Malaysian Journal of Computing, 9 (2): 1852-1866, 2024

eISSN: 2600-8238

DROWSY DRIVER DETECTION SYSTEM – VIA FACIAL RECOGNITION AND DRIVING DATA

Nur Iman Kamila Azharudin[1], **Hafizah Mansor**[2*] **and Shaila Sharmin**[3]

[1,2*,3]*Department of Computer Science, Kulliyyah of ICT, International Islamic University Malaysia*

[1]imankamila.a@live.iium.edu.my, [2*]hafizahmansor@iium.edu.my, [3]shailasharmin@protonmail.com

ABSTRACT

According to the National Highway Traffic Safety Administration, an estimated 17.6% of all fatal crashes in the years 2017–2021 involved a drowsy driver. This study proposes a drowsy driver detection system that uses both facial recognition and vehicular data to detect if a driver is feeling sleepy behind the wheel. We aim to address a lack of works in the literature that combine data measured from the driver (image or biological data) and vehicular data for drowsy driver detection. Our primary data was collected from simulated driving sessions in which a camera was used to record test drivers' faces while driving a virtual car in the CARLA simulator in both drowsy and non-drowsy states. The collected data consists of video of test drivers' faces from the camera and vehicular data from the simulated car. The video data was used to obtain facial features such as Mouth Over Eyes (MOE), Eyes Aspect Ratio (EAR), and Mouth Aspect Ratio (MAR), while the vehicle data yielded features such as speed, steering wheel movement and pedal readings. These features were used to train Support Vector Machine (SVM) and Random Forest (RF) models to detect drowsy drivers. The results indicate that RF is a better model to be used as compared to SVM in predictions of drowsiness in drivers with an accuracy of 96.24% and 86.85% respectively.

Keywords: *Driving Data, Drowsy Driver Detection, Facial Recognition, Machine Learning Technique*

Received for review: 04-03-2024; Accepted: 26-08-2024; Published: 01-10-2024
DOI: 10.24191/mjoc.v9i2.25694

1. Introduction

Drowsy driving or feeling sleepy while driving has been known to become a major contributor to road accidents. Human factors, which include drowsy driving are one of the factors contributing to road accidents (Bahari et al., 2023). According to the Bukit Aman Traffic Investigation and Enforcement Department, 20% of road accidents in Malaysia are caused by drowsy drivers annually ('Tiny naps' While Driving Leading Cause of Road Accidents, 2022). The National Safety Council (NSC) of the United States of America also mentions that this issue causes approximately 100,000 crashes per year (Rivelli, 2022). Such accidents caused by drowsy driving can be reduced with the implementation of a drowsy driver detection system.

There have been numerous proposals for drowsy driver detection that utilise different types of features and techniques. Although image recognition as well as driving data have been used many times in detecting fatigue in drivers, there is limited research which combines these two types of data for detecting drowsy drivers. To address this issue, we collected primary data from simulated driving sessions in which a camera was used to record test drivers' faces while driving a virtual car in the CARLA simulator in both drowsy and non-drowsy states. The collected data consists of video of test drivers' faces from the camera and vehicular data from the simulated car. The video data was used to obtain facial features such as Mouth Over Eyes (MOE), Eyes Aspect Ratio (EAR), and Mouth Aspect Ratio (MAR), while the vehicle data yielded features such as speed, steering wheel movement and pedal readings. These features were used to train two machine learning (ML) models, Support Vector Machine (SVM) and Random Forest (RF), to distinguish between drowsy and non-drowsy states. Evaluation metrics obtained from testing the trained models were finally used to determine which of the two models is more suitable for drowsy driver detection. The objective of this study is, thus, to examine whether features from image-based and vehicle-based data can be effectively used to detect drowsiness in drivers using ML methods. The contributions of this work may be summarised as follows: A drowsy driver dataset that consists of videos capturing the faces of test drivers as well as engine and driving data from simulated driving sessions in both drowsy and non-drowsy states. Additionally, a hybrid measure for detecting drowsiness that uses image-based and vehicle-based methods by combining both video data and driving data.

Numerous researchers have conducted work in detecting drowsiness. The algorithms used are either image-based, biological-based, vehicle-based, or hybrid-based measures (Albadawi et al., 2022). An image-based measure uses images as the data to train the algorithm. A biological-based measure uses heart rates or brain waves as data with the use of invasive sensors, while a vehicle-based measure uses in-vehicle data such as the speed of the vehicle and steering wheel angle. Lastly, a hybrid-based measure combines more than one type of data. Most studies are performed using image-based measures and biological-based data, while only a few studies are performed with vehicle-based and hybrid measures.

The work by Meda et al. (2021) explains the process of drowsiness detection using multiple ML models with image-based measures. The first step is feature extraction using multiple facial feature landmarks. The facial features are extracted using the Eye Aspect Ratio (EAR), the Mouth Aspect Ratio (MAR), the Pupil Circularity (PUR) and the Mouth Aspect Ratio over Eye Aspect Ratio (MOE). This research compares the accuracy and precision scores of six ML models. The models are k-nearest neighbour (KNN), logistic regression, decision tree classifier, Naïve Bayes classifier, long short-term memory (LSTM) networks and deep convolutional neural network (CNN). The accuracy scores of the models were somewhat on the lower end ranging from 64% to 78%, while the precision scores for all models range from 45% to 75%. It is seen from the results that the RF classifier and the LSTM model give the best scores for prediction.

The development of a drowsiness detection method based on facial features from video images was conducted by Junaedi and Akbar (2018). This is achieved firstly by recognising the presence of a face and eyes within the region of interest. Then, the eyes are used to measure the start of eye closure by measuring the time the eyes are closed and the percentage of eye closure (PERCLOS) for estimation of drowsiness. The experiment detects the face and eyes of the subject by using the Viola-Jones method which computes Haar-like features within the area of interest. Also, Adaptive Boosting (AdaBoost) is used to achieve rapid computation of the images. The result of this study indicates that while the iris area cannot be used to characterise drowsiness in drivers, PERCLOS has a better potential in measuring fatigue levels.

Abbas (2020) develops a hybrid fatigue detection system by integrating visual and non-visual features based on multi-cams and Electrocardiogram (ECG) sensors. The study uses hybrid-based measures, where a combination of facial feature recognition (image-based measure) and ECG data (and biological-based measure) are used to determine the fatigue levels of the driver. The face marks are calculated using the EAR and MAR values, similar to the work by Meda et al. (2021), and PERCLOS is used to measure drowsiness. The heart rate of the driver is read and collected by the ECG sensor. The CNN model is pre-trained with the visual features and the non-visual features combined as a single vector with the help of the Deep Belief Network (DBN) to connect the said vectors. The experiment yields an accuracy score for the CNN and DBN model combined, the pre-trained CNN and DBN model combined and the CNN model as a stand-alone. The paper concludes that the combination of image detection and ECG sensor data gives a satisfactory outcome for the hybrid fatigue system with 94.5% detection accuracy on 4250 images.

Machine learning techniques with hybrid-based measures have been used to explore actual human behaviour during drowsy drives (Vural et al., 2007). The work uses the Facial Action Coding System (FACS) to analyse facial motions and expressions. Another feature used to evaluate drowsiness and fatigue is placing a sensor on the subject's head and the steering wheel to read the motion of both aspects. The experiment is done by enabling subjects to play a driving video game on a Windows machine for three hours. The subjects' faces and crashes that had happened were recorded. From the facial recording, the study utilises 31 facial actions from FACS to detect fatigue during the experiment. An SVM model was trained with all facial actions to detect the actions when happening both alone and combined simultaneously with other facial actions. In addition, the study also does a classification experiment to determine levels of drowsiness according to facial movements. The paper uses Multinomial Ridge Regression (MLR) and AdaBoost to classify the facial features, which results in high accuracies of 94% and 92%, respectively.

The movement of the car (vehicle-based measure) to detect drowsiness has been studied by Arefnezhad et al. (2019). The work uses mainly the steering wheel angle of the vehicle. Also using a driving simulator, this work creates a dataset consisting of 20.5 hours of driving and 36 driving features. The paper uses the Adaptive Neuro-Fuzzy Inference System (ANFIS) to filter the features of the steering wheel data and select features that are most important. Then, an SVM model is used to classify the data into awake and drowsy classes. The accuracy of the model is compared between when the features are not filtered and when their proposed feature filtering is used. It is found that using SVM with the filtered feature set yields a higher accuracy of 98.12%.

In-vehicle sensors (vehicle-based measures) to detect drowsiness are used by Jeon et al. (2021). The sensors are placed at the steering wheel and pedal. The utilised features are the steering wheel angle, steering wheel velocity and two pedal pressure data. An ensemble network model which comprises the CNN is used. The model can categorise drowsiness during long drives and short drives. The number of non-drowsy driving data is reduced when training the ensemble CNN model. This is done to balance the number of drowsy and non-drowsy driving and efficiently train the proposed model. This enables a high accuracy value of 94.2% to be achieved in detecting drowsiness in long and short driving sessions.

Gupta et al. (2023) proposed a real-time drowsiness detection system through facial expressions using a deep CNN model as an image-based measure. The system analyses the driver's eyes, mouth, and head rotation pose with front angles, and left and right yaw angles up to 90°. The analysis was performed on a publicly available and manually captured image dataset. Three different CNN algorithms, namely EfficientDet D0, SSD MobileNet V2, and SSD ResNet50 V1, are used. SSD ResNet50 V1 is found to be the best model for this study because it is the most accurate and consistent.

From our review of the literature, we find that there are numerous works that use either data measured from the driver (image or biological) or the vehicle. However, there is a lack of detection systems that combine both driver and vehicle data, which is what we aim to do in the present study. The system has the benefit of being a non-invasive detection system since the drivers do not need to use any additional sensors attached to their bodies. Therefore, we consider image-based and vehicle-based data, and model drowsy driver detection as a binary classification problem. This work employs SVM and RF, which are common classification algorithms that have shown good results for drowsy driver detection.

The paper is organised as follows. Section 2 discusses the methodology which includes data collection, pre-processing and modelling. Section 3 discusses the results, while Section 4 concludes the paper.

2. Methodology

2.1 Data collection

To create an efficient detection system, a dataset is needed to train and test the system to correctly predict drowsiness. Due to the absence of a public dataset which includes image data and driving data, we started this study by creating a dataset that can be used for modelling.

The dataset used in this project consists of two parts, the video data and the driving data. The data was retrieved by using the CARLA Simulator, version 0.9.13. CARLA is an open-source driving simulator which enables users to drive around in settings of choice. The data was gathered by recording video data and driving data during each driving session.

The Logitech G29 Driving Force Steering Wheel and Pedals were used to interact with the CARLA Simulator, as shown in Figure 1. In addition, an iPhone 11 was used as a webcam, with the help of DroidCam and OBS Studio, to record the movement and facial expressions of the subjects.

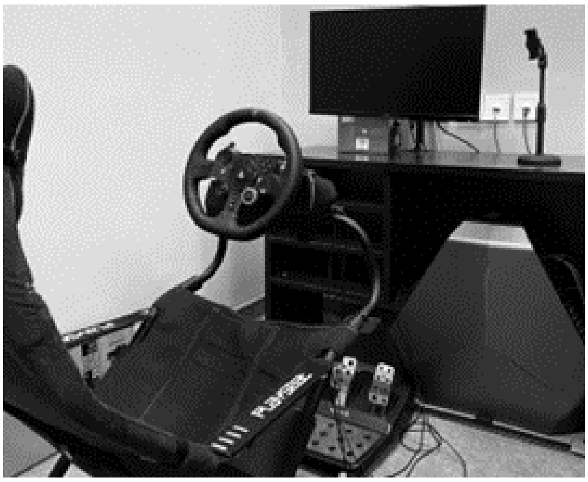

Figure 1. Setup of Simulator

The driving simulation was done on a highway setting, as seen in Figure 2 as fatigue and drowsiness are elevated on straight monotonous roads such as highways, rather than curved roads (Anwar & Sikander, 2019). Drivers were required to drive only on the highway without making any turns or going through any exits. The map setting of the simulation only had one traffic light, which required the drivers to stop when it was red. The remaining settings such as weather and car model were left to the default settings to ensure consistency of data. The drivers were required to continuously drive for one hour before the recording ended.

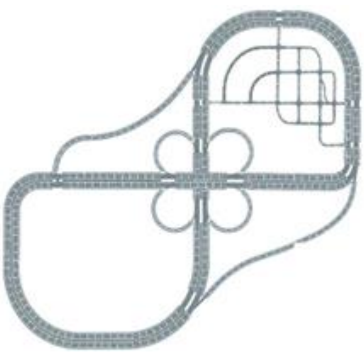

Figure 2. Map of the Driving Simulation

2.1.1 Data description

This study essentially aims to detect drowsiness by using image recognition and driving data. This dataset consists of two parts, video data and driving data, both of which were collected for four test drivers. The average age for all drivers was 22 years and the subjects consisted of two male and two female drivers. Each subject was required to drive in two different sessions, the first one being in a non-drowsy state and the second in a drowsy state.

The first session was conducted at any time of the day with a pre-requirement of having enough sleep, for at least six hours the night before the session (Drowsy driving: Asleep at the wheel, 2022). On the other hand, the second session was conducted between 13:00 to 15:00 hours as this period is when adults experience the largest dip in energy during the daytime (Anwar & Sikander, 2019). Subjects were also required to have slept less than six hours the night before as such adults are more prone to fall asleep while driving (Drowsy driving: Asleep at the wheel, 2022). In each session, the drivers were required to drive continuously for one hour, during which the video data and driving data were recorded. Since each subject is required to drive for a total of two hours in both sessions, there is a total of eight hours of video recordings and driving data recorded.

2.1.2 Video data

The videos of the subjects driving were taken at 60 frames per second (fps) for one hour continuously. Thus, the total number of frames from 8 hours of recordings is 1,728,000 frames. Due to limitations of computing power, the videos were pre-processed to retain only the first frame of each second, as done by Romera et al. (2015). Doing so yielded 3,600 frames from each hour-long video, resulting in a total of 28,800 frames from all eight hours of videos.

Figure 3 shows an example frame extracted from the dataset. The camera was placed from a slightly-sided angle to mimic the position of a rearview mirror dash camera. This way, when the system is adapted to real car settings, the system would still be able to detect the facial expressions of the driver.

Figure 3. Example Frame of Driver 4 from the Video Data

2.1.3 Driving data

Similar to the video data, the driving data is also designed to process 60 ticks per second. In CARLA, a tick is defined as a time when a single update is provided in the simulation (Synchrony and time-step - CARLA Simulator, n.d.). Thus, after each tick, the driving data was logged into two log files simultaneously. The data captured in the first file are the clock reading, throttle reading, steer reading, brake reading, reverse mode reading, handbrake reading, manual mode reading, gear reading, and the x, y, z vector components of the velocity of the car. The second file logged the rate of collision of the simulated car.

The data from both files were combined into a single data frame by using the clock value to match each row. In addition, to match the time sequence of the pre-processed video data, the first tick of every second was extracted. This was achieved by computing the cumulative value of the clock to get the time of simulation in milliseconds. The time in milliseconds was then converted into the 24-hour time format of hh:mm:ss. However, due to insufficient video memory provided by the graphics processing unit, there were inconsistencies in the duration of ticks when recording the driving data. Thus, the time difference attained is also uneven. Therefore, we took the value in the row which has the smallest delta time to the respective second, which was achieved by using the .nearest() function in Python.

Once the data was cleaned, the speed of the vehicle was calculated from the x, y, and z vector components available in the data. Also, the handbrake, manual mode, and gear readings were deleted from the dataset as they were insignificant to the study. This is because the values of the handbrake and manual mode did not change throughout the entire dataset. The gear column was deleted because the gear selection in the simulation was automatically changed by the program and not the driver, since the hardware of the gear component was not utilised in the data collection part.

2.2 Data labelling

An important part of this study is to be able to correctly label each row of data as drowsy or non-drowsy. Tüfekci et al. (2022) identify drowsy driving as an anomaly, explaining that driving while being drowsy is abnormal and driving while being awake is normal. Thus, by adapting the idea of drowsy driving as an anomaly to driving in an alert state, anomaly detection may be used to differentiate between drowsy and non-drowsy driving for labelling the driving data.

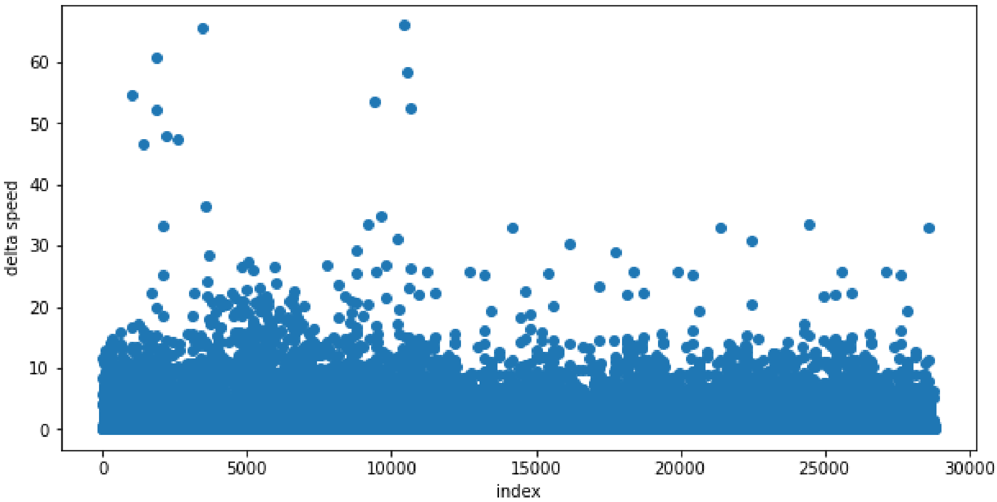

Figure 4. Scatterplot of Delta Speed

To identify which instances are anomalies, normal instances need to be identified first. Thus, the difference in speed, throttle reading, brake reading and steering wheel angle of each instance is calculated. Figure 4 shows the scatterplot of the speed difference in m/s between each instance and the previous second. As seen in

Figure 4, there is a higher density of instances when the delta speed is smaller and very few instances where the delta speed is high. We can, thus, consider points which have a higher delta speed as anomalies. However, since we need to identify a threshold delta speed for anomalies, we utilize the standard deviation as an anomaly threshold (Badr, 2019).

Once the standard deviation is calculated, the distribution can be visualised as in Figure 5. Values that are not within the normal range are considered as anomalies and labelled as drowsy. The same process is repeated for the delta brake, delta throttle and delta steer. The process of labelling the driving data has been illustrated in Figure 6.

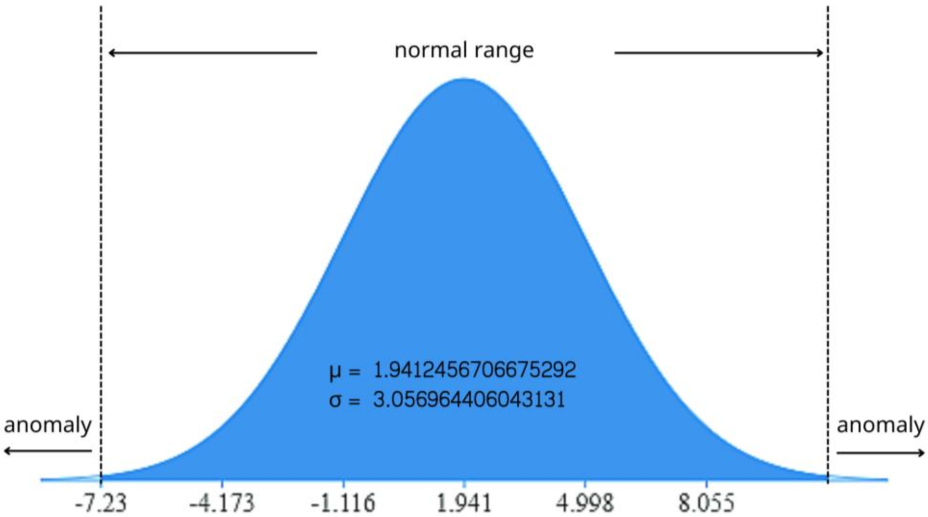

Figure 5. Distribution Graph for Delta Speed

In Figure 6, data is labelled as drowsy when the delta value is more than the anomaly threshold. Since all delta values are positive, negative anomaly threshold values are not needed. In addition, the reverse and collision features have been considered in data labelling. This is because there were instances where the drivers accidentally collided with other cars or road dividers when they fell asleep and needed to reverse to continue driving. To sum up, data is labelled as drowsy whenever the vehicle goes over the anomaly threshold or if the vehicle experiences a collision.

The next part is to label the video data. As explained before, the first frame of each second was extracted, from which the facial landmarks of the subjects' faces were extracted using the extract_face_landmarks() function from the mlxtend Python library. These facial landmarks can be seen in Figure 7. From among these facial landmarks, we retained those numbered 36 to 68 to focus on the eyes and mouth for eye blinks and yawns.

Using the eyes and mouth landmarks, the Eyes Aspect Ratio (EAR), Mouth Aspect Ratio (MAR), Pupil Circularity and Mouth Over Eye (MOE) values were calculated and used for modeling. The EAR value yields the distance of eye openings. The EAR value decreases when the eyes are closing. Next, the MAR value gives the distance of mouth openings—the bigger the mouth is opened, the higher the MAR is. Considering MAR enables the detection of yawns. The Pupil Circularity feature focuses on the location of the pupil. The experiment by Zhong (2021) shows that as the eyes droop due to drowsiness, the Pupil Circularity value should decrease. However, this value would not decrease if the person were only blinking. The last feature extracted the MOE value. This value would be the highest when a person yawns while closing their eyes. In this case, the higher the MOE value, the drowsier the person is. In Figure 8 there are many high and some low peaks in the MOE data. The higher the peaks in the MOE line plot, the drowsier the driver feels.

However, an important issue faced while extracting face landmarks and their features is that in certain frames, the faces of drivers could not be detected. This happens when drivers cover their mouths while yawning, rub their eyes or do any action that covers the face. When this happens, the landmarks cannot be extracted and are replaced with NULL values. These rows were completed using interpolation, whereby the average values of EAR, MAR, MOE, and Pupil Circulatory from the rows above and below were used to fill in the missing values.

Subsequently, the data was labelled according to the conditions shown in Figure 9. The labelling process is dependent on the threshold values from each feature, which are calculated using a method similar to that by Dewi et al. (2022). The EAR threshold is obtained by calculating the average of the EAR value when the eyes are opened and the EAR value when the eyes are closed. The same method was applied to all other features to find their respective threshold values.

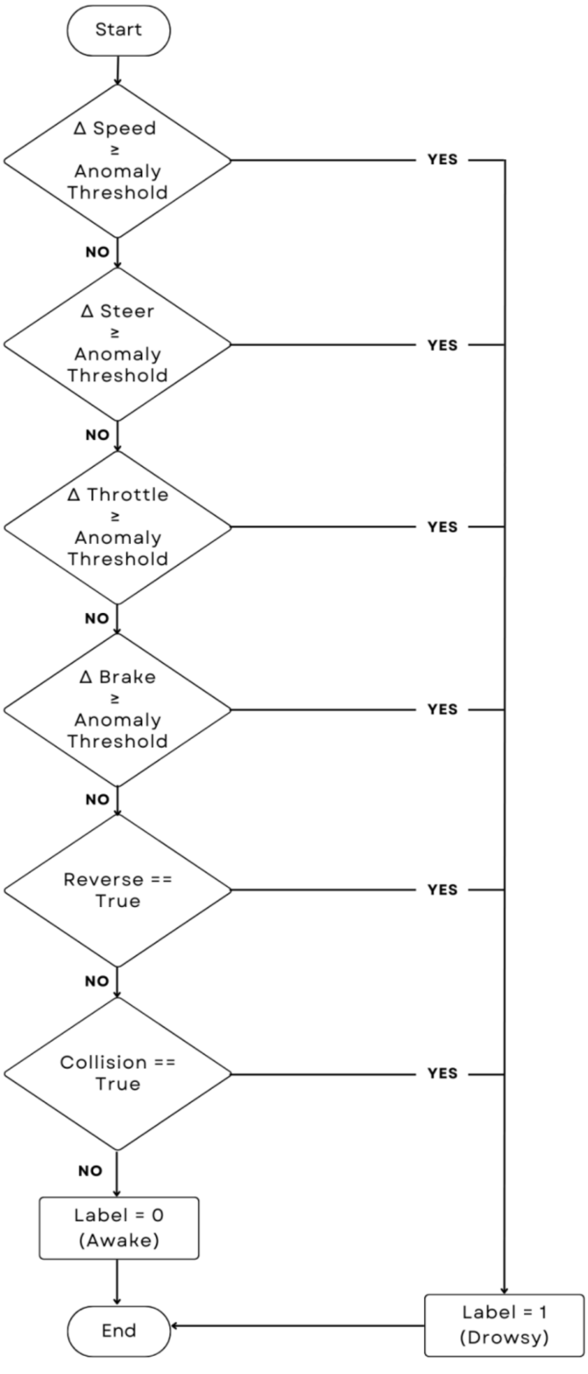

Figure 6. Flowchart of Driving Data Labelling Process

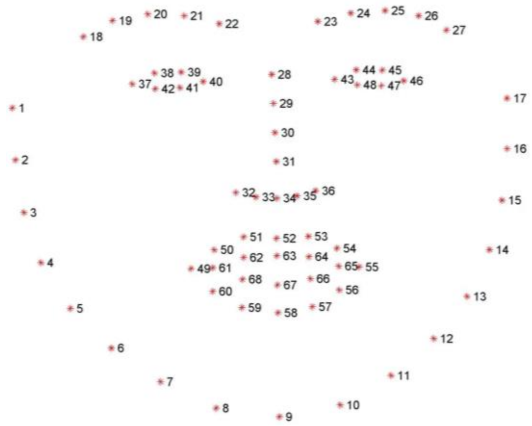

Figure 7. Facial Landmark Extracted Using extract_face_landmarks()

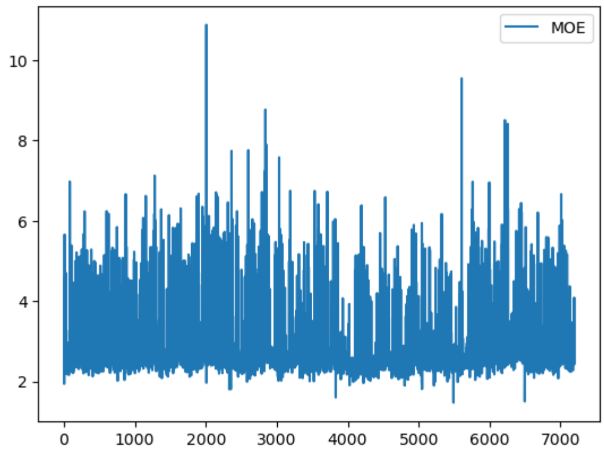

Figure 8. Line Plot of MOE Value of Driver 4

2.3 Data cleaning and pre-processing

In this section, the driving dataset and the video dataset were concatenated to yield one synched dataset. The datasets are joined by using the pandas.concat() function which joins two DataFrames by index. Since we labelled the driving and image data separately, we need to combine the labels to produce an overall label for each row that represents whether the driver is in a drowsy or non-drowsy state. We obtain the overall labels by taking the logical OR of the two labels, i.e., if either the driving label or facial label is drowsy, then the overall label is drowsy.

2.4 Modelling

Before we begin building the model, the data needs to be split into training and testing data. However, after data labelling is conducted, it is found that the number of drowsy and non-drowsy data is not balanced. The ratio of drowsy to non-drowsy data available is 4:21. Therefore, to avoid underfitting when training the model, under-sampling is done to balance the ratio of drowsy to non-drowsy data during training.

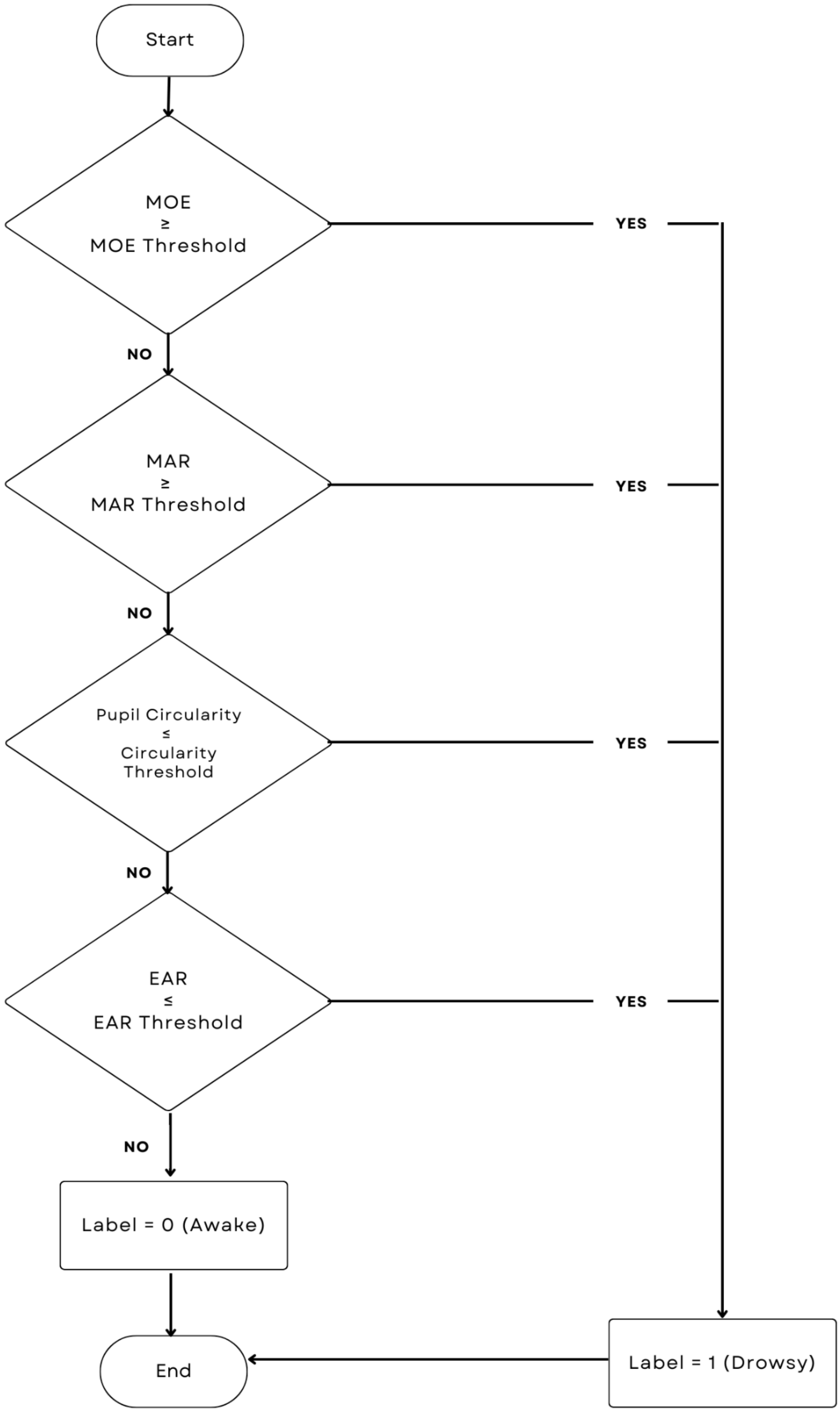

Figure 9. Flowchart of Video Data Labelling Process

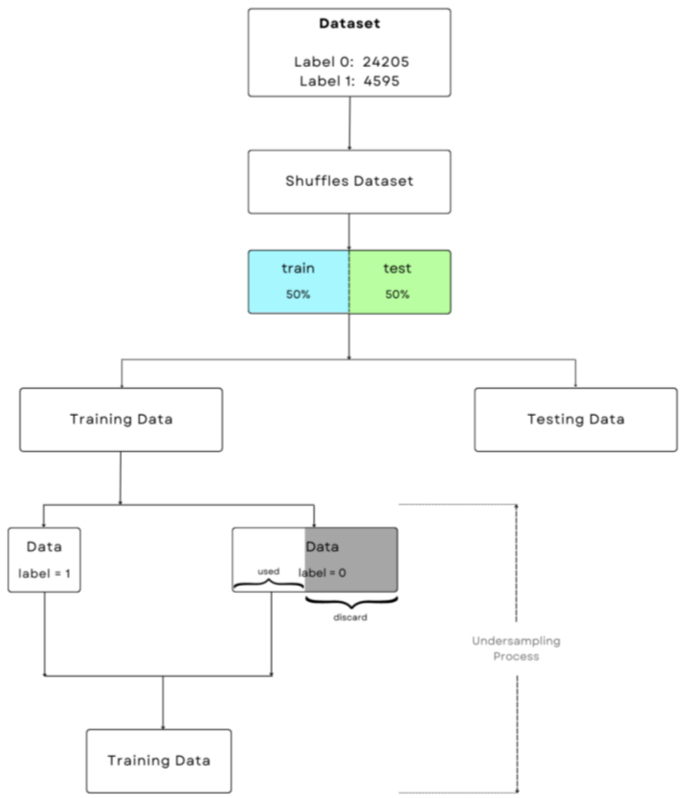

Figure 10. Flowchart of Under-Sampling Data

Under-sampling is a technique used to remove data from the majority group to even out the dataset. From under-sampling, a balanced ratio of 50:50 between the drowsy and non-drowsy dataset can be achieved. The undersampling process used has been represented in Figure 10. The original dataset was split into two equal parts for testing and training (50:50 for testing: training). The training dataset was then under-sampled by removing data from the majority class (non-drowsy) to match the amount of data in the minority class (drowsy). The testing dataset, on the other hand, was not undersampled and consisted of 2194 drowsy and 11606 non-drowsy datapoints.

Once data is balanced and divided into training and testing datasets, the model building process can be started. The ML algorithms used in this study are SVM and RF. These algorithms were chosen due to their high accuracy in predicting drowsiness in works by Meda et al. (2021), Vural et al. (2007), and Arefnezhad et al. (2019).

2.4.1 Support Vector Machine (SVM)

Support Vector Machine is a supervised learning method that can be used for classification, regression, and anomaly detection problems. During the training phase, the SVM algorithm finds a hyperplane in the feature space of the training data that can distinctly separate data points of different classes. The hyperplane is a decision boundary, whereby data points on one side of the plane belong to one class, and data points on the other side belong to another class. The hyperplane is chosen such that the distances between the hyperplane and the nearest data points on either side of it are maximised, i.e. the margin between instances of the two classes is the largest. During testing, this hyperplane is used to classify new datapoints into either class.

We assumed that the two classes in our dataset, drowsy and non-drowsy, are linearly separable, which is why we employed linear SVM. The implementation of support vector classification in the scikit-learn library was used in this study, which internally implements 5-fold cross-validation. The regularisation parameter C was set to the default value of 1.

2.4.2 Random Forest (RF)

Random Forest is a supervised learning algorithm which is used for classification and regression problems. RF improves upon the tendency of decision trees to overfit to the training data by using an ensemble of such trees and aggregating their decisions to produce an overall prediction. A decision tree for a classification task, such as that in the present work, is produced by taking the dataset and recursively partitioning it along different input features until each partition consists of data points with the same target label. The tree structure represents this partitioning with each node being the feature along which the dataset is split and each leaf consisting of datapoints of a single class. RF produces a "forest" of such decision trees during training. During testing, a new datapoint is run through each tree and the prediction yielded by the majority of trees is considered the overall predicted label for that datapoint.

In this study, the RF classifier implementation in scikit-learn has been used. It was set to train 100 decision trees with no limit on the maximum depth of each tree, i.e., the dataset is split until each leaf has only instances of the same class. Furthermore, the n_jobs hyperparameter was set to -1 so that training uses as many processors as is available to train trees in parallel. For all other hyperparameters, the default values have been used.

3. Results and Discussion

To understand the performance of both classification models, we report standard evaluation metrics derived from the confusion matrix, which gives the number of true negatives (TN), true positives (TP), false negatives (FN), and false positives (FP). These evaluation metrics are namely accuracy, precision, recall, and F1-score. Apart from these, we also report Matthews Correlation Coefficient (MCC) and balanced accuracy (bACC). We elect to report these additional metrics because, while our training dataset is balanced, the testing dataset is imbalanced with more instances of the negative class—non-drowsy datapoints. While accuracy can be an inadequate measure of performance for an imbalanced dataset and a biased model, the other metrics like precision, recall, and F1-score focus on TP and do not consider TN. For such datasets, balanced accuracy and MCC may be considered, which consider all four confusion matrix quantities and are thus more informative. Of these, MCC may be considered as the single best metric for determining the overall performance of a classifier—ranging from -1 (worst) to 1 (best), MCC is close to 1 only when a classifier correctly identifies both positive and negative values and when its positive and negative predictions are correct (Chicco et al., 2021).

The evaluation metrics obtained from both models after training and testing have been reported in Table 1. If we consider each evaluation metric, it is apparent that while both classifiers demonstrate reasonably good classification performance, the RF classifier is superior to the SVM classifier. All of the metrics for the RF classifier are higher than the corresponding metrics from the SVM classifier.

For both models, we observe higher recall values but lower precision values. This indicates that the models are biased to the positive or drowsy class because the models are able to classify most drowsy datapoints correctly but many of their drowsy predictions are not correct. However, RF yields higher precision and recall, and subsequently higher F1-score, which demonstrates the effectiveness of RF for drowsy driver detection. The high MCC and balanced accuracy scores of RF compared to SVM also show that the RF is better at correctly distinguishing between drowsy and non-drowsy datapoints and yields fewer incorrect predictions. It may thus be concluded that the RF algorithm is better suited for drowsy driver detection than linear SVM. Compared to Meda et al. (2021) which uses only facial features and achieves an accuracy of 78%, our RF classifier achieves a higher accuracy of 96.24%.

Table 1. Evaluation Metrics for Models on Testing Data

	SVM	**RF**
Accuracy	0.8685	0.9624
Precision	0.5566	0.8330
Recall	0.8491	0.9549
F1-score	0.6724	0.8898
Matthews Correlation Coefficient	0.6153	0.8702
Balanced Accuracy	0.8606	0.9594

An important factor in modelling and prediction is the dataset used. Since this study processes only one frame per second, increasing the value to between 2 and 4 frames per second could allow more immediate detection of drowsiness, which could be part of future work. The image recognition aspect may also be improved—while the head and hand movements of the subjects are not considered in this study, movements like covering the mouth while yawning or dropping the head can also indicate drowsiness and can be included to develop a more robust detection system.

Limitations of this work include the small size of the dataset, which is due to the limited availability of computing power for pre-processing the video data. This limitation can be addressed with more computing resources to process larger quantities of video data and produce a larger, more comprehensive dataset. Another limitation is the method used to label the dataset. Although labelling the dataset using the standard deviation is valid, some potential biases might be introduced to the model. More robust and direct labelling techniques can ensure the integrity of the dataset and improve the overall reliability of the results.

4. Conclusion

The purpose of this study is to apply machine learning to vehicular data as well as drivers' facial features to detect drowsiness in drivers. To achieve this, a dataset which consists of video data and the vehicle's driving data was collected from simulated driving sessions where test drivers were in two driving states, drowsy and non-drowsy. After pre-processing the dataset, anomaly thresholds derived from non-drowsy data were used to label the dataset. To prepare the dataset for machine learning, we split the dataset into training and testing sets and performed under sampling on the training set to balance the two classes. Finally, we trained SVM and RF classifiers and evaluated them using six metrics. We find that although both classifiers exhibit a bias for the drowsy class, the RF classifier is better than the SVM classifier at distinguishing between drowsy and non-drowsy instances, similar to the work by Meda et al. (2021). From the results, we can conclude that a combination of image- and driving-based data can be used with traditional ML to effectively detect drowsiness in drivers as suggested by Albadawi et al. (2022).

Acknowledgements

The work presented in this paper is supported by the Fundamental Research Grant Scheme for Research Acculturation of Early Career Researchers (FRGS-RACER) from the Ministry of Education Malaysia, ref no. RACER/1/2019/ICT03/UIAM//1 (no. RACER19-006-0006).

Funding

The author(s) received funding for this work from the Ministry of Education Malaysia under the Fundamental Research Grant Scheme for Research Acculturation of Early Career Researchers (FRGS-RACER), ref no. RACER/1/2019/ICT03/UIAM//1 (no. RACER19-006-0006).

Author Contribution

Author1 prepared the literature review, wrote the research methodology and performed fieldwork. Author1 conducted the statistical analysis and interpreted the results. Author 2 oversaw the article writing and the fieldwork. Author1 prepared and presented the paper at the conference, the 2nd International Postgraduate Colloquium (IPC 2023). Author 3 contributed to the data collection as well as the analysis and interpretation of results.

Conflict of Interest

The authors have no conflicts of interest to declare.

References

Abbas, Q. (2020). HybridFatigue: A Real-time Driver Drowsiness Detection using Hybrid Features and Transfer Learning. *International Journal of Advanced Computer Science and Applications, 11*(1), 585-593.

Albadawi, Y., Takruri, M., & Awad, M. (2022). A review of recent developments in driver drowsiness detection systems. *Sensors, 22*(5), 2069.

Anwar, S., & Sikander, G. (2019, June). Driver Fatigue Detection Systems: A Review. *IEEE Transactions on Intelligent Transportation Systems, 20*(6), 2339-2352.

Arefnezhad, S., Samiee, S., Eichberger, A., & Na, A. (2019, Feb). Driver Drowsiness Detection Based on Steering Wheel Data Applying Adaptive Neuro-Fuzzy Feature Selection. *Sensors, 19*(4), 943.

Badr, W. (2019, Mar 5). *5 Ways to Detect Outliers/Anomalies That Every Data Scientist Should Know*. (Medium) available at https://towardsdatascience.com/5-ways-to-detect-outliers-that-every-data-scientist-should-know-python-code-70a54335a623&sa=

Bahari, N., Mohd Nusa , F., Tarudin, N., Mohamad, N., & Md Murozi, A. F. (2023). Factors contributing to road accidents: case study of Jalan Seremban-Tampin. *Malaysian Journal of Computing (MJoC), 8*(1), 1436-1448.

Chicco, D., Tötsch, N., & Jurman, G. (2021). The Matthews Correlation Coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation. *BioData Mining*, 14(1), 1-22.

Dewi, C., Chen, R., Jiang, X., & Yu, H. (2022). Adjusting eye aspect ratio for strong eye blink detection based on facial landmarks. *PeerJ Computer Science, 8,* e943.

Drowsy driving: Asleep at the wheel. (2022, Nov 21). *Centers for Disease Control and Prevention*. available at https://www.cdc.gov/sleep/features/drowsy-driving.html

Gupta, N., Rajoo, R., & Jacob, P. (2023). Driver drowsiness detection system through facial expression using Convolutional Neural Networks (CNN). *Malaysian Journal of Computing (MJoC), 8*(1), 1375-1387.

Jeon, Y., Kim, B., & Baek, Y. (2021, Mar). Ensemble CNN to Detect Drowsy Driving with In-Vehicle Sensor Data. *Sensors, 21*(7), 2372.

Junaedi, S., & Akbar, H. (2018). Driver Drowsiness Detection Based on Face Feature and PERCLOS. *Journal of Physics: Conference Series*., 1090, p.012037

Meda, H., H., Ganesh, J., J. M. P., & Sahani, A. (2021). Machine Learning Models for Drowsiness Detection",. *IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, (pp. 1–5).

Rivelli, E. (2022, Sept 13). *Drowsy Driving Statistics and Facts 2022*. (Bankrate) available at https://www.bankrate.com/insurance/car/drowsy-driving-statistics/

Romera, E., Bergasa, L., & Arroyo, R. (2015). A Real-Time Multi-scale Vehicle Detection and Tracking Approach for Smartphones. *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, (pp. 1298-1303).

Synchrony and time-step - CARLA Simulator. (n.d.). available at https://carla.readthedocs.io/en/latest/adv_synchrony_timestep/

Tefft, B. C. (2024). *Drowsy Driving in Fatal Crashes, United States, 2017–2021*. Washington: Foundation for Traffic Safety available at https://aaafoundation.org/drowsy-driving-in-fatal-crashes-united-states-2017-2021/#:~:text=Results%20show%20that%20an%20estimated%2017.6%25%20of%20all,were%20killed%20in%20crashes%20that%20involved%20drowsy%20drivers.

'Tiny naps' While Driving Leading Cause of Road Accidents. (2022, June 25). The Star Newspaper, available at https://www.thestar.com.my/news/nation/2022/06/25/tiny-naps-while-driving-leading-cause-of-road-accidents

Tüfekci, G., Kayabaşı, A., Akagündüz, E., & Ulusoy, İ. (2022, October). Detecting driver drowsiness as an anomaly using lstm autoencoders. In *European Conference on Computer Vision* (pp. 549-559). Cham: Springer Nature Switzerland.

Vural, E., Cetin, M., Ercil, A., & Littlewort, G. (2007). Drowsy Driver Detection Through Facial Movement Analysis. *2007 IEEE International Conference on Human-Computer Interaction*, (pp. 6-18).

Zhong, G. (2021, Jun 10). *Drowsiness detection with machine learning*. (Medium) available at https://towardsdatascience.com/drowsiness-detection-with-machine-learning-765a16ca208a