Mistaken Pedal Pressing during Emergency Braking by Analyzing Pedal Behaviors

Rahadian Yusuf *, Ivan Tanev *, and Katsunori Shimohara *

Abstract: Affective computing has been used to improve computer usability and user interface by considering user’s emotion. There is still ample scope for exploration of affective computing especially for applications in a driver assistance system to recognize driver’s emotions upon an emergency braking that has been extensively studied by several pieces of research. Pressing an accelerator mistakenly during an emergency braking is a serious case that leads to accidents especially in the case of an elderly driver or a beginner. This paper reviews briefly the affective computing, emergency braking, and mistaken pedal pressing, proposing a possible approach for improving the existing driving assistance system by analyzing driver’s behavior of pedal pressing. The study is based on the idea that an emergency braking behavior should not mistakenly happen on the accelerator pedal, and we used evolutionary computing to investigate this idea. Results from our reviews and experiments showed that current affective computing technology might be insufficient to recognize a mistaken pedal pressing using facial expressions or similar emotions. However, we found that analyzing pedal pressing behaviors of a driver can recognize a mistaken pedal pressing during emergency braking. This can provide further alternative options to improve the safety of driving, especially for elderly and beginners, as we focus on the driver aspects.

Key Words: affective computing, emergency braking, mistaken pedal pressing.

1. Introduction

Affective computing, a term coined by Picard [1], is a field of computer science and information technology, focusing on recognition of expressions and emotions as well as improving human–computer interaction. Tao [2] describes that emotion recognition helps improve the agent’s accuracy in predicting user’s intention. It is an interdisciplinary field spanning computer science, psychology, and cognitive science.

There is ample scope for implementing affective computing in a number of practical applications, especially driving. Recognizing driver’s emotion might contribute to positive improvement on driving assistance systems.

In this research, we are particularly interested in an emergency braking situation leading to a mistaken pedal pressing, whereby a driver presses the accelerator instead of brake accidentally. There are several reasons for our interest as discussed below.

Our main reason is that a mistaken pedal pressing is a dangerous situation as it might lead to an accident. Because of the high risk associated with the situation, we decided to check the availability of a method of prevention of the situation by detecting the likely occurrence of emergency braking. In this context, the research on this field, which is still in progress, hints the possibilities of detection. However, as discussed in detail in the subsequent sections, a driver is likely to panic during an emergency situation, thereby making a mistake. Recognizing a mistake due to human panicking in an emergency then became our second reason of interest.

According to research by Suzuki [3], the proportion of accidents caused by mistaken pedal pressing is increasing despite the decreasing trend in the total number of accidents, and the proportion has especially been higher for elderly persons. One concern about emergency braking is that an accident happens when a driver presses the accelerator pedal instead of the brake pedal, thus increasing the speed of the car instead of decreasing. This might lead to a fatal accident, especially if the driver is panicking and pressing the accelerator pedal forcefully.

Our research on recognizing a mistaken pedal pressing starts with the idea that the pedal pressing behavior can be recognized by a simple behavioral aspect that differentiates between normal acceleration and emergency braking. We assume that the accelerator pedal behavior during normal driving is different from the brake pedal behavior during emergency braking. In other words, the way a driver presses the accelerator pedal during normal driving is assumed to be different from the way he presses the brake pedal during emergency braking.

We collected data of 15 subjects using a driving simulator and then used machine learning (genetic programming) to evolve a classifier that can recognize them. We demonstrated that the evolved classifiers can differentiate between the behavior of pressing an accelerator pedal during normal driving and that of pressing a brake pedal during emergency braking.

2. Related Work

In the preliminary stage of our research, we first studied the literature on affective computing, especially related to driving. Subsequently, we collected further details by studying the literature specifically on emergency braking and mistaken pedal pressing. Furthermore, we investigated various options regarding the safe environment for experimenting the mistaken pedal pressing. Further alternative options to improve the safety of driving, especially for elderly and beginners, as we focus on the driver aspects.

* Graduate School of Science and Engineering, Doshisha University, Kyoto, Japan
E-mail: yusuf2013@sil.doshisha.ac.jp, itanev@mail.doshisha.ac.jp, kshimoha@mail.doshisha.ac.jp
(Received October 29, 2018)
(Revised February 9, 2019)
pressing during emergency braking.

2.1 Affective Computing

Our previous work covered a survey on utilizing affective computing [1] for drivers, and it was found that there was ample scope to explore in this regard [4]. However, the cumbersome sensors, especially in respect of their time response, pose a major challenge in utilizing affective computing for drivers, despite several advances in this field including emotion classification and facial expressions [5]–[7]. In addition, our preliminary experiments revealed the difficulties in relying on facial expression analysis due to technical limitations, as explained in detail in Section 3.1.

We then decided to analyze the pedal pressing behavior of drivers, by assuming that the pattern of accelerator pedal pressing for normal acceleration is different from that of brake pedal during emergency braking. The following sections show that these two patterns are different and therefore our assumption can be used for further advances.

2.2 On Driver’s Distraction and Emergency Braking

There have been many pieces of research on emergency situations during driving, and there are two specific fields relevant to our research: driver’s distraction (or condition) and emergency braking.

Sharma [8] observed that cognitive load on a driver might impair him cognitively, which might in turn cause a steering oscillation leading to accidents. The research performed simulations to evolve a model of a driving agent that could control a car as well as recognize a possible distraction of a driver.

Several studies [9]–[11] suggested that visual and cognitive distractions are the two major types of driver distraction that can degrade performance, while Liang [12] and Liao [13] investigated cognitive distraction of a driver using support vector machine (SVM) and eye movement.

Regarding emergency braking, research studies indicate different approaches. Suzuki [3] investigated the pressure applied on brake pedals during emergency braking. Podusenko [14] investigated the detection of emergency braking by lifting the accelerator pedal instead of pressing the brake pedal and developed a driving assistant system applicable to a driving simulator. Tsukuda [15] investigated an advanced emergency braking system (AEBS) and proposed a method to estimate road friction coefficient, focusing more on vehicle dynamics (especially tire) using simulations on CarSim 8.2.1, and they also discussed on mechanical aspects on emergency braking.

There is still ample scope for research on emergency braking that necessarily incorporates the aspects of drivers in addition to those of vehicle.

2.3 Driving Simulator

Many types of research could easily use a real car on a real world because of relatively normal and safe driving; however, we could not adopt the same approach in our research due to the abnormal driving, which is to simulate a mistaken pedal pressing during emergency braking. Moreover, an emergency braking test using a real car in a real-world situation is too dangerous to perform.

Several pieces of research use a simulated environment of driving by setting up a computer and a camera in front of the users, in the same way as performed by Paschero [11]. However, we preferred to perform our research in an environment with more similarities to driving a real car.

In line with the scope of our research, we opted to use a driving simulator (Forum-8) that is already available for use at our institution. Figure 1 shows a photograph of the simulator.

3. Experiments

3.1 (Preliminary) Experiment 1: Expression

The purpose of this experiment is to simulate a case of mistaken pedal pressing during emergency braking [4]. Using the Forum-8 driving simulator, we could manipulate the pedals to simulate a different behavior. Accordingly, in this case, we prepared a manipulation by setting the brake pedal to act as an accelerator. The purpose of this manipulation was to simulate a mistaken pedal pressing during emergency braking and to examine the changes in the facial expressions of the subjects.

The subjects were asked to drive on a selected track that induces an emergency braking situation, as a response to which the driver must immediately press the brake, otherwise a crash happens. During the driving experiments, immediately before the induced emergency braking situation, the functionality of the brake pedal is superseded by that of the accelerator pedal without any notice to the driver. The subjects were not informed of the purpose of the experiment, and they were asked only to perform a normal (fast but safe) driving. This was done to simulate a mistaken pedal pressing. In most cases of mistaken pedal pressing, the driver did not realize that they were pressing the wrong pedal [3]. For this initial experiment, we used two cameras: one to record the view from the back side of the subject, showing the subject’s activities and screen of the driving simulator; and the other one to record the subject’s facial expression for detection of possible changes in emotions.

We then synchronized both the recorded videos to investigate the emotions aroused during the simulated mistaken pedal pressing.

Two subjects were required to use the Forum-8 driving simulator. During the normal driving and prior to an emergency braking situation, the brake pedal function was set to facilitate acceleration of the car immediately upon pressing the pedal. We studied the videos to observe the changes in emotions of the subjects based on their facial expressions. However, regrettably, both subjects did not show any obvious changes in their facial expressions. When we interviewed the subjects briefly, they expressed that they were neither panicking nor having a fear, as they were aware that there was no immediate real-life
threat of crashing. On the contrary, the subjects developed confusion because they were getting a feel as though the machines were not working properly.

One of the reasons of this preliminary result was the lack of involvement and conditioning of the subject. In our experiment, we had not asked the subjects to drive many times and try emergency braking on their own instead of a manipulated setting; thus, they might not have understood the normal situation. However, we indeed recognized obvious signs of change in the subject’s behavior, although they were not showing obvious changes in their facial expressions, as they seemingly became stiffer. We expected some physiological changes; however, recognizing them from their facial expressions might not be feasible or it might take time due to the nature of emergency.

We also performed a simple analysis from the synchronized video. We were specifically interested in the time-series information between the moments of mistaken pedal-pressing, crashing, and driver’s emotion. We used Microsoft Emotion API to recognize the emotions frame by frame, with a sampling interval of 100 ms.

Figure 2 shows a sample frame of synchronized video. Figures 3 and 4 show sample results from the two different subjects; only three obviously detected emotion changes are shown and other emotions that were barely detected by the API were omitted for easier comprehension.

The figures suggested that it was usually getting too late to detect an emotion. Because of this negative result, we decided to investigate the driver’s behavior based on a different measure rather than facial expression, as previously stated.

3.2 Experiment 2

3.2.1 Pedal behavior

We used the Forum-8 driving simulator to collect the data of normal driving and emergency braking. The subjects were asked to drive on several tracks, varying from a high-speed highway to rural area, from clear weather to stormy weather. Several times during the driving, the subjects were asked to perform emergency braking, simulated by a sudden bell or an alarm to represent an emergency case.

For our research, the raw data collected from the Forum-8 driving simulator were the positions of the accelerator pedal and the brake pedal at a sampling rate of $\sim 40$ ms. We also extracted initial features of local gradient for the respective positions of the accelerator pedal and the brake pedal.

From the 15 subjects, we then extracted the data on several features for each part of the time series: the maximum gradient, the maximum position, and the average gradient. A part of the time-series data represents a slope of the signal corresponding to pressing of the pedal (before it is released), which represents one row of extracted data. Subsequently, we separated the training and test data-sets, labeling them as either TRUE (for emergency braking data) or FALSE (for normal acceleration).

For performing the evolution of a classifier, we opted for genetic programming (GP) using our in-house GP engine named XGP [16],[17]. XGP is an in-house GP engine featuring extensible markup language (XML)-based genotype representations of candidate solutions (genetic programs), XML-schema that determines the allowed syntax of the genotypes, and a user datagram protocol (UDP) channel to communicate between a fitness evaluator and the XGP. XGP manages the population of genetic programs and performs the main genetic operations: selection, crossover, and mutation.

As an in-house GP engine, XGP has been used extensively at our laboratory to evolve various classifiers and genetic programs. In addition, XGP is advantageous in that it has a shorter development time owing to its versatility. Typically, it only took a reasonably little time to change the XML-schema and adapt...
the desired syntax of genotypes for a program. In other words, it is modular, and we need only editing an external file (representing the Backus–Naur form of the grammar) that describes the desired syntax of the evolved genetic programs.

Evolving programs using XGP requires that the engine runs as a Microsoft Windows application in parallel with another application to evaluate the fitness of each individual sent by XGP. Then, the XGP manages the population and GP operations. For genotype representation in our evolution process, we used terminals as shown in Table 1.

| Feature Name | Values |
|--------------|--------|
| Variables of 3 extracted features (terminal) | 0–100 (integer) |
| Random constants (terminal) | 1–10 (integer) |
| Arithmetic operations (non-terminal) | +, −, *, / |

Fig. 4 Decisive 10s of Experiment 1 (two instances of repeated experiments) from subject 2 (emotion reading based on time) [4].

The fitness function determines the mechanism of evaluating the quality of each individual in XGP (i.e., the evolved mathematical function for emotion recognition). In our work, we used Matthew correlation coefficient \( (MCC) \) [18] as a fitness function, as shown in Eq. (1):

\[
MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}},
\]

where

\( TP = \) True Positive,
\( TN = \) True Negative,
\( FP = \) False Positive,
\( FN = \) False Negative.

The range of \( MCC \) is \(-1\) to \(1\), but for our experiments, we normalized this \( MCC \) into a score of \(10,000\) (total mismatch) and \(0\) (total match).

The pedal behavior data collected by us were found to be considerably imbalanced; there were plenty of normal acceleration data but only a limited emergency braking data. Therefore, we evolved the system twice; first with the imbalanced data, and second with the balanced data.

It is natural that the number of extracted data for normal acceleration is much higher than that for emergency braking data. The data from the 15 subjects included \(\sim 500\) rows of emergency braking and \(\sim 2700\) rows of normal acceleration. Therefore, the normal acceleration data was downsized to \(\sim 800\) rows by way of random removal of partial data so as to obtain a reasonable proportion to the extent of preventing overfitting.

For evolving the program of classifiers, we ran the XGP iteratively using training data-set, which can simply be called as one independent run. Thus, one independent run of XGP is executed iteratively to select an individual program that provides the best fitness function, whereby the termination criteria were defined as satisfactory fitness (below \(20\) out of \(10,000\)), the number of generations reached \(\) (iterations, \(100\)), or stagnation that implies no improvements after \(32\) generations. As the generations increase, the fitness function is expected to be getting better; in this research, the lower the better.

Fitness convergence (of \(20\) independent runs of XGP) of each evolution is shown in Fig. 5. The vertical axis represents fitness value (the lower the better), while the horizontal axis represents generation. It is evident that the evolutions converged well, implying that both behaviors can be recognized by machine learning.

It is evident from Fig. 5 that removing possible redundant data of normal acceleration during normal driving yields a better result in terms of computational effort and effectiveness. The fitness landscape converges faster and yields better fitness as well.

3.2.2 Evaluation by cross-validation

As revealed by the fitness convergence, the genetic programming can be used to evolve a classifier to the problem. However, fitness convergence alone is not adequate to evaluate the evolved programs properly. We need to evaluate the test data-set that is different from training data-set, to verify that
We prepared our next experiment [19] by separating the balanced data-set into four parts randomly, as shown in Fig. 6. We then ran genetic programming on four batches with different training data-sets each consisting of three parts, and each batch consisting of 20 independent runs. Subsequently, we performed cross-validation for evaluation, using the best evolved programs from each batch.

We obtained results [19] as shown in Table 2. It is quite evident that the best evolved programs are capable of recognizing a pattern between normal acceleration and emergency braking. It is evident from Table 2 that the result is cross-validated and performed similarly. The average accuracy of recognition for test data-sets is 95.72%, that for False (normal driving) is 96.83%, and that for True (emergency braking) is 94.6%.

From the results, we conclude that the genetic programming is usable to evolve a good classifier to differentiate patterns between normal acceleration and emergency braking. However, we preferred to investigate further about this particular method, due to the possibility of overfitting.

3.2.3 Collaborative filtering

Genetic programming is a stochastic method that is not safeguarded against machine learning problems especially overfitting. It is possible that an evolved individual program is overfitted to a training data-set and could not perform well on test data-set or in a real-world application.

We conducted an experiment to investigate the results obtained by selecting the best five evolved individual programs instead of just a single best program. From these five bests of the runs, we performed collaborative filtering using a simple voting mechanism, whereby each of these programs would vote the prediction, and the one with the majority of voting wins. We were interested in investigating the results of such a simple method and comparing them with the predicted best, and with the average of the best five runs themselves. Results are shown in Table 3.

Using the results from the cross-validation experiments, we managed to obtain the results as shown in Table 3. There are four sub-tables, each corresponding to the cross-validation evaluation. Five columns (1A–1E) represent the best five evolved programs to the respective batch. The column ‘Col’ shows the result of collaborative filtering (simple voting), while ‘Avg’ represents the average of the five values.
At the bottom of each sub-table, there is a row titled ‘Average’ that represents the average accuracy percentage on test data-set; we have included the results of the training data-set only for the sake of comparison. Bolded results show a better value between collaborative filtering and the simple average of the best five programs.

For further comparison, we have underlined the individual results of each evolved program on each data-set (training and test). In most cases of this experiment, we can see that the best evolved programs do not always give the best performance on test data-set. This might be caused by overfitting, and a method such as collaborative filtering can be used to obtain a more general and robust classifier.

From the bolded results, we can see that collaborative filtering exhibits better performance than the average of the best five programs. It is still lower than the best results on test data-set, but it is difficult, if not impossible, to know which of the evolved programs would do better in test data-set or real-world data-set. Therefore, collaborative filtering on several best individual programs could be a good-enough approach to obtain satisfactory results.

4. Conclusions and Future Work

We briefly reviewed the research studies on implementation of affective computing for recognition of a mistaken pedal pressing during emergency braking, and we found that there was ample scope for further exploration. We used the Forum-8 driving simulator to create a more realistic driving environment yet without using a real car because of the risks or dangers associated with emergency braking.

From our first preliminary experiment, we used a simulated mistaken pedal pressing to investigate driver’s behavior and changes of facial expression. We used interviews, video analyses, and emotion recognition on time-series data and found that drivers were not involved or conditioned enough during the experiments.

In our second experiment, we analyzed pedal pressing behaviors of 15 subjects in respect of both normal acceleration and emergency braking. The basic idea was that the respective pedals should exhibit only these two behaviors. The results suggested that both behaviors can be recognized by using machine learning, which happens to be genetic programming in this case, as suggested by the fitness convergence. We then performed cross-validation for evaluation, and we obtained a stable result with an average accuracy of recognition of 96.83% for normal driving and 94.6% for emergency braking, the collective average of the accuracy being 95.72%. Furthermore, we investigated the implementation of collaborative filtering by using a simple voting method from best five evolved programs, instead of a single best evolved program.

From the above summary, we conclude that it is feasible to implement affective computing for recognition of a mistaken pedal pressing during an emergency braking by analyzing the pedal pressing behavior, despite the technical limitations in the recognition by using facial expressions.

For future work, we are looking forward to realizing an improved accuracy, especially in a real-time implementation. However, we also need to consider the differences in respect to mechanical aspects between brake pedal (nonlinear) and accelerator pedal (linear). So far, we assumed in our experiments that these differences do not influence the pedal-pressing behavior. It would indeed be another direction of research to investigate the mechanical dynamics of the pedals themselves.

References

[1] R. Picard: Affective computing, MIT Media Laboratory Perceptual Computing Section Technical Report, No. 321, 1995.
[2] J. Tao and T. Tan: Affective computing: A review, J. Tao, T. Tan, and R.W. Picard (Eds.), Affective Computing and Intelligent Interaction, LNCS 3784, pp. 981–995, Springer, 2005.
[3] T. Suzuki: Method for detecting operation mistakes with accelerator pedal, Proceedings of Fast-Zero 2017, TuB-P2-1, 2017.
[4] R. Yusuf, I. Taney, and K. Shimohora: Affective computing to help recognizing mistaken pedal-pressing during accidental braking, Artificial Life and Robotics, 10.1007/s10015-018-0515-1, 2019.
[5] P. Ekman and W.V. Friesen: Facial Action Coding System, Consulting Psychologist Press, 1977.
[6] J.A. Russel: A circumplex model of affect, Journal of Personality and Social Psychology, Vol. 39, No. 6, pp. 1161–1178, 1980.
[7] J.F. Cohn: Foundations of human computing: Facial expression and emotion, International Conference on Multimodal Interfaces, pp. 233–238, 2006.
[8] D.G. Sharma, R. Yusuf, I. Taney, and K. Shimohora: Effects of cruising speed on steering oscillations of car induced by modeled cognitively impaired human driver, SICE Journal of Control, Measurements, and System Integration, Vol. 10, No. 3, pp. 156–164, 2017.
[9] J.D. Lee, B. Caven, S. Haake, and T.L. Brown: Speech-based interaction with in-vehicle computers: The effect of speech-based e-mail on drivers’ attention to the roadway, Human Factors, Vol. 43, No. 4, pp. 631–640, 2001.
[10] D. Haigney and S.I. Westerman: Mobile (cellular) phone use and driving: A critical review of research methodology, Ergonomics, Vol. 44, No. 2, pp. 132–143, 2001.
[11] D.L. Strayer, F.A. Drews, and W.A. Johnston: Cell phone-induced failures of visual attention during simulated driving, J. Exp. Psychol., Vol. 9, No. 1, pp. 462–466, 2003.
[12] Y. Liang, L. Reye, and J.D. Lee: Real-time detection of driver cognitive distraction using support vector machines, IEEE Transactions on Intelligent Transportation Systems, Vol. 8, No. 2, pp. 340–350, 2007.
[13] Y. Liao, S.E. Li, W. Wang, Y. Wang, G. Li, and B. Cheng: Detection of driver cognitive distraction: An SVM based real-time algorithm and its comparison study in typical driving scenarios, IEEE Intelligent Vehicles Symposium, pp. 394–399, 2016.
[14] A. Podusenko, V. Nikulin, I. Taney, and K. Shimohora: Comparative analysis of classifiers for classification of emergency braking of road motor vehicles, Algorithms, Vol. 10, No. 4, 129, 2017.
[15] S. Tsukuda, Y. Shiozawa, H. Mouri: Proposal of advanced emergency braking system adapted to the road surface condition, Proceedings of Fast-Zero 2017, TuA-A1-2, 2017.
[16] I. Taney and K. Shimohora: XML-based genetic programming framework: Design philosophy, implementation, and applications, Artificial Life and Robotics, Vol. 15, No. 4, pp. 276–280, 2010.
[17] I. Taney and K. Shimohora: XGP: XML-based genetic programming framework, Proceedings of the 34th Symposium of the Society of Instrument and Control Engineers (SICE) on Intelligent Systems, pp. 183–188, 2007.
[18] B.W. Matthews: Comparison of the predicted and observed secondary structure of T4 phage lysozyme, Biochimica et Biophysica Acta: Protein Structure, Vol. 405, No. 2, pp. 442–451, 1975.
[19] R. Yusuf, A. Podusenko, I. Taney, and K. Shimohora: Recogni-
tion of mistaken pedal pressing based on pedal pressing behavior by using genetic programming, *Proceedings of the 2018 International Conference on Internet of Things and Intelligence System (IoTaIS 2018)*, pp. 104–108, 2018.

---

**Rahadian YUSUF** (Member)

He received his B.E. and M.E. degrees in Electrical and Informatics Engineering from Bandung Institute of Technology and Ph.D. of Engineering degree from Doshisha University, in 2005, 2011, and 2016, respectively. In 2016, he joined Doshisha University, Japan. His research interests include affective computing, human-system interaction, evolutionary systems, and socio-informatics.

**Ivan TANEV**

He received his M.S. (with honors) and Ph.D. degrees from Saint-Petersburg State Electrotechnical University (Russia) in 1987 and 1993, and Dr.Eng. from Muroran Institute of Technology (Japan) in 2001. In 2004, he joined Doshisha University (Japan), where he is currently a Professor at the Department of Information Systems Design, Faculty of Science and Engineering. His research interests include control systems design, artificial intelligence, evolutionary computations, and multi-agent systems.

**Katsunori SHIMOHARA** (Member)

He received the B.E. and M.E. degrees in Computer Science and Communication Engineering and the Doctor of Engineering degree from Kyushu University, Fukuoka, Japan, in 1976, 1978 and 2000, respectively. From 2001 to 2006, he was Director of the Network Informatics Laboratories and the Human Information Science Laboratories, Advanced Telecommunications Research Institute (ATR) International, Kyoto, Japan. He is currently a Professor at the Department of Information Systems Design, Faculty of Science and Engineering, and the Graduate School of Science and Engineering, Doshisha University, Kyoto, Japan. His research interests include human communication mechanisms, evolutionary systems, human-system interactions, and socio-informatics.