Classifying, Understanding, and Mimicking Driver Behaviour from Sensor Data Using LSTM-Based Models for Autonomous Driving

MSc Research Project
MSc in Artificial Intelligence

Metehan Dilekmen
Student ID: 23308907

School of Computing
National College of Ireland

Supervisor: Arundev Vamadevan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Metehan Dilekmen		
Student ID:	23308907		
Programme:	MSc in Artificial Intelligence	**Year:**	2024/2025
Module:	Research Project		
Supervisor:	Arundev Vamadevan		
Submission Due Date:	11.08.2025		
Project Title:	Classifying, Understanding, and Mimicking Driver Behaviour from Sensor Data Using LSTM-Based Models for Autonomous Driving		
Word Count:	8800	**Page Count**:	20

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: Metehan Dilekmen

Date: 09.08.2025

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	
Signature:	
Date:	
Penalty Applied (if applicable):	

Classifying, Understanding, and Mimicking Driver Behaviour from Sensor Data Using LSTM-Based Models for Autonomous Driving

Metehan Dilekmen
23308907

Abstract

The automatic detection of vehicle driving behaviour is an important step towards driver assistance systems and autonomous vehicle technologies. In this project, a model has been developed that uses physical data generated during driving to determine the driver's behaviour pattern.

The primary driving behaviours targeted are classified as aggressive, drowsy, and normal. The model aims to identify these behaviours and establish a system capable of making human-like decisions. In addition to classical methods, deep learning models that work on time series data were also used in the study. The system was tested using data obtained from real driving and achieved various classification successes. The model was tested in different driving scenarios and made incorrect predictions in some cases due to external factors. This showed that systems based solely on physical data may be limited.

This project demonstrates the potential of artificial intelligence-based classification systems to mimic human behaviour, while also pointing to future needs for the development of such systems.

Keywords: *Driver Behaviour Classification, LSTM-Based Deep Learning, Driving Pattern Recognition, Aggressive and Calm Driving Detection, Explainable AI in Autonomous Vehicles*

1 Introduction

Automatic detection of vehicle driving behaviour has become an important requirement for both driver assistance systems and autonomous vehicles. Whether a driver is driving aggressively, drowsy or normally plays a key role in accident prevention, improving road safety and developing systems that mimic human driving.

Therefore, the development of models that can recognise human behaviour based on real driving data is now one of the main areas of research. In this project, a system was designed that classifies and attempts to imitate human driver behaviour using data from physical sensors. The data consists of measurements such as speed, acceleration and jerks obtained from real driving.

This data, processed as time series, has been analysed based on the assumption that driving styles exhibit certain patterns. The goal of the model is to capture these patterns, accurately label driver behaviour and make decisions in a manner similar to humans.

The main question of this research is as follows:

'How accurately can time series-based deep learning models understand human driving behaviour based solely on physical sensor data, and how reliably can they replicate this behaviour?'

To answer this question, classical machine learning methods (KNN and Random Forest) were first tested. Next, LSTM-based deep learning models capable of directly processing time series data were developed. During the training process, feature engineering was applied, new inputs were generated from variables such as acceleration jerk and amplitude, and data imbalance issues were addressed. The methods used during the project, the results obtained, and the analyses conducted are explained in separate sections:

2. Related Work: This section summarises recent studies on driver behaviour classification and explains how this project differs from examples in the literature in terms of the methods and data structures used.

3. Research Methodology: The data set used is introduced, the features created, the pre-processing steps applied, and the data preparation process are detailed, and the processes that will be detailed in the rest of the report are briefly mentioned.

4. Design Specification: The architecture of the developed model is explained and the reasons for choosing a time series-focused approach are explained.

5. Application: This section presents the model training process, the machine learning algorithms used, and the structure of the model step by step.

6. Evaluation: This section evaluates the classification results of the model, scenario-based test outputs, and accuracy rates.

6.4 Discussion: The results are discussed, the strengths and weaknesses of the model are addressed, and a general evaluation is made in light of the current results.

7. Conclusion and Future Work: The overall results of the study are summarised, the gains obtained are conveyed, and recommendations for future improvements are presented.

Many studies in the literature use simulation data to classify driver behaviour, which means that models are trained only with concrete data. Camlica (2023) notes that this does not adequately reflect variables such as driver character, sudden decisions, and stress, which are important in real-life scenarios. Talebloo (2022) showed that aggressive driving behaviour can be successfully classified using deep learning, but emphasises that a better understanding of human behaviour and its inclusion in training is necessary for such models to be directly integrated into autonomous systems.

In this project, in addition to improving classification success, we discuss how the model's decisions can be made understandable and how it can mimic human movements. The results show that such systems can be technically powerful but limited due to unmeasurable factors such as external influences and human intent.

2 Related Work

2.1 Fundamentals of Driver Behaviour Detection

Traffic safety remains one of the most significant unsolved problems worldwide today. Every year, millions of people lose their lives or suffer serious injuries in traffic accidents. A large proportion of these accidents are directly attributable to human error. In particular, distracted driving, sudden decisions, and aggressive driving habits continue to cause accidents despite advanced vehicle technologies. Alkinani and Khan (2020) have pointed out that driver inattention directly contributes to accidents, especially in risky situations such as high speed and heavy traffic, and that detection using camera systems alone is insufficient.

Driver behaviour usually shows itself through physical reactions such as acceleration, braking, lane changing and steering movements. Analysing these behaviours can provide important information about the driver's mood, attention level and general driving style. Davoli et al. (2020)

emphasise that early detection of such behaviours is critical for safe driving policies. Thanks to in-vehicle sensors, risky situations can be detected automatically.

In the past, such behaviour analyses were carried out using methods such as observation, surveys or manual evaluation. However, these approaches are time-consuming and often rely on subjective interpretations. Fasanmade (2021) stated that measuring driver attention using manual methods is inefficient in terms of both accuracy and consistency, and drew attention to the necessity of automated systems. This is because a driver's assessment of their own driving is often based on personal perceptions, which do not yield objective results.

Artificial intelligence-based systems have emerged in recent years to address these shortcomings. With the development of deep learning models, driver behaviour can now be classified and analysed in real time. García et al. (2023) showed that behaviours associated with stressed or distracted drivers can be detected much more successfully when supported by biosignals. These biosignals can be obtained from data such as heart rate or EEG. However, advanced artificial intelligence systems are required to process this type of data.

Additionally, Hemmatyar and colleagues (2022) successfully classified aggressive driving using time-sensitive models. The main advantage of such systems is their ability to consider the driving context. For example, sudden acceleration may be considered aggressive in one situation but normal in another. Models that can make decisions based on the situation significantly reduce misclassification rates.

Finally, working with data obtained from real vehicle environments significantly improves the performance of models. Ou (2019) pointed out that simulation-based systems cannot fully reflect real traffic behaviour. This is because driver character, psychological states, or unexpected events cannot be adequately represented in simulations. Systems developed with real data, on the other hand, can adapt to more diverse driver profiles.

The common point of these studies is that they show that driver behaviour detection is no longer just a theoretical topic, but has become a technology-supported practical solution.

2.2 Learning-based Approaches

Although traditional machine learning methods have been used for many years to classify driver behaviour, these methods have not been sufficiently successful with time series data. Rule-based models or tree structures generally make decisions based on instantaneous data, while time-dependent problems such as driver behaviour require deeper structures. For this reason, deep learning approaches have come to the fore in recent years.

In particular, models compatible with time series data, such as LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit), have shown successful results in learning sequential patterns in driver behaviour. Sahoo and colleagues (2023) reported achieving high accuracy rates with an LSTM model based solely on speed and steering data, but noted that challenges remain in terms of prediction consistency. The success of LSTM stems from its ability to retain past data in memory and learn behavioural continuity.

Alternatively, the GRU architecture is also preferred for similar problems. Mozaffari and Azimi (2021) compared LSTM and GRU architectures in a fatigue detection system that analyses facial expressions and physical signals, observing that LSTM was more successful due to its deeper learning capacity. This makes LSTM particularly suitable for driver behaviour analysis involving multidimensional and complex data.

Kwon et al. (2021) developed a CNN-LSTM hybrid model capable of analysing both temporal and spatial data. This model can be integrated into V2X (vehicle-to-everything) communication systems and can classify complex driving data in a more meaningful way. Thus, both instantaneous data and behaviour patterns that develop over time can be analysed together.

However, the success of the model depends not only on the correct architectural choices but also on processing power and model performance. In real-time systems, the model must be fast and efficient. Dalloo and colleagues (2024) emphasised the necessity of lightweight architectures in this context and noted that models must be both fast and stable in environments that generate large amounts of data.

Finally, Debbarma and Pal (2025) showed that they classified aggressive driving with approximately 90% accuracy using a double-layer LSTM architecture. This success was made possible not only by speed but also by the combined use of multiple physical variables such as steering angle, braking, and speed change.

These studies show that both suitable deep learning architectures and rich and balanced data sets to feed these models are needed for the accurate classification of driver behaviour.

2.3 Challenges Related to Data, Simulation and Labelling

One of the most common problems in systems for classifying driver behaviour is how well the data used reflects real-life conditions.

Simulation data is often preferred in many academic studies; however, this data cannot adequately represent complex factors such as sudden decisions, psychological effects, or environmental conditions in real driving environments. Ou (2019) and Camlica (2023) have noted that models developed in simulation environments do not adequately reflect real traffic behaviour, thereby limiting model performance.

While the use of real driving data enables the learning of more diverse and natural behaviour examples, the processing and labelling of this data also presents a separate challenge. In particular, the interpretation of concepts such as ‘aggressive’ or ‘calm’ can vary from person to person. Ghoniemy and colleagues (2023) emphasised that there is no common standard for labelling driver behaviour, making direct comparisons between studies difficult.

In addition to the labelling process, class imbalance in datasets is another significant problem. Real-world data typically consists predominantly of ‘normal’ or ‘calm’ driving. Since “aggressive” or ‘drowsy’ driving is rare, such classes may not be adequately learned by the model. Karuppusamy and Bhuvaneswari (2023) demonstrated in a model developed using GPS and CAN-BUS data that data balancing techniques must be applied to ensure that the aggressive driving class is learned correctly.

In addition to labelling errors, context-independent evaluations can also lead to incorrect results. For example, high speed on a motorway is considered normal, while the same speed in a city may be considered aggressive. Chuang and Lin (2022) developed a model that takes these contextual differences into account by integrating variables such as traffic density and road conditions into the classification process.

Cojocaru and colleagues (2022) have pointed out that systems developed in simulation environments may not adequately represent sudden decision-making or natural driver reflexes, suggesting that the success of such systems in real-world scenarios may be limited.

The challenges outlined in this section are not only technical but also methodological. Therefore, standardisation in the labelling process, as well as data diversity and balance, are prerequisites for successful and fair classification systems.

2.4 Behaviour Modelling for Human-Like Autonomous Systems

Autonomous vehicles must not only perceive environmental data but also operate in harmony with human drivers. In real traffic environments, drivers make decisions not only based on rules but also on their own habits, attention states, and the immediate conditions they encounter. Therefore, it is crucial for autonomous systems to understand human driver behaviour and make decisions accordingly.

Recent studies have emphasised the need to evaluate driver behaviour not only through in-vehicle sensors but also in conjunction with the context. Chuang and Lin (2022) developed a model that considers traffic density and road conditions to assess the level of aggressiveness based on environmental factors. Similarly, Svyetlichnyy and Łach (2024) were able to perform behaviour analyses suitable for the driving environment using a similar contextual classification system.

The success of such models depends not only on physical data but also on data measuring the psychological and physiological state of the driver. Biological indicators such as EEG, eye movements, and heart rate are used for this purpose. Bellos and colleagues (2025) developed a system that classifies driver behaviour based on attention levels and aggression tendencies by analysing EEG data.

Integrating such analyses into the decision-making processes of autonomous vehicles enables them to move in a more harmonious manner with human drivers. Hosseini and colleagues (2024) emphasised that autonomous systems should not only act according to traffic rules but also according to the expected behaviour of human drivers. This allows decisions such as sudden braking, lane changing, or yielding to be made in a more natural way.

In this context, it is expected that behaviour modelling will not remain at the classification level but will be directly integrated into decision support systems. Voos and Distante (2022) developed a system that can mimic driver behaviour, enabling autonomous vehicles to move more naturally in cities. Alzahrani et al. (2024) showed that multi-task models can generate suitable responses based on driver behaviour.

Visual data is another source of information that supports these systems. Yeola and Barve (2024) developed a system that predicts a driver's aggressive tendencies based on head movement and facial expression analysis. These analyses provide important contributions to the early prediction of driver behaviour.

Similar to these studies, the LSTM-based model developed in this project not only classifies driving styles but also evaluates how these classes can be related to autonomous system decisions. The model's responses were observed through scenario-based analyses, and the impact of human behaviour on system decisions was discussed. Thus, the model has been able to contribute to the decision-making system by taking behaviour classification to a higher level.

2.5 Conclusion

The studies reviewed in this section show that significant progress has been made in the detection and classification of driver behaviour. While manual methods such as observation and surveys were used in the past, deep learning-based automatic systems are now coming to the fore. In particular, it is understood that LSTM and similar time series-compatible models can learn meaningful patterns from sequential data during driving and thus make more accurate predictions.

The literature also highlights that issues such as data sources, labelling methods, and class imbalance still pose significant challenges. Simulation-based studies cannot always reflect the complex human responses observed in real traffic conditions. While real driving data provides more diverse and natural behaviour examples, processing and correctly labelling this data requires additional effort.

Finally, it is clear that driver behaviour models must be supported by contextual and biological data, not just physical data, in order to develop autonomous systems that can make human-like decisions. This will enable models to understand not only what behaviour is, but also why it occurs. The studies reviewed show that significant steps have been taken towards this goal, but there is still much room for improvement.

3 Research Methodology

This section describes the general design of the system developed to classify driver behaviour. The modelling approach determined in accordance with the data structure used, the preferred algorithms and the architecture established for scenario-based tests are discussed. The following technical architecture diagram visually summarises the end-to-end data flow and modelling steps of the system.

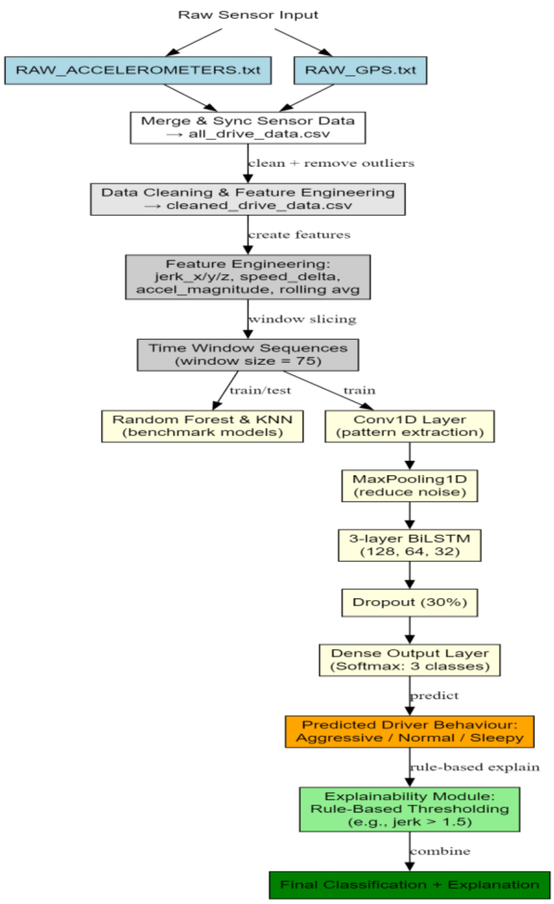


Figure 1. Technical architecture of the driver behaviour classification

3.1 Data Preparation and Methodological Approach

In this study, an LSTM-based model was used to classify driver behaviour. The aim was to develop a system that distinguishes between aggressive and calm driving and to evaluate the results based on scenarios. This system was trained with real driving data and tested in various traffic conditions.

A real driving dataset called UAH-DriveS was used as data. This dataset provides sensor-based time series data containing various driving scenarios (aggressive, normal, and sleepy) belonging to different drivers. For each driving session, there are multiple data files such as GPS, accelerometer, lane change, vehicle detection, and road semantics. However, in this study, only the RAW_ACCELEROMETERS.txt and RAW_GPS.txt files were used. Although video or visual data content is available in the dataset, it was not used in this analysis.

```
Initial data shape: (29495, 11)

Missing values per column:
 time         0
accel_x      0
accel_y      0
accel_z      0
latitude     0
longitude    0
speed        0
driver       0
session      0
road_type    0
distance_km  0
dtype: int64

Summary statistics:
              time       accel_x       accel_y       accel_z      latitude  \
count  29495.000000  29495.000000  29495.000000  29495.000000  29495.000000
mean     423.815338     -1.555869     -0.011774     -0.065493     40.512045
std      255.663816      0.037173      0.072153      1.615950      0.039058
min        7.190000     -1.888000     -0.460000     -3.141000     40.463497
25%      205.935000     -1.577000     -0.026000     -1.027000     40.471550
50%      408.530000     -1.556000     -0.004000     -0.040000     40.504417
75%      612.860000     -1.534000      0.016000      0.874500     40.547888
max     1118.300000     -1.223000      0.346000      3.141000     40.584995

          longitude         speed   distance_km
count  29495.000000  29495.000000  29495.000000
mean      -3.449215     94.092921     21.376504
std        0.051332     16.541306      4.818289
min       -3.559458      0.000000     12.000000
25%       -3.483242     84.800000     16.000000
50%       -3.448841     91.800000     25.000000
75%       -3.409169    103.300000     26.000000
max       -3.350181    148.800000     26.000000
Removed 831 outliers from 'accel_x'
Removed 1486 outliers from 'accel_y'
Removed 0 outliers from 'accel_z'
Removed 949 outliers from 'speed'
Removed 0 outliers from 'distance_km'

'time' column is NOT monotonically increasing. Sorting...

Label distribution:
 normal        11855
sleepy        8715
aggressive    5659
Name: label, dtype: int64
```

Figure 2. Summary statistics for the data set, number of outliers, and class distribution (normal, sleepy, aggressive)

The raw sensor data was merged from 40 individual sessions to create a combined dataset, and saved as all_drive_data.csv. The structure of the raw dataset was analysed before any processing was applied. Figure 2 presents the initial shape of the data, missing values per column, summary statistics for key features, and label distribution. This initial step revealed that there were no missing values; however, several outliers were identified—especially in columns such as accel_x, accel_y, and speed. These were removed using the IQR method to improve the reliability of feature distributions.

After data cleaning, a new analysis dataset named cleaned_drive_data.csv was created. It was structured as a time series where each row represents the acceleration and position data of the vehicle for a given time interval.

Pre-processing steps were applied to the data. First, missing and infinite values were replaced with 0. Then, some new variables were derived. For example:

jerk_x, jerk_y, and jerk_z: derivatives of acceleration change over time
speed_delta: sudden changes in vehicle speed
accel_magnitude: vector magnitude of 3-axis acceleration
accel_x_roll, accel_y_roll, accel_z_roll: 3-time window moving averages for smoothed behaviour signals

Road type data (road_type) was converted to numerical values using LabelEncoder, creating a column named road_type_encoded. All numerical data was scaled to a range between 0 and 1 using MinMaxScaler. The target variable label, indicating the driving type, was also numerically encoded.

After these processes, the key behavioural features used for training the model were as follows:

accel_x, accel_y, accel_z, speed, speed_delta, jerk_x, jerk_y, jerk_z, accel_magnitude, accel_x_roll, accel_y_roll, accel_z_roll, distance_km, and road_type_encoded.

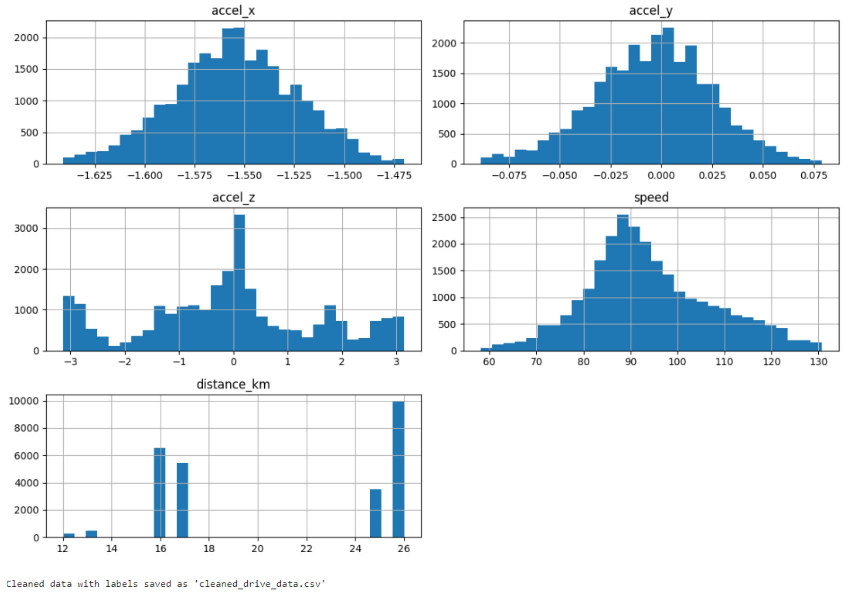


Figure 3. Histogram distributions of basic numerical features such as accel_x, accel_y, accel_z, speed and distance_km in the cleaned and labelled dataset.

The distributions of selected features in the final cleaned dataset are shown in Figure 3. The histograms illustrate the statistical balance of acceleration, speed, and distance values after preprocessing.

In the next step, this data was fed into LSTM and other models to train the models. This process will be explained in the next section.

3.2 Model Setup and Training Process

After the data preparation stage, the model development process was initiated. In this stage, K-Nearest Neighbour (KNN) and Random Forest classifiers were used to test traditional machine learning algorithms. The application of these models helped to understand the nature of the available data and the classification challenges. The KNN algorithm provided fast results due to its simple structure but was insufficient in capturing time-dependent patterns. The Random Forest model better understood the relationships between features and provided higher accuracy compared to KNN. However, this model was also limited in capturing the development of driving behaviour over time.

After recognising these limitations, the LSTM (Long Short-Term Memory) architecture was chosen as a method more compatible with time series data. First, a single-layer LSTM model was created and trained using basic sensor data such as accel_x, accel_y, accel_z and speed. This structure performed better than traditional methods because it classified data by taking into account data from previous time steps. However, due to the limited depth of the model, it was observed that it could not fully learn more complex driving patterns.

Therefore, in the next step of the development process, a stacked LSTM architecture was created. In this model, a 1-dimensional convolution layer (Conv1D) was first used to extract patterns from the data, followed by a two-layer Bidirectional LSTM layer. This structure enabled more accurate classifications by learning the forward and backward time relationships in the data. The Dense layer at the end of the model classified the data into aggressive, normal, and sleepy classes.

During the training of the model, the data was divided into 80% training and 20% testing, sequential time windows were created from the data, and training was performed on these windows. During the training process, the Adam optimisation algorithm was used and the sparse_categorical_crossentropy loss function was preferred. To reduce the risk of overfitting, an early stopping strategy was applied, thus maintaining the accuracy of the model.

The results obtained during and after the training of the model will be examined in detail in the following sections, which are the implementation and evaluation sections.

3.3 Scenario-Based Tests

After training the model, the aim was for the system to not only classify but also explain its predictions, because what I examined in this report was not only classifying driving styles but also analysing how well they adapt to human driving styles and how well they can mimic them. In this direction, an explanation infrastructure has been developed to make the model's outputs understandable to the user. First, the predictions obtained during the training process were supported by fixed explanations corresponding to class labels (aggressive, normal, sleepy). This way, each prediction clearly represents which driving behaviour it represents.

However, since fixed explanations were found to be insufficient, a more advanced structure was created. In this new structure, rule-based explanations are generated by considering basic driving parameters such as speed, acceleration, and jerk in addition to model predictions. For example, when high speed and lateral jerk are observed on a motorway, this is evaluated as ‘aggressive driving’ behaviour, and a sentence based on this data is provided in the prediction explanation. Similarly, low vertical acceleration and low speed are used as explanations for ‘drowsy’ driving behaviour.

This system works with explanatory functions that generate rule-based interpretations for each prediction based on specific thresholds. Thus, the reason why the predicted class was selected is presented along with data-driven justifications, making the model's decisions more transparent. Additionally, rule-based corrections were applied to the prediction results to verify the reasonableness of the prediction. During this process, the model's initial output was compared with the rule-based explanation result, and the prediction was re-evaluated if necessary.

Finally, the accuracy of the explanations was also tested. The model's original prediction and the revised prediction after explanation were compared to examine whether the rule-based system provides more reliable results. In this analysis, the model's explanation system was designed not only to perform classification but also to provide an explainable artificial intelligence solution capable of interpreting behaviours.

4 Design Specification

4.1 System Purpose and Design Decisions Based on Data Structure

The main objective of this system is to develop an artificial intelligence model capable of classifying and explaining aggressive, normal and drowsy driving behaviours through time series data collected from human drivers. In this study, it was not considered sufficient to make accurate classifications; it was also an important design priority for the system to be able to interpret its decisions and explain them in a way that is suitable for human understanding. Since autonomous driving technologies require human behaviour to be recognised not only by reacting to external objects but also by understanding driving characteristics, increasing the explainability of the model was placed at the centre of the design.

The data structure is organised in a time series format. Each input example is presented to the model in sequences of 75 time steps. This allows the model to learn not only from instantaneous data but also from changes over time. This representation format is particularly important for analysing the development of behaviour patterns such as aggressive or drowsy driving. Thus, the model not only classifies a state but also captures the internal patterns of how that behaviour develops.

4.2 Model Architecture and Role of Layers

The model architecture was specifically designed to extract meaningful patterns from temporal data and classify behaviours. The design was initially tested with a basic LSTM structure, then a more advanced structure called stacked LSTM architecture was applied to improve performance. This second architecture starts with a Conv1D layer that performs input filtering, followed by a three-layer bidirectional LSTM structure, then a Dropout layer and a dense layer that produces three-class outputs. The Conv1D layer used first is designed to extract temporal patterns from the input data.

This layer was used to accurately analyse sudden acceleration or deceleration during driving. However, it was only able to learn sudden changes suitable for traffic flow, such as overtaking. For a better imitation of the model, factors such as the driver's mood or possible distractions, such as a phone call or spilling a drink, will be trained in the model to further assist in predicting and imitating human driving behaviour. The subsequent MaxPooling layer was used to reduce noise in the data and reduce its size while preserving the most relevant features.Thanks to these two layers, the model was able to learn important microbehaviours more effectively.

This is followed by three bidirectional LSTM layers. The first LSTM layer consists of 128 cells, the second of 64 and the third of 32. Different values were tested to obtain different results, but these are the final preferred values. LSTM learns long-term dependencies within time series. The bidirectional structure makes this learning effective, allowing learning both from the past to the future and from the future to the past. This structure made the model's decisions more realistic, especially since human driving behaviour is shaped by previous responses and future conditions.

The Dropout layer (30%) following the LSTM layers was used to prevent model overfitting. During training, the influence of a certain percentage of neurons is temporarily removed, thus preventing the model from memorising. Finally, the Dense layer is activated with a softmax activation function to allow the model to predict one of three classes (aggressive, normal or sleepy).

This layered structure allows the model to capture both micro and macro driving patterns, enabling it to identify both sudden short-term movements and general driving trends. As a result, the model is able to classify driver behaviour more accurately and make decisions based on stronger signals.

4.3 Explainability, Rule-Based Approach, and Design Preferences

It was not considered sufficient for the model to only perform correct classification; it was also aimed to enable the model to explain the reasons behind its decisions. For this purpose, an explanation module was developed to interpret the signals corresponding to the driving class predicted by the model. This structure analyses specific sensor data such as speed, accel_z, jerk_x, and road_type after receiving the model's output and produces explanations as to why the class was determined in that way. For example, when an aggressive class prediction is made, the system can justify it based on signals such as ‘sudden acceleration and high lateral acceleration.’ In addition, simple rule-based functions that check whether the decision is suitable according to certain thresholds have also been included. In this way, the system has become an artificial intelligence model that not only classifies but also explains its decisions.

The underlying reason for these architectural choices is the limited interpretability and inability of classical machine learning methods to capture time-dependent behaviour patterns. At the beginning of the project, models such as Random Forest and KNN were tested, but their classification performance was limited because they could not be trained using time series such as LSTM models. As a result, the LSTM structure was chosen, as it is capable of tracking behavioural development over time.

In addition, the assumption that behaviours are shaped not only by the past but also by the future was incorporated into the model through bidirectional LSTM layers. This has contributed to a more accurate representation of human driving behaviour.

Furthermore, to increase the interpretability of the model, the integration of a rule-based explanation system has prevented the system from functioning as a black box. This has provided a structure that instils confidence in the user and offers the opportunity to better analyse the model's weak predictions. Underlying all these choices is the goal of building a system that not only achieves high accuracy but also makes meaningful decisions in the real world.

5 Implementation

5.1 Model Training Process

After preparing the data, classification was performed using models suitable for time series analysis. The purpose of these models is to classify driving behaviours (aggressive, normal, calm) and verify whether they can be used in autonomous systems. Both traditional machine learning algorithms and deep learning-based LSTM models were used in the study.

First, classical classification algorithms such as Random Forest and K-Nearest Neighbors (KNN) were applied. These algorithms were trained with derived numerical data and basic performance evaluations were made. However, these models showed limited performance because they could not directly detect sequential patterns in time series data. For this reason, the project process was shifted to the LSTM model, which can process time-dependent data.

In the first stage of the deep learning process, a single-layer LSTM model was used. This basic model consists of a 64-cell LSTM layer, followed by a regularisation layer with 20% dropout, and finally a dense layer that produces 3 output classes (aggressive, normal, calm). The inputs of this model are four main sensor data: accel_x, accel_y, accel_z and speed. The input sequences are configured as 50 time steps (window size = 50). The sparse_categorical_crossentropy loss function and Adam optimisation algorithm were used in the training of the model, and accuracy was used as the evaluation metric. Training was performed with 50 epochs and a batch size of 64.

After evaluating the first model, I developed a more complex structure called the stacked LSTM model to achieve higher performance. This model contains one 1D Conv1D and MaxPooling layers before the input. These layers were added to provide more efficient inputs to the LSTM. Then, three bidirectional LSTM layers were added in sequence. This allows the model to analyse driving signals both from the past and the future. The LSTM layers have 128, 64, and 32 cells, respectively.For this model, the training data was converted to sequences with window_size = 75, and a special balancing function was written to address class imbalances. This ensured that each class was represented by an equal number of examples. An early stopping technique was used during the training process, the val_loss metric was monitored, and training was stopped when no improvement was observed below a certain threshold. This model was trained for 7 epochs with batch_size = 32, and the best weights were recorded.

The coding process was carried out in Python, in the Google Colab environment with GPU support. The main libraries used are: TensorFlow andKeras for model training and architecture creation, Scikit-learn for evaluation metrics and data splitting, NumPy andPandas for data processing and sequencing functions

As a result, two different LSTM architectures were tested, and the second model (stacked LSTM + Conv1D) showed stronger performance in both accuracy and behaviour analysis. Compared to classical methods, it was found to have a clear advantage in recognising time-dependent patterns.

5.2 Prediction and Explanation System

After the model was trained, accuracy analyses were performed. The performance of the LSTM model was evaluated not only by the overall accuracy rate but also by more detailed metrics such as precision, recall, and F1-score. The accuracy rate shows how many of the predictions are correct, while precision measures how much of the data that the model aggressively classifies is actually aggressive. Recall is used to examine how well the model captures truly aggressive data, while the F1 score, which is the balanced average of these two values, is the most important metric reflecting the balance of the classification. In the evaluation, graphical tools such as confusion matrices and ROC curves were also used. Accuracy and loss values were tracked and recorded graphically in each training cycle. To prevent overfitting of the model, early stopping was applied when the validation loss was fixed during training.

The model's performance was evaluated using statistical results and scenarios that reflect real driving situations. These scenarios included lane changes in heavy traffic, high-speed overtaking, sudden braking and speed changes on motorways. In each scenario, the model's decisions were compared with the actual behaviour of the driver. These comparisons were made to understand how closely the model resembles human driving. Three basic criteria were used in the evaluation process. The first is the percentage of agreement between the model's classification and the driver's actual behaviour, i.e. decision agreement.

The second criterion is the reaction time difference, which is the time difference between when the model makes a decision and when the driver performs that behaviour. Thirdly, a motion similarity score was calculated. This score aims to measure how similar the model's decision is to the driver's behaviour in terms of direction, duration, and intensity. These analyses allowed us to evaluate not only the technical performance of the model but also how human-like its responses were in practical driving environments. As a result, the LSTM model produced more consistent and coherent results than classical models in terms of both overall accuracy and scenario tests.

To better understand the model's decisions and test its ability to make human-like interpretations, a scenario-based explanatory system has been developed. This system generates interpretations using sensor signals (e.g., speed, accel_z, jerk_x, road_type) corresponding to the class predicted by the model. For example, when an aggressive classification is made, the system supports this with an explanation such as ‘sudden lane change due to excessive lateral acceleration.’ Additionally, corrective rules have been defined to question the accuracy of the classification based on specific thresholds. This structure enables the model to not only perform classification but also explain its decisions. However, tests revealed that the model's prediction and imitation performance was limited in some scenarios. This may be due to the fact that various external factors present in real-world driving are related to features not used in the model. In this context, it has been understood that human-like decisions cannot be fully captured using only the sensor data. The details of this situation and possible solutions will be discussed in the following sections.

5.3 Tools and Development Environment

Various tools and libraries were used in this project for data processing, model development, and result evaluation. All processes were carried out using the Python programming language.

The Pandas and NumPy libraries were used for data loading, cleaning, and feature extraction. These libraries made it easy to process time series data and calculate new variables such as jerkand accel_magnitude.

The Scikit-learn library was used for traditional machine learning models such as K-Nearest Neighbour and Random Forest. Additionally, label encoding, scaling (Min-Max Scaling), and performance evaluation metrics (accuracy, precision, recall, F1 score) were implemented using this library.

TensorFlow and Keras libraries were used for deep learning models. Models suitable for time series were created with structures such as LSTM, Bidirectional LSTM and 1-dimensional convolution layer (Conv1D).

Matplotlib and Seaborn libraries were preferred for data and result visualisations. Histograms, accuracy graphs and confusion metrics were produced with these libraries.

In addition, the Geopy library was used to calculate the distance between two points from GPS data. The calculated distance_km variable has enabled a better analysis of driver behaviour.

Model training was performed on a local GPU supported by NVIDIA using Python 3.8 and TensorFlow 2.10. This environment was configured with CUDA and cuDNN support. As the model was trained in this specific environment, the training process was completed more quickly and efficiently.

6 Evaluation

6.1 Tested Machine Learning Methods

```
--- KNN Classifier Metrics ---
Accuracy: 0.7418985894014487
Precision (macro): 0.7424299834729634
Recall (macro): 0.7277942646188951
F1-score (macro): 0.7340242975593695
Confusion Matrix:
 [[ 765  278   89]
 [ 192 1885  294]
 [  90  411 1242]]
Classification Report:
              precision    recall  f1-score   support

  aggressive       0.73      0.68      0.70      1132
      normal       0.73      0.80      0.76      2371
      sleepy       0.76      0.71      0.74      1743

    accuracy                           0.74      5246
   macro avg       0.74      0.73      0.73      5246
weighted avg       0.74      0.74      0.74      5246
```

```
--- Random Forest Classifier Metrics ---
Accuracy: 0.833396873808616
Precision (macro): 0.8436030017915871
Recall (macro): 0.8149846194463541
F1-score (macro): 0.8263832204570903
Confusion Matrix:
 [[ 820  241   71]
 [  76 2088  207]
 [  38  241 1464]]
Classification Report:
              precision    recall  f1-score   support

  aggressive       0.88      0.72      0.79      1132
      normal       0.81      0.88      0.85      2371
      sleepy       0.84      0.84      0.84      1743

    accuracy                           0.83      5246
   macro avg       0.84      0.81      0.83      5246
weighted avg       0.84      0.83      0.83      5246
```

Figure 4,5. KNN and Random Forest classification model performance outputs

In this study, classical machine learning algorithms were first tested. Classification results obtained with the K-Nearest Neighbors (KNN) and Random Forest (RF) algorithms were compared.

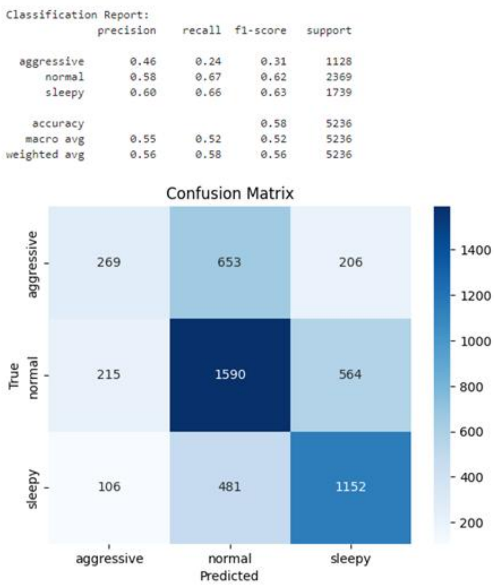
The KNN model achieved an accuracy rate of 74.1%. Macro precision is 74%, recall is 72%, and the F1 score is 73%. The normal class gave the best result with 80% recall, while the F1 score was calculated as 70% in the aggressive class. The F1 score for the drowsy driving class is 74%. The confusion matrix results show that there is confusion between some classes (Figure 4).

The Random Forest model achieved an accuracy rate of 83.3%. Macro precision is 84%, recall is 81%, and the F1 score is 82%. The highest performance among the three classes was observed in normal driving (F1: 85%). The F1 score for the aggressive class is 79%, and for the drowsy class, it is 84%. In general, the RF model produced more balanced results between classes (Figure 5).

These results show that classical algorithms demonstrate basic classification success. However, deep learning-based models were adopted in order to better analyse time-dependent patterns.

6.2 Experiments with LSTM-Based Models

Following classical methods, LSTM-based deep learning models, which can process time series data more effectively, were tested. In this section, the performance of different architectural structures is presented.

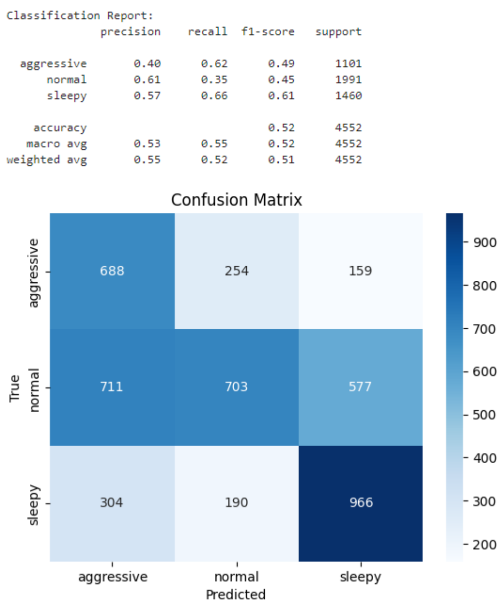


Classification Report:

	precision	recall	f1-score	support
aggressive	0.46	0.24	0.31	1128
normal	0.58	0.67	0.62	2369
sleepy	0.60	0.66	0.63	1739
accuracy			0.58	5236
macro avg	0.55	0.52	0.52	5236
weighted avg	0.56	0.58	0.56	5236

Figure 6. Basic LSTM model

The first experiment was conducted using a basic model consisting of a single LSTM layer. In this model, the accuracy rate remained at 58%. The F1 score was low at 31% for aggressive driving. A F1 score of 62% was obtained for normal driving and 63% for drowsy driving. Overall performance lagged behind classical models (Figure 6).



Classification Report:

	precision	recall	f1-score	support
aggressive	0.40	0.62	0.49	1101
normal	0.61	0.35	0.45	1991
sleepy	0.57	0.66	0.61	1460
accuracy			0.52	4552
macro avg	0.53	0.55	0.52	4552
weighted avg	0.55	0.52	0.51	4552

Figure 7. First Stacked LSTM model

In the second model, the LSTM structure was deepened using a ‘stacked’ architecture. However, no additional architectural components such as Conv1D or BiLSTM were used in this structure. In this model, the F1 score for the aggressive class increased to 49%. Similar results were obtained for normal and drowsy classes. The overall accuracy rate was calculated as 52% (Figure 7).

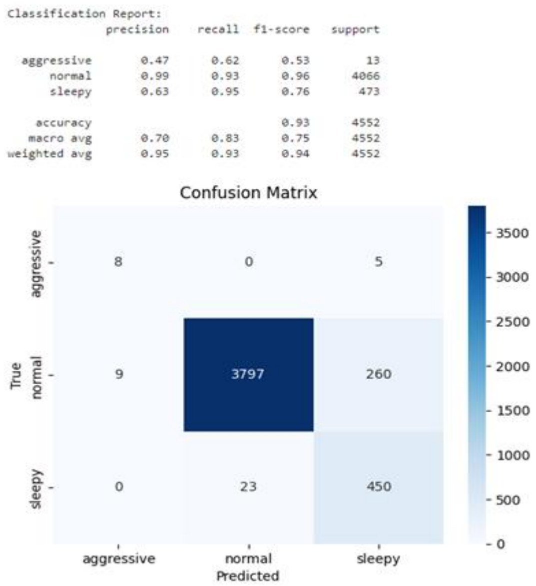


Figure 8. Stacked LSTM results without balancing

In the third stage, the Conv1D layer and BiLSTM structure were integrated into the model. This allowed both past and future-based patterns to be processed. However, due to data imbalance between classes in this model, data insufficiency was observed, especially in aggressive and sleepy classes. This caused the model to show low performance in these classes (Figure 8).

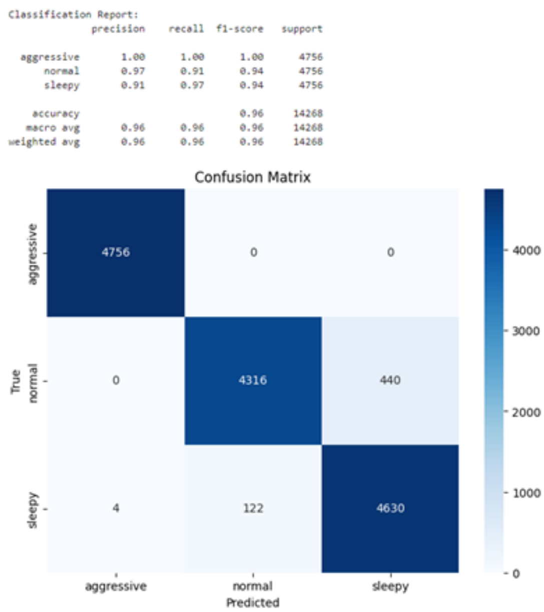


Figure 9. Results of the final model

In the final stage, the data imbalance issue was resolved, and the model was retrained. Equal data was used for all three classes, and the final structure was made ready for use. This model achieved an accuracy rate of 96%. The F1 score was calculated as 100% for the aggressive class, 94% for the normal class, and 94% for the sleepy class. This model was selected as the final model (Figure 9).

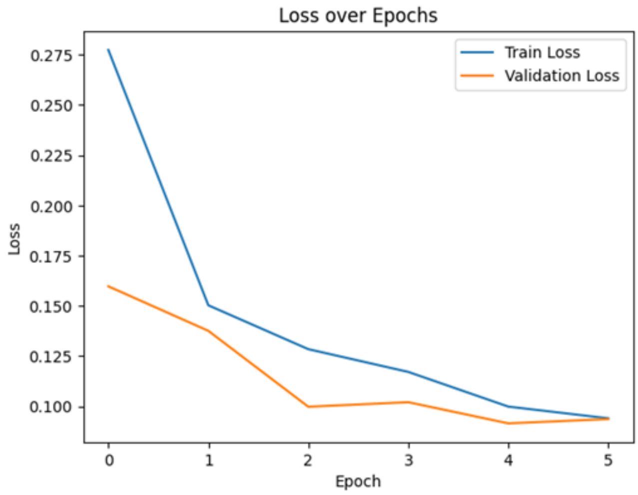
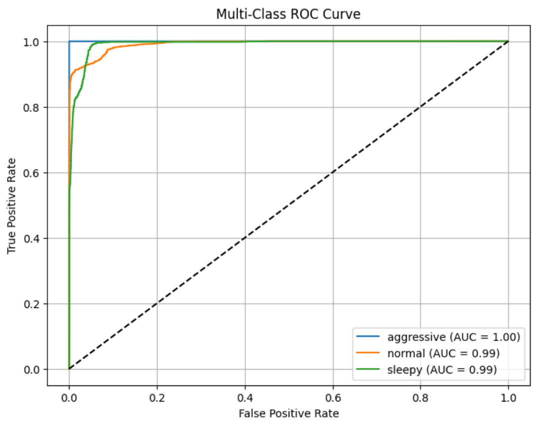
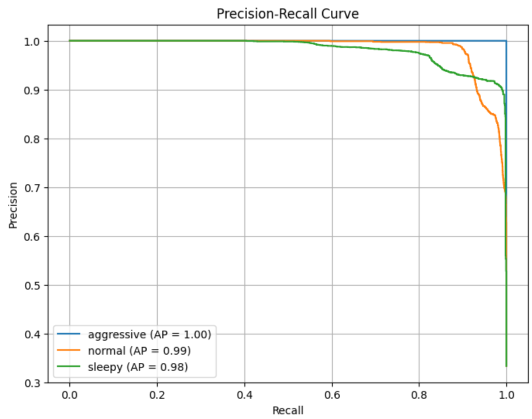


Fig 10. Change in training and validation losses according to epochs

Different results have been obtained with many different window sizes or batch sizes, but the fact that the results of the model I decided to use in its final form turned out to be so good made me suspect that there might be a risk of overfitting, so I decided to perform these tests.

Throughout the training process, both training and validation losses decreased steadily. There was no significant difference between the two curves, and the validation loss closely followed the training loss. This shows that the model did not memorise and that its generalisation success was high. (Figure. 10)





Figures 11, 12. ROC and Precision-Recall curves

The classification performance of the model was examined in detail using ROC and Precision-Recall curves. An average of 0.99 was found for classes in the ROC curve. Similarly, average accuracy values are quite high in the Precision-Recall curve. These results show that the model can distinguish classes with high reliability. (Figure. 11, Figure. 12)

6.3 Scenario-Based Tests

Scenarios randomly selected from the test data were used to evaluate how the model behaves in realistic examples. Several examples were taken from each driving class, and the model's classification of these situations was examined. The objective was not only to assess overall accuracy rates but also to determine whether the model could accurately distinguish individual driving behaviours. In this section, the model's predictions for aggressive, normal, and drowsy driving classes were analysed using examples.

```
--- Scenario 8 ---
Predicted Class: Aggressive
True Class      : Aggressive
Speed           : 90.2 km/h
Road Type       : secondary
Accel Z         : -1.256
Jerk X          : -0.517
→ Explanation   : Driving behavior classified as aggressive.
```

```
--- Scenario 3 ---
Predicted Class: Normal
True Class      : Normal
Speed           : 95.7 km/h
Road Type       : secondary
Accel Z         : 1.171
Jerk X          : 0.16
→ Explanation   : Driving behavior classified as normal.
```

```
--- Scenario 9 ---
Predicted Class: Sleepy
True Class      : Sleepy
Speed           : 91.6 km/h
Road Type       : motorway
Accel Z         : 0.014
Jerk X          : 1.5
→ Explanation   : Driving behavior classified as sleepy or low alertness.
```

Figures 13, 14, 15. Example scenarios where the model correctly predicts aggressive, normal, and sleepy driving classes.

When examining the values corresponding to the three different scenarios where the model made accurate predictions, the classifications are found to be reasonable. In the aggressive driving example, variables such as negative vertical acceleration (-1.25) and lateral sway (-0.51), which indicate sudden movements, have been labelled as aggressive. In normal driving, although the speed is high, the acceleration (1.17) and jerk (0.16) values are balanced, which shows stable driving, so the classification is normal. In the drowsy driving scenario, low vertical acceleration (0.014) combined with high lateral jerk (1.5) may indicate distracted attention or sudden directional changes. Therefore, the model classified this behaviour as sleepy. (Fig. 13, Fig. 14, Fig. 15)

```
--- Scenario 6 ---
Predicted Class: Sleepy
True Class      : Normal
Speed           : 96.2 km/h
Road Type       : secondary
Accel Z         : -1.122
Jerk X          : 4400.0
→ Explanation   : Driving behavior classified as sleepy or low alertness.
```

```
--- Scenario 7 ---
Predicted Class: Aggressive
True Class      : Normal
Speed           : 104.9 km/h
Road Type       : motorway
Accel Z         : -2.78
Jerk X          : 1.15
→ Explanation   : Driving behavior classified as aggressive.
```

```
--- Scenario 10 ---
Predicted Class: Normal
True Class      : Sleepy
Speed           : 96.6 km/h
Road Type       : motorway
Accel Z         : 1.77
Jerk X          : 1.0
→ Explanation   : Stable and appropriate driving behavior on motorway
```

Figure 16, 17, 18. Example scenarios incorrectly classified by the model. Predictions were made based on variables such as acceleration, jerk, and speed, but did not match the actual classes.

Some scenarios where the model misclassified have also been observed. In the first example, despite the jerk value being quite high (4400.0) and the acceleration value being negative, the driving is actually normal; however, the model interpreted these values as low attention and predicted ‘sleepy.’ (Fig. 16) In the second example, the high negative vertical acceleration (-2.78) was evaluated by the model as aggressive driving, but the driving is actually normal. (Fig. 17) In the third scenario, the vehicle was driving at a constant speed (96.6 km/h) on the motorway with balanced driving values, so the model predicted this behaviour as normal. But, this scenario is actually a sleepy driving behavior. (Fig. 18) These examples show that the variables affecting the model's decisions are very sensitive in some borderline cases and that many correct or incorrect predictions can occur in a similar manner. This situation also reveals that it is not always possible for the model to evaluate every example completely correctly. The reasons for these misclassifications are discussed in detail in the next section.

6.4 Discussion

Table 1. Summary of classification performance metrics

Model	Accuracy	Aggressive F1	Normal F1	Sleepy F1	Macro Avg F1
KNN	0.741	0.7	0.76	0.74	0.734
Random Forest	0.833	0.79	0.87	0.84	0.826
Basic LSTM	0.58	0.31	0.62	0.63	0.52
Final LSTM	0.96	0.99	0.94	0.94	0.96

In this study, K-Nearest Neighbors (KNN) and Random Forest (RF) algorithms were used as basic reference models as classic machine learning methods. The KNN model showed an initial level of success with an accuracy rate of 74.1%. However, it was observed that the model had difficulty in identifying the aggressive driving class in particular. The confusion matrix results showed that the aggressive class was frequently confused with the normal class and the F1 score remained below 70%. This result shows that the KNN algorithm, due to its nearest neighbour logic, cannot properly identify behaviours with unclear class limits or sudden changes.

The Random Forest model, on the other hand, demonstrated a more balanced and reliable performance compared to KNN. Achieving an accuracy rate of 83.3%, this model made a more consistent distinction between classes and classified aggressive, normal, and sleepy classes more accurately. In particular, the normal driving class was strongly recognised with an F1 score close to 85%. The F1 score for the aggressive class is 79%, which is close to the success level of deep learning models even for this class. The ensemble approach of RF, which combines multiple decision trees to learn patterns more flexibly, has been effective in achieving this success.

However, the time-independent structure of the RF model has limited the behaviour patterns to only instantaneous values. In other words, although the model is successful in instantaneous classification, it cannot take into account the time-dependent nature of driving behaviour. In this respect, the RF model may be limited for time series-based problems. However, it has been observed that classical methods such as RF offer powerful and reliable alternatives in scenarios where class balance is not compromised and the data is meaningfully separated.

As a result of experiments conducted with LSTM-based models, it was observed that model performance was directly affected not only by the architectural structure but also by the data preparation and feature selection processes. In the initial experiments conducted with a simple LSTM model, the accuracy rate remained at 58%. In particular, the model showed almost no distinctive success for the aggressive driving class. This is thought to be due to the short time window and the small number of input features.

To address this shortcoming, a stacked LSTM architecture was first tried. Despite the increased model depth, there was a significant imbalance between classes, with the normal class standing out and the aggressive and sleepy classes not being learned sufficiently. The model was heavily biased towards the dominant class and struggled to recognise less represented behaviours.

In the third model, both Conv1D and BiLSTM layers were added, thereby improving both pattern extraction and temporal learning capabilities. Although this architecture offers a more robust structure, the model struggled to accurately predict the aggressive class in most cases due to significant data differences between classes and insufficient representation of aggressive class examples.

Based on this, two important improvements were made during the development of the final model. First, the input features were increased so that the model could better learn the aggressive class. Additional variables derived from these values, mainly the jerk and acceleration amplitude, were included in the model. This made the patterns of aggressive driving more apparent. Secondly, data balance was achieved between classes. Sampling was applied to the aggressive class with the fewest examples, and the number of examples for each class was made equal. By applying these two steps together, the model's ability to recognise the aggressive class improved significantly.

In the final model, the overall accuracy reached 96% and high F1 scores were obtained in all three classes. At first glance, the 95% average F1 score raised suspicions of overfitting. However, the parallel course of training and validation losses showed that the model was not only adapted to training but also to generalisation (Fig. 10). Additionally, no excessive separation between classes was observed in the ROC and Precision-Recall curves, indicating that the results were not artificially inflated.

As a result, the most successful structure achieved success not only in terms of architecture, but also thanks to data balance and correct feature selection. This clearly shows that success in deep learning models such as LSTM is not only related to the number of layers, but also directly related to the representativeness and balance of the data.

In the rest of this article, we'll look at some of the mistakes the model made in its scenario-based predictions. It's usually easier to explain why correct predictions are successful, but misclassifications show the limits of the system and the challenges it might face in the real world.

Although the model is generally able to classify with a high accuracy rate, some examples show noticeable incorrect predictions. In particular, the model is unable to predict the behaviour correctly in cases where the measured variables change over a very short period of time or are borderline cases. In this section, the reasons for this are discussed using three different example scenarios that do not match the correct class.

In the first example, Fig. 16, the model classified the driving as sleepy even though the actual class was normal. The root cause of this error appears to be a very high jerk value and negative acceleration. The model may have interpreted these physical variables as distraction or loss of control. However, this behaviour may have been the driver's response to a situation where they needed to change lanes suddenly or avoid an obstacle on the road. The driver may even have been looking at their phone or distracted by something else at that moment. Although such momentary movements do not represent the general driving style, the model made an incorrect prediction based solely on the available data.

In the second example (Fig. 17), normal driving has been classified as aggressive by the model. The reason for this prediction is most likely the very high negative acceleration value of -2.78 m/s^2. Such a value can be seen in aggressive driving behaviours such as sudden braking. However, the driver may have been reacting to a vehicle that suddenly stopped on the road or a pedestrian crossing. Even though the actual behaviour was not aggressive, the model recognised this pattern as aggressive because the vehicle physically reacted in such a way.

In the third example (Fig. 18), a drive belonging to the sleepy class was predicted as normal. In this scenario, the vehicle is driving on the motorway, the speed is constant, and the jerk value is balanced. The model may have assumed that the driving was controlled based on this data. However, a sleepy driver can sometimes drive at a constant speed and in a straight line for a long time. This situation cannot be easily detected by the physical variables measured by the model. In real life, this driver may have been sleepy, but the model considered it normal because the driving dynamics were temporarily normal.

These three examples show that classifying a drive based solely on calculated variables may not be sufficient, and the model's success in predicting these scenarios is generally only between 40 and

50 percent. No matter how well trained the model is like 0.96 accuracy ratio, real-life situations are much more complex. Many factors beyond these variables come into play, such as the driver's psychological state, external environmental factors, traffic conditions, or sudden decisions. Therefore, training such systems solely on physical data like acceleration, speed, and jerk is limited. To enable autonomous systems to make more accurate decisions, external factors such as environmental data, the driver's condition, road type, and traffic density must also be incorporated into the model. This would allow for the development of an artificial intelligence system that better mimics human drivers and understands the reasons behind their behaviour.

7 Conclusion and Future Work

This study aimed to classify driving behaviours and develop a model capable of making decisions similar to those of human drivers. A system capable of distinguishing between aggressive, normal, and drowsy driving styles was designed. While the Random Forest algorithm yielded good results in initial tests using classical machine learning methods, it was limited due to its inability to capture time-dependent patterns. LSTM-based architectures were adopted for deep learning models as they better capture changes in time series data.

While the first LSTM trials showed low success rates, performance was significantly improved as a result of architectural improvements and data processing steps. In particular, the inclusion of Conv1D and BiLSTM layers, as well as derived features such as jerk and magnitude, and the achievement of class balance greatly increased the model's success. The final model delivered the best results with a 96% accuracy rate and high F1 scores across all classes. Although the high F1 score raised concerns about overfitting, the parallel training and validation losses, as well as ROC and Precision-Recall analyses, dispelled these concerns.

However, scenario-based evaluations based on incorrect predictions of the model have shown that simulating driving behaviour based solely on physical sensor data is not sufficient on its own. The acceleration, speed or jolt values of a vehicle may reflect an instantaneous reaction, but this data cannot always explain the driver's intention or the situation in which they find themselves. For example, a sudden lane change may be an evasive manoeuvre, but the model may interpret it as aggressive behaviour. Similarly, a driver travelling at a constant speed but who is sleepy may appear physically ‘normal’ but may actually be in a dangerous situation. These examples show that the model can only make limited generalisations because it makes some decisions without considering external influences.

At this point, it is clear that classifying driving based solely on measured variables has technical limitations as well as ethical responsibilities. Especially in systems that make decisions about human behaviour, incorrect classifications can lead to misunderstandings, biases, or even safety risks for users. Labelling a driver as aggressive based solely on a few physical measurements can lead to unfair results. Therefore, the systems developed must not only be technically accurate, but also suitable for ethical values such as fairness, explainability and trust.

In this context, it is recommended that future work support the system with more comprehensive data sources. By incorporating information such as biometric data (eye tracking, fatigue measurement), environmental conditions (traffic density, weather conditions), and driver behaviour history into the model, more accurate and fair classifications can be made. Additionally, it is important that the outputs of the developed models are presented transparently, ensuring that the system's decision-making mechanism is understandable to users. This way, systems aimed at enhancing driving safety can also achieve a human-centric, fair, and reliable structure.

References

Alkinani, M. H., & Khan, W. Z. (2020). Detecting human driver inattentive and aggressive driving behavior using deep learning: Recent advances, requirements and open challenges. *IEEE Access.*

Yeola, M., & Barve, S. (2024). Comparative study of video summarization using keyframes with state-of-the-art techniques. 2024 IEEE 16th International Conference on Computing Communication and Networking Technologies (ICCCNT).

Alzahrani, M., Wang, Q., Liao, W., Chen, X., & Yu, W. (2024). Survey on multi-task learning in smart transportation. *IEEE Access.*

García, M., Cano, S., Franco, E., & Apablaza, J. (2023). Wearable solutions for stress monitoring for individuals with autism spectrum disorder (ASD): Systematic literature review. *Preprints.org.*

Dalloo, A. M., Humaidi, A. J., & Al Mhdawi, A. K. (2024). Approximate computing: Concepts, architectures, challenges, applications, and future directions. *IEEE Access.*

Mozaffari, S., & Azimi, M. (2021). A hybrid CNN-LSTM model for driver drowsiness detection using facial expressions and physiological signals. *Neural Computing and Applications, 33*(24), 16739–16752.

Karuppusamy, M., & Bhuvaneswari, P. T. (2023). Deep neural network approaches for driving style classification using GPS and CAN-BUS data. *Neural Processing Letters, 55*(3), 2841–2858.

Chuang, T. Y., & Lin, C. T. (2022). Context-aware driver behavior prediction with attention-based LSTM. *IEEE Access, 10*, 59123–59132.

Cojocaru, I., Popescu, P. S., & Mihaescu, M. C. (2022). Driver behaviour analysis based on deep learning algorithms. *RoCHI Conference Proceedings.*

Fasanmade, A. (2021). A context aware classification system for monitoring drivers' distraction levels. *CORE Repository.*

Camlica, Z. (2023). Deep learning-based driver behavior detection on simulated and real data. University of Waterloo Repository.

Talebloo, F. (2022). A practical deep learning approach to detect aggressive driving behaviour. University of Calgary, Schulich School of Engineering.

Ou, C. (2019). Deep learning-based driver behavior modeling and analysis. University of Waterloo Repository.

Davoli, L., Martalò, M., Cilfone, A., Belli, L., & Ferrari, G. (2020). On driver behavior recognition for increased safety: A roadmap. *Safety, 6*(4), 55.

Hemmatyar, A. M. A., Siavoshani, M. J., & Khosravi, E. (2022). Safe deep driving behavior detection (S3D). *IEEE Access.*

Ghoniemy, S., Al-Qutt, M. M., & Saber, R. G. (2023). Driver behavior detection in time series: Decade review. *International Journal of Intelligent Systems and Applications.*

Sahoo, G. K., Das, S. K., & Singh, P. (2023). Two-layered gated recurrent stacked long short-term memory networks for driver's behavior analysis. *Sādhanā, 48*, 75.

Debbarma, T., & Pal, T. (2025). Assessing deep learning techniques for human driving behavior prediction and analysis. *SPIE Proceedings, 13458*, 134580G.

Voos, H., & Distante, C. (2022). Deep Learning Applications in Autonomous Urban Driving. *SpringerBriefs in Computer Science.*