

**INDIRA GANDHI
DELHI TECHNICAL UNIVERSITY
FOR WOMEN**



Title of Project

Driver Drowsiness Detection

Submitted By

Millennium Gupta - 013

A.Vaishnavi - 056

Rashmita Yadav - 058

Rhea Prasad - 061

Mahima Patel - 062

Under the supervision of

Mr. Rishabh Kaushal

Assistant Professor

Department of Information Technology

Abstract

In the present times driver sleepiness is a primary element of vehicle misfortunes and accidents in the world. A straightforward method of estimating driver weariness is measuring the status of the driver that means driver drowsiness. So, considering this to be very critical to respond about the drowsiness of the driver to save lives. This project is proposed towards manifesting a model of the drowsiness detection system. A concurrently working framework that can identify the state of the eye is what we have come up with, from the input feed that are the images. Although there are many approaches for measuring the drowsiness this approach is absolutely nonintrusive and does not feign the driver in any means, hence producing the exact situation of driver.

This system operates by observing the mouth and eyes of the driver. If the driver's eyes remain shut for more than a specified duration, the driver labelled drowsy. The accuracy of the algorithms was highest (for KNN = 83.4 % and for Decision Tree = 83.7 %) when we split our data into data from 13 videos and data from 9 videos respectively i.e. for 60:40 configuration. The programming is done in OpenCV for the apprehension of facial features.

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 1 |
| 1.1 | Problem Statement & Objectives | 2 |
| 1.2 | Scope | 2 |
| 1.3 | Limitation | 2 |
| 2 | Literature Review | 3 |
| 2.1 | What is Drowsiness? | 3 |
| 2.2 | Questions | 3 |
| 3 | Counter Measures | 4 |
| 3.1 | Vehicle-based measures | 4 |
| 3.2 | Physiological measures | 4 |
| 3.3 | Subjective Methods | 4 |
| 4 | Proposed Methodology | 5 |
| 4.1 | Behavioral Measures | 5 |
| 4.2 | Data Source and Pre-processing | 5 |
| 4.3 | Feature Computation | 6 |
| 4.3.1 | Feature Extraction | 6 |
| 4.3.2 | Eye Aspect Ratio | 6 |
| 4.3.3 | Mouth Aspect Ratio | 6 |
| 4.3.4 | Pupil Circularity | 6 |
| 4.3.5 | Mouth Over Eye Ratio | 7 |
| 4.4 | Feature Normalization | 7 |
| 5 | Exploration of Data | 8 |
| 5.1 | Metrics | 8 |
| 5.2 | Visualisations | 8 |
| 6 | System Descriptions | 9 |
| 6.1 | OpenCV | 9 |
| 6.2 | Dlib Libraries and their dependencies | 9 |
| 6.3 | Keras | 9 |
| 6.3.1 | Tensorflow | 9 |
| 6.4 | Python Libraries | 9 |
| 7 | Algorithms | 10 |
| 7.1 | Basic Classification Methods | 10 |
| 7.1.1 | Description | 10 |
| 7.1.2 | Logistic Regression | 11 |
| 7.1.3 | KNN | 12 |
| 7.1.4 | Naive Bayes | 13 |
| 7.1.5 | Decision Tree | 14 |
| 7.1.6 | Random Forest | 15 |
| 7.2 | Convolutional Neural Networks (CNN) | 16 |

| | | |
|----------|---|-----------|
| 7.2.1 | Description | 16 |
| 7.2.2 | CNN Parameters | 16 |
| 7.3 | Long Short-Term Memory Networks | 17 |
| 7.3.1 | Description | 17 |
| 7.3.2 | Internal memory space of RNN while processing new data: | 17 |
| 7.3.3 | Why LSTM? | 17 |
| 7.3.4 | LSTM Parameters | 18 |
| 7.3.5 | Graphical Representation Of Results | 19 |
| 7.4 | Comparison Of Algorithms | 20 |
| 7.4.1 | 50:50 Configuration | 20 |
| 7.4.2 | 60:40 configuration | 22 |
| 7.4.3 | 70:30 configuration | 24 |
| 7.4.4 | 80:20 configuration | 26 |
| 7.4.5 | 95:5configuration | 28 |
| 8 | Conclusion | 30 |
| 8.1 | Limitations | 30 |
| 8.2 | Future Scope | 31 |
| 9 | Related Work | 32 |

1 Introduction

Drowsiness Detection of a Driver is a car security technology that prevents accidents if suppose the driver gets drowsy. It is evident from our research work that “road accident due to driver fatigue” about 20 percent of all road accidents are related to it. Up to 50percent on certain specific roads. According to the “Data and Statistics division of CDC The Centers for Disease Control and Prevention- 76k injuries and 1.2k deaths annually is related to unattentive reported crashes”. Because of this hazard that drowsiness possess on the road, new ways necessitates to be produced for offsetting its affects.

This happens when an object or experience attracts a person’s attention apart from driving. Driver drowsiness involves no triggering effect but, is distinguished by a progressive removal of attention from the road. Both driver distraction and drowsiness, might have the identical effects, i.e., reduced driving performance, more delayed reaction time, and increased risk of collision. Based on Procurement of video of the camera that is in presence of driver, performs real-time processing of a video stream to understand the driver’s level of sleepiness if the drowsiness is calculated. We have used OpenCV for the apprehension of facial features. The main aim is to measure physical changes (i.e. open/closed eyes and mouth to detect fatigue) is well suited for real world conditions. Under this technique eye aspect ratio, mouth aspect ratio, etc. of a person is monitored and then we detect the drowsiness symptoms.

1.1 Problem Statement & Objectives

Our problem statement required studies from a variegated set of data ²⁵Due to the quintessentiality of this problem, it is important to produce a resolution for drowsiness detection, especially in the early phase to prevent mishappenings. We investigated the problem and took our objective as to measure physical changes (i.e. open/closed states of eyes and mouth) a well suited paradigms for real-world conditions in order to detect changes. We came up with a solution to this problem statement and decided to build a detection system, that will in identifying key attributes of drowsiness to distinguish them between alert or drowsy. In this effort of ours, we came up with a project idea of Driver Drowsiness Detection via two important detailing aspects the eye and mouth monitoring, using Python and OpenCV.

1.2 Scope

Our solution to this problem is to build a detection system that identifies key attributes of drowsiness to detect whether alert or drowsy, by continuously monitoring the eyes and mouth of the driver one can detect the sleepy state of the driver.

1.3 Limitation

If the user uses specs than it is difficult to detect the actual state of eye. As it is highly dependent on light, hence reflection of specs may give output for a closed eye as an opened eye. Hence for this purpose the eye needs to be close to the camera in order to avoid light. If multiple face arises in the window then the camera may detect more than one face. There is chance for Undesired output due to different faces at different conditions. So, we need to make sure that only the driver face come within the range of camera. The speed of detection is reduced due to operation on multiple faces.

2 Literature Review

2.1 What is Drowsiness?

Drowsiness is represented as a reduced level of information represented by sleepiness and difficulty in staying alert but the person wakes with unadulterated excitement by stimuli. It might be created by inadequacy of sleep, medicine, or a cerebral distribution. It is frequently the result of unattentiveness which can create both mental and physical hazards. Physical weakness, or muscle fatigue, can be a short dynamic failure of a muscle to perform normally. The origin of mental fatigue amid any rational action is growing and relies on an individual's subconscious function, moreover upon various elements, for instance, lack of rest and general well-being. Mental exhaustion has additionally been appeared to diminish physical performance. It can present as lethargy, dormancy, or ordered consideration dizziness. In the past times according to accessible data driver drowsiness has grown to be one of the real causes for street accidents implying demise and severe injuries and loss of economy. It can be correctly stated that a driver who falls asleep or becomes unattentive of his conditions while driving his vehicle is at a brink of losing his control, which results in accidents, crashes and loss of live and property.

2.2 Questions

Following information from the driver about more than one incident which may or may not be critical:

- How much sleep did you get before this drive?
- Were you more tired once the adrenaline wore off?
- When you stopped driving was it as you were drowsy or another reason?
- Did you nap at all when you stopped?
- Do you think that you drove at all with your eyes closed? for how long?
- Were you tired when you started?
- What speed were you going?
- What time of the day or night was it?
- What were the road conditions?
- What were the traffic conditions?
- What type of road was it? E.g. curvy, straight, two-lane, etc.
- At what point during your drive did this happen?
- Where were you going?
- How long did you keep driving?
- What did you do to try to become more alert?
- What else would have helped you become more alert?

3 Counter Measures

Our study states that if a mishap happens, it's reason can be one of these accompanying classes: (1) humans(driver) (2) vehicular (3) surrounding factor A 91% of car accidents came from driver errors, while the other classes corresponded to 4% for vehicular and 5% for the surrounding factors. Various Measures for measurement of drowsiness are as follows:

3.1 Vehicle-based measures

Vehicular measures survey the path position, which monitors the vehicle's position in order to identify the path markings, in order to determine the driver's weakness, and acquire information of movement of the steering wheel to characterize fatigue level from low to high.

Disadvantages:

- Vehicle-based measures depend on the geometry of road which can at times activate the alarming system unnecessarily.

- The way the current driver drives should be learned and modeled so that the system is efficient.

- Issues like micro-sleeping which can happen on straight highways isn't detected.

3.2 Physiological measures

These are the main metrics of the physical changes occurring in our body due to fatigue. These changes can be measured by their respective instruments like ECG, EMG, EOG, EEG.

Disadvantages: -Several electrodes are to be placed on head, chest and face which is not at all a convenient for a driver as it is very annoying. -For perfect result, extremely careful placement is required.

3.3 Subjective Methods

These methods involve assessment of the level of drowsiness of driver by rating them based on a form of questionnaire. These ratings are self-evaluated or evaluated by some experts who observe the driver while he/she is in action. To detect changes in the drowsiness state of the driver, a pre experimental, a mid-experimental, and a post experimental Karolinska Scale for Sleepiness tests were conducted. Methods such as Stanford Scale for sleepiness and Karolinska Sleepiness Scale are the two scale that are most widely utilized subjective measures. SSS is a 7-point measurement scale that describe the current state of drowsiness of an individual which is used to categorize driver drowsiness into two classes. This is because of each scale is closely related. On the other hand, KSS is a 9-pointer scale. As a contrast to Stanford Scale, KSS is a robust scale that is capable of categorizing the drowsiness of driver into different levels.

4 Proposed Methodology

4.1 Behavioral Measures

The main aim is to measure physical changes (i.e. open/closed eyes and mouth to detect fatigue) is well suited for real world conditions. Under this technique eye aspect ratio, mouth aspect ratio, etc. of a person is monitored and then we detect the drowsiness symptoms

4.2 Data Source and Pre-processing

A team of research students from the University of Texas at Arlington created the REAL-LIFE DROWSINESS DATASET (RLDD) for detection of multi-stage drowsiness. So, for training and testing, we used this dataset. They had uploaded data of 60 unique participants comprising of 30 hours of video. The data created by the research team was uploaded in form of zip files, each file consists of five or six folders in which there are videos of unique individuals named as 0 for alert state video and 10 for drowsy state video. From that dataset, sufficient amount of data of some participants for both the states i.e. alert and drowsy, was chosen. For each video, OpenCV was used to extract 1 frame/second, generally starting at a 3-minute mark up till the end of the video. Size of each clip was around 10 minutes, so around 240 frames/video were extracted. We labeled the frames from alert videos as 0 and from the drowsy videos as 1. There were total 68 landmarks/frame but we only kept the landmarks for features that are eyes and mouth. These were some important data points that helped in extraction of features for our model.

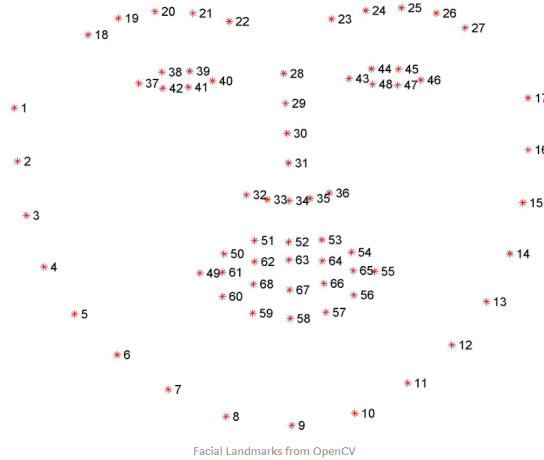


Figure 1: 68 Total Landmarks

4.3 Feature Computation

4.3.1 Feature Extraction

We ventured into development of features suitable for the classification model. We tested and hypothesized several features and included pupil circularity, mouth aspect ratio, and finally, mouth aspect ratio over eye aspect ratio as our main features from which we were able to conclude for our finalized model.

4.3.2 Eye Aspect Ratio

It is the ratio of the height of the eyes to the diameter of the eyes. The calculation of the length of the eyes is done by taking average of two distinct vertical lines drawn across the eyes, p_1 and p_4 are diametrically end points horizontally.

$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2 \|p_1 - p_4\|} \quad (1)$$

We hypothesized that when driver is feeling drowsy, his/her eyes is most likely

to get smaller than normal and he/she will tend to blink more. Therefore, our model should do the job of predicting the class as drowsy in case Eye-Aspect-Ratio for a driver over successive frames declines i.e. their eyes starts to become more closed or they start to blink faster.

4.3.3 Mouth Aspect Ratio

On the same lines as EAR, MAR also measures ratio of height of mouth to the diameter of mouth. Our hypothesis was that as a driver becomes drowsy, they are likely to yawn more and lose control over the actions of their mouths, which makes the MAR higher than usual. Here, EF and AB are lines joining vertical and horizontal end points.

$$MAR = \frac{|EF|}{|AB|} \quad (2)$$

4.3.4 Pupil Circularity

It is a metric which is interrelated to EAR, but it places a greater emphasis on the pupil and not the entire eye. For instance, someone who has half-opened their eyes or almost closed is going to have a much lower PUC value as compared someone who has fully opened their eyes due to the squared term in the denominator. Just like the EAR, the hypothesis was that when a driver feels drowsy, their pupil circularity will decline.

$$\text{Circularity} = \frac{4 * \pi * \text{Area}}{\text{perimeter}^2} \quad (3)$$

$$\text{Area} = \left(\frac{\text{Distance}(p_2, p_5)}{2} \right)^2 * \pi \quad (4)$$

$$\text{Perimeter} = D(p1, p2) + D(p2, p3) + D(p3, p4) + D(p4, p5) + D(p5, p6) + D(p6, p1) \quad (5)$$

4.3.5 Mouth Over Eye Ratio

MOE is the ratio of Mouth-Aspect-Ratio to the Eye-Aspect-Ratio. The reason we make use of this feature is simply that EAR and MAR behave in oppositely if individual changes his/her states. In contrast to EAR and MAR, MOE responds strongly to changes as it captures even slightest changes in both EAR and MAR and will overemphasize the changes as the numerator and denominator move in opposite directions. As the MAR acts as numerator and EAR is the denominator, our hypothesized that as the driver gets drowsy, the MOE tends increase

$$MOE = \frac{MAR}{EAR} \quad (6)$$

4.4 Feature Normalization

When we tested our models with the main features, whenever we split randomly the frames in our training and test, the model yielded high accuracy. However, whenever the frames were split by individuals (i.e. an individual that is in testing set will not be in training set), our performance of our model would be poor. This means that our model was having trouble with newer faces and the main reason for this issue was as simple as that each individual has different set of core features in their alert state because person A may naturally have smaller eyes as compared to person B. That is why, that it's important to normalize these features for each individual in order to yield better results. For normalizing the features for each driver, the first three frames from each driver's alert video were taken and we used them as a baseline for our normalization. Then we calculated the standard deviation and mean for each of the feature from frames and used those for normalizing each feature for each and every driver. Mathematically, this is what the normalization equation should look like:

$$\text{Normalised Feature}_{n,m} = \frac{\text{Feature}_{n,m} - \mu_{n,m}}{\sigma_{n,m}} \quad (7)$$

where n is feature, m is the person. the standard deviation and mean are taken from first three frames of 'Alert' state. After normalizing all of the four main features, our feature set comprised of eight features, each main feature being

complemented by its normalized version. After testing all eight features in our models, the results automatically improved significantly.

5 Exploration of Data

5.1 Metrics

| Parameter | 50:50 | 60:40 | 70:30 | 80:20 | 95:5 |
|---------------|-------|-------|-------|-------|-------|
| MAR_N | 0.253 | 0.218 | 0.218 | 0.232 | 0.220 |
| MAR | 0.137 | 0.184 | 0.204 | 0.197 | 0.213 |
| Circularity | 0.124 | 0.112 | 0.09 | 0.106 | 0.08 |
| MOE_N | 0.123 | 0.148 | 0.145 | 0.148 | 0.14 |
| MOE | 0.120 | 0.093 | 0.122 | 0.094 | 0.100 |
| Circularity_N | 0.088 | 0.091 | 0.074 | 0.106 | 0.078 |
| EAR | 0.077 | 0.087 | 0.06 | 0.060 | 0.060 |
| EAR_N | 0.732 | 0.062 | 0.082 | 0.083 | 0.099 |

Table 1: Feature Importance according to different configurations

5.2 Visualisations

We analyzed feature importance using Random Forest Model, out of all features, as evident from table, we found MAR (Mouth Aspect Ratio) is most important, since we frequently yawn more when we are feeling sleepy and normalization of MAR feature made it more clear that it better indicates the alertness or drowsiness in various participants.

6 System Descriptions

We used following tools and technologies:

6.1 OpenCV

OpenCV is an open source PC vision library which was intended for computational proficiency and also comprising of a high spotlight on constant and continuous picture discovery. One of the preliminary objectives OpenCV is to provide with an easy to-utilize PC vision framework which encourages individuals to construct exceptionally refined vision applications quickly. The OpenCV library, containing more than 500 capacities, ranges numerous territories in vision. Since PC vision and AI frequently goes connected, OpenCV likewise has a total, broadly useful, Machine Learning Library (MLL). This sub library is centered around factual example acknowledgment and grouping. The MLL is extremely valuable for the vision capacities that are the premise of OpenCV's handiness, yet is sufficiently general to be utilized for any AI issue.

6.2 Dlib Libraries and their dependencies

It is a C++ toolbox for AI, it likewise gives a python API to utilize it in your python applications. Probably the best component is an incredible documentation for C++ and Python API. It is utilized in the code to identify faces and get facial tourist spots facilitates particularly the 12 focuses which characterize the two eyes left and right. S

6.3 Keras

To construct our classification model, Keras utilizes TensorFlow as backend.

6.3.1 Tensorflow

It is an open-source programming library for programming of dataflow over a scope of errands. It is a math library, which is also utilized for AI applications. TensorFlow calculations are computed as stateful charts of dataflow. The name TensorFlow gets from the tasks that such neural systems perform on multidimensional information exhibits. These exhibits are alluded to as "tensors".

6.4 Python Libraries

Python is plainly perhaps the best language for AI. Python contains exceptional libraries for AI in particular SciPy, pandas and NumPy which are incredible for direct variable-based math and becoming acquainted with portion strategies for AI. The language is extraordinary to utilize when working with AI calculations and has simple punctuation generally.

7 Algorithms

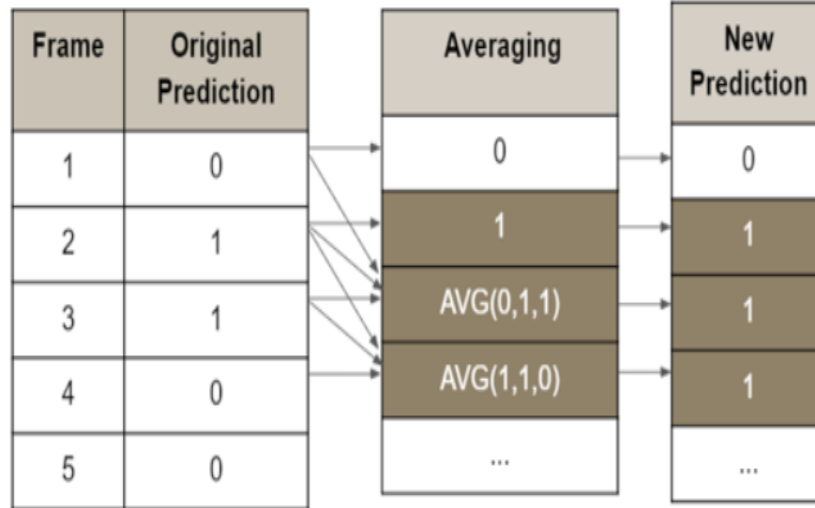
7.1 Basic Classification Methods

7.1.1 Description

After all the features are extracted and normalized, we want to try out some modeling techniques, starting from the most basic classification models like logistic regression, Naive Bayes, KNN, Decision Tree, Random Forest Model moving on to more complex models which includes Large Short Term Memory and CNN.

For prediction of the label of individual frame in sequence:

The method which we use to sequence the basic classification methods was to get the average of prediction results along with prediction results from the previous two frames. Our data set has been divided into training data and test data based on participants and also the data set and points are in order of sequence of time. Averaging is used in such case and it allows us to deliver predictions with more accuracy.



Introducing Sequence to Basic Classification Models

Figure 2: Averaging

7.1.2 Logistic Regression

The Logistic regression algorithm comes under the class of Supervised Learning methods. It may be used for Regression or for Classification problems, but mostly used in Classification problems. Logistic regression is generally used in predicting the categorically dependent variables using independent variables. The output of Logistic Regression problem can only lie between 0 and 1. This regression is used where we need the probabilities between two classes. Like, whether it is drowsy or not, either 1 or 0, false or true etc. In logistic regression, weighted sum of inputs is passed to a function, called activation function, that has the ability to map values into the range of 0 and 1. Such a function is called sigmoid function.

Algorithm 1 Logistic Regression

```
1: procedure MYFUNCTION( $A, S$ )
2:   for ( $j=1$  to  $n$ ) do
3:      $B = 0$ 
4:   end for  $j = \min S$ 
5:    $j = \text{smallest } (A, a)$ 
   return min of  $S$ 
6: end procedure
```

| Logistic Regression | 50:50 | 60:40 | 70:30 | 80:20 | 95:5 |
|---------------------|-------|-------|-------|-------|-------|
| Accuracy | 0.699 | 0.747 | 0.689 | 0.723 | 0.595 |
| F1 score | 0.622 | 0.742 | 0.677 | 0.707 | 0.484 |
| ROC | 0.705 | 0.805 | 0.736 | 0.766 | 0.592 |

Table 2: Comparison of Different Configurations for Logistic Regression

For logistic regression, Best accuracy score is for 60:40 configuration (0.747), Best F1 score is for 60:40 configuration(0.742) and Best AUC (Area Under ROC Curve) is for 60:40 configuration(0.805)

7.1.3 KNN

K-Nearest Neighbors is an algorithm based on Supervised Learning technique. K-NN algorithm considers the similarity in between new case or data or available cases and put this new case into a category which is similar to the available categories. This algo stores all of the available data and tries to classify the new data points based on similarity. This means that whenever any new data is encountered then that can also be easily classified in a well-suited class with the use of K- NN algorithm. K-NN is used for Classification as well as for Regression. K-NN is non-parametric, that means it doesn't make presumptions on the underlying data. It is also termed as a lazy-learner algo as it stores the dataset and at the time of classification itself, it performs an action on dataset, instead of learning from the training set immediately. This algorithm at the training phase stores the dataset and whenever it receives a new data, it classifies the data into a class that is much similar to the new data.

Algorithm 2 KNN

```
1: procedure MYFUNCTION( $A, B$ )
2:   for ( $j=1$  to  $n$ ) do
3:      $d = A$  to  $a$ 
4:   end for
5:    $j = \text{smallest}(A, a)$ 
   return majoritylabel
6: end procedure
```

| Parameter | 50:50 | 60:40 | 70:30 | 80:20 | 95:5 |
|-----------|-------|-------|-------|-------|-------|
| Accuracy | 0.745 | 0.834 | 0.783 | 0.717 | 0.729 |
| F1 score | 0.783 | 0.803 | 0.776 | 0.631 | 0.628 |
| ROC | 0.759 | 0.908 | 0.776 | 0.746 | 0.706 |

Table 3: Comparison of Different Configurations for KNN

For KNN, Best accuracy score is for 60:40 configuration (0.834), Best F1 score is for 60:40 configuration(0.803) and Best AUC (Area Under ROC Curve) is for 60:40 configuration(0.908)

7.1.4 Naive Bayes

It is a classification algorithm used for 2 class and multi class problems. It is easy to understand when we use binary values (0 or 1). It is called naive Bayes because it involves calculation of probabilities for each hypothesis. This is done to make our calculations traceable. Instead of calculating values for each attribute value $p(d1, d2, d3|h)$, it is assumed as conditionally independent given the target value is calculated as $p(d1|h)*p(d2|h)$ and so on. Possible representation for "naive Bayes" using probabilities is herewith:

- 1)"Probability of Class" - can be defined as the probability of each class in the training dataset.
- 2)"Probability of Conditions" - It is referred to as the conditional probability of every input value given to every class value.

Algorithm 3 Naive Bayes

Input: Training dataset p

```

1:  $A = (A1, A2, A3, An)$ 
2: procedure MEAN DEVIATION( $A$ )
3:   for ( $j=1$  to  $n$ ) do
4:      $d = (A \text{ to } An)$ 
5:   end for
6:    $j = \text{smallest}(A, An)$ 
   return likelihood
7: end procedure

```

| Parameter | 50:50 | 60:40 | 70:30 | 80:20 | 95:5 |
|-----------|-------|-------|-------|-------|-------|
| Accuracy | 0.671 | 0.675 | 0.631 | 0.627 | 0.487 |
| F1 score | 0.700 | 0.685 | 0.644 | 0.690 | 0.621 |
| ROC | 0.659 | 0.709 | 0.627 | 0.591 | 0.476 |

Table 4: Comparison of Different Configurations for Naive Bayes

For Naive Bayes, Best accuracy score is for 60:40 configuration (0.675), Best F1 score is for 60:40 configuration(0.685) and Best AUC (Area Under ROC Curve) is for 60:40 configuration(0.709)

7.1.5 Decision Tree

Decision Tree is a Supervised learning for unraveling Classification issues. It is a tree-organized classifier, where the interior hubs speak to the highlights of dataset, branches speak to the choice standards and each leaf hub speaks to the result. In a Decision tree, there are two hubs, Decision Node what's more, Leaf Node. Choice hubs are utilized to settle on a choice and for having numerous branches, while Leaf hubs are the yield of those choices what's more, for not containing any longer branches. The choices or the test are performed dependent on highlights of the given dataset. It is called as a choice tree since it is like a tree, it begins with the root hub, which grows further into branches and afterward become a tree-like structure. A choice tree essentially poses an inquiry, and dependent on answer whether (Yes/No), it further parts the tree into subtrees.

Algorithm 4 Decision Tree

Input: Set of classified dataset S

Output: Decision Tree

```

1: A = (A1,A2,A3,An)
2: procedure BUILDTREE(A)
3:   for (j=1 to n) do
4:     maxgain =0
5:     splitA = null
6:     e = Entropy (attri)
7:     g = Infogain(A,e)
8:   end for
9:   partition (S,splitA)
   return partition
10: end procedure

```

| Parameter | 50:50 | 60:40 | 70:30 | 80:20 | 95:5 |
|-----------|-------|-------|-------|-------|-------|
| Accuracy | 0.756 | 0.837 | 0.787 | 0.719 | 0.709 |
| F1 score | 0.781 | 0.808 | 0.744 | 0.710 | 0.595 |
| ROC | 0.746 | 0.916 | 0.849 | 0.771 | 0.714 |

Table 5: Comparison of Different Algorithms for Decision Tree

For Decision Tree, Best accuracy score is for 60:40 configuration (0.837), Best F1 score is for 60:40 configuration(0.808) and Best AUC (Area Under ROC Curve) is for 60:40 configuration(0.916)

7.1.6 Random Forest

The main reasons of using random forest algorithm are: It may be used for both classification and regression problems. Overfitting is a critical problem which can make the results worse, but with random forest algorithm, if there are enough no. of trees in forest, the classifier will not be able to overfit this model. It also handles missing values. This classifier can also be used for categorical values.

Algorithm 5 Random Forest

```

1:  $A = (A_1, A_2, A_3, A_n)$ 
2: procedure MYFUNCTION( $A, B$ )
3:   for ( $j=1$  to  $n$ ) do
4:      $D = 0$ 
5:   end for( $Q_{ip}$ )
6:    $Q = \text{add}(D, i)$ 
   return Trees
7: end procedure

```

| Parameter | 50:50 | 60:40 | 70:30 | 80:20 | 95:5 |
|-----------|-------|-------|-------|-------|-------|
| Accuracy | 0.787 | 0.823 | 0.776 | 0.697 | 0.717 |
| F1 score | 0.819 | 0.844 | 0.080 | 0.753 | 0.636 |
| ROC | 0.860 | 0.939 | 0.900 | 0.836 | 0.727 |

Table 6: Comparison of Different Algorithms for Random Forest

For Random Forest, Best accuracy score is for 60:40 configuration (0.823), Best F1 score is for 60:40 configuration(0.844) and Best AUC (Area Under ROC Curve) is for 60:40 configuration(0.939)

7.2 Convolutional Neural Networks (CNN)

7.2.1 Description

It's a type of deep neural network which works very well for image classification. CNN generally comprises of an input layer, output layer and a hidden layer which might have multiple no. of layers. CNN is generally used for analyzing the image data. We decided to build a 1 dimension CNN and send some numerical features along with it as a input for better understanding of spatial relationship between features of 2 states. CNN model, in total have 5 layers: 1 convolution layer, 1 flatten layer, 2 fully connected dense layer and 1 dropout layer. Flatten layer flattens the the output from convolution layer and makes it linear before it is passed to the first dense layer. Dropout layer is randomly dropped to 20% of output nodes from second dense layer in order to prevent the model from overfitting. Dense layer at end has single output node which gives 0 as output for alert state and 1 for drowsy state.

7.2.2 CNN Parameters

| | |
|---------------------|---------------------|
| Activation Function | Relu Sigmoid |
| Optimiser | Adam |
| Loss Function | Binary Crossentropy |
| Number Of Epochs | 100 |
| Learning Rate | 0.00001 |

Table 7: CNN Parameters

| CNN | 50:50 | 60:40 | 70:30 | 80:20 | 95:5 |
|----------|---------|-------|-------|-------|-------|
| Accuracy | 0.725 | 0.759 | 0.748 | 0.695 | 0.617 |
| F1 score | 0.0.773 | 0.780 | 0.762 | 0.711 | 0.59 |
| ROC | 0.724 | 0.844 | 0.809 | 0.735 | 0.626 |

Table 8: Comparison of Different Configurations for CNN

For Convolutional Neural Network, Best accuracy score is for 60:40 configuration (0.759), Best F1 score is for 60:40 configuration (0.780) and Best AUC (Area Under ROC Curve) is for 60:40 configuration (0.844)

7.3 Long Short-Term Memory Networks

7.3.1 Description

There is another way of dealing with sequential data that is by using a LSTM model. LSTM networks are a special kind of Recurrent Neural Networks (RNN), which is capable of learning the long-term dependencies in data. RNN are the feed-back neural networks which has memory internally that allows the information to persist.

7.3.2 Internal memory space of RNN while processing new data:

While taking a decision, RNNs considers output as well as the current input, which it has learned from the inputs obtained in earlier situations. This is the major difference between RNNs and other neural networks. Except RNN remaining neural networks are independent to each other.

7.3.3 Why LSTM?

We planned to use LSTM network as it allows us to study the long sequences without thinking about the gradient vanishing problems which are faced when we use traditional RNNs. Within the LSTM network itself, there are in total 3 gates for each time. step: Input gate, Output Gate and forget gate. Forget Gate: from the name itself, it is clear that this gate tries to forget the part of memory from it's previous output.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (8)$$

Input Gate: This gate takes the decision about what should be kept away from the input in order to modify the memory.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (9)$$

$$Ct = \tanh(WC \cdot [h_{t-1}, x_t] + b_C) \quad (10)$$

Output Gate: This gate takes the decision about what will be the output obtained by merging the input and memory.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (11)$$

$$h_t = o_t * \tanh(C_t) \quad (12)$$

Firstly, we will convert our videos into a batch of data. Then, each batch will be sent via a fully connected layer which has 1024 units which are and we are also using the the sigmoid function. The following layer is the LSTM layer which has 512 units which are hidden and it is followed by 3 FC layers till we reach the final layer.

7.3.4 LSTM Parameters

| | |
|------------------|---------|
| Number Of Epochs | 50 |
| Learning Rate | 0.00005 |
| Timestamp | 5 |

Table 9: LSTM Parameters

| Parameter | 50:50 | 60:40 | 70:30 | 80:20 | 95:5 |
|-----------|-------|-------|-------|-------|-------|
| Accuracy | 0.637 | 0.657 | 0.625 | 0.520 | 0.463 |
| F1 score | 0.651 | 0.695 | 0.659 | 0.660 | 0.557 |
| ROC | 0.621 | 0.646 | 0.584 | 0.549 | 0.440 |

Table 10: Comparison of Different Configurations for LSTM

For LSTM, Best accuracy score is for 60:40 configuration (0.657), Best F1 score is for 60:40 configuration(0.695) and Best AUC (Area Under ROC Curve) is for 60:40 configuration(0.646)

7.3.5 Graphical Representation Of Results

ROC CURVE (Between true positive rate and false positive rate)

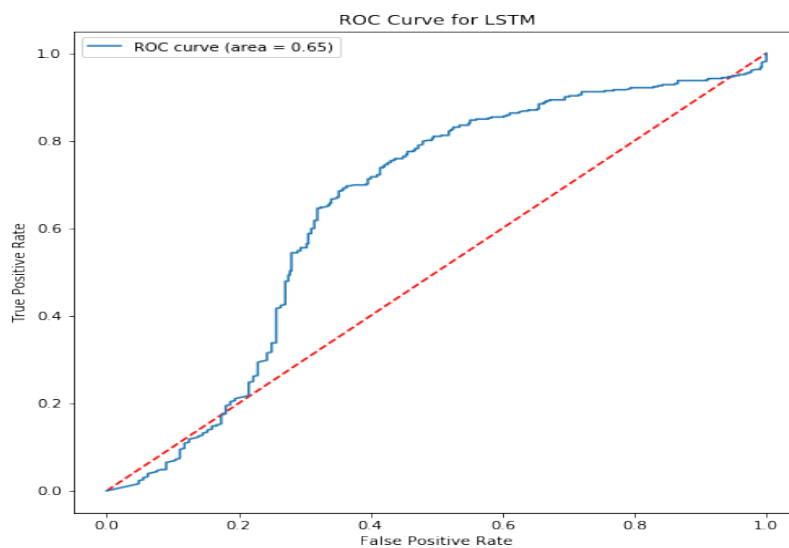


Figure 3: LSTM ROC curve for 60:40 configuration

Calibration CURVE (Between mean predicted values and fraction of positives)

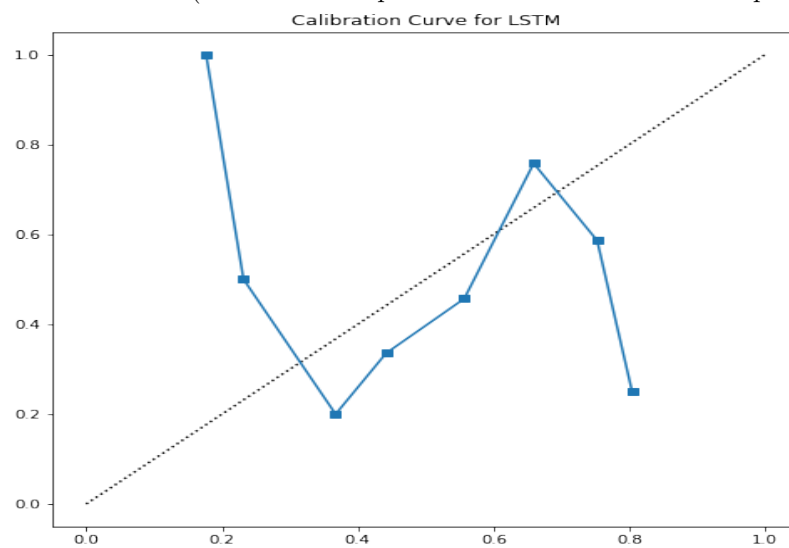


Figure 4: LSTM calibration curve for 60:40 configuration

7.4 Comparison Of Algorithms

7.4.1 50:50 Configuration

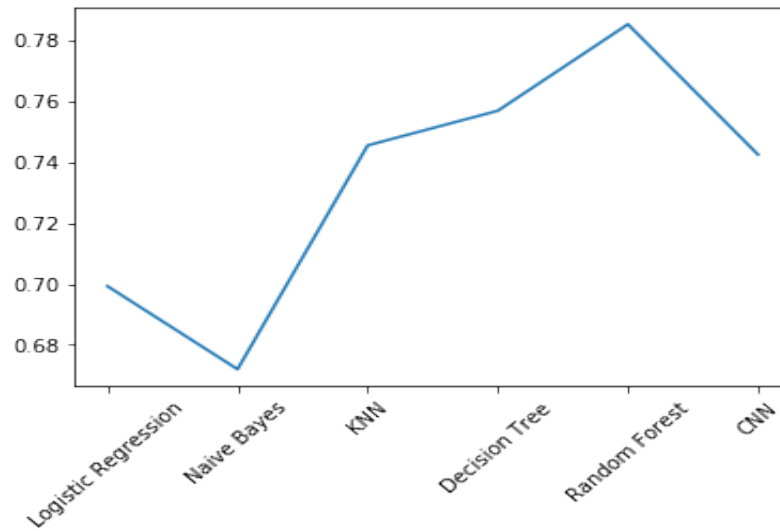


Figure 5: accuracy curve comparison for 50:50 configuration

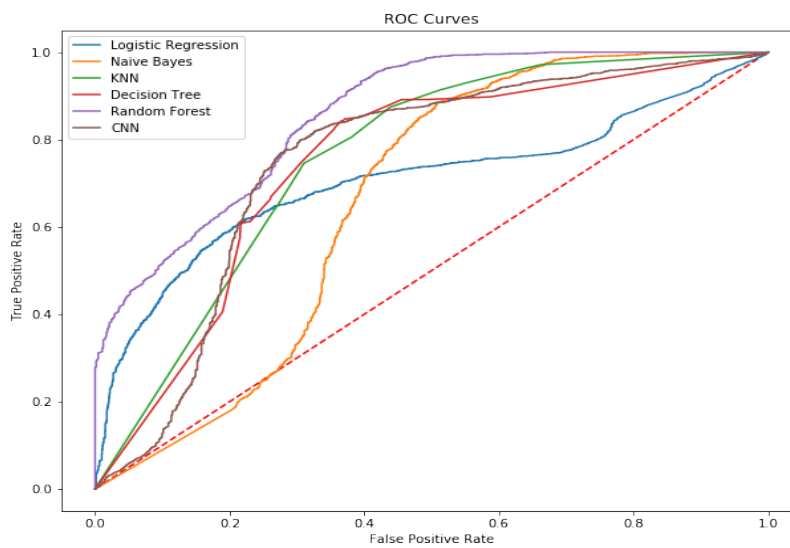


Figure 6: ROC curve comparison for 50:50 configuration

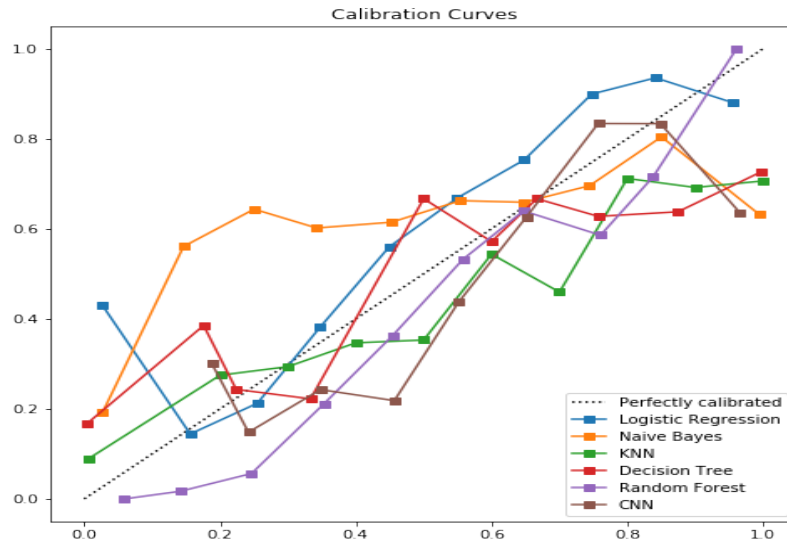


Figure 7: Calibration curve comparison for 50:50 configuration

For 50:50 configuration, accuracy of Random Tree(0.787) and Decision Tree(0.756) was highest and lowest for logistic regression(0.699) and Naive Bayes(0.671). AUC score from the ROC curve was highest for Random Forest and KNN.

7.4.2 60:40 configuration

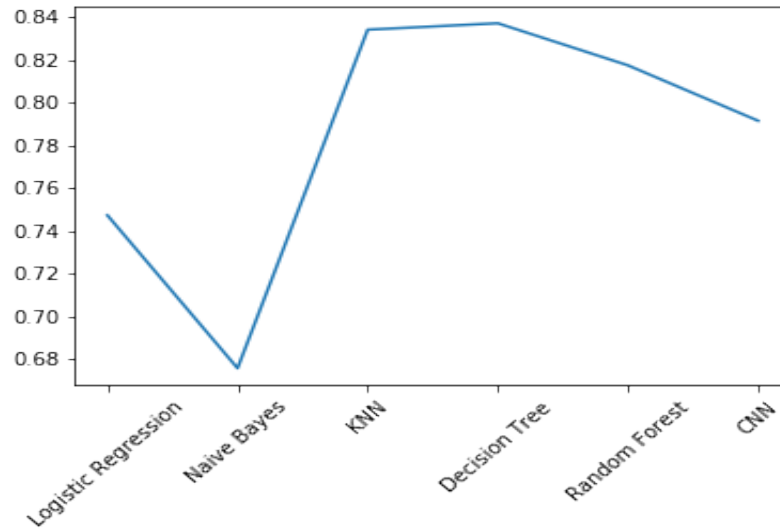


Figure 8: accuracy curve comparison for 60:40 configuration

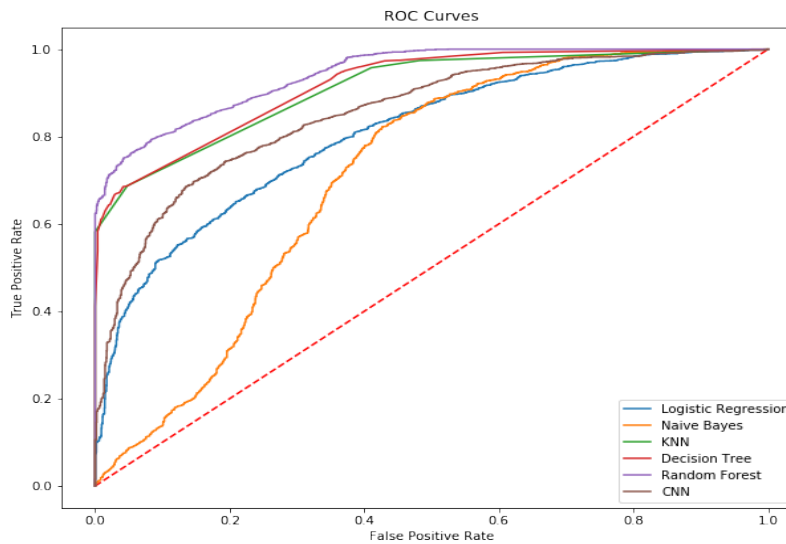


Figure 9: ROC curve comparison for 60:40 configuration

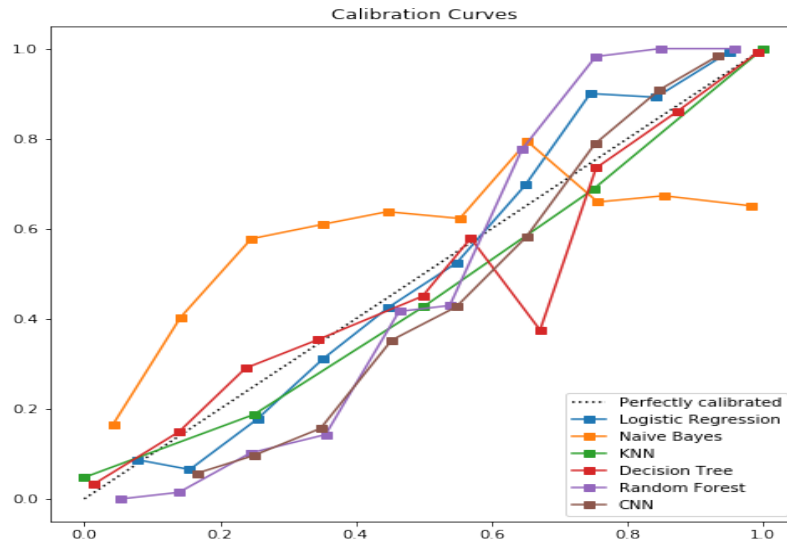


Figure 10: Calibration curve comparison for 60:40 configuration

For 60:40 configuration, accuracy of Decision Tree(0.837) and KNN(0.834) was highest and lowest for logistic regression(0.747) and Naive Bayes(0.675). AUC score from the ROC curve was highest for Random Forest and Decision tree.

7.4.3 70:30 configuration

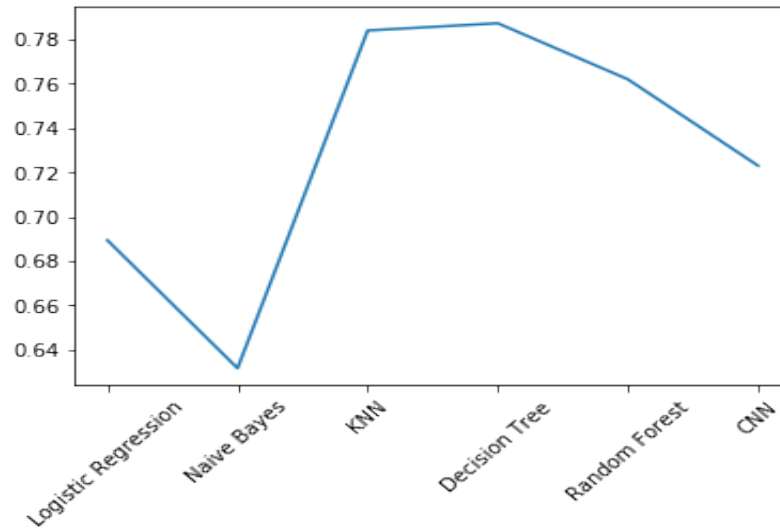


Figure 11: Accuracy curve comparison for 70:30 configuration

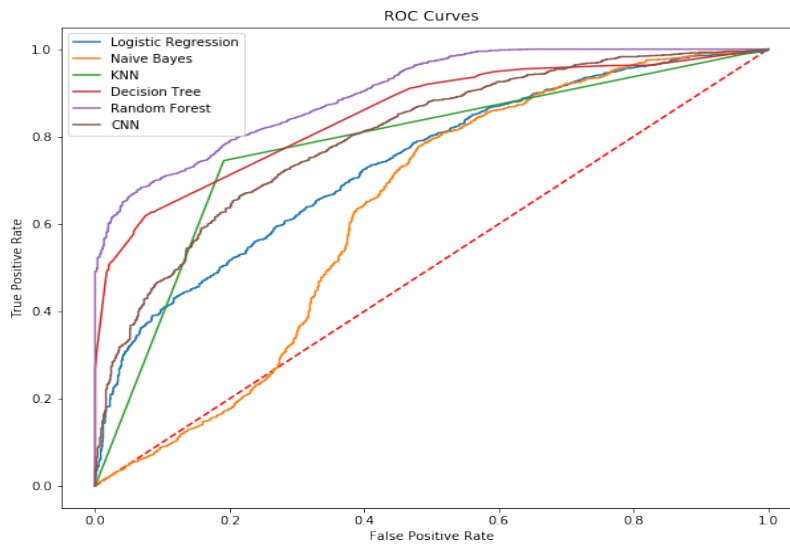


Figure 12: ROC curve comparison for 70:30 configuration

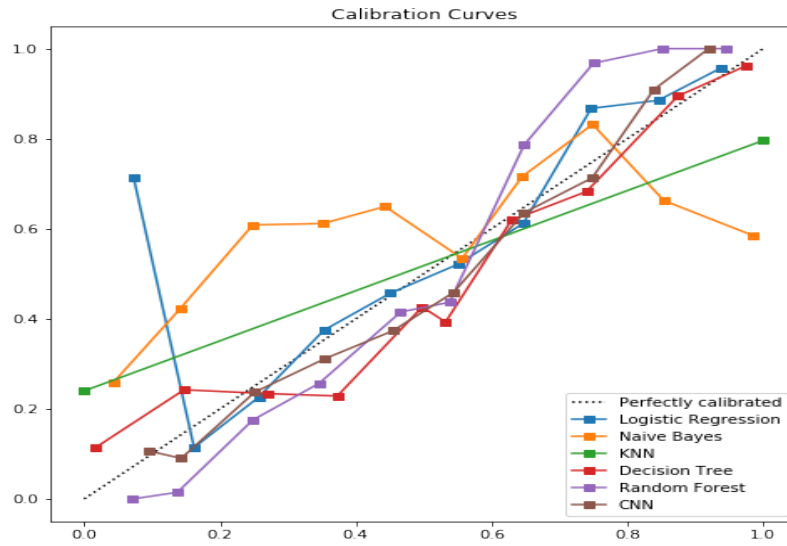


Figure 13: Calibration curve comparison for 70:30 configuration

For 70:30 configuration, accuracy of KNN(0.783) and Decision Tree(0.787) was highest and lowest for logistic regression(0.689) and Naive Bayes(0.631). AUC score for ROC curve was highest for Random Forest and Decision Tree.

7.4.4 80:20 configuration

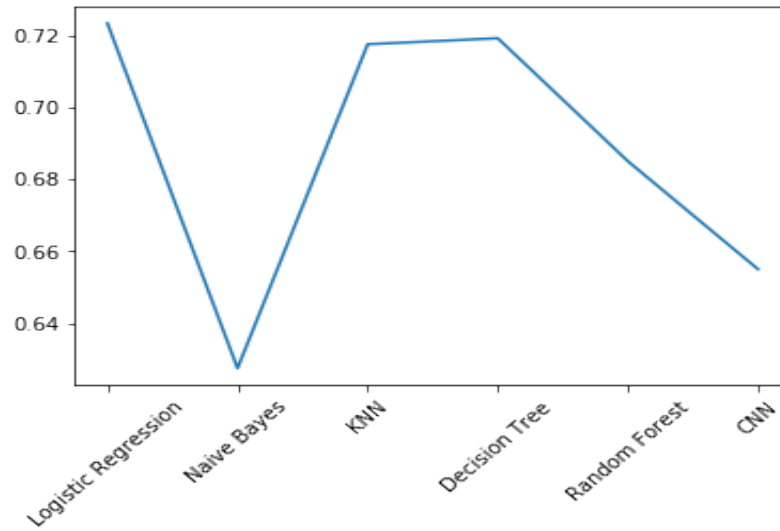


Figure 14: Accuracy curve comparison for 80:20 configuration

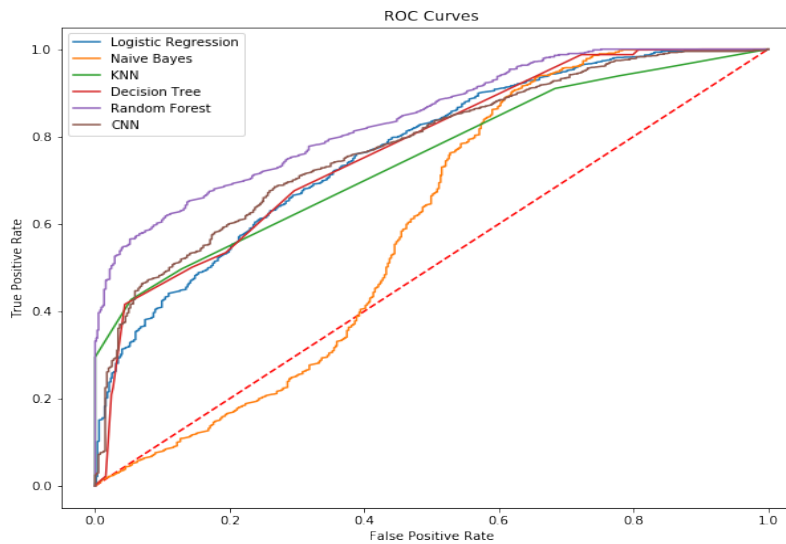


Figure 15: ROC curve comparison for 80:20 configuration

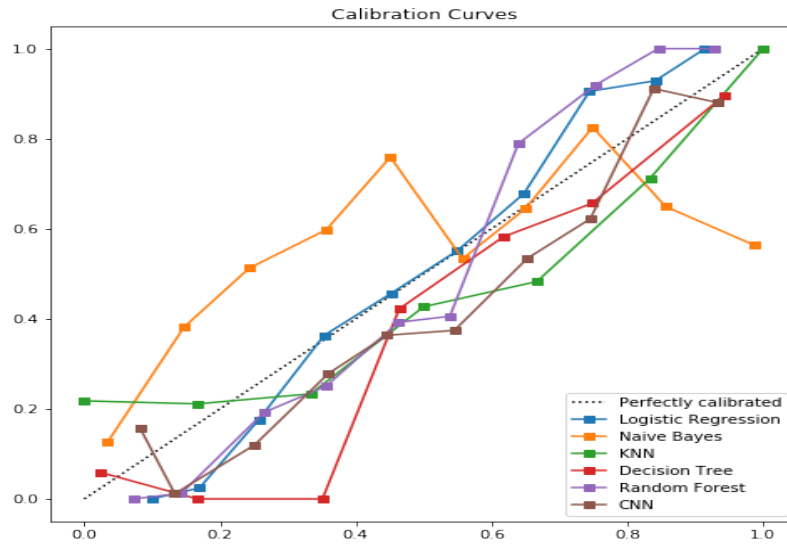


Figure 16: Calibration curve comparison for 80:20 configuration

For 80:20 configuration, accuracy of Logistic regression (0.723) and Decision Tree (0.719) was highest and lowest for CNN(0.695) and Naive Bayes(0.627).AUC score of ROC curve was highest for Random Forest and Decision Tree.

7.4.5 95:5 configuration

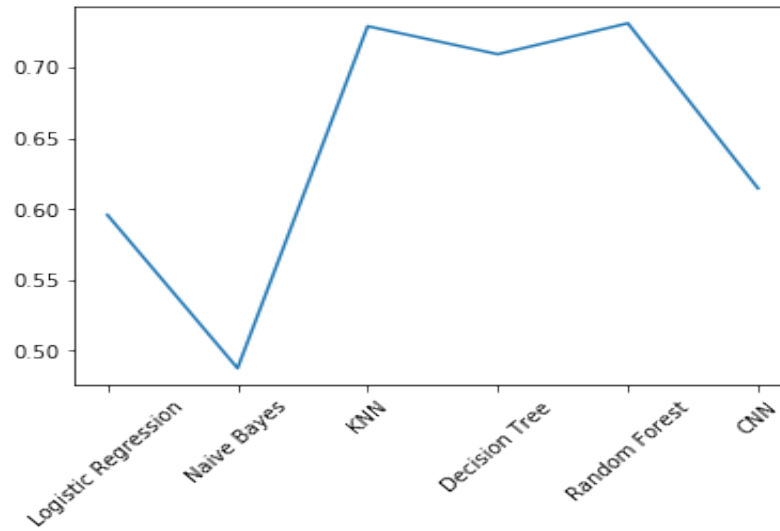


Figure 17: Accuracy curve comparison for 95:5 configuration

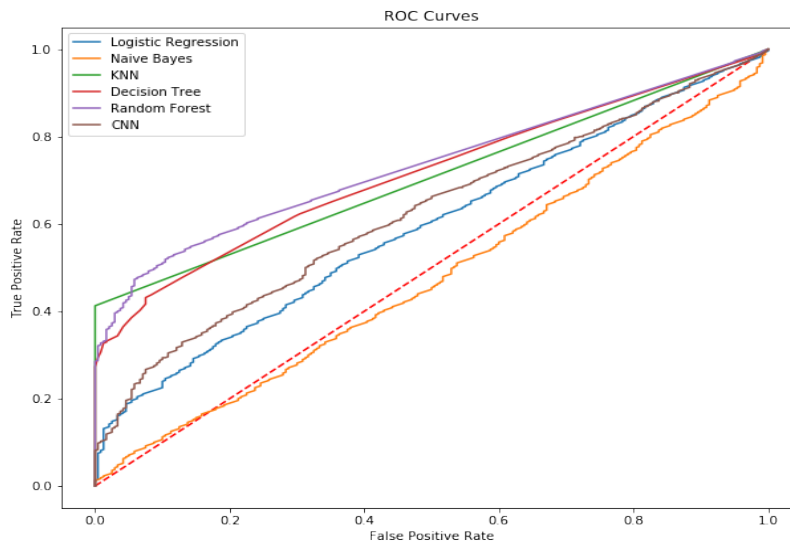


Figure 18: ROC curve comparison for 95:5 configuration

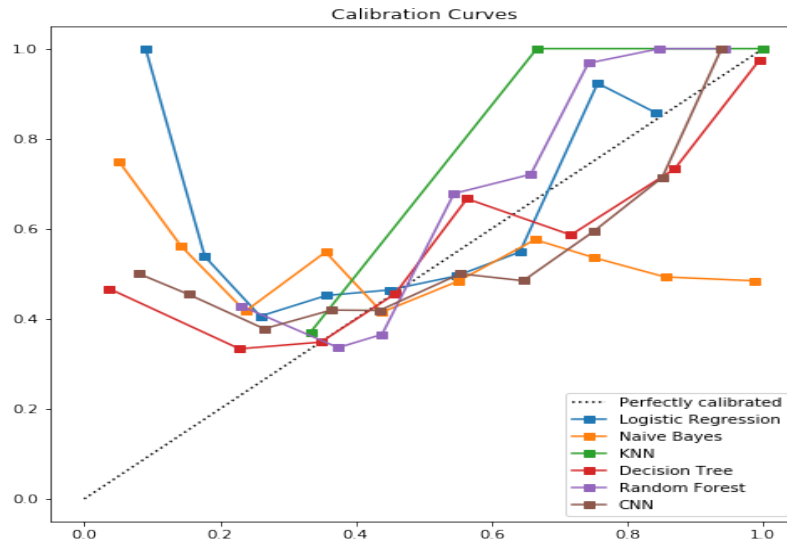


Figure 19: Calibration curve comparison for 95:5 configuration

For 95:5 configuration, accuracy of KNN (0.729) and Random Forest(0.717) was highest and lowest for logistic regression(0.595) and Naive Bayes(0.487).AUC score for ROC curve was highest for Random Forest and Decision Tree.

8 Conclusion

We had tried different classifiers, from all these we concluded, when we trained our data and tested our data according to different configurations, KNN and Decision Tree both had shown the highest accuracy. Also, on comparing accuracies we found out it was highest for Decision Tree (83.7 percent) and KNN (83.4 percent) when we split our data into 60:40 configuration i.e. data from 13 videos and 9 videos respectively. We tried lowering the data from 0.5 to 0.4, KNN and Decision Tree still reported the highest accuracy. We concluded following things for this project: One is, for our performance normalization of features was crucial. This is because different people have different baselines for eye aspect ratio and mouth aspect ratio because some people may naturally have big or small eyes and similarly other features. That is why, normalizing of features for each participant was necessary. Other is, efficiency of simple models like KNN be better or nearly equal to more complex models. In case of our project, K- Nearest Neighbour and Decision tree model are giving better accuracies as compared to LSTM and CNN model. But a complex model with less false and incorrest-negative rate is better to use compared to a simple model which may be easier and cheaper while deploying.

8.1 Limitations

Multiple face detected: In the camera or window, if more than one face arises then it may detect multiple faces. This can surely give undesired outputs because of various faces under various conditions. That is why, it is crucial to take care that only driver face should come in window or camera. Also one more reason is detection speed also reduces if it operates on multiple faces at the same time.

Use of eye glasses: It is difficult to detect eye aspect ratios and state in case a person uses eye glasses. In such cases, to avoid light, closeness to camera to eye, is needed. Because reflection of eye glasses can give result for an opened eye as closed eye, as it depends on light a lot.

8.2 Future Scope

To fine tune our models and improve the outputs, there are certain things we can do. First, to account the movement of the person in the video, if there is any, we need to put distance between the landmarks of face. This is because in reality a person will not be static on screen and these sudden movements of participants can give output as drowsy or signal of micro-sleep. Also we can create more data on our own and can use that data to prevent the models from overfitting. Second, accident may occur if person is slow while responding to the warning signals that means person's response sometimes is not enough to prevent accidents, even after being warned. So, we can avoid these kind of situations by designing and fitting a motor driven system. Also, we can synchronize that with some alarm system that slow down the vehicle automatically after getting warning or alarm signals. Third, we can make an Android Application for the users and can provide all information regarding the drowsiness or alertness levels during the journey, using the information based on frames number which are captured so that whether the person is drowsy or alert he/she will get to know.

9 Related Work

Shuyan et al. [2] performed the drowsiness detection using a Support Vector Machine with parameters related to eyelids.

Sayed et al. [4] proposed a study to detect driver drowsiness for prevention of accidents and for safer highway travel using artificial neural networks(ANN). They pre-processed signals of the angles of steering which was then passed through the ANN that classifies them drowsy or alert.

Flores et al. [1] presented a study for real time driver drowsiness detection which was based upon Artificial Intelligence and visual information. The main motto was to find, to keep track of and to analyze features of the face(eyes and mouth) to compute an index of drowsiness.

Wang et al. [5] developed a model for detection of drowsiness that studies the effects of drowsiness on the driver's performance. They made use of the Karolinska Sleepiness Scale to determine the level of drowsiness.

Omidyeganeh et al. [3] presented a robust model that detects the drowsiness of a driver using detection methods for closure of eyes and yawning. This study captured these features via a camera that must be installed in the car. Firstly, using the computer vision technology, the face region was detected and analyzed which helped in detection of the eye and mouth of the driver. Afterwards the driver fatigue level was detected.

References

- [1] Marco Javier Flores, José María Armingol, and Arturo de la Escalera. Real-time warning system for driver drowsiness detection using visual information. *Journal of Intelligent & Robotic Systems*, 59(2):103–125, 2010.
- [2] Shuyan Hu and Gangtie Zheng. Driver drowsiness detection with eyelid related parameters by support vector machine. *Expert Systems with Applications*, 36(4):7651–7658, 2009.
- [3] Mona Omidyeganeh, Abbas Javadtalab, and Shervin Shirmohammadi. Intelligent driver drowsiness detection through fusion of yawning and eye closure. In *2011 IEEE International Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems Proceedings*, pages 1–6. IEEE, 2011.
- [4] Riaz Sayed and Azim Eskandarian. Unobtrusive drowsiness detection by neural network learning of driver steering. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 215(9):969–975, 2001.
- [5] Xuesong Wang and Chuan Xu. Driver drowsiness detection based on non-intrusive metrics considering individual specifics. *Accident Analysis & Prevention*, 95:350–357, 2016.