Driving Event Detection using 1D CNN

Potchara Vinitwattanakoon[1*] Ekarat Rattagan[2]

Abstract – This study explore the use of 1D Convolution Neural Network (CNN) to predict occurred driving event from motion sensor data recorded by mounted smartphone inside a moving car. Contrary to most state-of-the-art algorithm for Time Series Classification (TSC) which use a 2D CNN architecture such as ResNet or FCN that often require a large amount of training data, a compact 1D CNN could have an advantage over 2D CNN where the labeled training data is scarce due to its less complexity in term of model architecture which is what this study want to explore. Since deep learning algorithm often require a large amount of datasets during training process, when dealing with the small datasets, the data augmentation is one of the method to increase the availability of training data and could help improve the performance of the algorithm, so this study will also explore, implement, and evaluate the use of data augmentation that is applied in the preprocessing step.

Keywords: Time series classification, Deep learning, Convolution neural network, Driving event detection, Motion sensor data, Data augmentation

[*]Corresponding author. E-mail: potchara.vin@stu.nida.ac.th

[1]Graduate Students at Business Analytics and Data Science Graduate School of Applied Statistics National Institute of Development Administration

[2]Assistant Professor Dr. at Business Analytics and Data Science Graduate School of Applied Statistics National Institute of Development Administration

1. Introduction

With road accident in Thailand on the rise (The Office of Transport and Traffic Policy and Planning of Thailand, 2019) along with the amount of motorist that kept increasing each year has caused the numerous damage to properties and lives. While most of the accidents are caused by distracted or inattentive driving, some are caused by driver's aggressiveness driving. Study has pointed out that one of the key-trait that associate with road accident is aggressive driving (American Automobile Association, 2009), of which consists of aggressive breaking, aggressive accelerating, aggressive turning, and aggressive lane changing which are the events that could occur while driving. However, with the help of machine learning algorithm that can learn to detect aggressive driving event from the pattern of data collected from motion sensor device, it is possible to automate the aggressive driving event detection in real-time and in turn could be implemented as a way to deter aggressive driving. The automate driving monitoring system could also be used to incentivize safe driving and reward good driving behavior in the form of road tax rebate or use as a metric to evaluate driver workforce in transportation service which in turn could help reduce the amount of road accidents.

Though there are many studies that propose and implement a way to use machine learning to classify events from time series data that is recorded during driving, such as the study that explore the use of 2-Dimension (2D) CNN deep learning to classify driving behavior style of driver that use data from On-Board Diagnostic (OBDII) such as speed, acceleration, throttle position to classify the driving style (Mohammad Shahverdy et al., 2020). While other study explores the use of classical machine learning and multilayer perceptron (MLP) to classify driving style using data gathered from motion sensor device within modern smartphone device (Jair da Silva Ferreira Junior, 2017). Not many has explored the use of 1-Dimension (1D) CNN to classify driving event yet. In the case where training data is scarce or low-cost computation is required for real-time implementation purpose, 1D CNN could prove to have an advantage over 2D CNN or classical ML algorithm due to its less complexity when compared to 2D CNN while retaining the ability to detect and extract features hidden inside the time series data due to the nature of deep learning that have hidden layer. Although, the 2D CNN TSC such as Residual Network (ResNet) or Fully Convolution Neural Network (FCN) (Zhiguang Wang, et al., 2016) have proven to be a very strong baseline and high-performance classifier for TSC (Hassan Ismail Fawaz, et al., 2019), in the environment where labeled training data is limited, the 1D CNN could have a comparable performance to the 2D CNN counterpart, which is the main topic of this study. This study will also explore the use of data augmentation that occur in the data preprocessing step to increase the size of training data to achieve higher accuracy.

2. Motivation

To be able to classify and detect aggressive driving event that has occurred while driving as driving monitoring system and could use those data to incentivize safe driving practice and reward good driving behavior through road tax rebate and/or use as a metric to evaluate driver that work in transportation service with the goal to promote safer driving and reduce road accident by the use of data gathered from motion sensor that locate inside modern smartphone device or modern OBDII device that has built-in motion sensor. The data include acceleration in 3 axis, gyroscope data in 3 axis, magnetometer data in 3 axis, with the total of 9 features to be use as an input data.

3. Related Work

Jair da Silva, Jr. Ferreira et al. (2017) have explore the use of both classical ML and neural network (MLP) to predict occurred driving event using a mounted smartphone that record motion sensor data during simulated driving session. The study has concluded that one of the key factor for good performance algorithm to achieve good accuracy score is to have a large sliding window.

Mohammad Shahverdy et al. (2020) have explore the use of 2D CNN to predict driving behavior (normal, drowsy, drunk, distracted, and normal using recurrence plot to convert time series data (acceleration, throttle position, speed, engine RPM) into image-like 2D data structure for 2D CNN classifier using a recurrence plot.

Ismail Fawaz, H., et al. (2019) have review multiple TSC algorithm and compare their performance on each UCR/UEA dataset achieve. The study then ranked each algorithm using a critical difference diagram and conclude that the end-to-end deep learning algorithm such as ResNet and FCN are the current state-of-the-art TSC algorithm. Due to the nature of deep learning model being a black box hence making them uninterpretable, a Class Activation Map (CAM) visualization could be used to highlight which part of the input data the model used to identify which class it belongs to.

Terry Um et al. (2017) have explore the use of various data augmentation method to boost the performance of TSC algorithm that use wearable sensor data to monitor Parkinson’s Disease patient in order to determine the dosage for each patient. Due to limitation of the amount of data that require an expert within the field to identify which class the data belongs to, many augmentation method were used on the dataset and was evaluate and compare. Study then conclude that while data augmentation can be used on TSC dataset when the amount of data is limited, only some data augmentation method can be used and require a good domain knowledge within the field to select a proper data augmentation method.

Zhiguang Wang et al. (2016) have propose a strong baseline for DL TSC algorithm. Considered to be one of the state-of-the-art algorithm called Fully Convolution Neural Network (FCN) that consist of 2D CNN layer with batch normalization, and global average pooling layer while also explore the use of GRAD-CAM to monitor which region the algorithm use to decide what class label the data is.

Serkan Kiranyaz et al. (2021) have review the application use of 2D CNN and compare to its newer variant, 1D CNN. In some application 1D CNN could perform better than 2D CNN in a scenario where data is scarce or when low-cost computation is required due to their minimal structure.

4. Methodology

4.1 Dataset

Data are gathered from motion sensor inside smartphone that consist of acceleration in X, Y, Z axis, gyroscope data in X, Y, Z axis, magnetometer data in X, Y, Z axis, with the total of 9 features that collected while driving a vehicle at sampling rate of 50 Hz. The dataset used in this study is obtained from Jair da Silva Ferreira Junior work in 2017. The aggressive and non-aggressive driving event are recorded as a timestamp of when the event has started and ended by assistance who sit beside the driver. The driving event type consist of aggressive accelerating, aggressive braking, aggressive left-turn, aggressive right-turn, aggressive left-lane change, aggressive right-lane change, and non-aggressive with the total of 7 types.

4.2 Data Preparation

The data, which is a time series data consist of multiple driving session with arbitrary length with the total driving time of 50 minutes depending on how long the trip is, will need to be transformed to fit within the CNN architecture that require fixed size of input will be achieved by data transformation via window slicing method with the time length of 21 as shown in **Fig. 1**.

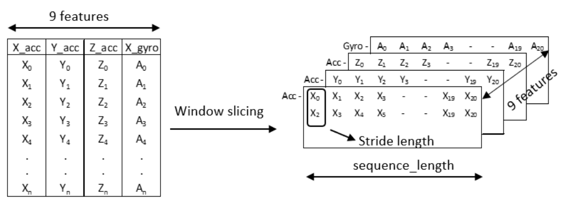

Fig. 1 Window Slicing of time series data

Time series data will be preprocessed with window slicing using method from keras called timeseries_dataset_from_array( ) with seqeunce_stride of 2, sequence_length of 21, and sampling_rate of 1, as parameters. **Fig. 2** outline approach to avoid data leakage where the window sequence that has an area that overlapped with Test Set will be discarded when doing train-test-split using purging method (Marcos de Prado, 2018).

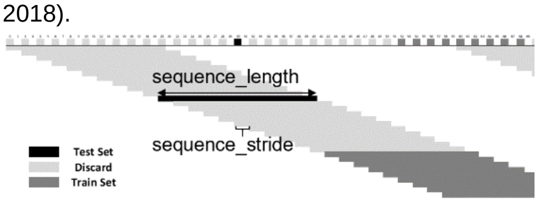

Fig. 2 Train-Test-Split for Sliced window with purging (discard)

Due to the limitation of the amount of data which consist of events count of 12 for aggressive accelerating, 12 for aggressive braking, 22 for aggressive left-turn, 11 for aggressive right-turn, 4 for aggressive left-lane change, 5 for aggressive right-lane change, and 14 for non-aggressive with total of 80 events that occurred during 50 minutes of driving, the training data after train-test-split will then be augmented, added to increase training data size, to cover the unexplored input space (Terry Um, 2017), and evaluated whether the data augmentation help improve the deep learning model performance or not.

class	Event	Count
0	non-aggressive event	14
1	aggressive right turn	11
2	aggressive left turn	22
3	aggressive right lane change	5
4	aggressive left lane change	4
5	aggressive braking	12
6	aggressive acceleration	12
	total	80

Table 1 Dataset event count

The data augmentation methods that were chosen from Terry Um's study in 2017 are scaling, time warping, and random sampling. **Scaling** (DA_Scaling) is a simple augmentation where a magnitude of the data is multiplied by a random scalar. **Time-warping** (DA_Timewarping) is another way to smoothly distort the time-length of data where some area is lengthen while some is shorten. **Random sampling** (DA_RandSampling) is similar to time-warping method, but the distort of data occur in a much smaller area.
Fig. 3 summarize the use of augmented data in training set which added as additional training data. However, test set will use only the un-augmented data to preserve and maintain correct label.

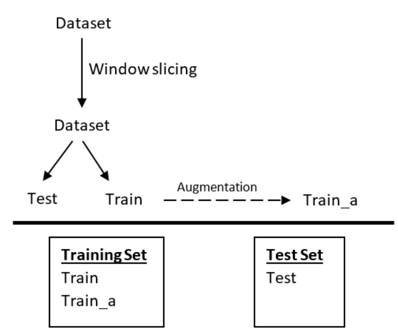

Fig. 3 Training & Test Set post data augmentation

4.3 Architecture

The first layers of the deep learning neural network usually learn the low-level features whereas later layers combine multiple low-level features from earlier layers to a more higher-level features before send to last layer that classify and predict which class label the data belong to. In this study, 1D CNN will be used for driving event classification. The architecture consist of a 1D convolution layer with rectified units (ReLUs) followed by average pooling layer with dropout layer at the end of each unit. The parameters of interests in this study consist of filter size at input layer, number of convolution layer, probability of dropout layer and finally the type of data augmentation of input data, all of which will be explored during hyper parameters tuning in searching for the combination that output the highest evaluation score.

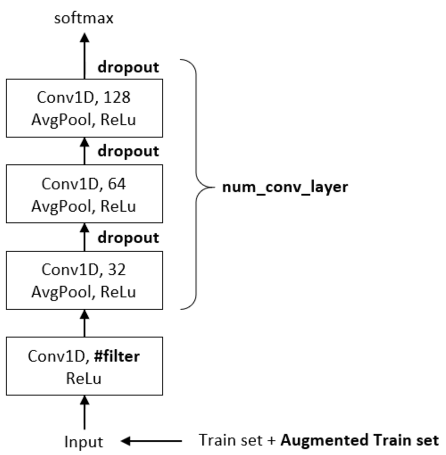

Fig. 4 1D CNN Model Summary

The output classification of driving event that occurred within sliced window of input data will use the middle point of the window as the event that has occurred and will be 1 of the 7 types of driving events that consist of aggressive accelerating, aggressive braking, aggressive left-turn, aggressive right-turn, aggressive left-lane change, aggressive right-lane change and non-aggressive.

4.4 Evaluation

Once the best combination of parameters for 1D CNN is obtained from hyper parameters tuning process, the 1D CNN model of this study will then be compared with the baseline state-of-the-art FCN algorithm proposed by Zhiguang Wang, et al. work in 2016 and a support vector machine (SVM) algorithm in the algorithm comparison process. The architecture of FCN is shown in **Fig 5**. The metric that will be used to evaluate the performance of this multiclass on each algorithm is macro average F1-score or F-measure in both hyper parameters tuning and algorithm comparison process which is calculated based on **Equation 1**.

$$\mathrm{F-measure} = \frac{2 \times \mathrm{Recall} \times \mathrm{Precision}}{\mathrm{Recall} + \mathrm{Precision}} \quad (1)$$

The process of hyper parameters tuning and algorithm comparison are described in pseudo **Algorithm 1** and **Algorithm 2** respectively.

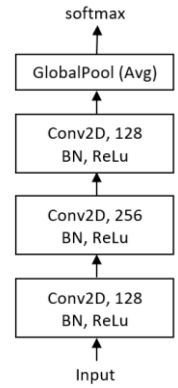

Fig. 4 FCN Model Summary that will be used as a baseline

Algorithm 1: Pseudo algorithm for hyper parameter tuning of CNN

Input : (X_{train}, Y_{train}) and (X_{test}, Y_{test})
Initialize (parameters) : batch_size = 32, kFold(k=5), epochs = 100, num_classes = 7
with dynamic epoch using early stopping callback function: patience=10, min_delta=0.00001

```
for no. of conv layer ∈ {1, 2, 3} do
 for initial filter size ∈ {32, 64, 128} do
  for augmentation method ∈ {False, RandScaling, TimeWarp, Scaling} do
   for dropout ∈ {0, 0.2, 0.5} do
    for skfold.split do
     model = Sequential( )

     model.fit( )
     model.evaluate( )
    end
   end
  end
 end
end
```

Algorithm 2: Pseudo algorithm for algorithm comparison

Input : (X_{train}, Y_{train}) and (X_{test}, Y_{test})
Initialize (parameters) : batch_size = 32, repeats = 10
epochs = 100, num_classes = 7
with dynamic epoch using early stopping callback function:
patience=10, min_delta=0.00001,

```
for augmentation method ∈ {False, RandScaling, TimeWarp, Scaling} do
  for initial filter size ∈ {32, 64, 128} do
    for algorithm ∈ {1D_CNN, FCN, SVM} do
      for repeat in range (10) do
        if algorithm == 1D_CNN then:
          for dropout ∈ {0, 0.2, 0.5} do
            model = Sequential( )

            model.fit( )
            model.evaluate( )
          end
        else:
          model = Sequential( )

          model.fit( )
          model.evaluate( )
        end
      end
    end
  end
end
```

5. Results

The result shown in **Table 2** that rank top 20 combination of hyper parameters has shown that scaling method of data augmentation help increase the performance of the model while other data augmentation method do nothing or lower the performance score of the model. For a small dataset in this study, 2 convolution layers is enough to obtain a good result while having 3 convolution layers is unsuitable due to insufficient training data to fine-tune the weight.

Table 3 summarized the result of algorithm comparison between this study's 1D CNN algorithm and 2 baseline namely FCN and SVM. The 1D CNN performance is comparable to FCN on a small dataset like in this study. The improvement of performance on algorithm that use scaling data augmentation method although low, is still noticeable as shown in the result from doing a repeated run of algorithm 10 times.

no. conv layers	initial filter	dropout	Augmentation Method	Mean		
				Precision	Recall	F1-Score
2	32	0.5	DA_Scaling	83.5%	88.2%	85.3%
2	32	0.2	None	84.3%	87.1%	85.0%
2	32	0	None	84.5%	85.1%	84.6%
3	32	0.2	None	83.8%	86.1%	84.2%
3	64	0.2	DA_Scaling	83.4%	86.1%	84.1%
1	32	0.5	None	82.9%	86.0%	84.0%
1	64	0	DA_Scaling	82.9%	84.6%	83.6%
2	64	0.5	DA_Scaling	81.8%	87.1%	83.5%
1	64	0.5	DA_Scaling	82.5%	85.6%	83.4%
1	32	0.5	DA_Scaling	82.8%	85.2%	83.3%
1	32	0.2	None	83.8%	83.5%	83.3%
1	64	0.2	DA_Scaling	83.9%	83.7%	83.2%
1	32	0.2	DA_Scaling	82.1%	85.6%	83.2%
3	128	0.2	DA_Scaling	83.0%	83.9%	82.9%
1	64	0.5	DA_TimeWarp	81.4%	85.2%	82.8%
3	32	0.2	DA_Scaling	82.5%	84.4%	82.6%
0	128	0	None	81.7%	85.7%	82.6%
2	32	0.2	DA_Scaling	81.7%	85.6%	82.6%
0	128	0.2	DA_Scaling	81.4%	86.9%	82.5%
3	32	0.2	None	81.9%	84.9%	82.5%

Table 2 Hyper Parameters Tuning Result of 1D CNN

Algorithm	no. conv layers	initial filter	dropout	Augmentation Method	Mean		
					Precision	Recall	F1-Score
1D CNN	2	32	0.2	DA_Scaling	83.2%	85.6%	83.9%
1D CNN	2	32	0.2	None	82.5%	84.7%	83.1%
1D CNN	2	32	0.2	DA_TimeWarp	82.6%	84.6%	83.0%
1D CNN	2	32	0.5	DA_Scaling	80.8%	87.0%	82.5%
FCN	-	-	-	None	82.0%	85.7%	82.1%
FCN	-	-	-	DA_Scaling	82.2%	83.2%	81.9%
1D CNN	2	32	0.5	None	80.6%	85.0%	81.8%
1D CNN	2	32	0	None	81.3%	82.8%	81.5%
FCN	-	-	-	DA_RandSampling	80.9%	84.3%	81.0%
1D CNN	2	32	0	DA_TimeWarp	80.0%	82.6%	80.3%
1D CNN	2	32	0.2	DA_RandSampling	79.6%	81.7%	80.1%
1D CNN	2	32	0	DA_Scaling	80.3%	81.5%	80.0%
SVM	-	-	-	DA_Scaling	78.6%	81.9%	79.8%
1D CNN	2	32	0	DA_RandSampling	78.1%	79.7%	78.2%
FCN	-	-	-	DA_TimeWarp	77.0%	82.0%	78.1%
1D CNN	2	32	0.5	DA_TimeWarp	76.3%	82.7%	77.9%
SVM	-	-	-	None	74.9%	81.8%	77.0%
1D CNN	2	32	0.5	DA_RandSampling	74.1%	83.0%	76.9%
SVM	-	-	-	DA_TimeWarp	71.7%	74.1%	72.4%
SVM	-	-	-	DA_RandSampling	69.8%	79.7%	72.1%

Table. 3 Algorithm Comparison Result between 1D CNN and other baseline algorithm (FCN and SVM)

6. Conclusion

For small TSC dataset, a smaller model such as 1D CNN in this study that has a lower trainable parameter could have a performance comparable to the state-of-the-art algorithm and with the help of a right data augmentation method, the performance could be improved further especially in the setting where labeled data is scarce since data augmentation can help increase the amount of training data and help the algorithm to explore all the input space while training.

References

American Automobile Association. (2009). Aggressive Driving: Research Update April 2009. American Automobile Association Foundation for Traffic Safety, Washington, DC (2009)

de Prado, Marcos., (2018), Advances in Financial Machine Learning. John Wiley & Sons Inc. Chapter 7, ISBN10 1119482089.

Ferreira, Jair da Silva, Jr. Git-Hub. Retrieved 4 January 2021 from https://github.com/jair-jr/driverBehaviorDataset (2021).

Ferreira, Jair, Jr. (2017). Driver behavior profiling: An investigation with different smartphone sensors and machine learning. PLoS ONE 12(4): e0174959. https://doi.org/10.1371/journal.pone.0174959

Ismail Fawaz, H., Forestier, G., Weber, J. et al. (2019). Deep learning for time series classification: a review. Data Min Knowl Disc 33, 917–963 (2019). https://doi.org/10.1007/s10618-019-00619-1

Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., Inman, Daniel, J., (2021) 1D convolutional neural networks and applications: A survey, Mechanical Systems and Signal Processing, Volume 151, 2021, 107398, ISSN 0888-3270, https://doi.org/10.1016/j.ymssp.2020.107398.

Osafune, T., Takahashi, T., Kiyama, N. et al. (2016). Analysis of Accident Risks from Driving Behaviors. International Journal of Intelligent Transportation Systems Research 15(3).

Schuh, Günther., Reinhart, Gunther., Prote, Jan-Philipp., Sauermann, Frederick., Horsthofer, Julia., Oppolzer, Florian., Knoll, Dino. (2019). Data Mining Definitions and Applications for the Management of Production Complexity. Procedia CIRP, Volume 81, 2019, Pages 874-879, ISSN 2212-8271, https://doi.org/10.1016/j.procir.2019.03.217.

Shahverdy, Mohammad., Fathy, Mahmood., Berangi, Reza., Sabokrou, Mohammad. (2020). Driver behavior detection and classification using deep convolutional neural networks. Expert Systems with Applications Volume 149, 1 July 2020, 113240

The Office of Transport and Traffic Policy and Planning of Thailand. (2019). Road Accident 2018 Situation Report of Thai Ministry of Transport. Department of Safety Improval, Bureau of Safety

Um, Terry., Pfister, Franz., Pichler, Daniel Christian., Endo, Satoshi., Lang, Muriel., Hirche, Sandra., Fietzek, Urban., Kulic, Dana. (2017). Data augmentation of wearable sensor data for parkinson’s disease monitoring using convolutional neural networks. 216-220. 10.1145/3136755.3136817.

Wang, Z., Yan, W., Oates, T. (2016). Time Series Classification from Scratch with Deep Neural Networks: A Strong Baseline. 578-1585. 10.1109/IJCNN.2017.7966039.