Effective feature extraction from driving data for detection of danger awareness

Kotaro Nakano
Graduate School of Software and Information Science
Iwate Prefectural University
152-52 Sugo, Takizawa, Iwate 020-0693, JAPAN
Email: g236p002@s.iwate-pu.ac.jp

Basabi Chakraborty
Faculty of Software and Information Science
Iwate Prefectural University
152-52 Sugo, Takizawa, Iwate 020-0693, JAPAN
Email: basabi@iwate-pu.ac.jp

Abstract—In recent years, the importance of driver’s support system is increasing as a solution for dealing with care related accidents. These driving support systems are equipped with functions for avoiding various hazards when the driver drives the vehicle, reducing the risk of causing an accident. In this research, we focus on the time series data of the driving behaviour of the driver, and based on these data, experiments aiming at development of the dangerous driving detection system due to cognitive distraction of the driver have been conducted. The driving behaviour data have been collected from driving simulator which contain driver’s actions mainly steering, accelerator and foot brake operations. It has been observed that the driving behaviour of each driver changes while driving in the state of distraction from while driving attentively and by analyzing these changes, the driver’s distraction from the normal state can be detected. The objective of this paper is to find the effective features for detection of distracted driving of specific driver in real time (specific short intervals). From the collected data of driving behaviour of multiple subjects, static feature based driving model and dynamic feature based driving model for individual drivers and all drivers for attentive driving and distracted driving have been developed. It can be shown from the results that distracted driving can be identified for individual in real time with stable accuracy using dynamic feature based models.

Index Terms—cognitive distraction, driving data, feature extraction, dynamic feature, recurrence plot,

I. INTRODUCTION

In recent years, the importance of cars equipped with driver’s support systems is increasing in the field of elderly people’s driving support and some researches on driver’s support system are still going on. These existing driver’s support systems are designed based on the behaviour of an average driver. However, driving by each individual driver has different characteristics depending on his or her preferences and driving skill and there is a possibility that sufficient support cannot be done with the system developed on the basis of average driving.

From the above background, it is important to design a personal model for each driver’s driving behaviour and detection of distracted driving from normal driving should be developed on the basis of individual models which can greatly improve the accuracy of the dangerous driving detection.

Generally, the distracted state during driving refers to a state in which the driver cannot concentrate on driving due to some factors, for example, for operating other things than a steering or a gear, or the driver looks away from the road or the driver’s mind is busy with something not directly related to driving. In this research, we aim to detect distracted driving that may lead to accidents due to cognitive load [1] being imposed by some factors and it becomes difficult to perform attentive driving. The driving data have been collected from driving simulator (D3sim) that represents driver’s driving behaviour (steering angle, steering torque, acceleration stroke, car speed, engine speed, brake stroke etc). These data are time series data sampled at regular intervals from sensors incorporated in the driving simulator.

There are mainly three approaches for time series classification. In this work, feature based approaches are followed for analysis of driving simulator data and a comparative study has been made with static features and dynamic features. Time series driving simulator data have been represented by static and dynamic features separately. Static features refer to statistical parameters of the time series taken over a window of subsequences and expressed as the average value over the window such as the mean value and mean variance of subsequences. As the average is taken, the instantaneous changes of the time series cannot be taken into account. Dynamic features refer to the features that retains the information on changes with time of the time series data such as recurrence plot etc.

The main objective of this work is two fold, one is to verify whether there is a difference in driving behaviour for each individual and the other is to check whether it is possible to detect a distracted state from the driving behaviour from classification results of the time series. For the first objective, the classification problem is to distinguish individual subjects and for the second objective, the classification problem is to distinguish attentive state and distracted state of the driver. Samples (sub sequences) used in classification are extracted from the original time series data at regular intervals. The next section presents some related works followed by methods and experiments in the following section. Section 4 describes the results followed by final section of conclusion.

II. RELATED WORKS

Currently, various research works on the automatic detection of driver’s cognitive distraction based on various sensor data are carried out. In [2] [3], the image of the driver’s face...
obtained from the camera is used for detecting driver’s cognitive distraction by convolutional neural network. Measuring cognitive distraction from cameras or any nonintrusive sensors is difficult as many parameters need to be integrated for detecting cognitive distraction. In [4] [5] Support Vector Machines (SVM) and Dynamic Bayesian Networks (DBN) are used respectively for detecting cognitive distraction from driver’s visual behaviour and driving performance.

In [6], detection of vigilance level of the driver has been done from the combination of driver’s EEG signals and the data of driving contexts. In [7] real-time distracted state of distraction has been detected based on eye movement [8] and another approach using Machine learning technique is presented in [9]. In [10] time series analysis of sensor data is used for detection of cognitive distraction. In [11], authors developed a method of detecting cognitive distraction from driver’s eye movements and driving data using Inductive Logic Programming.

Also, with regard to distraction state detection from these sensor data, in particular, methods using deep learning have been widely used in recent years due to increase of computational resources by evolution of hardware and development of methods using deep learning. The computational cost of using deep learning method is very high, but it has been proved to be able to detect distracted states with high accuracy. In particular, many methods using images and convolution neural networks (CNN) have been proposed and high classification accuracy can be obtained.

III. METHODS AND EXPERIMENTS

In order to verify the possibility of cognitive distraction detection by developing individual models, classification is performed using time series data on driver’s behaviour obtained from multiple drivers.

A. Data collection

The definition of driver’s distracted state in this experiment is that the driver is not concentrating on driving and in this experiment, we defined talking or simple chatting with co-passerenger (talking) and doing simple arithmetic operation like addition, multiplication without pencil and paper (mental arithmetic) as the cognitive load. We collected driving behavior data from several different drivers using driving simulator (D3sim). The detail of the data collection environment is as follows.

1) The age of the subjects is in their twenties and they have a driving license.
2) Two different courses through city roads were set for driving.
3) All subjects practiced 10-20 minutes before collection of actual data.
4) When the scenario started, subjects drove following a car in front of the subject’s car.
5) The subjects did not stop at traffic light but followed other traffic rules.

B. Data set details

In this section, details of feature extraction from the collected data is described.

After collection of all the data, we checked the value of each sensor data and decided to use the data of Acceleration stroke, Brake stroke, Engine speed, Car speed, Steering torque and Steering angle. samples are shown in Fig. 1 to Fig. 6. These features are recorded at 120 Hertz, but to reduce computation time we converted these data to 30 Hertz and finally the data were normalized.
Features extracted from the simulator data:

1) Steering angle  
2) Steering torque  
3) Acceleration stroke  
4) Brake stroke  
5) Car speed  
6) Engine speed 
7) Rate of turn  
8) Engine output torque  
9) Posture angle (3 axis)  
10) Velocity vector (3 axis)  
11) Angular velocity vector (3 axis)  
12) Acceleration vector (3 axis)  
13) Angular acceleration vector (3 axis)

C. Prepossessing

First 20 sec and last 20 sec of the data has been deleted. The features with no significant change over time also has been removed. Finally, a sliding window is applied to each sample for each feature. Window width has been taken as 5 second to 17 second with overlap 0.2 to 0.5.

D. Classification

In classification experiments, classification is performed for two objectives. 1) verification of whether each user has a characteristic of driving, which makes it possible to identify a specific user by machine learning, 2) detection of distracted state with high accuracy by developing a driving model for each individual. Training data and test data are split at 80:20 ratio before each experiment. The ratio of class labels in the data set is equal for all the 4 users for user identification.

1) Classification with static features: In the classification using static features, one sample is created by calculating each value from one sub sequence as shown below.

Static features:

1) Mean value  
2) Variance  
3) Largest value in the array  
4) Smallest value in the array  
5) Interquartile range  
6) Sum of the squares divided by the number of values  
7) Max index  
8) Minimum index

2) Classification using dynamic feature: In this work, recurrence plot [12] is used as the dynamic feature and Convolutional Neural Network (CNN) [13] as the classifier. Recurrence plot is a tool for visualizing attractors in chaotic time series analysis, and it is used as one of time series data representation methods in time series classification. By using the recurrence plot, the dynamic characteristics of time-series data can be represented by a single image that is a different representation from raw data. Also, when generating the recurrence plot, one needs to estimate embedding parameters $m$ and $tau$. In the experiment, we set parameter $m$ in the range from 2 to 5 and set parameter $tau$ in the range from 1 to 5.
<table>
<thead>
<tr>
<th>Feature</th>
<th>Classifier</th>
<th>Target</th>
<th>Data separation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic</td>
<td>FCN</td>
<td>USER</td>
<td>all data</td>
</tr>
<tr>
<td>Static</td>
<td>SVM, KNN</td>
<td>USER</td>
<td>all data</td>
</tr>
<tr>
<td>Dynamic</td>
<td>FCN</td>
<td>Drivers state</td>
<td>all data</td>
</tr>
<tr>
<td>Static</td>
<td>SVM, KNN</td>
<td>Drivers state</td>
<td>all data</td>
</tr>
<tr>
<td>Dynamic</td>
<td>FCN</td>
<td>Drivers state</td>
<td>User individual</td>
</tr>
<tr>
<td>Static</td>
<td>SVM, KNN</td>
<td>Drivers state</td>
<td>User individual</td>
</tr>
</tbody>
</table>

The figure 7 shows the full convolutional network (FCN) architecture used for classification. It is the basic architecture of CNN that includes two convolution layers. The input recurrence plot is resized to $30 \times 30$ before input to CNN to decrease computational cost.

The classification experiments are shown in Table III.

The results of user recognition, classification accuracy versus window size is shown in Fig. 8, Fig. 9 and Fig. 10 for static feature with SVM, KNN and for dynamic features respectively. From the results, both features have higher recognition accuracy as window size is extended because characteristic patterns can be captured well by increasing window size, optimum window size being 12-13 second. With regard to the value of overlap, it is better to use as small as possible, because a large difference in accuracy is obtained particularly when using dynamic feature. Considering the best situation it is found that individual user can be classified with 90% or a little more accuracy using the dynamic feature.
window size compared to the case of user classification. From the results of the Fig. 13, it is understood that the classification can be performed with high accuracy if dynamic feature is used. Also, it is found that accuracy changes with the change of overlap.

C. Results of cognitive distraction by individual user

Accuracy of cognitive distraction detection for individual user with static features is shown in Table IV and Table V (the best obtained values are shown). From the tables it is found

**Table IV**

<table>
<thead>
<tr>
<th>accuracy</th>
<th>window size(sec)</th>
<th>overlap(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>All USERS</td>
<td>0.578</td>
<td>15</td>
</tr>
<tr>
<td>USER1</td>
<td>0.594</td>
<td>15</td>
</tr>
<tr>
<td>USER2</td>
<td>0.746</td>
<td>14</td>
</tr>
<tr>
<td>USER3</td>
<td>0.631</td>
<td>14</td>
</tr>
<tr>
<td>USER4</td>
<td>0.735</td>
<td>11</td>
</tr>
<tr>
<td>Average of USER indiv</td>
<td>0.677</td>
<td></td>
</tr>
</tbody>
</table>

**Table V**

<table>
<thead>
<tr>
<th>accuracy</th>
<th>window size(sec)</th>
<th>overlap(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>All USErS</td>
<td>0.615</td>
<td>5</td>
</tr>
<tr>
<td>USER1</td>
<td>0.692</td>
<td>17</td>
</tr>
<tr>
<td>USER2</td>
<td>0.725</td>
<td>12</td>
</tr>
<tr>
<td>USER3</td>
<td>0.631</td>
<td>7</td>
</tr>
<tr>
<td>USER4</td>
<td>0.72</td>
<td>17</td>
</tr>
<tr>
<td>Average of USER indiv</td>
<td>0.693</td>
<td></td>
</tr>
</tbody>
</table>

...
that the detection accuracy increases for individual user’s data compared to mixed data for all users and accuracy varies from user to user for both the classifiers. Next, the result of cognitive distraction by individual user using dynamic feature is shown in Table VI. This table is a summary of the results of cognitive distraction detection by individual users. The accuracy in this table represents the highest accuracy among parameter of each $m$ and $tau$. It is seen that the accuracy has been greatly improved for individual user. Moreover, the detection accuracy varies from user to user. The details of the results for individual users are shown in Table VII, Table VIII, Table IX and Table X respectively. It seems that user 4 has the highest detection accuracy among all the users. Fig. 14 represents change of distracting detection accuracy with window size and overlap for user 4.

V. CONCLUSION AND FUTURE WORK

In this work, driving data from different sensors attached to the driving simulator have been collected for several users for attentive and distracted driving for an analysis. The objective has been to explore the possibility of detecting distracted driving in real time from the driving behavior reflected in the data from simulator. We considered only cognitive distraction caused by occupation of mind with things other than driving. From our investigation with the collected data and their analysis, it is found that there is a considerable difference between attentive and distracted driving that can be classified from driving behaviour data. We also found that the driving behaviour varies from person to person and the difference of attentive driving and distracted driving varies from person to person.

The maximum detection accuracy of 92% has been achieved by considering individual user’s driving pattern and dynamic feature with CNN classifier. It is also verified that the smaller the overlap of the subsequence, the greater the accuracy and a window size of 15 sec is optimum for the maximum detection accuracy. Now in the next step, we need to lower computational cost as much as possible for developing a real time application.

ACKNOWLEDGMENT

This research was supported by PRML Laboratory of department of Software and Information Science, Iwate prefectoral university.

REFERENCES