Traffic Lane Change Detection using mobile phone sensors and geolocation

Masoud Ghasemi[1], Amir Golroo[1*], Mohammad Javad Amani[1], Mehdi Rasti[2]

[1] Department of Civil and Environmental Engineering, Amirkabir University of Technology, Tehran, Iran.
[2] Department of Computer Engineering, Amirkabir University of Technology, Tehran, Iran.

* (Corresponding Author Email: agolroo@aut.ac.ir)

Abstract

Driver behavior is a critical factor in traffic safety. Detecting abnormal driver behaviors through appropriate indicators and enforcing driving regulations will reduce high-risk driving behaviors and increase traffic safety. Detecting dangerous driver behavior is beneficial for developing warning systems and preventing accidents. Some high-risk driving behaviors, such as sudden lane changes, are dependent on determining the movement direction of the vehicle which has not received enough attention. The objective of this study is to determine the direction of movement of the vehicle and lane changes, using the sensors in the smartphone mounted on a vehicle. To achieve this goal, first, by using the Samsung Galaxy S6 smartphone and an accurate Global Positioning System (GPS), longitudinal and angular accelerometer data and GPS data are sampled as a dataset and combined by different types of neural networks. Then, combined data is fed into a suggested neural network and lane changes are detected. Finally, the GPS data is used as the ground truth for the training of the neural network. If the GPS is not accessible, this neural network, just by receiving smartphone accelerometer data, can estimate the vehicle's direction of movement with an accuracy of 0.5 to 4.8 meters compared to the ground truth up to 8 seconds after the GPS is shut down. Using the vehicle travel path, an algorithm is proposed that can correctly detect the change of driving lane in the sample data set with 94% accuracy, 93.62% precision, 88.00% recall, and F1-score 90.72% which are acceptable values.

Keywords:

Driver Behavior, Lane Change Detection, smartphone, Transportation Safety, Artificial Intelligence

1. Introduction

Nowadays, transportation management is one of the most important concepts in every country. One of the most crucial criteria of transportation management is traffic safety. Traffic safety has received significant attention in transportation due to the increasing fatality rates in recent decades. Traffic safety refers to the methods used to prevent accidents or incidents in transportation networks. Traffic safety is influenced by many factors, such as driver behaviors, vehicles, and environmental conditions. Several studies have shown that more than 90% of accidents are due to mistakes made by road users or driver fatigue [1, 2]. Road users can include pedestrians, cyclists, drivers, vehicle passengers, and public transport passengers [3]. The behavior of road users, particularly drivers, is one of the most influential factors in traffic safety.

Driver behavior refers to the way a driver drives in real traffic conditions. Many research studies have been conducted to identify driver behaviors. Some studies only consider driving style to identify driver behavior, while others take into account weather and vehicle conditions [4, 5]. Most of these research studies could only identify one aggressive behavior, and only a few studies existed that could identify more than one abnormal behavior. To date, no investigation has been able to identify all driver behaviors [6].

To identify driver behavior, it is essential to understand the vehicle's position relative to other cars and obstacles. Depending on the direction of movement of the vehicle under study, driver behavior can be investigated under two categories: 1) transverse movement (lane change) and 2) longitudinal movement (sudden change of speed). These categories can be determined by analyzing the speed of the car, its acceleration, and steering angle [7]. One of the best methods that can be used to identify driver behaviors is lane change and vehicle trajectory. In addition to driver behavior, vehicle trajectory can be used to evaluate travel time, fuel consumption, etc. [8]. Also, it can be used to provide digital maps of roads [9]. Developing a system that assists researchers in detecting vehicle trajectory and direction can be used to identify aggressive behaviors such as lane changes and improve traffic safety. Several methods can be employed to identify lane change as a sign of driver behavior, including the Internet of Things (IoT).

The IoT is a network of machines, devices, and other things that are interconnected and able to communicate with each other without the need for human intervention [10]. The IoT acts as an interface between computers and other objects [11]. In such methods, input information and data are collected by automated systems and sent to a central server. The IoT can be applied in transportation engineering, especially in determining lane changes using different devices. The application of IoT in identifying driver behavior can be studied in three main fields: (1) understanding traffic conditions, (2) making appropriate decisions under perceived conditions, and (3) detecting distracting behaviors of drivers, such as drinking, eating, and speaking. In addition, the path of the vehicle can be combined with research that has been done on pavement distress detection to determine the location of these distresses [12].

2. Literature review

Automobile companies are constantly improving their vehicles to enhance driver safety. There are generally two groups of car safety systems: passive safety systems such as airbags and seatbelts, and active safety systems. The active safety system is designed to control the car in a safe manner and prevent accidents [13]. Li et al. (2017)[14] developed a Driver's Smart Advisory System (DSAS) to warn drivers near unsignalized intersections. The impact of the system's alarms on driver behavior, such as braking and speeding, was evaluated. The results showed that the system could improve driver awareness in unsignalized intersections. One of the most well-known active safety systems is the Advanced Driver Assistance System (ADAS), which can warn drivers and automatically intervene in handling cars if necessary [15]. These systems employ a specific series of sensors and have the capability to add more sensors for further assistance [16]. For instance, Julia Carrillo (2015), and Shaout et al. (2011) [17, 18] used laser and radar to capture data and applied image processing techniques to develop speed controller systems. One of the first systems designed to control driver behavior is the Adaptive Cruise Control (ACC) system, which allows drivers to adjust their speed and distance from the car in front of them without using the accelerator or brake pedal. Early versions of ACC systems were laser-based, but long-range radars are now used due to the shortcomings of lasers. More complex ACC systems use ultrasonic sensors for more precise data at low speeds. These systems are particularly applicable in heavy traffic [12, 19, 20].

Crossing protection systems and lane change assistant systems are designed based on road signs to reduce the risk of high-speed collisions, particularly on highways and roads. These systems use optical detection of road signs and markings, but their efficiency depends on weather conditions, and they perform weaker in rainy weather. Most cars now use multi-function mono cameras and multi-function stereo cameras. An advantage of the stereo camera is that it can recognize 3-D objects, lines, and obstacles [21, 22]. The more intelligent the driver assistance systems are and the less driver intervention is required, the closer they come to the concept of a fully automatic vehicle. Self-driving cars are classified into five levels of intelligence, where level five is fully automatic. Level one, also called driver assistance systems, controls only one part of the vehicle, such as cruise control, automatic braking, and lane-keeping systems. In such systems, the driver's reaction is still needed. However, instead of assisting the driver in making decisions, other systems analyze the driver's decisions, such as the intensity of the brakes or changing lanes, and detect any abnormal behavior. Galarza et al. (2018) designed a monitoring system to detect driver drowsiness and alert the driver to consciously control the vehicle. In this system, a smartphone was used to detect driver behaviors and properly interact with them [7]. Lee and Chung (2012) [23] proposed a method for monitoring driver safety levels using data fusion based on multiple discrete data sources. These data sources consisted of eye properties, the temperature inside the car, car speed, etc. Carmona et al. (2015) [24] developed a tool for analyzing driver behavior based on low-cost hardware and advanced fusion-based software capabilities. This device employed the information provided by in-vehicle sensors along with the unit of inertia measurement and Global Positioning System (GPS).

Yu et al. (2017) [25] divided the driver's abnormal behaviors into six categories: helical movements, rotation, slip, fast detour, long-range detour, and sudden braking. By capturing the required information over six months of real-world driving conditions, they concluded that each of the six abnormal behaviors listed had a unique pattern of acceleration and orientation. They proposed a system that deployed smart cell sensors to enable high-precision monitoring of driver abnormal behaviors. They used two machine learning algorithms, including support vector machines and neural networks, to train and detect drivers' abnormal behaviors.

Other systems evaluate drivers' distraction. Eating, drinking, speaking with others, and speaking on smartphone are common factors that distract drivers. Siuhi and Mwakalonge (2016) [26] reviewed the application of smartphones in improving traffic safety, including examples such as eliminating drivers' temptation to use smartphones, providing information to drivers, and blocking calls to prevent distractions. Fitch et al. (2014) [27] gathered data from 204 drivers by installing cameras and sensors in their cars and identified risky driving behaviors. Researchers concluded that using a smartphone while driving is a dangerous behavior that can lead to distraction due to multitasking [28]. They claimed that utilizing smartphones leads to significant changes in vehicle speed and challenges in maintaining the lateral position of the car [28, 29]. Moreover, using a smartphone while driving can cause drivers to reduce their speed and maintain a larger headway, which can be unsafe. In an attempt to compensate for the larger headway, drivers may increase their speed, resulting in reduced reaction times and potential danger [30]. Generally speaking, using a smartphone while driving can increase braking distance to three times [31].

The most common methods used to evaluate drivers' safety while using smartphones include driving simulators, real driving tests, and analyzing factors that affect accidents [27, 30]. Dingus et al.(2016) [32] utilized a dataset including 906 accidents and estimated that using a smartphone can double the risk of accidents.

It is significantly important to determine the exact position of a vehicle in order to identify driver behaviors. GPS can be used to approximate the vehicle's position, but in many cases, it encounters errors. Xiao et al. (2020) [8] proposed a systematic solution for car path detection. They combined the GPS data with internal diagnostic information including speed, steering angle, and vehicle acceleration to achieve higher positioning accuracy. When the GPS data was unavailable or inaccurate, the vehicle's trajectory was estimated using machine learning techniques. To calibrate the logic unit of this system, in places where the GPS was available, the location of the device was indicated using this unit and compared with the position shown by the GPS.

Havyarimana et al. (2018) [33] proposed a combined approach to solve the problem of predicting the position of the vehicle when the GPS was inaccessible, including GPS partial or complete inaccessibility in the short or long term. Ahmed et al. (2019) [34] developed a method for the fusion of the GPS and the unit of inertia data using the Kalman filter. In this method the calibration was continuous and when there was no access to GPS, the coefficients of the model were estimated using the unit of inertia. Driessen et al. (2022) [35] applied only the GPS data of a mobile devices including two smartphones and a GPS-equipped GoPro camera to recognize the changes in the car trajectory. They came up with a high accuracy of 90% in the lane-change classification. They claimed that the use of the method for highway engineering and traffic behavior research that employs floating car data appears promising. The occasional occurrence of false positives may limit the applicability of this method to real-time advisory systems.

To sum up, the diagnosis of abnormal behaviors of drivers based on the vehicle movement direction has not received enough attention. This study's novelty is to determine the vehicle's movement direction to be able to add vehicle location data to the driver's unusual behavior features, resulting in more accurate and precise driver behavior analysis. In other words, the research gap is to detect lane changes to distinguish dangerous driver behavior using vehicle trajectory and GPS data.

3. Objective and scope

The objective of this research is to detect vehicle movement direction using smartphones and machine learning techniques, specifically, vehicle lane changing. Smartphones are commonly used devices, and employing them as a low-cost data collection tool can benefit researchers. In this study, the convolutional and recurrent neural networks were employed as machine learning techniques to detect lane changing. These two models were chosen based on the type of data. The dataset provided in this study is a temporal dataset used to store time and time-interval information. Convolutional neural networks and recurrent neural networks are two of the best models that can be employed to analyze temporal datasets and extract patterns from them. Recurrent neural networks, in particular, are powerful tools for analyzing time series data.

4. Methodology

After a thorough literature review, suitable smartphone sensors for collecting data for the study's purpose were identified. Subsequently, a smartphone application was developed to record and transmit the data to a server. In the third step, the smartphone longitudinal accelerometer data was collected using a smartphone mounted on a vehicle. In the fourth step, the smartphone data was combined with GPS location data to eliminate noises from acquired data. In this step, a neural network-based model was designed and implemented The model could take in the accelerometer data and the GPS data's initial point to forecast the vehicle's next position. The model was tested and validated in the fifth step. In the sixth step, an algorithm was designed to detect lane changes. lastly, the neural network-based model was verified. Figure 1 illuminates the research methodology.

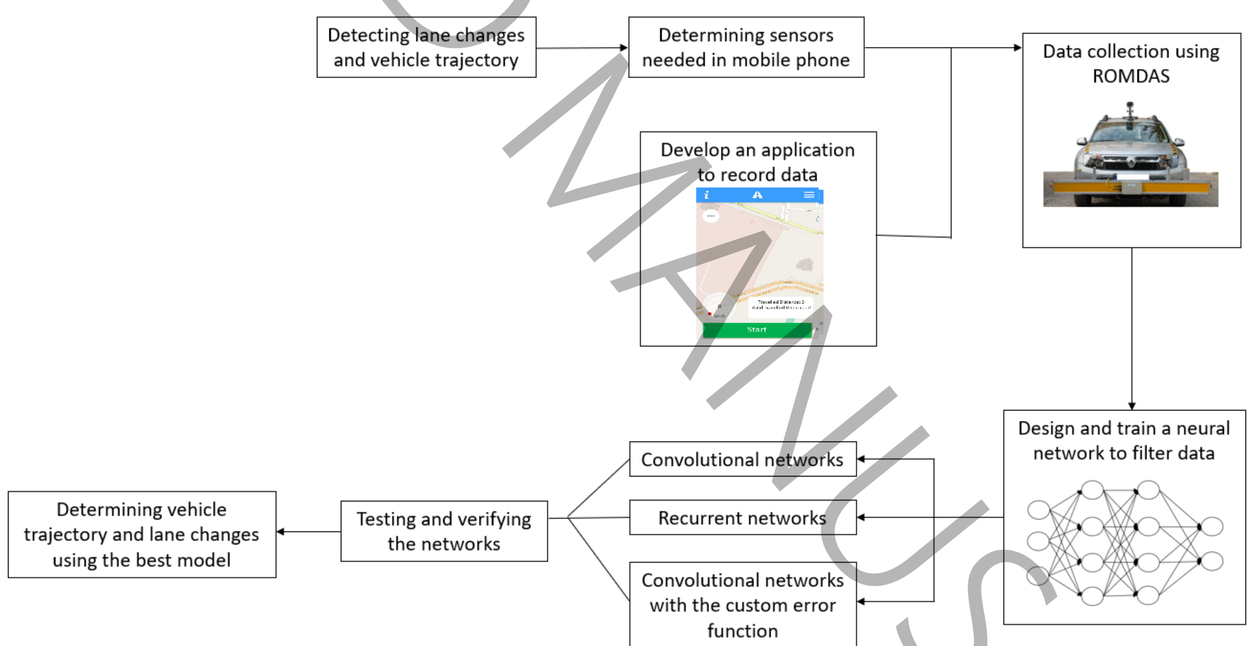
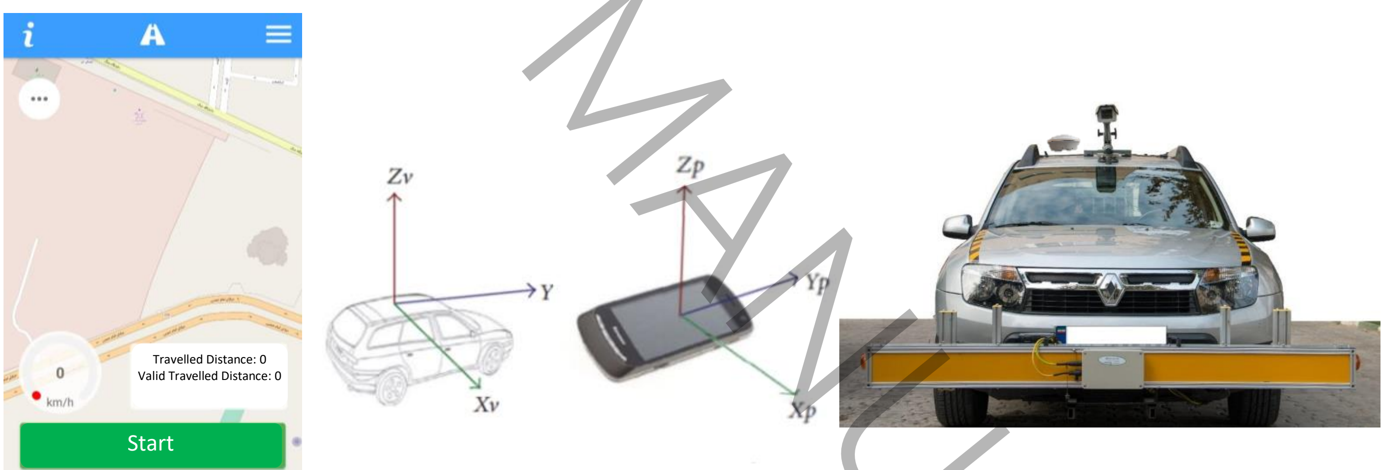


Figure 1- Research Methodology

5. Choosing sensors and data collection

In this study, for the sake of simplicity, the starting point was considered as the origin of the coordinate system. The geographic north direction was assumed as the positive direction of the y-axis, while the east direction was considered as the positive direction of the x-axis. This coordinate system is referred to as the reference coordinate system throughout the study. Among the various accelerations, the longitudinal acceleration of the smartphone in the y-direction and the angular acceleration around the z-axis were used to identify the vehicle's trajectory. The data recorded by the GPS and smartphone were labeled by time. The frequency of the smartphone accelerometer was 60 Hz, while that of the GPS mounted on a Road Surface Profiler (RSP) was 5 Hz. An application was developed to record the acceleration, gyroscope, and GPS data, as shown in Figure 2(a). This application received time from the GPS module with an accuracy of 1 ms. [36]. The application was installed on a Samsung Galaxy S6 smartphone, which was placed and fixed parallel to the ground on the RSP chassis (see Figure 2(b)). It was positioned directly below the GPS mounted on the RSP roof (see Figure 2(c)) to avoid interference with data collection by these two devices. Data collection took place for 2 hours in the morning, resulting in 180,000 records. After the initial preparation of the recorded data, specifically 90,000 records were employed.

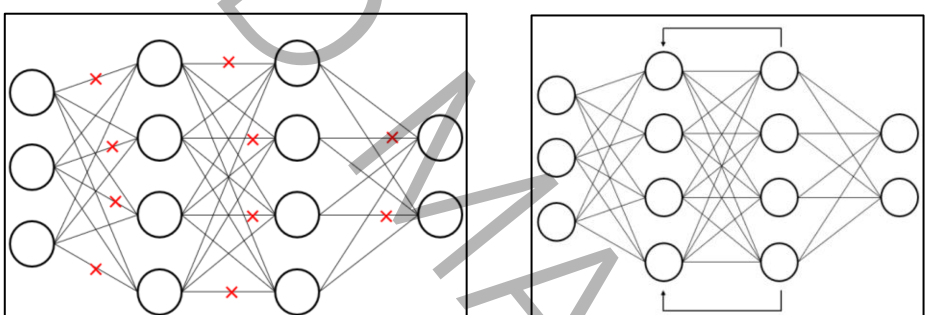


(a) (b) (c)

Figure 2- (a) the application developed to collect and send data to the server (b) smartphone and vehicle coordinate systems (c) RSP (the Road Surface Profiler)

6. Design, implementation, and training of the neural network

Neural networks have become popular recently and widely applied by researchers. Their applications in transportation engineering cover a wide range of subjects. Examples of areas where neural networks have been deployed include pavement distress detection, travel time estimation, trip mode choice prediction, etc. [37, 38]. Various types of neural networks with different architectures were designed and implemented to combine the GPS data and smartphone longitudinal and angular acceleration. In this study, two types of convolutional and recurrent neural networks were employed to filter the accelerations received from the smartphone and aggregate them. The convolutional neural network was trained in two different ways. Therefore, a total of three different neural networks were trained on the acquired data. Figure 3 illustrates the structure of these networks.



(a) (b)

Figure 3- (a) convolutional neural networks schematic (b) recurrent neural networks schematic

First, the optimal number of iterations (epochs) was determined for each type of network based on the changes in network error on the training and validation datasets. Subsequently, ten similar networks were separately trained with this number of iterations, and the average error of these ten networks was reported as the error of the associated architecture. The optimal architecture for each of the three proposed networks was obtained through a trial-and-error process. The Adam optimizer was used with a learning rate of0.001. After selecting the depth, width, and type of network layers through trial and error, the test data set was fed to the network to obtain model performance metrics along with the estimated acceleration values.. Using the estimated accelerations and the start point of movement, the maximum time duration for which the vehicle position could be estimated without GPS data was approximated. Additionally, the decrease in accuracy was assessed in case of prolonged inaccessibility to GPS data.

In the convolution neural networks, the velocity recorded by the GPS was used to calculate the average acceleration between two consecutive data points. Also, by calculating the changes in car direction in the reference coordinate system, the average angular acceleration between two consecutive GPS data points was determined. Using these two types of acceleration, two separate neural networks were trained to estimate longitudinal and angular accelerations. These networks utilized a combination of convolutional layers and average pooling layers.. The convolutional layer in convolutional neural networks serves as the core of the network. It takes an input such as a vector of numbers and applies a convolution filter to produce outputs. In a one-dimensional convolutional layer (Conv1D), the input is a single temporal or spatial dimension. The pooling layers reduce the size of the convolutional layers' output, which is called the feature map, by taking the average over its values. In these networks, an activation function is required to activate neurons. Various activation functions are available, but in this study, ReLu and tanh were used as two common activation functions in the convolutional and recurrent neural networks. Figure 4 and 5 depict the structure of one of the convolution networks.

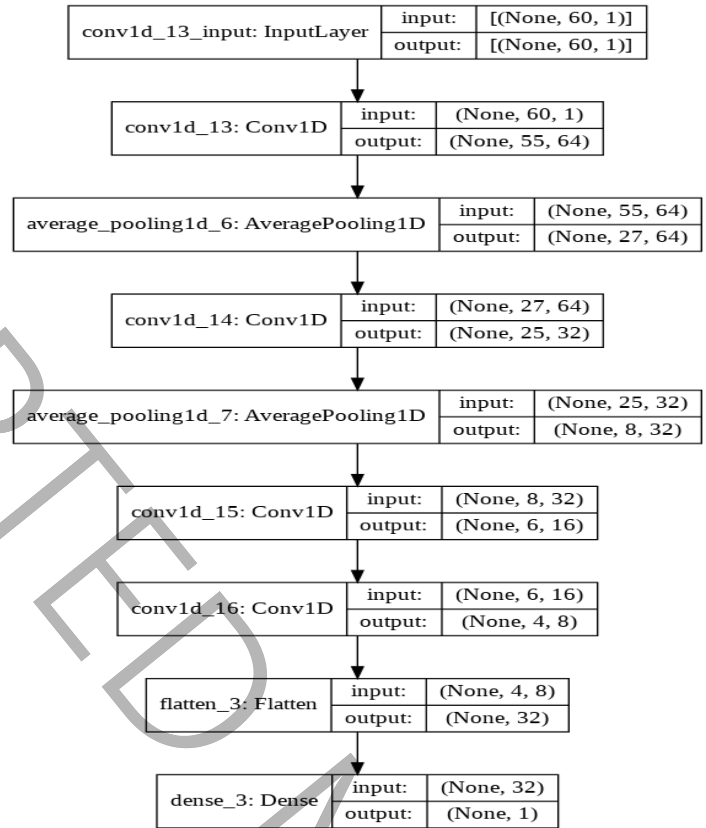


Figure 4- A sample of a convolutional neural network used to estimate longitudinal and angular accelerations

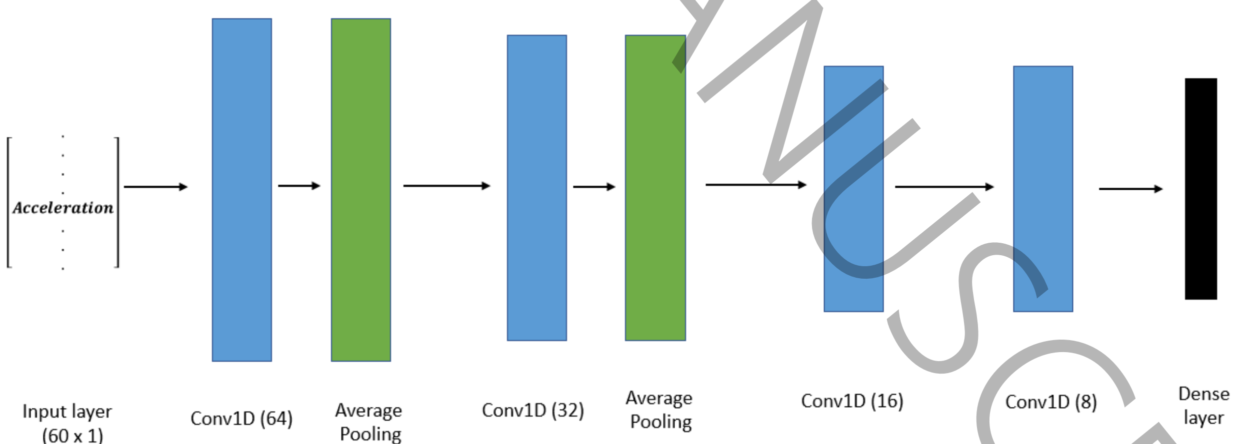


Fig. 5- A convolutional neural network (network number five in Table 1)

Mean squared error was utilized as the error function. The network inputs were fed into the model as an array of length 60. Samples of the smartphone acceleration, which contained noise and were sampled within one second, were then input to the desired network. The network produced an aggregated and denoised acceleration for that second.

The second architecture applied in this study is the recurrent neural network. In this architecture, Similar to the previous architecture,, two separate networks were used to determine longitudinal and angular accelerations. However, the type of layers employed in this network differed from the previous network. Long Short-Term Memory (LSTM) units were used in this architecture. In LSTM units, the outputs of some neurons can affect the input of others neurons. Recurrent neural networks (RNN) are suitable for sequential and time series data. The same as the convolutional neural network, mean squared error was applied as the error function. The input of the network was set as an array of length 60 whose elements were noisy and unfiltered accelerations and the output was a filtered acceleration value. Figures 6 and 7 illuminate the structure of the recurrent neural network.

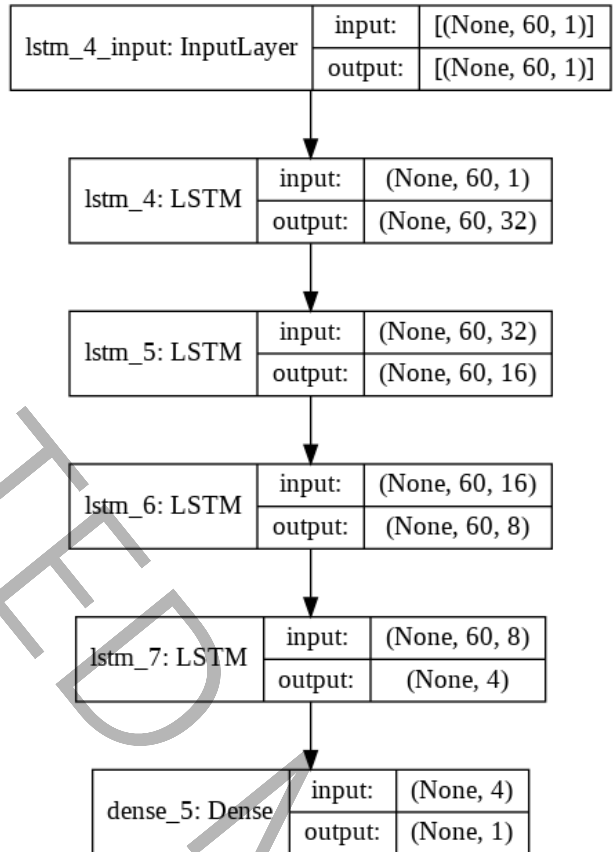


Figure 6- A sample of a recurrent neural network used to estimate longitudinal and angular accelerations

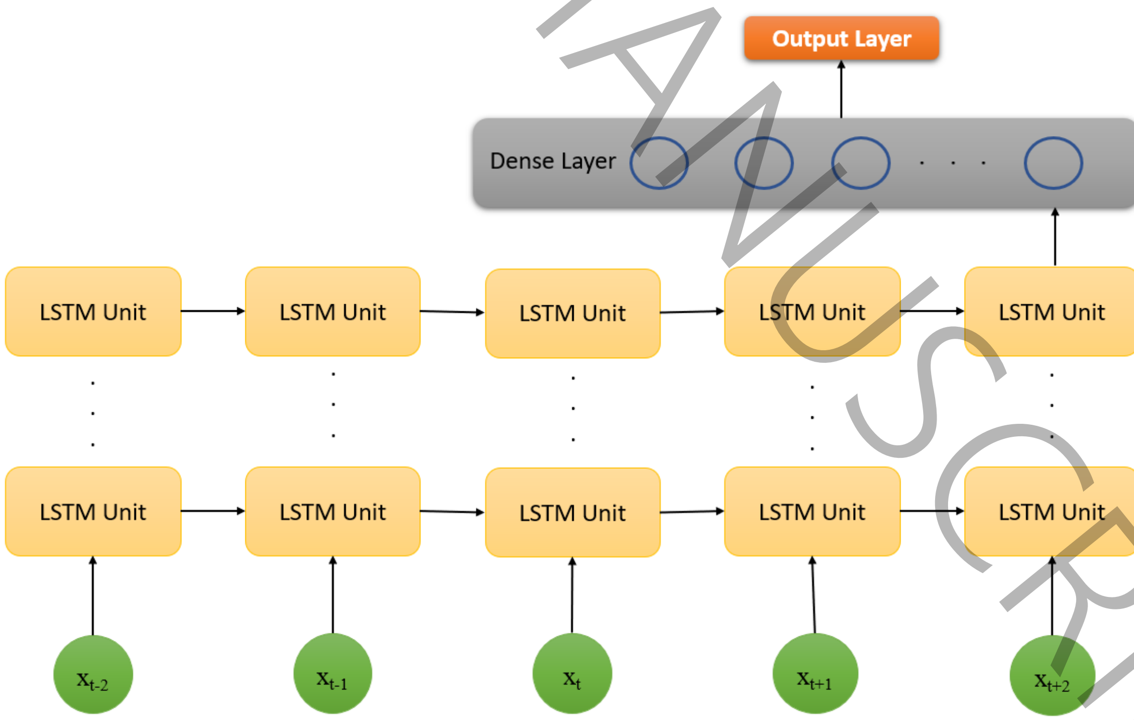


Figure 7- A recurrent neural networks

The accuracy of the GPS is about 1 meter. Therefore, if the model can estimate the vehicle positions within a one-meter radius of the position reported by the GPS, its performance will be acceptable. This idea was used in the third architecture. In the third architecture instead of training two separate networks to estimate each acceleration, a single convolutional neural network was used. The input in this architecture was a 2×60 matrix of accelerations and the output was a vector in which each of its arrays was a denoised angular and longitudinal acceleration. By receiving filtered and aggregated accelerations through the network and the start point of movement, the next point of the route was calculated. The error function of this network was defined in such a way that if the distance between the calculated position and the position recorded by the GPS is less than 1 meter (i.e., the GPS accuracy), then it would report a small error and otherwise it would report a significant amount of error. Equation 1 shows the error function.

$$f(x)=\begin{cases} C & ,\sqrt{\left(x_p-x_a\right)^2+\left(y_p-x_a\right)^2}\le 1 \\ \sqrt{\left(x_p-x_a\right)^2+\left(y_p-x_a\right)^2} & ,\sqrt{\left(x_p-x_a\right)^2+\left(y_p-x_a\right)^2}>1 \end{cases} \tag{1}$$

In this equation, x_p, x_a, y_p, y_a, and C represent the predicted length in the reference coordinate system, the actual length, the predicted width, the actual width, and the constant value, respectively.

7. Algorithms and model verification

7-1- Lane change detection algorithm

The outputs of the three models were longitudinal and angular acceleration per second. Angular acceleration per second indicated how much the car deviated from its previous direction of movement. Therefore, using Equation 2, the amount of transverse displacement of the vehicle could be calculated.

$$d_L=\frac{2V+a}{2}\times\sin\left(\theta\right) \tag{2}$$

In this equation, d_L, V, a, and θ are transverse displacements relative to the initial position of the vehicle, initial vehicle velocity, and one-second longitudinal and angular accelerations of the model output, respectively.

In the same way, the amount of transverse displacement at different times relative to the initial time was calculated. Relative transverse displacement was calculated in the next 5 seconds. The reason for selecting this time interval is that it is assumed to be the proper amount of time to complete a lane change. The transverse displacement would be more or less than a lane width (i.e., 3.6 m) which resulted in the following two conditions:

If the transverse displacement was less than 3.6 meters in 5 seconds, the vehicle was considered to not have changed lanes. If the transverse displacement was more than 3.6 meters within 5 seconds starting at time tn, the vehicle was considered to have changed lanes.

7-2- Verification and Validation

The verification of the selected model for filtering and combining smartphone accelerometer and GPS data was carried out in two steps. In the first step, the estimated accelerations of the model were checked to ensure they were within the expected acceleration range for a vehicle, i.e., they should not represent large positive or negative accelerations. In the second step, the estimated vehicle path was compared with the real path from which the data was collected to ensure consistency. To validate the model, the model outputs were compared with the results of the GPS. This means that outputs with equal initial conditions should be similar to each other. Additionally, the model was tested by a test set (unseen) that was not a part of the training set.

8. Results and discussion

Neural networks employed in this study with different architectures performed differently. Each network described in the previous section was tested. In this section, the model test results were presented. Using these results, the networks were verified and validated. Also, the results of the lane change detection algorithm were analyzed.

8-1- Convolutional neural networks

To select an optimum architecture, networks with different architectures were developed to estimate two types of acceleration.

8-1-1- Determination and verification of longitudinal acceleration filter convolution network architecture

To select the best convolutional neural network for estimating accelerations and predicting vehicle positions, six different architectures were developed and compared. These architectures are described in Table 1.

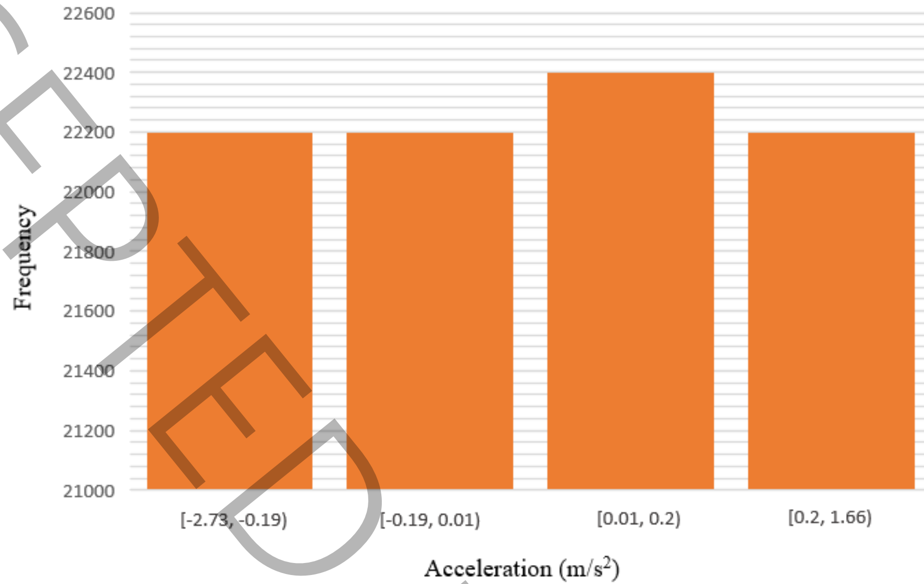
Table 1- proposed architectures for convolutional neural networks

Architecture #	Layer type	Layer width	Layer depth	strider	Activation function	Optimum architecture
1	Conv1D	128	6	1	ReLu	No
	Conv1D	64	6	1	ReLu	
	Conv1D	32	6	1	ReLu	
	Conv1D	16	6	1	ReLu	
	Conv1D	8	6	1	ReLu	
2	Conv1D	64	6	1	ReLu	No
	Conv1D	32	6	1	ReLu	
	Conv1D	16	6	1	ReLu	
	Conv1D	8	6	1	ReLu	
3	Conv1D	64	10	1	ReLu	No
	Conv1D	32	10	1	ReLu	
	Conv1D	16	10	1	ReLu	
	Conv1D	8	10	1	ReLu	
4	Conv1D	64	3	1	ReLu	No
	Conv1D	32	3	1	ReLu	
	Conv1D	16	3	1	ReLu	
	Conv1D	8	3	1	ReLu	
5	Conv1D	64	3	1	ReLu	Yes
	Average Pooling1D	-	2	2	ReLu	

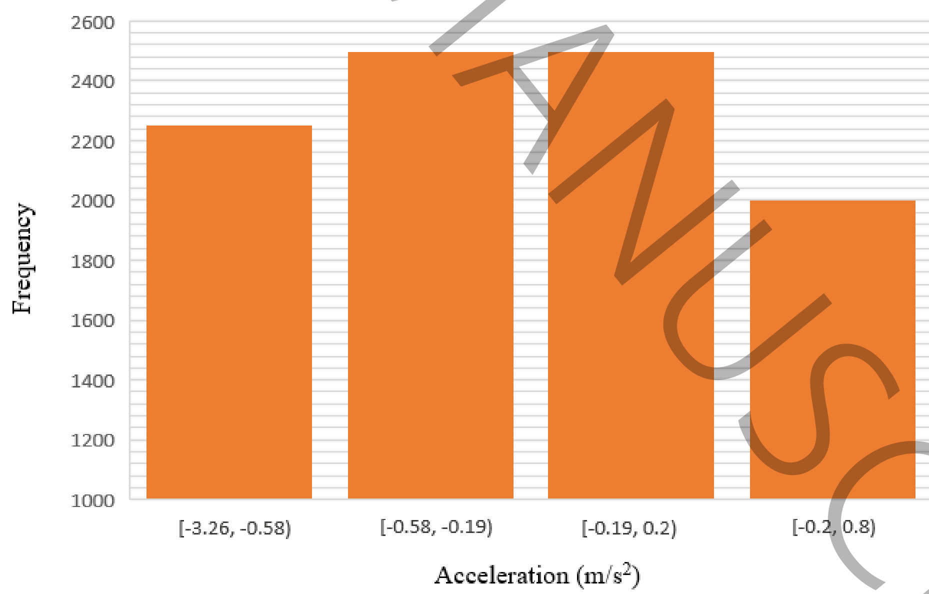
	Conv1D	32	3	1	ReLu	
	Average Pooling1D	-	3	3	ReLu	
	Conv1D	16	3	1	ReLu	
	Conv1D	8	3	1	ReLu	
6	Conv1D	64	3	1	ReLu	No
	Average Pooling1D	-	2	2	ReLu	
	Conv1D	32	3	1	ReLu	
	Average Pooling1D	-	3	3	ReLu	
	Conv1D	16	3	1	ReLu	
	Conv1D	8	3	1	ReLu	
	Average Pooling1D	-	2	2	ReLu	

By comparing different architectures expressed in Table 1, it was concluded that the less the number of one-dimensional convolution layers, the more generalized the architecture is. Different window sizes were tested for the filter dimensions, including 10 for a large filter layer, 6 for a medium filter layer, and 3 for a small filter layer. After selecting the number of one-dimensional convolution layers and the filter dimension, the number of average pooling layers was examined. Architecture five consisted of two layers of average pooling, while architecture six has three layers of average pooling. To ensure that the model achieves good generalization, it is crucial to strike a balance between its performance on the training set and its performance on the test and validation sets. If the model performs well on the training set but poorly on the test and validation sets, this indicates overfitting and a loss of generalization.

After reviewing all the cases, architecture five, with two layers of average pooling, four layers of one-dimensional convolution, and a filter window size of three, had the best performance on the test and validation data set. Architecture five demonstrated better generalization capabilities than architecture six, meaning that it was less prone to overfitting and was better able to apply what it learned from the training data to new data. The selected architecture on the training dataset had a mean squared error equal to 0.0022. If the square root of the reported error was computed, the value of 0.043 m/s^2 was obtained as the mean error of the longitudinal acceleration estimate. Figure 8 illustrates the distribution of longitudinal acceleration estimates generated by the selected network on both the training and test datasets. The similar distribution on both sets suggests that the network was able to generalize well to new data.



(a)



(b)

Fig. 8- (a) distribution of longitudinal accelerations estimated by convolutional neural network (b) distribution of longitudinal accelerations estimated by recurrent neural network

The distribution of longitudinal acceleration estimates shown in Figure 8 indicates that all values fell within the range of [-3.88, 2.87] m/s^2. According to the literature review, this is an acceptable interval for vehicle acceleration [39]. Also, the selected architecture in the previous section on the test data set had a mean squared error equal to 0.04. All acceleration estimates in Figure 8 fell within the acceptable range of [-3.88, 2.87] m/s^2.

To determine the optimal convolutional neural network for the angular acceleration filter, a similar procedure was followed as for the longitudinal acceleration filter. Six networks with different architectures were developed based on the design principles outlined in Table 1, and mean squared error values were computed for each network. Architecture five was identified as the best option, having achieved the lowest mean squared error on the test dataset and exhibiting balanced performance between the training and test datasets, indicating appropriate generalization. To evaluate a model, it is necessary to assess its performance on the training set by determining the difference between the estimated values by neural networks and their real values. This difference can be determined by an error function such as the mean squared error function. The selected network on the training data set had an average squared error equal to 0.026 , resulting in an average error of 0.16 degrees per second for estimating angular acceleration. Also, to validate a neural network, the performance of the model on the test set was examined. The selected network on the test dataset had a mean squared error equal to 0.47. The value of 0.385 degrees per second was obtained as the average error of estimating the angular acceleration.

8-1-2- Hybrid model using the convolutional networks

8-1-2-1- evaluate the model in estimating the vehicle trajectory

After selecting appropriate architectures for the convolutional neural networks, longitudinal and angular accelerations were provided as inputs to the networks, and the vehicle trajectory was estimated using the model's outputs. To reconstruct the vehicle's path, the first one-second interval was evaluated. In other words, at the beginning of each second, the initial position of the movement was determined by GPS, and then the endpoint of movement after one second was estimated by the model. In the following steps, the duration of unavailability of GPS data was increased. In this condition, using the initial point of movement, the endpoint of movement after one second was estimated, just as before. This point was set as the initial point for the next second. The accuracy of the smartphone GPS, in the case of normal access to satellites, is about five meters. If the access to GPS for three consecutive seconds was limited, the proposed hybrid model could maintain an accuracy of fewer than 5 meters in 75% of cases and had an average accuracy of about three meters. If the access was limited for six consecutive seconds, the model would have an accuracy of fewer than five meters with a probability of 50%.

8-1-2-2- evaluate the networks in estimating longitudinal and angular accelerations

The neural network accuracy in estimating longitudinal acceleration even after 10 seconds at worst was limited to 1.38 m/s^2. This network had an accuracy of 0.34 m/s^2 with a probability of 75% after 10 seconds. On the other hand, the network reports an angular acceleration that was estimated after 5 seconds with an accuracy of about 4.8 degrees per second and an average accuracy of about 9 degrees with a probability of 50%. Comparing the performance of the two networks, it can be concluded that the weakness of the model was more due to the performance of the convolution neural network in estimating angular accelerations.

8-2- Recurrent neural networks

The optimum architecture for recurrent neural networks was selected for filtering and estimating both angular and longitudinal accelerations. The sum of squared error was then calculated for each architecture. Subsequently, a model was developed using recurrent neural networks, taking into account the time of inaccessibility to GPS data, to evaluate its accuracy in estimating the vehicle trajectory. The performance of each network was evaluated separately.

8-2-1- Determining the recurrent filter network architecture for longitudinal acceleration

To determine the optimum architecture for recurrent neural networks, three different networks with different numbers of layers and depths were developed. The different architectures are shown in Table 2.

Table 2- proposed architectures for recurrent neural networks

Number of architecture	Layer type	Layer depth	Optimum architecture
1	LSTM	16	No
	LSTM	8	
	LSTM	4	
2	LSTM	32	Yes
	LSTM	16	
	LSTM	8	
	LSTM	4	
3	LSTM	64	No
	LSTM	32	
	LSTM	16	
	LSTM	8	
	LSTM	4	

The four-layer deep network showed better generalization capability compared to its shallower competitors. Increasing the network depth from four to five improved the performance of the model on training and validation datasets; However, it led to poorer performance on test data. Therefore, an architecture with a depth of four was selected in this section.

The recurrent neural network was verified and validated following a similar procedure as the convolutional networks. The selected network demonstrated an error of 0.12 m/s^2 on the training dataset and 0.18 m/s^2 on the test dataset. Additionally, it was ensured that all estimated acceleration values across all datasets were within the acceptable range of [-3.88, 2.87] m/s^2 [39].

8-2-2- Determining the recurrent filter network architecture for angular acceleration:

To determine the optimum architecture for estimating angular acceleration, three different networks were developed, similar to the recurrent filter network for longitudinal acceleration. Architectures with depths of three to five layers were evaluated for filtering and estimating angular and longitudinal accelerations. The four-layer depth architecture had the lowest error on the test and validation datasets. Although the five-layer depth architecture had less error on the training dataset, it showed larger errors on the validation and test datasets compared to the four-layer depth neural network, indicating lower generalizability. Therefore, the four-layer depth network was selected as the optimal architecture. Similar to the hybrid model built using convolutional neural networks, by combining recurrent networks, a model was developed to estimate vehicle trajectory. This model greatly reduced the model's dependence on GPS data, by up to 25%. The summary of the performance of the convolutional and recurrent neural networks in filtering and estimating angular and longitudinal accelerations are shown in Table 3.

Table 3- the summary of neural networks results evaluation

network	Acceleration	Mean squared error value	Acceptable range (m/s^{2})
Convolutional neural networks	Longitudinal	0.0022	[-3.88, 2.87]
	Angular	0.026	
Recurrent neural networks	Longitudinal	0.12	
	Angular	0.8	

8-3- Changing the method of training model

To train models described in previous sections, the longitudinal and angular accelerations were first calculated. Using these accelerations, two separate networks were developed, with the sum of squared error serving as the error function. In the third model, a new error function was used as explained in section 6. Among the three proposed models, the first model with five convolutional layers and one average pooling filter layer had the best performance after eight seconds of inaccessibility to the global positioning system compared to the other models. Finally, this model with its new error function was chosen to detect the lane changes. All three models are presented in Table 4.

Table 4- different architectures for convolutional neural networks with the proposed error function

Architecture #	Layer type	Layer width	Layer depth	strider	Activation function	Optimum architecture
1	Conv1D	64	3	1	ReLu	Yes
	Conv1D	32	3	1	ReLu	
	Average Pooling1D	-	8	8	linear	
	Conv1D	16	3	1	ReLu	
	Conv1D	8	3	1	ReLu	
	Conv1D	4	3	1	ReLu	
2	Conv1D	64	3	1	tanh	No
	Conv1D	32	3	1	tanh	
	Average Pooling1D	-	8	8	linear	
	Conv1D	16	3	1	tanh	
	Conv1D	8	3	1	tanh	
	Conv1D	4	6	1	tanh	
3	Conv1D	64	3	1	tanh	No
	Average Pooling1D	32	3	1	tanh	
	Conv1D	-	8	8	linear	
	Average Pooling1D	16	3	1	tanh	
	Conv1D	8	3	1	tanh	

8-4- Evaluate the performance of the lane change detection algorithm

During the data collection period, the RSP had 53 lane changes. The confusion matrix can present the performance of a model in accurate prediction of model input. It clearly shows the number of input cases which is predicted either right or wrong by the model. The confusion matrix of this algorithm in the detection of changing the passing lane is shown in Table 5.

Table 5- the confusion matrix of the vehicle lane change detection by the proposed algorithm

		Lane changes in recorded by an expert	
		Change	No change
Algorithm output	Change	44	3
	No change	6	0

It is observed that the proposed algorithm in the path detected 47 lane changes, of which 44 lane changes were correct and 3 were incorrect. In other words, in 3 cases, the vehicle did not change the lane and the algorithm incorrectly detected the lane change, while in 6 cases, it actually changed the lane but the model predicted that there was no lane change. There are different methods to present the results of the model evaluation, such as accuracy, precision, recall, and F1-score, which can be calculated using equations 1 to 4, respectively

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{6}$$

In these equations, TP, TN, FP, and FN refer to True Positive, True Negative, False Positive, and False Negative, respectively. They represent the cases where lane changes were detected correctly, no lane changes were detected correctly, lane changes were detected wrongly, and no lane changes were detected wrongly, respectively. Therefore, it can be said that the model accuracy is equal to 94% which means 94% of the lane changes detected by this algorithm were also correct. To validate the model performance accuracy, two approaches can be applied (1) using test (unseen) data to validate the accuracy of the model which has been carried out in this research (2) comparing with other research results in the same condition with the same objective and scope which have been rarely available in the related literature (Driessen et al., 2022). Furthermore, according to Table 1, the selected model has precision equal to 93.62% and recall equal to 88.00%. The F1-score in this study is equal to 90.72%. It should be noted that the high value of precision compared to other metrics shows that the number of false positives is low in the model meaning that the model does not predict something correct which is not actually right. However, the recall seems to be less than accuracy (by almost 5%) i.e., the model expresses false negative more than false positive. This result means that the model misses some correct patterns. In this study, having a higher precision level is more important than recall to ensure that the data which was applied in the modeling process is correctly annotated and employed. It should be noted that the confusion matrix is obtained from the test dataset. It shows that the model's performance accuracy on a dataset other than the training dataset is valid and acceptable.

9. Conclusion

The objective of this study was to develop a model to detect lane changes using a combination of longitudinal and angular accelerometer sensor data. This model could be used to reduce the reliance on GPS data in places where satellite access is limited such as city centers. It also saves on mobile power consumption and increases the battery life of the device by using fewer global positioning system sensors. The proposed model in this study consisted of convolutional layers. The error function of this model was applied to correct the weights of the model according to the distance of the estimated position from the actual position of the vehicle path. An algorithm was also proposed that could use model outputs to detect lane change. This algorithm makes the necessary decisions based on vehicle lateral displacements. The developed model could successfully predict the lane change with precision, recall, and F1-score equal to 93.62%, 88.00%, and 90.72%, respectively.

10. References

[1] A. Bener, E. Yildirim, T. Özkan, T. Lajunen, Driver sleepiness, fatigue, careless behavior and risk of motor vehicle crash and injury: Population based case and control study, Journal of Traffic and Transportation engineering (English edition), 4(5) (2017) 496-502.
[2] S. Reinhold, C. Laesser, D. Bazzi, The intellectual structure of transportation management research: A review of the literature, in, University of St. Gallen St. Gallen, 2015.
[3] R.D. Guide, American Association of State Highway and Transportation Officials, Washington, DC, (1996).
[4] S. Al-Sultan, A.H. Al-Bayatti, H. Zedan, Context-aware driver behavior detection system in intelligent transportation systems, IEEE transactions on vehicular technology, 62(9) (2013) 4264-4275.
[5] E.E. Galarza, F.D. Egas, F.M. Silva, P.M. Velasco, E.D. Galarza, Real time driver drowsiness detection based on driver's face image behavior using a system of human computer interaction implemented in a smartphone, in: Proceedings of the International Conference on Information Technology & Systems (ICITS 2018), Springer, 2018, pp. 563-572.
[6] M. Shahverdy, M. Fathy, R. Berangi, M. Sabokrou, Driver behavior detection and classification using deep convolutional neural networks, Expert Systems with Applications, 149 (2020) 113240.
[7] J. Wang, K. Li, X.-Y. Lu, Effect of human factors on driver behavior, in: Advances in Intelligent Vehicles, Elsevier, 2014, pp. 111-157.
[8] Z. Xiao, F. Li, R. Wu, H. Jiang, Y. Hu, J. Ren, C. Cai, A. Iyengar, TrajData: On vehicle trajectory collection with commodity plug-and-play OBU devices, IEEE Internet of Things Journal, 7(9) (2020) 9066-9079.
[9] X. Liu, J. Biagioni, J. Eriksson, Y. Wang, G. Forman, Y. Zhu, Mining large-scale, sparse GPS traces for map inference: comparison of approaches, in: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, 2012, pp. 669-677.
[10] S. Tanwar, S. Tyagi, S. Kumar, The role of internet of things and smart grid for the development of a smart city, in: Intelligent Communication and Computational Technologies: Proceedings of Internet of Things for Technological Development, IoT4TD 2017, Springer, 2018, pp. 23-33.
[11] V. Chauhan, M. Patel, S. Tanwar, S. Tyagi, N. Kumar, IoT Enabled real-Time urban transport management system, Computers & Electrical Engineering, 86 (2020) 106746.
[12] J. Masino, J. Thumm, M. Frey, F. Gauterin, Learning from the crowd: Road infrastructure monitoring system, Journal of Traffic and Transportation Engineering (English Edition), 4(5) (2017) 451-463.
[13] K.W. Wevers, M.L. Lu, R. van der Heijden Heijden, Technical feasibility of advanced driver assistance systems (ADAS) for road traffic safety, (2005).
[14] Q. Li, F. Qiao, X. Wang, L. Yu, Drivers' smart advisory system improves driving performance at STOP sign intersections, Journal of traffic and transportation engineering (English edition), 4(3) (2017) 262-271.
[15] D. Sunehra, K. Jhansi, Implementation of microcontroller based driver assistance and vehicle safety monitoring system, in: 2015 International Conference on Information Processing (ICIP), IEEE, 2015, pp. 423-428.
[16] A. Ziebinski, R. Cupek, D. Grzechca, L. Chruszczyk, Review of advanced driver assistance systems (ADAS), in: AIP Conference Proceedings, AIP Publishing, 2017.
[17] M.J. Carrillo, Robotic cars test platform for connected and automated vehicles, University of California, Irvine, 2015.
[18] A. Shaout, D. Colella, S. Awad, Advanced driver assistance systems-past, present and future, in: 2011 Seventh International Computer Engineering Conference (ICENCO'2011), IEEE, 2011, pp. 72-82.
[19] U. Hofmann, A. Rieder, E.D. Dickmanns, Radar and vision data fusion for hybrid adaptive cruise control on highways, Machine Vision and Applications, 14 (2003) 42-49.
[20] A. Kesting, M. Treiber, M. Schönhof, D. Helbing, Adaptive cruise control design for active congestion avoidance, Transportation Research Part C: Emerging Technologies, 16(6) (2008) 668-683.

[21] S. Cafiso, A. Di Graziano, G. Pappalardo, In-vehicle stereo vision system for identification of traffic conflicts between bus and pedestrian, Journal of traffic and transportation engineering (English edition), 4(1) (2017) 3-13.
[22] C. Ilas, Electronic sensing technologies for autonomous ground vehicles: A review, in: 2013 8th International Symposium on Advanced Topics in Electrical Engineering (ATEE), IEEE, 2013, pp. 1-6.
[23] B.-G. Lee, W.-Y. Chung, A smartphone-based driver safety monitoring system using data fusion, Sensors, 12(12) (2012) 17536-17552.
[24] J. Carmona, F. García, D. Martín, A. de la Escalera, J.M. Armingol, Data fusion for driver behaviour analysis, Sensors, 15(10) (2015) 25968-25991.
[25] J. Yu, Z. Chen, Y. Zhu, Y. Chen, L. Kong, M. Li, Fine-grained abnormal driving behaviors detection and identification with smartphones, IEEE transactions on mobile computing, 16(8) (2016) 2198-2212.
[26] S. Siuhi, J. Mwakalonge, Opportunities and challenges of smart mobile applications in transportation, Journal of traffic and transportation engineering (english edition), 3(6) (2016) 582-592.
[27] G.M. Fitch, K. Grove, R.J. Hanowski, M.A. Perez, Compensatory behavior of drivers when conversing on a cell phone: Investigation with naturalistic driving data, Transportation research record, 2434(1) (2014) 1-8.
[28] M.M. Haque, S. Washington, The impact of mobile phone distraction on the braking behaviour of young drivers: a hazard-based duration model, Transportation research part C: emerging technologies, 50 (2015) 13-27.
[29] W.J. Horrey, M.F. Lesch, M.J. Dainoff, M.M. Robertson, Y.I. Noy, On-board safety monitoring systems for driving: review, knowledge gaps, and framework, Journal of safety research, 43(1) (2012) 49-58.
[30] W. Consiglio, P. Driscoll, M. Witte, W.P. Berg, Effect of cellular telephone conversations and other potential interference on reaction time in a braking response, Accident Analysis & Prevention, 35(4) (2003) 495-500.
[31] S.G. Klauer, F. Guo, B.G. Simons-Morton, M.C. Ouimet, S.E. Lee, T.A. Dingus, Distracted driving and risk of road crashes among novice and experienced drivers, New England journal of medicine, 370(1) (2014) 54-59.
[32] T.A. Dingus, F. Guo, S. Lee, J.F. Antin, M. Perez, M. Buchanan-King, J. Hankey, Driver crash risk factors and prevalence evaluation using naturalistic driving data, Proceedings of the National Academy of Sciences, 113(10) (2016) 2636-2641.
[33] V. Havyarimana, D. Hanyurwimfura, P. Nsengiyumva, Z. Xiao, A novel hybrid approach based-SRG model for vehicle position prediction in multi-GPS outage conditions, Information Fusion, 41 (2018) 1-8.
[34] H. Ahmed, I. Ullah, U. Khan, M.B. Qureshi, S. Manzoor, N. Muhammad, M.U. Shahid Khan, R. Nawaz, Adaptive filtering on GPS-aided MEMS-IMU for optimal estimation of ground vehicle trajectory, Sensors, 19(24) (2019) 5357.
[35] T. Driessen, L.L.B. Prasad, P. Bazilinskyy, J. de Winter, Identifying Lane Changes Automatically using the GPS Sensors of Portable Devices, Human Factors in Transportation, 60 (2022) 50.
[36] S. Stieger, D. Lewetz, U.-D. Reips, Can smartphones be used to bring computer-based tasks from the lab to the field? A mobile experience-sampling method study about the pace of life, Behavior Research Methods, 50 (2018) 2267-2275.
[37] A. Fani, H. Naseri, Travel Time Modelling of Urban Roads By Application of Coyote Optimization-based Machine Learning Method, Amirkabir Journal of Civil Engineering, 53(9) (2021) 3649-3664.
[38] S. Ranjbar, F. Moghadas Nejad, H. Zakeri, Asphalt Pavement Bleeding Evaluation using Deep Learning and Wavelet Transform, Amirkabir Journal of Civil Engineering, 53(11) (2022) 4577-4598.
[39] P.S. Bokare, A.K. Maurya, Acceleration-deceleration behaviour of various vehicle types, Transportation research procedia, 25 (2017) 4733-4749.