model is loaded as a singleton at the start of the application and working on making predictions for any data while the application is running. Some details about Platform components as described in the next subsections.

5.1 Materials used

Spring Framework [Johnson et al. (2004)] is a Java Framework that allows Server applications to run like regular Java applications. Beyond this fact, this framework has a significant number of features (submodules) for handling essential requirements from Server Side systems. Questions like availability, security and others features are provided by Spring submodules and can be used.

Eclipse IDE, Photon-version is used as a development environment for the DBCA. Using a simple new Spring Boot starter application, it is possible to find the web module responsible for Rest API endpoints and the module H2, to store data over memory (simplest database).

5.2 Classes and Packages

Packages for the prototype platform are made under certain concepts, some of these concepts are: separate things according to the type of “job” the class do; Make sure that classes are made using best Software Engineering practices and patterns, without losing the goal of the platform; use methods and attribute names with defaults pointed by Java patterns.

Packages made for this project are:

- **dbca**: main package of application. It has only the main class of application;
- **dbca.business**: package that contains all classes for drive business rules;
- **dbca.mockdata**: package that contains classes used to create mock data inside platform;
- **dbca.repositories**: it contains the classes for manage store and recover data from database;
- **dbca.resources**: set of classes that holds and defines the REST api endpoints and also do some data manipulation;
- **dbca.tests**: classes used to test some features and changes inside application during its development;
- **dbca.utils**: classes to handle and transform data, mainly between Weka API Instances and Java object.

Figure 9 shows the organization of packages, underlining “DBCA project” (with blue line), “DBCA Main class” (with orange line), “Car Data class” (with sky blue line), “ML model class” (with purple line) and “ML .model file” (with red line).

![Figure 9: Packages and classes developed](image)

### 5.3 Principal classes

The two principal classes of this platform are showed in Figures [10](image) and [11](image). The class named “DbcaData” defines the attributes already mentioned for Java class, allowing them the use of the object to send and receive information through the platform. The second class named “MachineLearningModel” is a class that loads and file with .model extension and keeps the object ready to classify new data. These classes are the core of the application and are responsible for define data and classify any new data from insurance companies.

The entire code of DBCA Application can be view at the link [https://github.com/anujroxx/DBCA](https://github.com/anujroxx/DBCA)
6 Evaluation

The expected output for the proposed model is get the driving behavior of an insurance policy holder. The model selected present better results for indexes and accuracy and the main purpose of platform is possible which proves the initial hypothesis that present the possible feasibility of using of Machine learning in cloud applications for identifying driver behavior.

After some use of data mocked inside the application, one test was made to prove that any insurance company or business can use the proposed platform to make predictions of driver behaviour and apply results to its pricing model. The next section will briefly explain develop in the Prototype testing.
6.1 Prototype testing - Research Experiment

After development, one test performed with the client-side application (Postman API Development Environment). In this experiment, data was sent by the postman and classified by DBCA application on the server side. The Figure 12 shows the JSON data selected to be sent to DBCA application, classified by it and sent back to Postman. Note that the attribute “driver behaviour” was sent with value “?” and returns with class predicted by ML model inside DBCA application, that was “unsafe”.

6.2 Discussion

Results present by Subsection 6.1 shows that it is possible to run and deploy a Machine Learning model over the cloud to classify driver behaviour using data from cars. It is possible to see the business model and the pricing of insurance companies more just. Alternatively, people that drives car with safe behaviour can have insurance service with less expensive than people that drive using car under the unsafe behaviour.
Another possible benefit coming from this model of business is that, due to its business model, possibly used by various insurance companies and business, government entities can use the data for allowing people with safe behaviour get discount when they need to buy fuel or thin related to vehicles. It is true, people with less risky behaviour uses less fuel and then produces less pollution wish can means economy in areas such as health, infrastructure and recovery of the natural environment.

7 Conclusion and Future Work

The main focus of this research is to utilise the cloud technology integrated with machine learning model to analyse driving behavior where the cloud technology is used as vehicular telematics platform to analyse and process the massive amount data generated by vehicle sensors and predict the driving behavior of driver.

This work is supported by insurance companies using User-Based-Insurance or Pay-How-You-Drive pricing model and other business related to driver behaviour. The proposed prediction model was made by using clustering and classifying methods and achieves good validation index values (Silhouette and DB indexes) and also good value for accuracy at the classifying step. Using this model, we are sure that many insurance companies
and other business can get a more personalized pricing model and with it, get benefits also for its clients.

Some limitations of this work are that the application depends of receiving data from platform evolving hardware (car, ECUS, sensors, OBD and others) and software inside some smartphone (the most comfortable way to get data from the car). The other thing is that a right approach is needed to re-train the ML method according to new data from new drivers (making it more robust and wise) but it can be a problem according to insurance companies policies.

At final, the overall result is robust enough with some variations of this work can be done while a business model of this research is under evaluation. Mentioned variation is the use of OBD device with the Internet connection and an approach using streaming analysis for data coming from cars.

References


Configuration Manual

MSc Research Project
Cloud Computing

Anuj Kumar
Student ID: X17157641

School of Computing
National College of Ireland

Supervisor: Victor Del Rosal
I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author’s written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:  
Date: 19th December 2018

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

- Attach a completed copy of this sheet to each project (including multiple copies).
- Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).
- You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Signature:  
Date:  
Penalty Applied (if applicable):
1 Pre-requisite

1. Java is required to run application.

2. Java Eclipse IDE (Integrated development Environment) to run the Java application [Murphy et al. (2006)].

3. Postman (API development environment) for testing the data flow [Schreier and Hayn (2018)].

2 Run Application

1. Need to Download/clone in eclipse workspace from the git hub link https://github.com/anujroxx/DBCA.

2. After installing Eclipse, use it to create IDE for Java application and import DBCA project in Eclipse IDE. we need to browse the root directory to select the location of DBCA.

3. After importing DBCA, Run DbcaApplication.java present in dbca package under the path "src/main/java" as shown in Figure 2 to initiate the proposed model.

4. After running the application, Sprint boot is initiated and that can seen in figure Figure 3 the application is running on localhost:8080 port.

5. Internal processing of the application is already explained in main report (Sec:5) how all classed are packages are used to process the data.

3 DBCA is used to Analyze the input data

1. Open the Postman API development environment to send a post request with input data on http://localhost:8080/dbca. Input data should be in Json format to be analyzed by machine learning model as shown in figure Figure 4.

2. It is observable in figure Figure 5 for the input values of "drivingBehaviour" in Figure 4 is marked as "?" and the return value for the the query driving behaviour is "unsafe".

3. Similarly in figure it is observable, for different input values of telematics data DBCA is proving results based on analysis as safe and unsafe.
Figure 1: Import DBCA in Eclipse IDE

Figure 2: Run DbcaApplication.java
Figure 3: Application listening on port localhost:8080

```javascript
POST http://localhost:8080/dbece

http://localhost:8080/dbece

<table>
<thead>
<tr>
<th>Params</th>
<th>Authorization</th>
<th>Headers</th>
<th>Body</th>
<th>Pre-request Script</th>
<th>Tests</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>form-data</td>
<td>x-www-form-urlencoded</td>
<td>raw</td>
<td>JSON (application/json)</td>
<td></td>
</tr>
</tbody>
</table>

```json
{
  "insuranceCompanyIdentification": "Inscomp1",
  "driverIdentification": "abcdef",
  "engineLoad": 0.425,
  "maf": 0.425,
  "speed": 0.3,
  "throttlePos": 0.257,
  "timingAdvance": 0.325,
  "driverBehavior": "safe",
  "date": null
}

```
Figure 4: Value sent using Postman

Figure 5: Return value for the Postman input
Figure 6: Return value for the Postman input

References
