

## Modelling Intelligent Driving Behaviour Using Machine Learning

Qura-Tul-Ain Khan<sup>1</sup>, Sagheer Abbas<sup>1</sup>, Muhammad Adnan Khan<sup>2,\*</sup>, Areej Fatima<sup>3</sup>,  
Saad Alanazi<sup>4</sup> and Nouh Sabri Elmitwally<sup>4,5</sup>

<sup>1</sup>School of Computer Science, NCBA&E, Lahore, 54000, Pakistan

<sup>2</sup>Riphah School of Computing & Innovation, Faculty of Computing, Riphah International University,  
Lahore Campus, Lahore, 54000, Pakistan

<sup>3</sup>Department of Computer Science, Lahore Garrison University, Lahore, 54000, Pakistan

<sup>4</sup>Department of Computer Science, College of Computer and Information Sciences, Jouf University,  
Skaka, Aljouf, 72341, Saudi Arabia

<sup>5</sup>Department of Computer Science, Faculty of Computers and Artificial Intelligence, Cairo University, 12613, Egypt

\*Corresponding Author: Muhammad Adnan Khan. Email: adnan.khan@riphah.edu.pk

Received: 21 November 2020; Accepted: 04 March 2021

**Abstract:** In vehicular systems, driving is considered to be the most complex task, involving many aspects of external sensory skills as well as cognitive intelligence. External skills include the estimation of distance and speed, time perception, visual and auditory perception, attention, the capability to drive safely and action-reaction time. Cognitive intelligence works as an internal mechanism that manages and holds the overall driver's intelligent system. These cognitive capacities constitute the frontiers for generating adaptive behaviour for dynamic environments. The parameters for understanding intelligent behaviour are knowledge, reasoning, decision making, habit and cognitive skill. Modelling intelligent behaviour reveals that many of these parameters operate simultaneously to enable drivers to react to current situations. Environmental changes prompt the parameter values to change, a process which continues unless and until all processes are completed. This paper model intelligent behaviour by using a 'driver behaviour model' to obtain accurate intelligent driving behaviour patterns. This model works on layering patterns in which hierarchy and coherence are maintained to transfer the data with accuracy from one module to another. These patterns constitute the outcome of different modules that collaborate to generate appropriate values. In this case, accurate patterns were acquired using ANN static and dynamic non-linear autoregressive approach was used and for further accuracy validation, time-series dynamic backpropagation artificial neural network, multilayer perceptron and random sub-space on real-world data were also applied.

**Keywords:** Machine learning; artificial neural network; ann; time series; intelligent behaviour; agent



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## 1 Introduction

Human error is the main cause of driving problems. Accordingly, driver behaviour modelling has emerged to improve the driving experience by predicting environmental factors, driver intent, and driver and vehicle state. Sensory information and other predictors can helpfully warn the driver of potential dangers. Meanwhile, vehicle behaviour can determine driver state, measuring elements including driving competence and attention level [1]. Understanding a driver's cognitive load is critical. Autonomous systems can help to automatically identify driver cognitive workload, enabling the development of robust evaluation tools. Driver performance and physiological data can be assessed and measurements collected in real-time. A classification system can detect cognitive load, with physiological data allowing greater classification accuracy [2]. In vehicular systems, drivers usually demonstrate different driving styles. To analyse explicit links between a driver's dynamic demand and driving style, driving style can be classified as low, moderate and high. This involves recognising driving style according to the vehicle, driving route, design task and driver selection [3].

All aspects of a human's life are defined by their routine behaviour [4]. Routine describes deliberate behaviour constituting goal-oriented actions performed in different situations [5], actions that are acquired, learned and developed through repeated practice [6]. Good routines enable efficient completion of frequently repeated tasks through predictable behaviours. Variations in routine behaviour importantly indicate human response because behaviour is not static [7].

Current research is directed towards finding driver behaviour solutions that can avoid road safety problems, especially by identifying and understanding the relationship between road safety and driver behaviour. The major contributing factors to traffic accidents are weather, traffic, vehicle control, and driver sensitivity to complex environments. However, traffic accidents are ultimately largely dependent on the rational judgement and decision making of drivers [8].

Behaviour analysis involves two different approaches to understanding decision-making: analysing the beliefs and the values behind the process and evaluating decision-making according to personality traits and individual habits. This method is quantifiable and can be represented in applications of classifying drivers based on their driving style [9].

## 2 Literature Review

Thinking essential to understanding the world and managing different situations, problems and relationships [10]. To encourage environmentally responsive behaviour, researchers have used people-oriented approaches which reduce feelings of helplessness and provide sources for motivations [11]. Such approaches allow people to accomplish repetitive tasks at different levels using the human routines that are the blueprints of behaviour. Human routines are expressed through actions performed in particular situations, with behaviour modelling enabling people to improve their inexpert behaviour and change bad habits [4].

Driving output is determined by various factors, with drivers considered the most unstable factors because they exhibit different driving styles. Driving style constitutes the habitual long-term behaviour of drivers, contributing to the real-time adjustments they make to environmental information [3]. Driver behaviour variables require analysis to represent concepts related to driving habits. Defining driver behaviour as one of several different driving behaviours enables identification of behavioural trends and allows accurate measurement of driving style, with the main goal of driver style classification being to detect behaviour, recognise different techniques and

enable learning [12]. For example, some researchers have classified eco-driving according to three categories: trajectory planning, route planning and driver behaviour improvement [13].

Elsewhere, driving styles have been categorised as normal, safe, inattentive and aggressive [9]. Risky speeding is a behavioural pattern pertaining to an aggressive driving style, describing irregular, abrupt or instantaneous changes in vehicle speed, improper vehicle positioning and inconsistent acceleration or braking [14]. Aggressive driving behaviour increases the collision risk and can arise from driver annoyance or attempts to minimise travel time [15]. Although repeated behaviour in a stable context can promote automatic habits which are resistant to change through information-based techniques, such behaviours are context-dependent, meaning a change in context can weaken the habit's strength and facilitate reflection on that behaviour [16].

Tab. 1 presents studies on the effects of emotional, sensorimotor, cognitive and mixed stressors on driver behaviour and performance. Notably, another study recognised the effective copying mechanism could reduce behavioural errors caused by cognitive or emotional conflict [17].

**Table 1:** Comparison of previous work

S. No	Title of paper	Author and year of publication	Methodology used
1	Know your master: Driver profiling-based Anti-theft method	[18] (Kwak et al., 2017)	Driver's behaviour patterns were formulated by game simulation and the Hidden Markov Model.
2	A new formal approach to model human driving behavior in vehicular networks	[19] (Bouhoute et al., 2014)	The formalism of hybrid input-output automata was applied to observed driver behaviour.
3	A study of individual characteristics of driving behavior based on Hidden Markov model	[20] (Zhang et al., 2014)	The Hidden Markov Model was employed to model the individual characteristics of driver behaviour.
4	Improving driver assistance in intelligent transportation systems: An agent-based evidential reasoning approach	[21] (Benalla et al., 2020)	A novel agent-based evidential reasoning system using contextual information was used to process driver behaviour information.

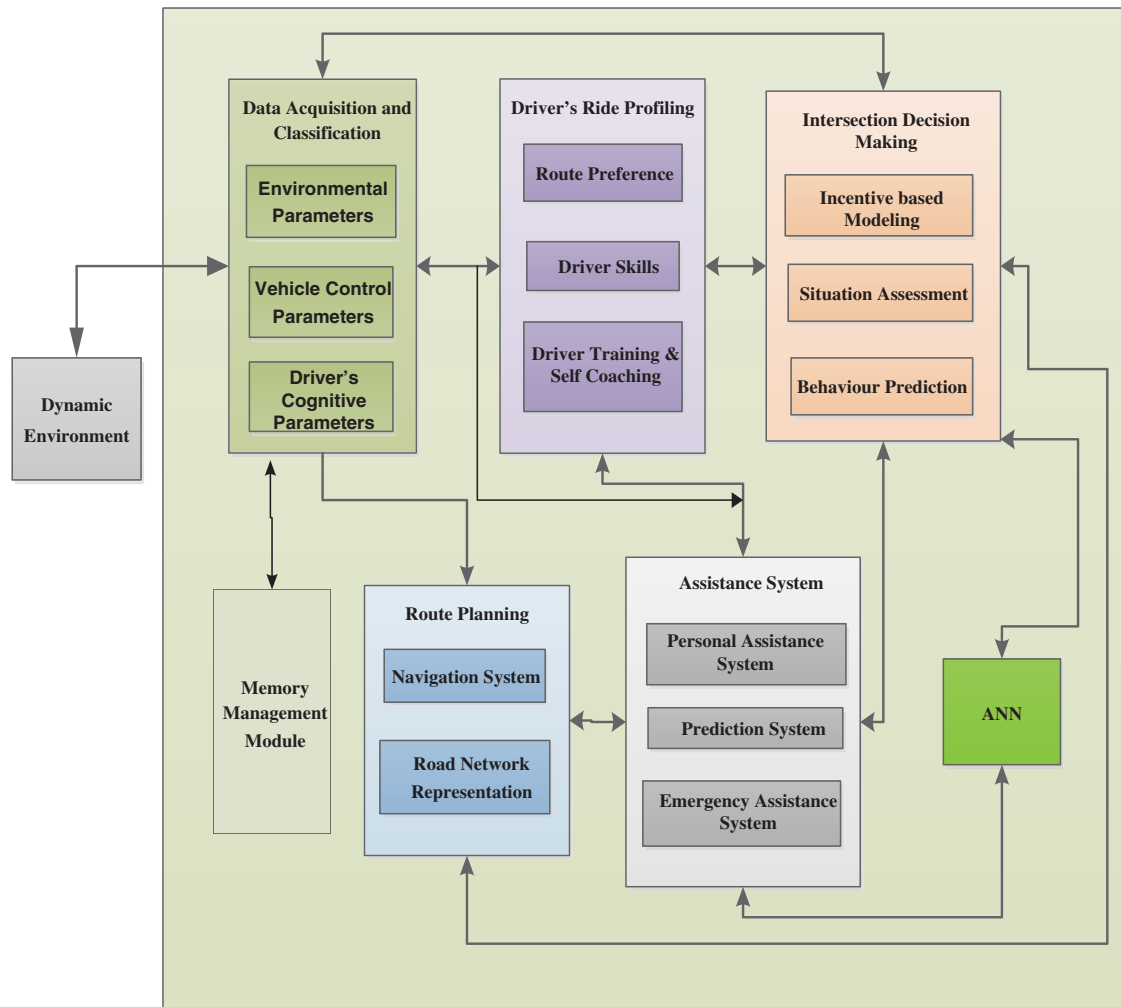
### 3 Methodology

The methodology comprised multiple modules (see Fig. 1 for a visual representation): a data acquisition and classification module, a driver ride profiling module, a decision-making module, a memory management module, a route planning module, an assistance system module and an Artificial neural network (ANN) module. Dataset validation involved a dynamic nonlinear autoregressive approach.

#### 3.1 Data Acquisition and Classification

Data acquisition and classification involves collecting environmental, vehicle and cognitive data. Environmental data include weather condition and time of day. Vehicle data include the condition of the vehicle, utilising inputs collected from all parts of the vehicle, including

accelerator pressure, braking, steering wheel movement, gear changes and vehicle turns. Cognitive inputs consider parameters such as intentions, motivations, emotions, knowledge, learning and decision-making. This research only analysed environmental and vehicle data.



**Figure 1:** Proposed driver behaviour model

### 3.2 Driver Ride Profiling

Driver ride profiling includes route preferences, driver skill, driver training and self-coaching. Route preferences includes long, medium and short routes, as well as factors such as terrain. Driver skill refers to driver expertise for a specific route. Driver training and self-coaching incorporate iterative learning, which describes driving training for a specific route.

### 3.3 Decision-Making Module

The decision-making module incorporates incentive-based modelling, situation assessment and behaviour prediction. Incentive-based modelling is responsible for decisions utilising working

progress, which includes the driver's past experience and how a driver operates a vehicle in a particular scenario, with scenario describing speed and weather condition.

Situation assessment considers the current environment using inputs from the data acquisition and classification, route planning and driver ride profiling modules. The behaviour prediction sub-module predicts behaviour after each complete iteration. Behaviour prediction interacts with the assistance system module to derive data patterns from that module's personal assistance and prediction systems.

### 3.4 Memory Management

Memory management stores relevant data and provides requested data to different modules.

### 3.5 Route Planning

Route planning considers speed limits, road types, traffic jams and weather conditions. Its navigation system includes online maps, available paths and alternative paths, with road network representation used to calculate the path according to road and location conditions.

### 3.6 Assistance System

The assistance system comprises the personal assistance system, prediction system and emergency assistance system. The personal assistance system guides the driver along their selected route, suggests changes to vehicle speed and assesses driver behaviour. The prediction system helps the driver to predict the next best route, the time of arrival at the destination and the driver's behaviour in particular scenarios. The emergency assistance system is only activated in case of emergencies, including sudden severe changes in the situation or the driver's behaviour.

### 3.7 Driver Behaviour Model Empowered by an Artificial Neural Network

The driver behaviour model uses an ANN to enable smooth data flow and dynamic and intelligent driver behaviour. The ANN is divided into static techniques and dynamic techniques. A dynamic nonlinear autoregressive approach was used to validate the model because driver behaviour is a constantly changing phenomenon. Other validation techniques used were multilayer perceptron and random subspace.

The ANN applied recognised human activity using an artificial backpropagation neural network and featured three layers: an input, output and hidden layer. Every neuron of the hidden layer used the Sigmoid(x) activation function. The proposed ANN is represented mathematically as:

$$\eta_{\Lambda} = Y_1 + \sum_{m=1}^l (t_{m\Lambda} * j_m) \quad (1)$$

$$\varphi_{\Lambda} = \frac{1}{1 + \beta^{-\eta_{\Lambda}}} \quad \text{where } \Lambda = 1, 2, 3, \dots, n \quad (2)$$

Combining Eqs. (1) and (2)

$$\eta_{\ddot{\gamma}} = Y_2 + \sum_{\Lambda=1}^n (\dagger_{\Lambda\ddot{\gamma}} * \varphi_{\Lambda}) \quad (3)$$

Eq. (3) provides backpropagation error, where  $\pi_{\check{y}}$  &  $out_{\check{y}}$  demonstrate the foreseen output and assessed output. The output layer's actuation function is described by Eqs. (4) and (5)

$$\varphi_{\Lambda} = \frac{1}{1 + \beta^{-\eta_{\check{y}}}} \quad \text{where } \check{y} = 1, 2, 3, \dots, j \quad (4)$$

$$S = \frac{1}{2} \sum_{\check{y}} (\pi_{\check{y}} - \varphi_{\check{y}})^2 \quad (5)$$

Weight change is described by Eq. (6):

$$\Delta\chi \propto -\frac{\dot{\Gamma}S}{\dot{\Gamma}\chi}$$

$$\Delta\dot{\Gamma}_{\Lambda, \check{y}} = -\epsilon \frac{\dot{\Gamma}S}{\dot{\Gamma}\underline{D}_{\Lambda, \check{y}}} \quad (6)$$

From Eq. (6), the chain rule method is applied:

$$\Delta\dot{\Gamma}_{\Lambda, \check{y}} = -\epsilon \frac{\dot{\Gamma}S}{\dot{\Gamma}\varphi_{\check{y}}} \times \frac{\dot{\Gamma}\varphi_{\check{y}}}{\dot{\Gamma}\eta_{\check{y}}} \times \frac{\dot{\Gamma}\eta_{\check{y}}}{\dot{\Gamma}\underline{D}_{\Lambda, \check{y}}} \quad (7)$$

From Eq. (7), values are substituted to provide the value of weight change according to Eq. (8)

$$\Delta\dot{\Gamma}_{\Lambda, \check{y}} = \epsilon (\pi_{\check{y}} - \varphi_{\check{y}}) \times \varphi_{\check{y}} (1 - \varphi_{\check{y}}) \times (\varphi_{\Lambda})$$

$$\Delta\dot{\Gamma}_{\Lambda, \check{y}} = \epsilon V_{\Lambda} \varphi_{\Lambda} \quad (8)$$

where

$$V_{\check{y}} = (\pi_{\check{y}} - \varphi_{\check{y}}) \times \varphi_{\check{y}} (1 - \varphi_{\check{y}})$$

The chain rule is applied to update weight between the input and hidden layers:

$$\Delta t_{m, \Lambda} \propto - \left[ \sum_{\check{y}} \frac{\dot{\Gamma}S}{\dot{\Gamma}\eta_{\check{y}}} \times \frac{\dot{\Gamma}\varphi_{\check{y}}}{\dot{\Gamma}\eta_{\check{y}}} \times \frac{\dot{\Gamma}\underline{U}_{\check{y}}}{\dot{\Gamma}\varphi_{\Lambda}} \right] \times \frac{\dot{\Gamma}\varphi_{\Lambda}}{\dot{\Gamma}\eta_{\Lambda}} \times \frac{\dot{\Gamma}\eta_{\Lambda}}{\dot{\Gamma}\dot{\Gamma}_{m, \Lambda}}$$

$$\Delta t_{m, \Lambda} = -\epsilon \left[ \sum_{\Omega} \frac{\dot{\Gamma}S}{\dot{\Gamma}\varphi_{\check{y}}} \times \frac{\dot{\Gamma}\varphi_{\check{y}}}{\dot{\Gamma}\eta_{\check{y}}} \times \frac{\dot{\Gamma}\eta_{\check{y}}}{\dot{\Gamma}\varphi_{\Lambda}} \right] \times \frac{\dot{\Gamma}\varphi_{\Lambda}}{\dot{\Gamma}\eta_{\Lambda}} \times \frac{\dot{\Gamma}\eta_{\Lambda}}{\dot{\Gamma}\dot{\Gamma}_{m, \Lambda}}$$

where  $\epsilon$  represents the constant and

$$\Delta\dot{\Gamma}_{m, \Lambda} = \epsilon \left[ \sum_{\check{y}} (\pi_{\check{y}} - \varphi_{\check{y}}) \times \varphi_{\check{y}} (1 - \varphi_{\check{y}}) \times (\underline{D}_{\Lambda, \check{y}}) \right] \times \varphi_{\check{y}} (1 - \varphi_{\check{y}}) \times \bar{\xi}_m$$

$$\Delta t_{m, \Lambda} = \epsilon \left[ \sum_{\check{y}} (\pi_{\check{y}} - \varphi_{\check{y}}) \times \varphi_{\check{y}} (1 - \varphi_{\check{y}}) \times (\underline{D}_{\Lambda, \check{y}}) \right] \times \varphi_{\Lambda} (1 - \varphi_{\Lambda}) \times \bar{\xi}_m$$

$$\Delta \mathbf{t}_{m, \Lambda} = \epsilon \left[ \sum_{\ddot{\gamma}} V_{\ddot{\gamma}} \left( \underline{D}_{\Lambda, \ddot{\gamma}} \right) \right] \times \varphi_{\Lambda} (\mathbf{1} - \varphi_{\Lambda}) \times \bar{\mathbf{g}}_m$$

Upon rearranging the previous equations, the condition can be calculated as

$$\Delta \theta \mathbf{t}_{m, \Lambda} = \epsilon V_{\Lambda} \bar{\mathbf{g}}_m \quad (9)$$

$$V_{\Lambda} = \left[ \sum_{\ddot{\gamma}} V_{\ddot{\gamma}} \left( \underline{D}_{\Lambda, \ddot{\gamma}} \right) \right] \times \varphi_{\Lambda} (\mathbf{1} - \varphi_{\Lambda})$$

$$\underline{D}_{\Lambda, \ddot{\gamma}}^+ = \underline{D}_{\Lambda, \ddot{\gamma}} + \dot{F}_{\bar{\mathbf{g}}} \Delta \dagger_{\Lambda, \ddot{\gamma}} \quad (10)$$

Eqs. (9) and (10) refresh the weights for the output and hidden layers; finally, Eq. (11) derives the weights for the hidden and input layers.

$$\mathbf{t}_{m, \Lambda}^+ = \mathbf{t}_{m, \Lambda} + \dot{F}_{\bar{\mathbf{g}}} \Delta \dagger_{m, \Lambda} \quad (11)$$

This ANN included one hidden layer and 20 neurons, with six inputs and one output. This ANN used the two-layer feed-forward method and was applied to test the framework using data categorised as training, approval, or testing. The ANN's execution was assessed using a regression investigation. To assess outcomes, we analysed the mean square error and regression fit. If the required result was not attained, the ANN was retrained with a different dataset.

### 3.8 Dynamic Environment

Dynamic environment incorporates vehicle status, traffic flow, weather conditions, road conditions and the driver's past behaviour.

## 4 Analysis and Results

The dataset was taken from [18] and featured 54 parameters and 94,380 values. Validation was conducted using the validation tools MATLAB and Weka.

### 4.1 Dataset Parameters

The dataset included two basic parameter types: input and output parameters.

### 4.2 Input Parameters

Tab. 2 presents the external parameters, which pertain to elements external to the driver.

### 4.3 Output Parameters

The output parameters presented in Tab. 3 pertain to driver elements.

### 4.4 Validation Techniques

Driver behaviour changes continuously and involves dynamic aspects. Accordingly, a dynamic ANN was used to analyse input values, with results presented according to the real-time scenario.

**Table 2:** Input parameters [18]

S. No	Variable	S. No	Variable
1	Fuel_consumption	22	Minimum_indicated_engine_torque
2	Accelerator_Pedal_value	23	Maximum_indicated_engine_torque
3	Throttle_position_signal	24	Flywheel_torque
4	Short_Term_Fuel_Trim_Bank1	25	Torque_scaling_factor(standardisation)
5	Intake_air_pressure	26	Standard_Torque_Ratio
6	Filtered_Accelerator_Pedal_value	27	Requested_spark_retard_angle_from_TCU
7	Absolute_throttle_position	28	TCU_requests_engine_torque_limit_(ETL)
8	Engine_soaking_time	29	TCU_requested_engine_RPM_increase
9	Inhibition_of_engine_fuel_cut_off	30	Target_engine_speed_used_in_lock-up_module
10	Engine_in_fuel_cut_off	31	Glow_plug_control_request
11	Fuel_Pressure	32	Activation_of_Air_compressor
12	Long_Term_Fuel_Trim_Bank1	33	Torque_converter_speed
13	Engine_speed	34	Current_Gear
14	Engine_torque_after_correction	35	Engine_coolant_temperature
15	Torque_of_friction	36	Wheel_velocity_front_left-hand
16	Flywheel_torque_(after_torque_interventions)	37	Wheel_velocity_rear_right-hand
17	Current_spark_timing	38	Wheel_velocity_front_right-hand
18	Engine_coolent_temperature	39	Wheel_velocity_rear_left-hand
19	Engine_Idel_Target_Speed	40	Torque_converter_turbine_speed_-_Unfiltered
20	Engine_torque	41	Clutch_operation_acknowledge
21	Calculated_LOAD_value	42	Converter_clutch

**Table 3:** Output parameters [18]

S.NO	Variables
1	Gear_Selection
2	Vehicle_speed
3	Acceleration_speed_-_Longitudinal
4	Indication_of_brake_switch_ON/OFF
5	Master_cylinder_pressure
6	Calculated_road_gradient
7	Acceleration_speed_-_Lateral
8	Steering_wheel_speed
9	Steering_wheel_angle
10	Time (s)
11	Class
12	PathOrder

#### 4.5 Validation Using the MATLAB Time-Series Neural Network

The MATLAB time-series neural network was used to validate the dataset, as presented in Tab. 4. This ANN was chosen because of its dynamic nature; given behaviour is a constantly changing phenomenon, static techniques cannot provide accurate results. Furthermore, the



scaled conjugate gradient was chosen for accuracy; this stops automatically when an ANN stops improving following increases in validation values.

**Table 4:** Validation using MATLAB time-series neural network

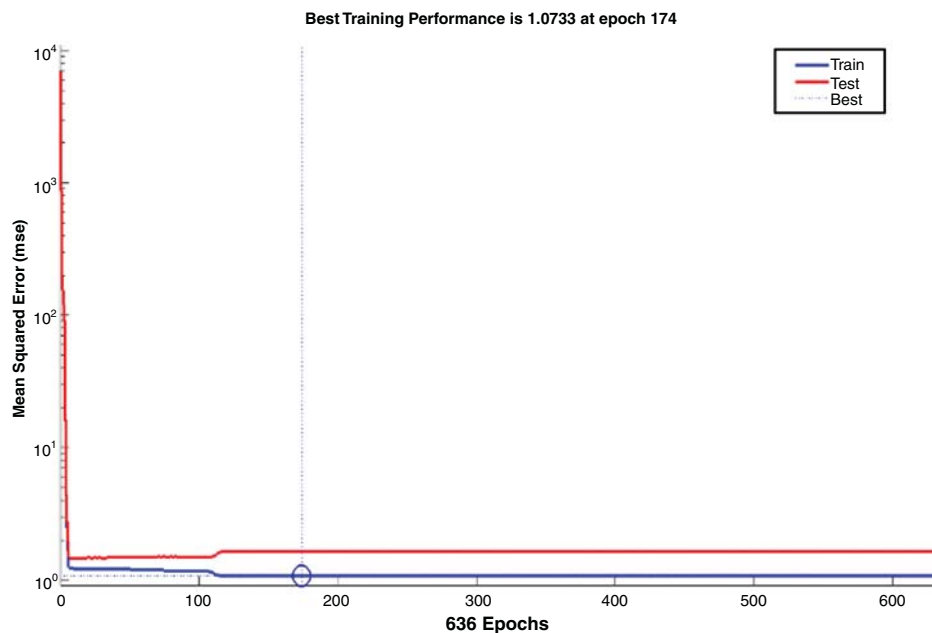
	Target values	Mean square error	R (regression values)
Training	361	1.07333	0.997687
Validation	78	0.00000	0.00000
Testing	78	1.64220	0.996836

#### 4.5.1 Artificial Neural Network Training

The ANN was trained using scaled conjugate gradient. Data division was random, with results presented for 636 epoch iterations. Results included performance, training state, error histogram, regression, time-series response and error autocorrelation.

#### 4.5.2 Training State

The training state graph shows results for epoch 636 and the validation checks at epoch 636. Gradient is on the y-axis, and epoch is on the x-axis. The first graph indicates that values begin at the peak and gradually decrease or minimise.



**Figure 2:** Validation performance: Mean square error decreases as the number of iterations increases

#### 4.5.3 Best Validation Performance

The best validation performance was calculated, as presented in Fig. 2, with the x-axis at epoch 174 and mean square error shown on the y-axis. The best validation performance was

calculated as the point at which the best line and the validation line intersected. Overall mean square error tended to decrease as the number of iterations increased.

#### 4.5.4 Error Histogram

The error histogram was computed with 20 bins. At the beginning of the iterations, the minimisation of error tended to increase gradually because training data values increased, and error was totally removed, as shown by the plain orange line. Following the zero error, the training process gradually minimised and totally stopped with only the training state remaining.

#### 4.5.5 Regression Plots

Different regression outputs are shown in Fig. 3, with graph 1 showing the training of data at regression value 0.99769, with the bunching of data indicating the fit line, graph 2 showing the test for fittest data regression values—that is, values nearest to target values—at 0.99684, and graph 3 showing the overall combined result for data flow and regression values at 0.99752; these results are highly refined and accurate.

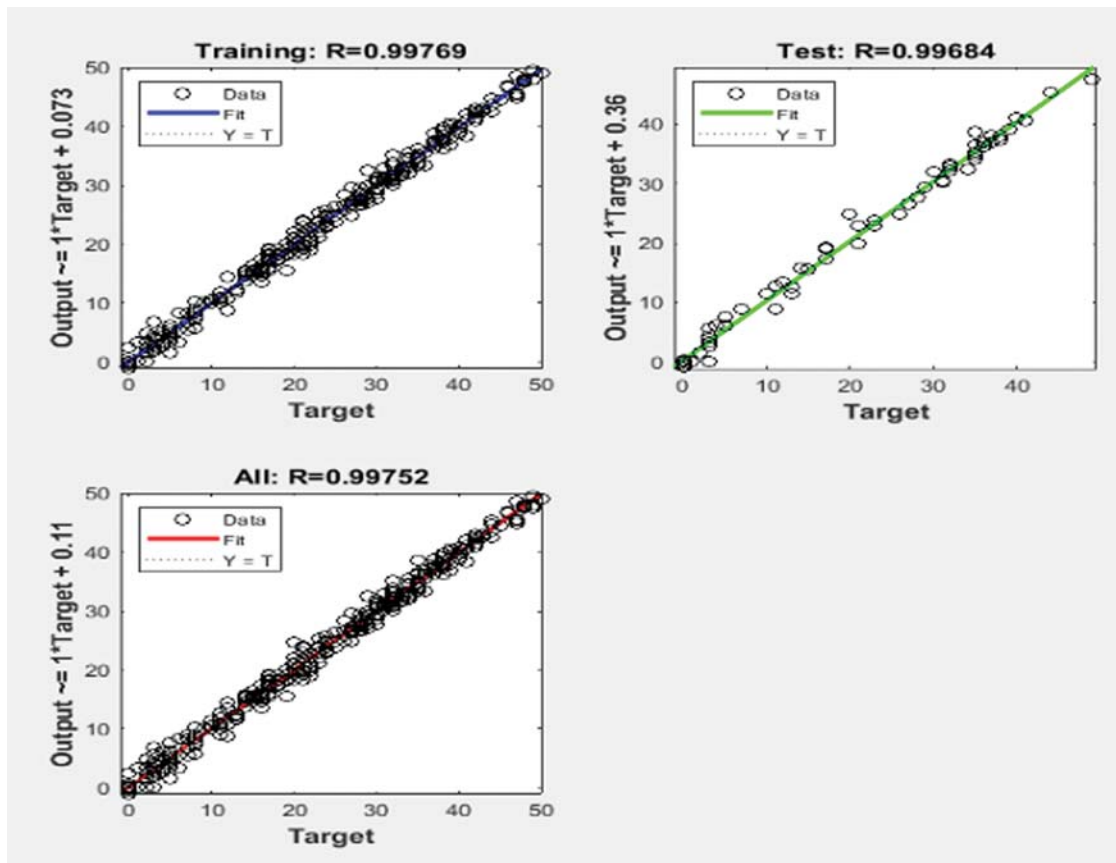
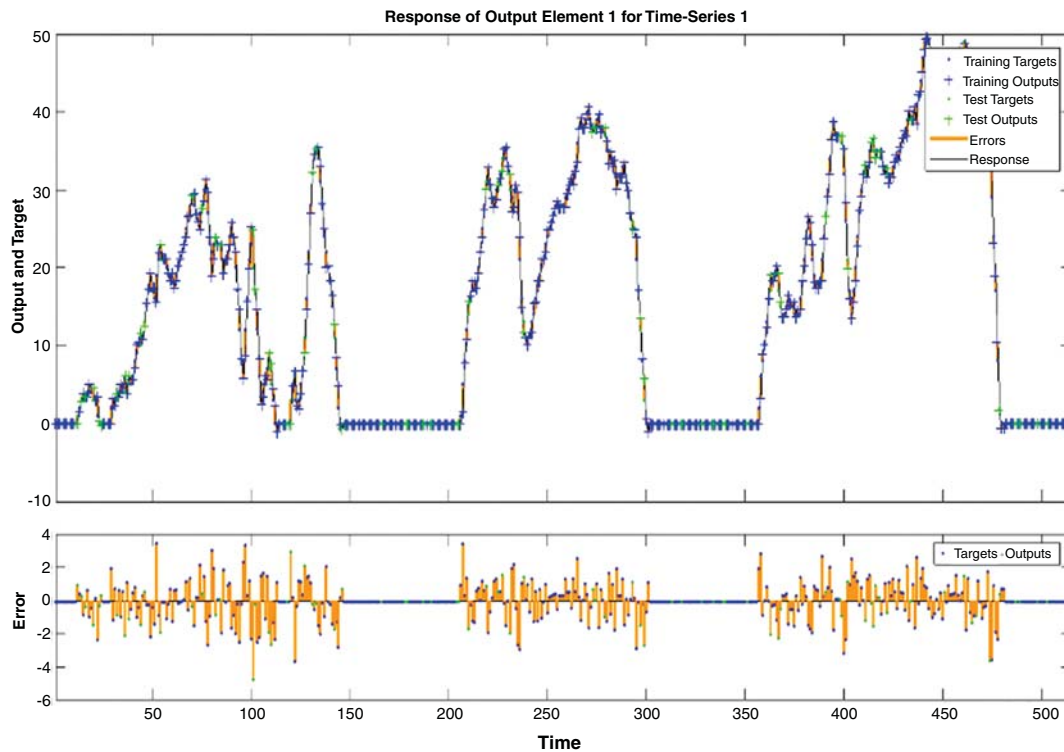


Figure 3: Regression plots

#### 4.5.6 Time-Series Response

Fig. 4 shows the time-series values sequentially by showing the time-series gap between target outputs and training outputs. Variations are indicated clearly for the first half of training; at the beginning of the second half, fluctuations tended to stabilise again.



**Figure 4:** Time-series response showing the time-series gap between target outputs and training outputs

### 4.6 Validation Using Weka Multilayer Perceptron

Weka is a tool for validating or training neural networks using different training algorithms. This research also used Weka's multilayer perceptron to train its data.

#### 4.6.1 Pre-Processing

The first step for training in Weka first step is pre-processing the dataset. This involves selecting all filters and attributes to direct classification and clustering.

#### 4.6.2 Classification

The second step is classifying the dataset. There were 54 attributes that could be trained for 31 sigmoid nodes. Results are given in the following sections.

#### 4.6.3 Classification Results

Tab. 5 presents the results for the classification of training data. The correlation coefficient range was near zero; the closer the result is to zero, the more accurate the ANN training.

Mean absolute error and root mean square error are the average means of the error values and range from 0 to 100. Relative absolute error and root relative square error range from 0 to 10.

**Table 5:** Validation using Weka multilayer perceptron

Validation parameters	Results
Correlation coefficient	0.9962
Mean absolute error	30.3982
Root mean squared error	69.4428
Relative absolute error	5.5875%
Root relative squared error	9.1207%
Total number of instances	94380

#### 4.6.4 Cluster

The third step is to cluster the given attributes. This can be conducted using a simple expectation maximisation class, which assigns probability distribution for each instance, indicating the probability of its belonging a different cluster and involves the following steps:

- 1) The number of clusters is set to one.
- 2) The training set is split randomly into ten folds.
- 3) Expectation maximisation is performed ten times using the ten folds in the usual CV manner.
- 4) The loglikelihood is averaged across the ten results.
- 5) If loglikelihood has increased, the number of clusters is increased by one, and the program continues at step 2.

During simulation, the number of clusters selected by cross-validation was nine, the number of iterations performed was three and the loglikelihood value was  $-47.07798$ . [Tabs. 6](#) and [7](#) present the cluster and cluster instance results.

**Table 6:** Cluster results

Attributes	Cluster
0	(0.17)
1	(0.02)
2	(0.01)
3	(0.12)
4	(0.23)
5	(0.06)
6	(0.1)
7	(0.05)
8	(0.06)

**Table 7:** Cluster instances

Attributes	Cluster	Percentage (%)
0	15062	16
1	1724	2
2	902	1
3	9543	10
4	31596	33
5	2445	3
6	3989	4
7	22693	24
8	6426	7

#### 4.7 Validation Using Weka Random Subspace

The Weka random subspace approach constructs a decision-tree-based classifier that maintains the highest accuracy for training data and improves generalisation accuracy as it grows in complexity. The classifier comprises multiple trees constructed systematically by pseudo-randomly selecting subsets of components of the feature vector that is tree constructed in randomly chosen subspace.

##### 4.7.1 Classification Results

Results are shown for ten folds and cross-validation techniques. The total classes processed are organised from A to J, with each class showing the individual output values of training iterations in a hierarchical form that indicates tree format.

```

REPTree
=====

Engine_coolant_temperature < 76.5
| Engine_coolant_temperature < 80.5
| | Engine_coolant_temperature < 62.5
| | | Engine_coolant_temperature < -15.5 : 1346.33 (2/4225) [1/2859481]
| | | Engine_coolant_temperature >= -15.5
| | | | Engine_coolant_temperature < 47.5
| | | | Short_Term_Fuel_Trim_Bank1 < -5.9
| | | | | Fuel_consumption < 441.6
| | | | | | Activation_of_Air_compressor < 0.5
| | | | | | Engine_coolant_temperature < 24 : 119.93 (9/340.02) [5/439.55]
| | | | | | Engine_coolant_temperature >= 24 : 76.85 (14/142.17) [6/120.38]
| | | | | | Activation_of_Air_compressor >= 0.5
| | | | | | Engine_coolant_temperature < 39.5 : 134.67 (33/331.73) [15/306.04]
| | | | | | Engine_coolant_temperature >= 39.5 : 189.38 (33/233.94) [20/307.7]
| | | | | Fuel_consumption >= 441.6
| | | | | | Wheel_velocity_rear_right-hand < 2.05
| | | | | | Fuel_consumption < 480
| | | | | | | Engine_soaking_time < 245 : 68.33 (52/277.84) [23/230.15]
| | | | | | | Engine_soaking_time >= 245
| | | | | | | | Engine_coolant_temperature < 35.5 : 80.95 (36/271.16) [25/650.64]
| | | | | | | | Engine_coolant_temperature >= 35.5 : 173.43 (4/1092.75) [3/2143.58]
| | | | | | | Fuel_consumption >= 480 : 42.87 (69/313.59) [39/303.22]
| | | | | | Wheel_velocity_rear_right-hand >= 2.05
| | | | | | Engine_coolant_temperature < 31
| | | | | | | Engine_coolant_temperature < 27.5

```

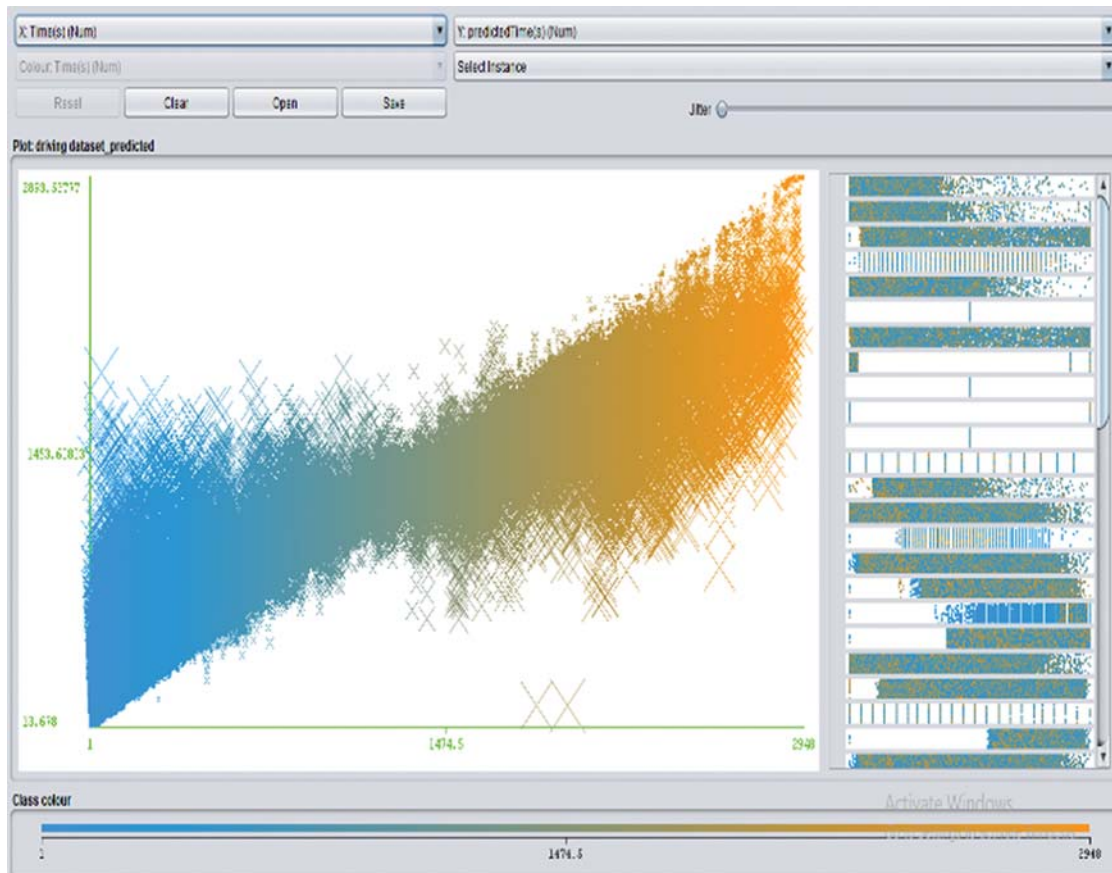
**Figure 5:** Individual parameter evaluations

The sequence of results for different parameters is shown in Fig. 5; each parameter is evaluated individually for more refined results.

Tab. 8 shows the classification of training data, with the correlation coefficient range being near zero; the closer the result is to zero, the more accurate the ANN training. Mean absolute error and root mean square error are the average means of the error values and range from 0 to 300. Relative absolute error and root relative square error range from 0 to 150.

**Table 8:** Validation results for Weka random subspace

Validation parameters	Results
Correlation coefficient	0.9261
Mean absolute error	194.895
Root mean squared error	287.5999
Relative absolute error	33.9581%
Root relative squared error	42.1055%
Total number of instances	94380



**Figure 6:** Visualisation comparing time and predicted time: Patterns show the predicted time at different iteration stages

#### 4.7.2 Visualisation

Fig. 6 presents a visualisation of the results in the form of time series. Time is given at the x-axis and predicted time is given at the y-axis. Patterns show the predicted time at different iteration stages.

#### 4.8 Validation Using Linear Regression Analysis

Tab. 9 shows the simulation results for linear regression analysis.

**Table 9:** Validation using linear regression analysis

Validation parameters	Results
Correlation coefficient	0.9855
Mean absolute error	64.6902
Root mean squared error	129.0136
Relative absolute error	11.8907%
Root relative squared error	16.9449%
Total Number of Instances	94380

#### 4.9 Validation Using Decision Tree

Tab. 10 shows the simulation results for the decision-tree approach.

**Table 10:** Validation using the decision-tree approach

Validation parameters	Results
Correlation coefficient	0.9884
Mean absolute error	62.5417
Root mean squared error	115.5902
Relative absolute error	11.4958%
Root relative squared error	15.1818%
Total number of instances	94380

**Table 11:** Comparison of results

Validation parameters	Multilayer perceptron	Random subspace	Linear regression	Decision tree
Correlation coefficient	0.9962	0.9261	0.9855	0.9884
Mean absolute error	30.3982	194.895	64.6902	62.5417
Root mean squared error	69.4428	287.5999	129.0136	115.5902
Relative absolute error	5.5875%	33.9581%	11.8907%	11.4958%
Root relative squared error	9.1207%	42.1055%	16.9449%	15.1818%

## 5 Discussion

Tab. 11 combines the results of all of the validation techniques used: multilayer perceptron, random subspace, linear regression analysis and decision tree. The differences between the results are indicated by comparing mean absolute errors, root mean square errors, relative absolute errors and root relative squared errors.

Tab. 11 clearly demonstrates that multilayer perceptron produces better results than the other validation techniques.

## 6 Conclusion

Intelligent agents can represent most human properties due to the similarity in cognitive processes, which enable the completion of deliberate, repetitive tasks ranging from routine to specific tasks. These processes are expressed through intelligent behaviour and actions performed in particular environments. Modelling intelligent driving behaviour is a complex task requiring consideration of many internal and external parameters. These parameters are activated simultaneously to transfer driving patterns from one situation to another, with these patterns constantly evolving to refine driving patterns.

This paper's 'driver behaviour model' can model intelligent driver behaviour in vehicular systems. According to this model, data from a dynamic environment is collected and refined through combination with the driver's profile and the route details. Refinements of behaviour require intersecting of the model's decision-making and assistance-system modules, which manage mechanisms internal to driver behaviour and behaviour in emergency situations. The external environment, a driver's past experience and their current route strongly impact output patterns.

To validate this model, driver datasets comprising multiple values related to different vehicular systems were evaluated, using Weka, by time-series ANNs using MATLAB backpropagation, multilayer perceptron, random subspace, linear regression and decision trees. Results produced means, regressions and correlations using classification and clustering techniques and indicated that the multilayer perceptron approach generates better results than other validation techniques.

**Funding Statement:** The authors received no specific funding for this study.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- [1] N. A. Ali and H. A. Zeid, "Driver behavior modeling: Developments and future directions," *International Journal of Vehicular Technology*, vol. 2016, pp. 1–14, 2016.
- [2] E. T. Solovey, M. Zec, E. A. Perez, B. Reimer and B. Mehler, "Classifying driver workload using physiological and driving performance data: Two field studies," in *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, France, pp. 4057–4066, 2014.
- [3] Y. Lei, K. Liu, Y. Fu, X. Li, Z. Liu *et al.*, "Research on driving style recognition method based on driver's dynamic demand," *Advances in Mechanical Engineering*, vol. 8, no. 9, pp. 1–14, 2016.
- [4] N. Banovic, T. Buzali, F. Chevalier, J. Mankoff and A. Dey, "Modeling and understanding human routine behavior," in *ACM Chi Conf. on Human Factors in Computing Systems*, Santa Clara, California, pp. 248–260, 2016.
- [5] M. A. Khan, M. Umair and M. A. Choudry, "Island differential evolution based adaptive receiver for mc-cdma system," in *IEEE Int. Conf. on Information and Communication Technologies*, Karachi, Pakistan, pp. 1–6, 2015.



- [6] M. Zubair, M. A. Choudhry and I. M. Qureshi, "Multiuser detection using soft particle swarm optimization along with radial basis function," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 22, no. 6, pp. 1476–1483, 2014.
- [7] M. Zubair, M. A. Choudhry and I. M. Qureshi, "Multi-user detection using soft particle swarm optimization for asynchronous mc-cdma," *International Information Institute Information*, vol. 16, no. 3, pp. 2093–2102, 2013.
- [8] S. B. Bezerra, M. I. Kaiser and G. R. A. Battistelle, "Road safety implications for sustainable development in Latin America," *Development*, vol. 18, no. 1, pp. 1–14, 2015.
- [9] G. Albertus, M. Meiring and H. C. Myburgh, "A review of intelligent driving style analysis systems and elated artificial intelligence algorithms," *Sensors*, vol. 29, pp. 1–15, 2015.
- [10] M. Asif, M. A. Khan, S. Abbas and M. Saleem, "Analysis of space & time complexity with pso based synchronous mc-cdma system," in *2nd IEEE Int. Conf. on Computing, Mathematics and Engineering Technologies*, Karachi, Pakistan, pp. 1–5, 2019.
- [11] M. Saleem, M. A. Khan, S. Abbas, M. Asif, M. Hassan *et al.*, "Intelligent Fso link for communication in natural disasters empowered with fuzzy inference system," in *IEEE Int. Conf. on Electrical, Communication, and Computer Engineering*, Karachi, Pakistan, pp. 1–6, 2019.
- [12] I. Mohammad, M. Ali and M. Ismail, "Abnormal driving detection using real time global positioning system data," in *IEEE Int. Conf. on Space Science and Communication*, Malaysia, pp. 1–6, 2011.
- [13] C. Chen, X. Zhao, Y. Yao, Y. Zhang, J. Rong *et al.*, "Driver's eco-driving behavior evaluation modeling based on driving events," *Journal of Advance Transportation*, vol. 12, pp. 1–15, 2018.
- [14] A. Aljaafreh, N. Alshabat and M. N. A. Din, "Driving style recognition using fuzzy logic," in *IEEE Int. Conf. on Vehicular Electronics and Safety*, Istanbul, Turkey, pp. 460–463, 2012.
- [15] J. H. Hong, B. Margines and A. K. Dey, "A smartphone-based sensing platform to model aggressive drivig behaviors," in *32nd Annual ACM Conf. on Human Factors in Computing Systems*, New York, NY, USA, pp. 4047–4056, 2014.
- [16] G. O. Thomas, W. Poortinga and E. S. Welsh, "Habit discontinuity, self-activation and the diminishing influence of context change: Evidence from the UK understanding society survey," *Plos One*, vol. 11, no. 4, pp. 1–16, 2016.
- [17] I. Pavlidisl, M. Dcostal, S. Taamnehl, M. Manser, T. Ferris *et al.*, "Dissecting driver behaviors under cognitive, emotional, sensorimotor, and mixed stressors," *Scientific Reports*, vol. 6, no. 12, pp. 25651–25663, 2016.
- [18] B. I. Kwak, J. Y. Woo and H. K. Kim, "Know your master: Driver profiling-based anti-theft method," in *4th IEEE Annual Conf. on Privacy, Security and Trust*, Izmir, Turkey, pp. 211–218, 2016.
- [19] A. Bouhoute, R. Oucheikh, I. Berrada and L. Omari, "A formal model of human driving behavior in vehicular networks," in *IEEE Int. Wireless Communications and Mobile Computing Conf.*, Nicosia, Cyprus, pp. 1–7, 2014.
- [20] X. Zhang, X. Zhao and J. Rong, "A study of individual characteristics of driving behavior based on hidden markov model," *Sensors and Transducers*, vol. 167, no. 3, pp. 194–205, 2014.
- [21] M. Benalla, B. Achchab and H. Hrimech, "Improving driver assistance in intelligent transportation systems: An agent-based evidential reasoning approach," *Journal of Advanced Transportation*, vol. 2020, pp. 1–14, 2020.