Contents lists available at ScienceDirect

Accident Analysis and Prevention

journal homepage: www.elsevier.com/locate/aap

Safety critical event prediction through unified analysis of driver and vehicle volatilities: Application of deep learning methods

Ramin Arvin [a], Asad J. Khattak [a,*], Hairong Qi [b]

[a] Department of Civil and Environmental Engineering, The University of Tennessee, United States
[b] Department of Electrical Engineering and Computer Science, The University of Tennessee, United States

ARTICLE INFO

Keywords:
Deep Learning
CNN
LSTM
Volatility
SHRP2
Crash prediction
Naturalistic driving study
Neural Network
Big Data

ABSTRACT

Transportation safety is highly correlated with driving behavior, especially human error playing a key role in a large portion of crashes. Modern instrumentation and computational resources allow for the monitorization of driver, vehicle, and roadway/environment to extract leading indicators of crashes from multi-dimensional data streams. To quantify variations that are beyond normal in driver behavior and vehicle kinematics, the concept of volatility is applied. The study measures driver-vehicle volatilities using the naturalistic driving data. By integrating and fusing multiple real-time streams of data, i.e., driver distraction, vehicular movements and kinematics, and instability in driving, this study aims to predict occurrence of safety critical events and generate appropriate feedback to drivers and surrounding vehicles. The naturalistic driving data is used which contains 7566 normal driving events, and 1315 severe events (i.e., crash and near-crash), vehicle kinematics, and driver behavior collected from more than 3500 drivers. In order to capture the local dependency and volatility in time-series data 1D-Convolutional Neural Network (1D-CNN), Long Short-Term Memory (LSTM), and 1DCNN-LSTM are applied. Vehicle kinematics, driving volatility, and impaired driving (in terms of distraction) are used as the input parameters. The results reveal that the 1DCNN-LSTM model provides the best performance, with 95.45% accuracy and prediction of 73.4% of crashes with a precision of 95.67%. Additional features are extracted with the CNN layers and temporal dependency between observations is addressed, which helps the network learn driving patterns and volatile behavior. The model can be used to monitor driving behavior in real-time and provide warnings and alerts to drivers in low-level automated vehicles, reducing their crash risk.

1. Introduction

In 2016, around 7.27 million vehicle crashes were recorded across the United States which lead to 37,914 fatalities and more than 2.17 million injuries (Federal Highway Administration, 2018), while human error is the leading cause of crashes, with a contribution in 94% of crashes (Anon 2008). Although the occurrence of a crash is an outcome of several factors, these statistics suggest that researchers should provide great attention to the human behavioral aspect of crashes, however the literature mainly focuses on roadway and infrastructure factors. In addition, conventional data sources including police-reported crashes, are the major source of the literature and these sources suffer from under-reported crashes. Based on the report by National Highway Traffic Safety Administration (NHTSA) (NHTSA 2009), 50% of property damage only (PDO) crashes and 25% of minor injury crashes are not reported to the police and are thus not recorded. Also, these crashes may be truncated due to state's monetary threshold (Hauer 2006). Given all these limitations, the police-reported data contains limited information on the pre-crash events, vehicular movements, driver state and decision, and maneuvers.

With the emergence of new sources for data collection, high-frequency naturalistic driving data is emerging which enables researchers to conduct in-depth crash analyses with the incorporation of instantaneous driving decisions and behavior prior to near-crash and crash involvement. Furthermore, these datasets contain detailed information not only on PDO and minor crashes, but also on near-crash events where a critical event happened but did not lead to a crash. This information can be coupled with driver behavior and vehicular movements to aid in the real-time prediction of a crash or near crash occurrence to potentially prevent it. In this context, since the prediction accuracy is vital, the desirable outcome is a model that can accurately predict crash risk before its occurrence.

* Corresponding author.
E-mail address: akhattak@utk.edu (A.J. Khattak).

https://doi.org/10.1016/j.aap.2020.105949
Received 9 August 2020; Received in revised form 12 November 2020; Accepted 9 December 2020
Available online 29 December 2020

Referring to the methodological standpoint, several studies have studied the correlates of pre-crash driving behavior, roadway, and environmental factors on crash risk and severity utilizing a traditional statistical approach. While these methods are very beneficial by providing insights regarding the association of factors, they usually suffer from low accuracy outside of the sample data and prediction. Therefore, in the context of real-time crash warning generation, other supervised methods are needed to perform better in terms of accuracy. Furthermore, due to high dimensionality of the data, traditional statistical methods might not be appropriate in this context.

Deep learning methods have recently gained significant attention in the literature due to their promising performance in several fields. In this context, the Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) are mainly utilized to process visual-related and time-series problems. With the recent improvements in the CNN and RNN (LeCun et al. 1998, Simard et al. 2003, Ciresan et al. 2011), and emergence of large-scale data integrated with efficient implementation of computational powers (i.e. graphics processing units (GPUs)), they outperformed not only conventional methods but also human performance (Sermanet and LeCun 2011).

The main contribution of this study is the development of a deep learning framework which integrates multiple data streams including vehicular kinematics in terms of speed, longitudinal and lateral accelerations, driving instability, and driver behavior to predict the occurrence of a crash/near-crash (CNC). The framework can process micro-level driving data, extract volatility features and learn driving patterns and behavior to quantify crash risk which can have a substantial impact on the field. The developed framework has several advantages:

1 The architecture configuration of the model is compact, making the model easy to be implemented for real-time safety performance monitoring and failure detection.
2 Its ability to capture temporal variations in the input data generated from multiple sensors.
3 The capability of the model to efficiently train the model using limited training dataset and back-propagation iterations (Eren et al. 2019).

With the emergence of new data sources, this study is timely and original as it harnesses this big data and incorporates it in the instantaneous driving behavior analysis by developing a deep learning framework to warn drivers regarding the risk of crash involvement. The compact configuration of the developed model would help agencies easily implement it in real-time applications.

2. Literature Review

Several studies have explored the contribution of driver behavior, vehicle factors, roadway, and environmental characteristics on the probability of crash using statistical methods (Dingus et al. 2011, Arvin et al. 2019b, Arvin and Khattak, 2020, Khattak et al., 2021). Although most of this research relies on police-reported data, it provides insightful inferences regarding the association of driving behavior and crash risk. The emergence of naturalistic driving data and high-resolution driving decisions has allowed for the exploration of microscopic driving behavior prior to crash occurrence. In our previous research (Arvin et al. 2019b, Kamrani et al. 2019), we have shown that instability in driving not only increases the likelihood of a crash involvement but also severity of a crash.

On the other hand, deep learning methods have recently received lots of attention due to the emergence of big data which is generated by multiple sources and the availability of computational power. It has been shown that deep learning methods are a great tool for representation learning as it requires little effort for manual feature extraction (Goodfellow et al. 2016). Referring to the transportation field, deep learning and reinforcement has applied to several areas including macroscopic traffic conflict prediction (Zeng et al. 2016, Parsa et al. 2019, Yuan et al. 2019, Cai et al. 2020, Formosa et al. 2020, Li et al. 2020), transportation planning (Cai et al. 2019), demand prediction (Lin et al. 2018, Xu et al. 2018a, Bao et al. 2019), network assignment (Xu et al. 2019), transportation maintenance (Wei et al. 2019), traffic surveillance and congestion detection (Li et al. 2019, Cui et al. 2020, Ding et al., 2020, Nasr Esfahani and Song, 2020), travel time prediction and reliability (Ghanim and Abu-Lebdeh 2015, Tang et al. 2019b, Khadhir et al. 2020), behavior prediction (Liu and Shi 2019, Osman et al. 2019, Tang et al. 2019a, Zhang et al. 2020), signal control (Jeon et al. 2018, Xu et al. 2018b, Gong et al. 2020), driver impairment detection (Ye et al. 2017, de Naurois et al. 2018), and vehicle classification (Nezafat et al. 2019). The main advantage of deep learning architecture over traditional statistical methods is the ability to model complex non-linear relations between associated factors and a dependent variable by incorporating distributed and hierarchical features (Ma et al. 2015).

Referring to the micro-level analysis of crash risk, few studies have attempted to identify crash risk level in a real-time manner. Shi et al (Shi et al. 2019) performed a Discrete Fourier Transform and performed XGBoost and K-mean to detect critical events. Kluger et al. (Kluger et al. 2016) performed a Discrete Fourier Transform and K-means clustering on longitudinal acceleration to detect critical events on a sample of 49 crashes and 42 near-crashes. Perez et al (Perez et al. 2017) utilized thresholds to identify boundaries for the detection of crash/near-crash events. Gao et al. (Gao et al. 2018) predicts the longitudinal conflicts between vehicles with CNN using vehicle kinematics and front-camera videos. However, their analysis only captures a commercial truck fleet, and the results might not be generalizable to other drivers and vehicle types. Osman et al. (Osman et al. 2018) tried to predict the safety critical events based on the vehicle kinematics information using multiple machine learning approaches. From the methodological standpoint, it seems that their method cannot capture the complexity embedded in the data, which can potentially be improved by Deep Learning methods.

After reviewing the literature, several gaps were found. First, the previous studies mainly incorporate raw vehicular movements in the analysis, while ignoring driver behavior and instability in driving. Second, the temporal nature of the dataset is ignored, and simple machine learning or neural network models are used. Finally, the proposed models might not perfectly capture the non-linear relationships between the input and output of the model. This study tries to address the aforementioned gaps by utilizing several deep learning methods including 1D-CNN, LSTM, and 1DCNN-LSTM models to capture time dependency between observations and incorporate driver behavior and instability in driving by utilizing the seconds of distracted driving and concept of driving volatility, respectively. Furthermore, the developed models can be used for real-time crash prediction and providing warnings to the drivers.

3. Data description and pre-processing

3.1. Data description

This study utilized the second Strategic Highway Research Program (SHRP2) naturalistic driving study data. The original data contains more than 4 petabytes of information, which known as the most comprehensive driving study. The data collection was performed from 2010 to 2013 and contains high-quality and high-resolution data from six states. The data contains information of 3500 drivers with more than 50 million vehicles miles traveled (Hankey et al. 2016). For the data collection, onboard data acquisition system (DAS) along various sensors such as front and cabin camera, forward sensor, and accelerometers are used to obtain information such as vehicular movements at 10-Hz frequency, video of the vehicle cabin and surrounding environment, and offset from lane center. (Hankey et al. 2016). The data reduction and resampling process was conducted by the Virginia Tech Transportation Institute

(VTTI) team to resample thousands of baseline events from 5 million trips. The VTTI team extracted all near-crashes and crashes with detailed information, while 7.5 K baseline events (normal driving events) are sampled by stratifying drivers and driving times using case-cohort and case-crossover random sampling methods (Hankey et al. 2016). This paper used a subset of SHRP2 NDS dataset which contained 7566 baseline, 1307 near-crash events, and 617 crash events. For each near-crash and crash, 30 seconds of vehicular movements, and driver behavior are available. The data contains evasive maneuver seconds, which used by the driver to avoid the safety critical event, and after its occurrence.

Since this paper is predicting the critical event before its occurrence, the unintentional driving decisions is needed to be included in the analysis and intentional behavior arising from drivers' behavior to avoid these events should be excluded. Furthermore, since we are using the information on driver distraction, we need to extract the seconds that driver was distracted, which was obtained from the summary of each trip. This study combines the crash and near-crash events as both events are critical, considering that near-crashes would become a crash if not for the appropriate response of the drivers to avoid the collision. After combining crashes and near-crashes, we focused on at-fault drivers by excluding not-at-fault drivers from the analysis. These not-at-fault drivers are defined in which the crash occurrence may not be the result of their driving behavior, such as multi-vehicle crashes in which the vehicle was hit by another vehicle, or single vehicle crashes in which the vehicle was hit by an animal, falling rock, etc. It should be noted that we are predicting the CNC events using the subject vehicle movements and none of the data on surroundings, e.g., presence or distance to other vehicles, and atypical events such as animal-involved crashes and falling rocks. As a result, we excluded crashes involving not-at-fault drivers from the analysis. Therefore, the sample size of CNC events reduced from 1925 to 1315 events, while the total of 7566 baseline events remained the same.

3.2. *Exclusion of Evasive Maneuvers*

As discussed, it is vital to remove the seconds that driver is attempting to avoid a crash, since the goal of this study is to detect critical event occurrence in real-time. To elaborate further, speed and acceleration profiles of a random crash are provided in Fig. 1. According to the figure, the crash happened at the 23rd second of data, while the driver reacted to it at 22nd second. Thus, we need to not only exclude the seconds after the crash occurrence, but also exclude the seconds that driver is reacting to the stimuli. The speed and longitudinal and lateral acceleration 15 seconds before the reaction time are used to measure driving volatility (which will be discussed in the next section) which help us to quantify driving instability.

3.3. *Measures of driving volatility*

The concept of volatility is introduced to quantify variations in instantaneous driving decisions through the extraction of useful information from microscopic vehicular kinematics. In the literature, these variations are quantified through several functions that are applied to vehicle speed (Arvin et al. 2019a, Arvin et al. 2019c, b, Kamrani et al. 2019, Arvin et al. 2020, Mohammadnazar et al., 2021), longitudinal acceleration (Liu and Khattak 2016, Arvin et al. 2019c, b, Kamrani et al. 2019, Wali et al. 2019, 2020), and lateral acceleration (Arvin et al. 2019c). This study applied several volatility functions to extract additional features from the data. In general, three volatility groups are extracted:

1 Speed volatility
2 Longitudinal acceleration volatility
3 Lateral acceleration volatility

In the following, the mathematical formulation of each volatility function is provided and discussed in detail.

Exponentially Weighted Moving Average Volatility (EWMA): This

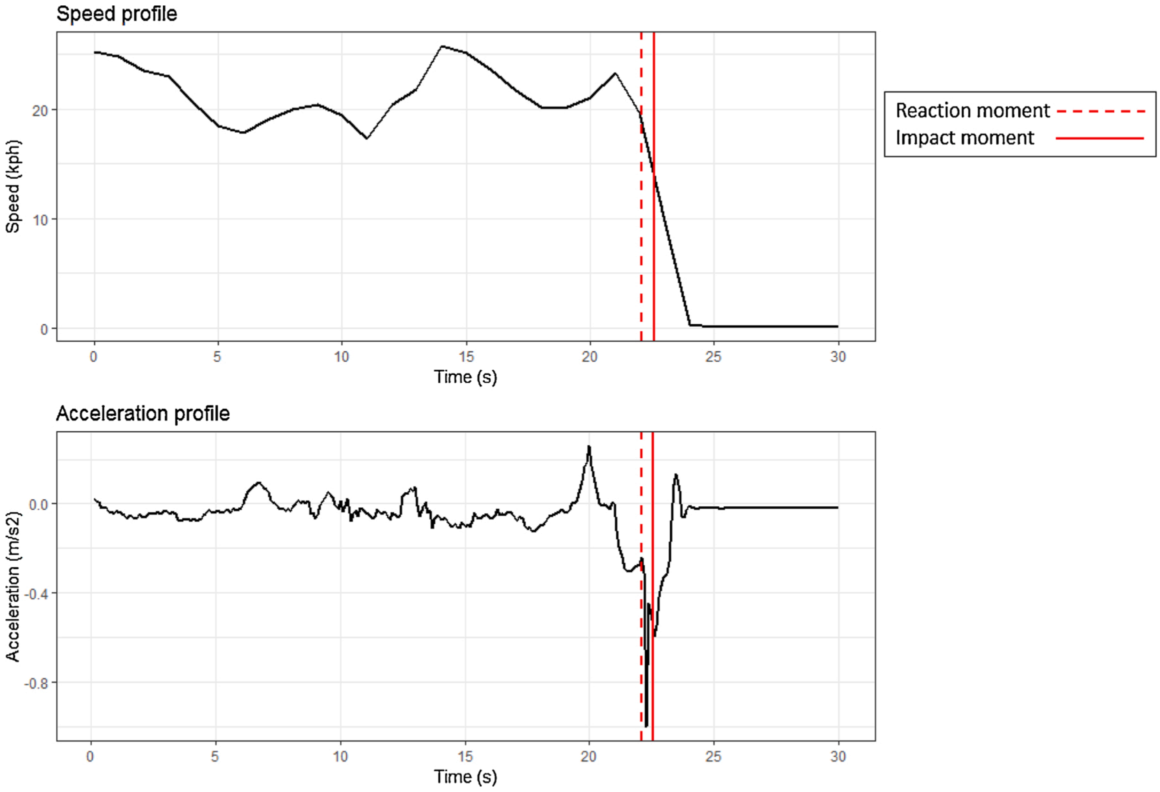

Fig. 1. Speed and acceleration profile of a random crash.

measure was introduced by RiskMetrics in 1996 (Longerstaey and Spencer 1996) which considers volatility as a weighted average of volatility observations over time. We can write (Longerstaey and Spencer 1996):

$$EWMA = \sigma_t^2 = \lambda\sigma_{t-1}^2 + (1-\lambda)\in_{t-1}^2 \tag{1}$$

where $\in_{t-1}$ is the return at time *t-1*, σ_{t-1}^2 is the EWMA volatility at time *t-1* and lambda is user defined weight (assumed 0.94 in this paper).

Time-varying stochastic volatility: which quantifies dispersion in the vehicular movements by considering changes in the ratio of observations. We can write (Figlewski 1994):

$$V_f = \sqrt{\frac{1}{n-1}\sum_{n}^{i=1}(r_i - \bar{r})} \qquad \text{from } t = 1 \text{ to } n \tag{2}$$

where V_f denotes the time-varying stochastic volatility, n is number of observations, and r_i is:

$$r_i = \ln\left(\frac{x_t}{x_{t-1}}\right) \tag{3}$$

where *ln* is natural logarithm, x_t and x_{t-1} are the observations at t and $t-1$, respectively. Since this measure requires a positive time-series input, it is only applied to vehicle speed.

Mean absolute deviation: which quantifies variations in the data by measuring the distance between observations and their central tendency (mean in this paper). We can write (Huber 2005):

$$MAD = \frac{1}{n}\sum_{n}^{i=1}|x_i - \bar{x}| \tag{4}$$

Coefficient of variation: which calculates the ratio of standard deviation over average (Everitt and Skrondal 2002):

$$C_v = \frac{S_{dev}}{|\bar{x}|} \tag{5}$$

Standard Deviation: which is the most common and basic approach to quantify dispersion in the data. We can write:

$$S_{dev} = \sqrt{\frac{1}{n-1}\sum_{n}^{i=1}(x_i - \bar{x})^2} \tag{6}$$

where x_i and $\bar{x}$ denotes the observed value i and the average of observations.

Next, each volatility measure is calculated at two levels: Level 1: Event-based volatility, Level 2: Temporal driving volatility. Event-based volatility applies the functions on 150 deci-second observations and

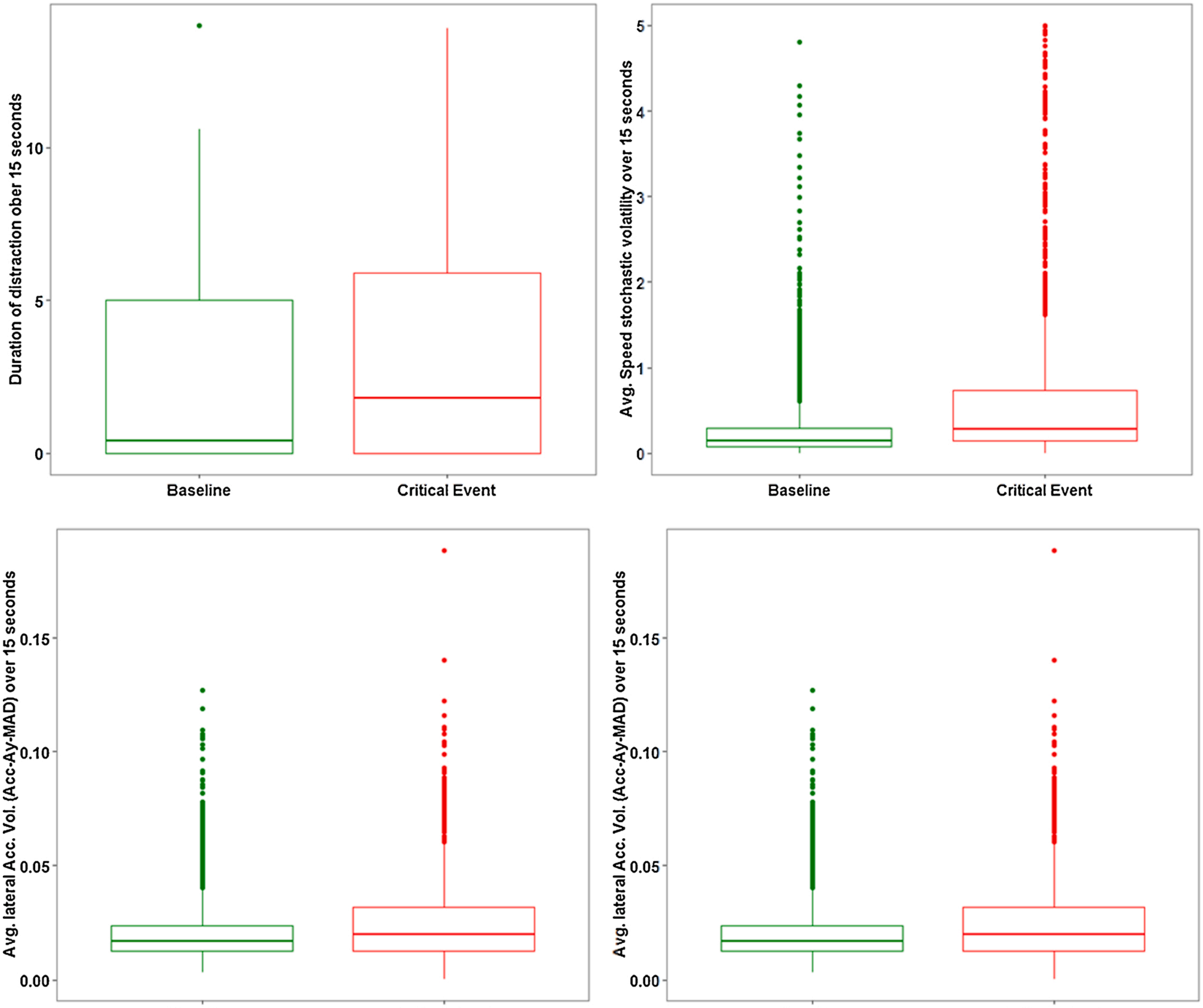

Fig. 2. Boxplot of distracted driving, speed, longitudinal and lateral volatilities for the baseline and critical events.

returns one value as a volatility measure. On the other hand, temporal volatility utilizes the concept of a moving average and applies a 3-second (30 deci-second) time-window to calculate temporal volatility for each second of driving and generates a time-series volatility profile.

3.4. Concept illustration and descriptive statistics

This section provides some statistical analysis to illustrate the positive association of driving volatility and distracted driving on the crash risk. The previous sections have discussed the procedure for calculating event-based and temporal driving volatility for speed and longitudinal, and lateral acceleration. Here, the contribution of the volatility on the crash risk is shown using a boxplot analysis, provided in Fig. 2. It can be noted that there is a substantial difference in these two groups. The analysis of variance (ANOVA) tests whether the difference between means is statistically significant (at 95% level) or not. The ANOVA test results show that with 99.99% confidence, there is a difference between the means of baseline and safety critical events. It can be inferred that in critical events, drivers were more distracted and volatile compared to the baseline events.

The descriptive statistics of the variables are provided in Table 1. The feature space contains information on three dimensions: 1- Seconds that the driver was distracted with a secondary task, 2- Event-based, and 3- Temporal driving volatility indices for speed, longitudinal, and lateral accelerations. It can be observed that the seconds of distraction and the driving volatility is significantly higher in the critical events compared with baselines.

4. Technical Approaches

4.1. Conceptual Framework

Fig. 3 provides the conceptual framework of the study. It has three main phases. The first phase is sensing, which collects driver information (i.e. in terms of distraction) and vehicular movements (i.e. speed, longitudinal and lateral acceleration). As discussed in the previous section, the data is preprocessed and cleaned by excluding the evasive maneuvers of critical events and considering 15 seconds for each event. In the second phase, the raw data is fed to the feature extraction phase to obtain volatility indices at the event and temporal levels. Fifteen volatility indices are extracted to quantify speed and longitudinal and lateral acceleration variations. Finally, the raw data and extracted features are fed to the deep-learning phase. Deep NN, 1D-CNN, LSTM RNN, and 1DCNN-LSTM models are developed to classify events as either baseline or critical event and evaluate the performance of the models.

4.2. Problem formulation

As discussed, three deep-learning methods are utilized to classify events: Deep Neural Network, 1D-Convolutional Neural Network (1D-CNN), Long Short-Term Memory (LSTM), and 1DCNN-LSTM model. While the multi-layer deep neural network processes the input data and information through interconnected neurons, it suffers from the limitation of the base assumption that all inputs are independent from each other, which is not the case in different fields, such as image classification, language processing, and time-series problems. Therefore, several methods are proposed to address the dependency in the input of the network (in this paper time dependency) by including local information (temporal information) in the input data.

4.2.1. Deep Neural Network (DNN)

A DNN model is known as a feed-forward artificial neural network with more than one hidden layer (Hinton et al. 2012a). These models process the information through a series of fully connected layers and associated with other layers through weighted connections. Each node, called a neuron, transforms the input with a non-linear function to create a decision boundary. Each neuron can be considered as a non-linear computational unit which applies activation function (e.g. sigmoid function). The neurons can be defined as:

$$a^{l+1} = f\left(W^l a^l + b^l\right) \tag{7}$$

where a^l and a^{l+1} denotes the activation value in level l and $l+1$, respectively, W^l is a weight matrix, b^l is the bias, and $f(.)$ represents the activation function. The special case is $l=1$ which denotes the input layer, and we denote it by $a^l = x$. The last layer of the DNN is a softmax

Table 1
Descriptive statistics of the baseline and critical events.

	Baseline events (N = 7566)				Critical events (N = 1315)			
Variable (feature)	Mean	SD	Min	Max	Mean	SD	Min	Max
Speed (mph)	62.36	31.22	0	125.81	38.48	29.24	0	215.41
Acceleration$_x$ (m/s^2)	−0.01	0.04	−0.23	0.25	−0.01	0.05	−0.2	0.26
Acceleration$_y$ (m/s^2)	0	0.04	−0.2	0.33	0	0.04	−0.18	0.24
Seconds of distraction	1.852	2.19	0	14.00	3.11	3.26	0	13.90
L1-Speed-S_{dev}	1.51	1.46	0	31.88	2.15	1.64	0	10.52
L1-Speed-D_{mean}	1.28	1.27	0	27.05	1.84	1.47	0	9.33
L1-Accleration$_x$-S_{dev}	0.05	0.03	0.01	0.2	0.08	0.04	0.01	0.28
L1-Accleration$_x$-D_{mean}	0.04	0.03	0	0.18	0.06	0.03	0.01	0.22
L1-Accleration$_y$-S_{dev}	0.04	0.04	0.01	0.24	0.06	0.05	0	0.4
L1-Accleration$_y$-D_{mean}	0.03	0.03	0	0.21	0.04	0.04	0	0.31
L2-Speed-V_f	0.01	0.04	0	0.68	0.03	0.06	0	0.6
L2-Speed-S_{dev}	2.68	2.32	0	96.22	3.68	2.45	0	20.06
L2-Speed-D_{mean}	2.28	1.98	0	81.91	3.13	2.08	0	16.89
L2-Speed-C_v	0.04	0.07	0	1.15	0.14	0.17	0	1.16
L2-Speed-EWMA	0.01	0.04	0	0.68	0.03	0.06	0	0.6
L2-Accleration$_x$-S_{dev}	0.02	0.01	0	0.12	0.04	0.02	0.01	0.15
L2-Accleration$_x$-D_{mean}	0.02	0.01	0	0.1	0.03	0.02	0	0.13
L2-Accleration$_y$-S_{dev}	0.03	0.01	0	0.15	0.03	0.02	0	0.23
L2-Accleration$_y$-D_{mean}	0.02	0.01	0	0.13	0.03	0.02	0	0.19

*L1: Event-based volatility measure; L2: Temporal volatility measure; S_{dev}: Standard deviation; V_f : Time-varying stochastic volatility; C_v: coefficient of variation; D_{mean}: mean absolute deviation; Acceleration$_x$: longitudinal acceleration; AccDec$_x$:both longitudinal acceleration and deceleration; Accleration$_y$: lateral acceleration; EWMA: Exponentially Weighted Moving Average Volatility.

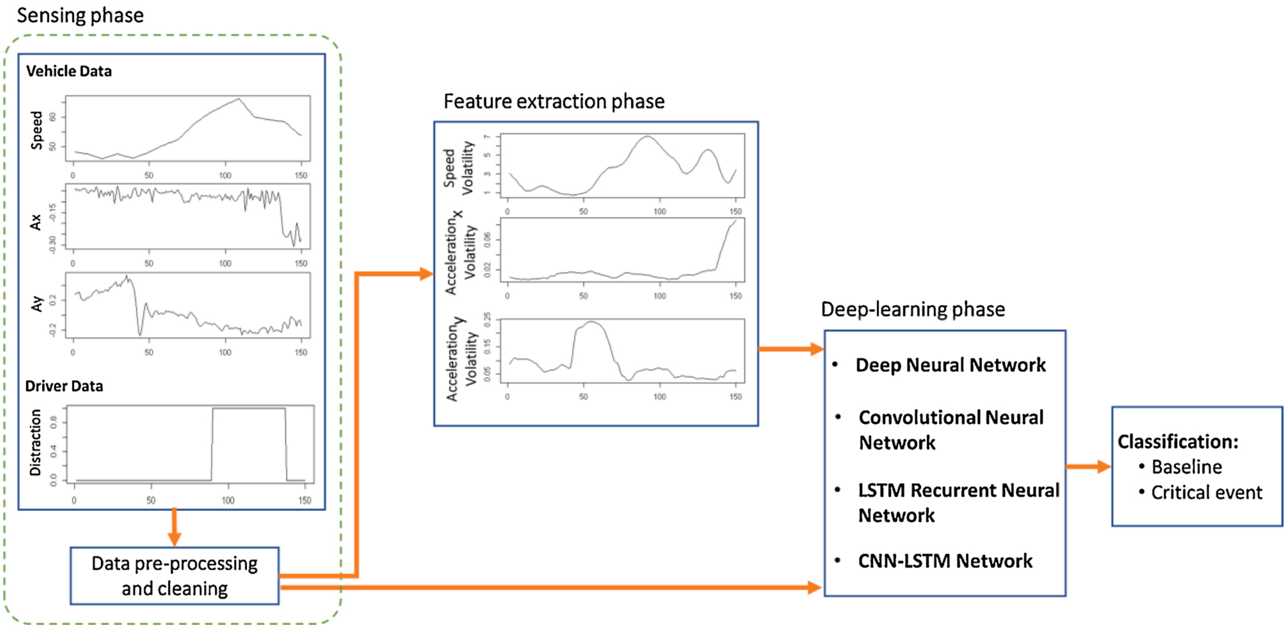

Fig. 3. Conceptual framework of the study.

classifier and the output is the two classes (i.e. Baseline and Critical events). Here, we considered a fully connected network with four dense layers including 150 nodes on each layer, following a softmax layer. While DNN models are prone to overfit the data, dropout regularization is utilized to penalize the weights. Dropout is a powerful technique for improvement in the generalization error of large NNs, introduced by Hinton et al. (Hinton et al. 2012b). The training procedure applies forward and backward propagations. While forward propagation computes actual classifications based on the input data, the aim of backward propagation is to update model parameters to minimize discrepancy between predicted and observed values.

4.2.2. 1D-CNN classifier

Compared to simple NN models that perform feature extraction by only taking a vector of inputs to the model, 1D-CNNs allow us to operate in a multi-scale manner and further investigate the time-series dependency between the observations. Fig. 4 and Table 2 illustrate the structure of the 1D-CNN model used in this study. The time-series motions of vehicles, driver distraction profile, and driving volatility measures are the model input, and the output layer classifies the output event. In the time-series data analysis, we can treat the input as a picture with the size of *(n,1)* pixels with *v* bands, where *v* is number of input streams. The convolutional layers in the model extract additional features from the data. In the convolutional layer, the model applies the convolution operation on the local input data in order to generate the corresponding 1D features, while applying different convolutions will generate several features from the input data (Song et al. 2019). In each convolutional layer of the model, 1D forward propagation is performed which can be formulized as:

$$x_k^l = b_k^l + \sum_{N_{l-1}}^{i=1} conv_{1D}\left(w_{ik}^{l-1},\ s_i^{l-1}\right) \tag{8}$$

Table 2
Structure of the 1D-CNN model.

Layer type	Output shape	Number of parameters
1D-CNN 1	(None, None, 75, 12)	492
1D-CNN 2	(None, None, 75, 12)	300
1D-CNN 3	(None, None, 100)	100100
1D-CNN 4	(None, None, 100)	100100
Dense 1	(None, 100)	10100
Dense 2	(None, 100)	10100
Dense 3	(None, 2)	202

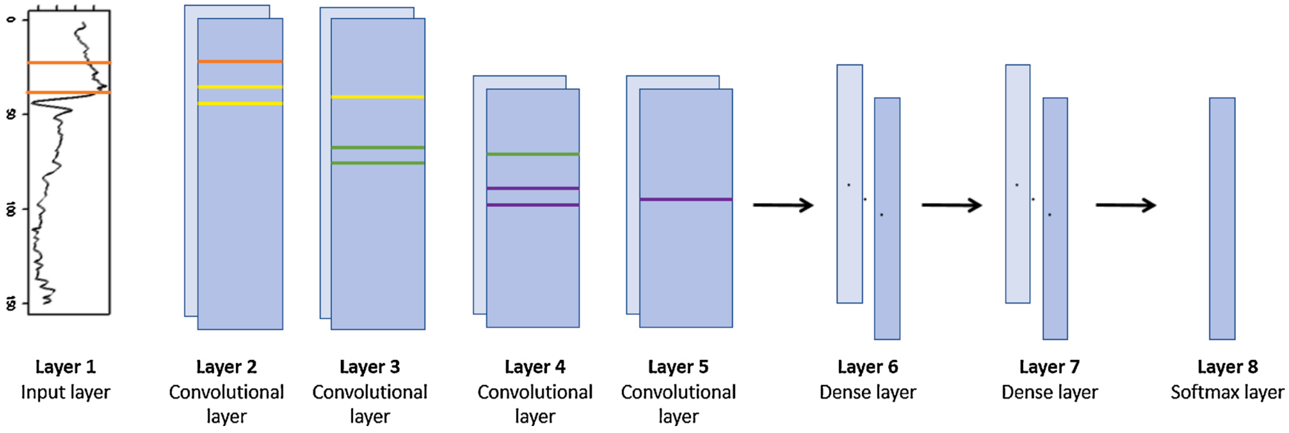

Fig. 4. Representation of 1D-CNN network used in this study.

where x_k^l and b_k^l represent the k^{th} neuron at layer l input and bias, respectively. w_{ik}^{l-1} is the kernel function in layer l-1.

Since the feature map resulting from each convolutional layer is sensitive to the position of features in the input and the number of feature maps increase the dimensionality of data, a pooling layer is utilized to down-sample the feature map. Among two common pooling methods, i.e. average-pooling and max-pooling, the latter is used to summarize the most activated node of a feature. At the end, the high-level features are flattened and fed into a fully connected network to perform classification.

While in the conventional DNN, a neuron is fully connected to neurons of a next layer, CNN structure is different as neurons are sparsely connected to each other based on their relative position. Therefore, in a DNN, values of the next layer (hidden neuron i in layer j, $h_{i,j}$) are obtained by multiplying all the neurons of the previous layer (h_{j-1}). On the other hand, the CNN model computes each activation by multiplying a subset of local input to the weight matrix (W). It should be noted that the weight matrix is shared across the entire layer which helps the model reduce the number of estimated parameters and allows for efficient training. A max-pooling layer is frequently used after a convolutional layer.

4.2.3. LSTM-RNN

Recurrent Neural Network (RNN) models aim to tackle DNN limitations by using recurrent connection in every neuron to include temporal information and feeding back this information to itself with a unit of time delay (Ordóñez and Roggen 2016). This helps the model to learn the temporal dynamics of time-series input. The RNN model requires an input sequence $a^l = (a_1^l, a_2^l, \ldots, a_T^l)$ where a_i^l is the activation of unit i at layer l, and T is the length of input sequence. By performing the following recursive equation, a sequence of activations of the next layer, $a^{l+1} = (a_1^{l+1}, a_2^{l+1}, \ldots, a_T^{l+1})$, will be obtained:

$$h_i^l = \sigma\left(W_{xh}^l a_t^l + h_{t-1}^l W_{hh}^l + b_h^l\right) \tag{9}$$

where σ denotes the activation function, W_{xh}^l and W_{xh}^l are the input-hidden and hidden-hidden weight matrix, respectively, and b_h^l is the vector of bias. The a_t^l can be obtained as following:

$$a_t^{l+1} = h_t^l W_{ha}^l + b_a^l \tag{10}$$

where W_{ha}^l is the hidden-activation weight matrix, and b_a^l is the vector of activation bias. Although RNN is designed to deal with time-series data, this model faces the problem of a vanishing and exploding gradient which can affect the model fit for the long-time lag models (Hochreiter and Schmidhuber 1997, Zhao et al. 2017) and face several challenges in real-world sequence modeling (Gers et al. 2002).

The LSTM models, introduced in 1997 by Hochreiter and Schmindhuber (Hochreiter and Schmidhuber 1997), attempt to extend the conventional RNN models which are capable of learning long-term time dependency in the input. Similar to RNN models, LSTM have the chain structure, but the difference is in the design of a neurons. While RNNs have a single learning neuron (e.g. tanh), there are three gates interacting with each other. The LSTM utilizes the concept of gating to provide a mechanism that defines the behavior of each memory cell in the network. The cell state is updated according to the gates' activations. The input data of the LSTM is fed into the write gate (input), read gate (output), and reset gate (forget). The LSTM layer can be written as:

$$i_t = \sigma_i(W_{ai}a_t + W_{hi}h_{t-1} + b_i) \tag{11}$$

$$f_t = \sigma_i\left(W_{af}a_t + W_{hf}h_{t-1} + b_f\right) \tag{12}$$

$$c_t = f_t c_{t-1} + i_t \sigma_c(W_{ac}a_t + b_c) \tag{13}$$

$$o_t = \sigma_o(W_{ao}a_t + W_{ho}h_{t-1} + b_o) \tag{14}$$

$$h_t = o_t \sigma_h(c_t) \tag{15}$$

where i_t, f_t, and o_t denotes the input, forget, and output gates, respectively, c_t represents cell activation vectors, and σ is the activation function.

In this study, the three LSTM classifier layers followed by three dense fully connected hidden layers, and softmax layer. The structure is provided in the Table 3.

4.2.4. 1DCNN-LSTM

Typically, DNN and RNN models receive the raw input data, while it has been shown that by applying feature-derived layers, their accuracy can be improved significantly (Palaz and Collobert 2015). Convolutional layers have been suggested to extract additional features from the raw data (Yang et al. 2015, Brownlee 2018). A convolution layer extracts features from the input data by applying a kernel (filter). By applying these kernels to different regions of input data, possible additional patterns are recognized and captured. It is worth noting that these kernels are optimized during the training process.

The application of a convolution layer relies on the input dimension. In the 1D context, a kernel can be considered as a filter which can removes the outliers, and the feature detector (Ordóñez and Roggen 2016). Feature map extraction using one-dimensional convolution can be written as:

$$a_j^{l+1}(\tau) = \sigma\left(b_j^l + \sum_{F^l}^{f=1} K_{jf}^l(\tau) * a_f^l(\tau)\right) \tag{16}$$

where a_j^{l+1} is the feature map j in layer $l+1$, σ is the kernel non-linear function, F^l denotes the feature maps of layer l, K_{jf}^l is the kernel mapping f to feature map j in layer $l+1$.

The structure of the 1DCNN-LSTM model is shown in Fig. 5 and Table 4. The input data is fed to the two convolution layers and the max-pooling layer is applied and to flatten the outputs. Next, the features are fed into three layers of LSTM with 100 nodes, and a dense fully connected network with a softmax classifier performs the classification.

5. Experimental evaluation

In real-world problems, highly imbalanced data is a common issue and is discussed widely in different contexts such as prediction of fraud transactions, cyber security attacks, and system failure prediction. Similarly, occurrence of a safety critical event in driving is a rare event which leads to imbalanced data. In the literature, several approaches are discussed to alleviate the issue by treating the data and methodology (Qian et al. 2014, Johnson and Khoshgoftaar 2019, Brownlee 2020). Referring to data, resampling is one of the common approaches to lessen this issue by sampling from the majority class (i.e. baselines). Referring to the methodology treatment, we need to deal with the accuracy paradox where high accuracy can be gained by model that predicts most of observations as baseline events (Brownlee 2020). Therefore, we are utilizing other performance metrics along with accuracy to score model performance both on baseline and safety critical events, such as

Table 3
Structure of the LSTM model.

Layer type	Output shape	Number of parameters
LSTM 1	(None, 150, 100)	48400
LSTM 2	(None, 150, 100)	80400
LSTM 3	(None, 150, 100)	80400
Dense 1	(None, 100)	10100
Dense 2	(None, 100)	10100
Dense 3	(None, 2)	202

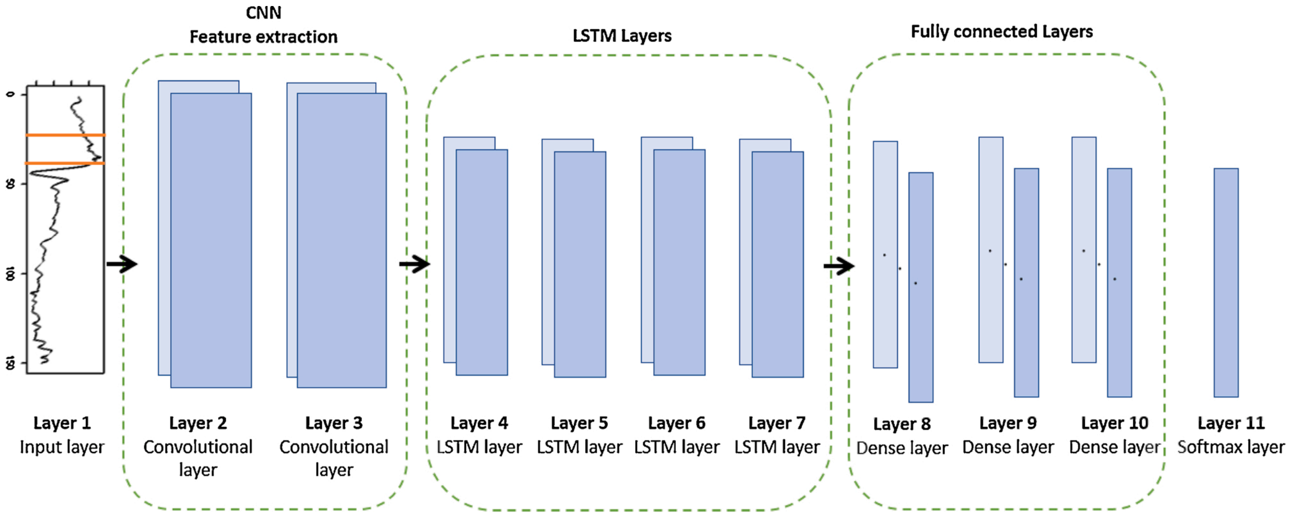

Fig. 5. Structure of the 1DCNN-LSTM model.

Table 4
Structure of the 1DCNN-LSTM model.

Layer type	Output shape	Number of parameters
1D-CNN 1	(None, None, 75, 12)	492
1D-CNN 2	(None, None, 75, 12)	300
LSTM 1	(None, None, 100)	400400
LSTM 2	(None, None, 100)	80400
LSTM 3	(None, None, 100)	80400
LSTM 4	(None, 150)	80400
Dense 1	(None, 150)	15150
Dense 2	(None, 150)	22650
Dense 3	(None, 150)	22650
Dense 4	(None, 2)	302

precision, recall, F1-score, and AUC. Finally, penalizing models using different weight regularization can be used to enhance model fitting on the minor group.

5.1. Training procedure

The dataset is randomly divided into the train and test datasets, where 70% of the data is used for training and 30% for testing. 20% of the training dataset is randomly separated for validations. All the weight matrices and bias vectors were randomly initialized to begin the training. The dropout and L2 regularization approaches are utilized to prevent overfitting of the training sample. Optimal dropout value for each model is obtained in a manner to prevent the risk of overfitting while obtaining the highest accuracy. To improve the efficiency, the data is segmented to mini-batches with a size of 256.

All the models are trained by the Adam optimizer (Kingma and Ba 2014) which is known for its efficient stochastic optimization approach with only requiring the first-order derivatives and low memory for analyzing. Moreover, it has the advantage of high computational efficiency, low memory requirement, straightforward implementation, and invariant feature to diagonal gradients rescaling, and also it is appropriate for problems that the data and parameters are large scale (Kingma and Ba 2014). Furthermore, it has the advantage of suitability problems with noisy and sparse derivatives and non-stationary objectives (Kingma and Ba 2014). The Adam optimizer is utilized to minimize the loss of objective function, which we used cross entropy function, by finding the optimal weights and bias terms.

In order to perform training, the dataset is randomly divided into train and test datasets. This study took advantage of the Keras and TensorFlow deep learning tools. Keras and TensorFlow are open-source python libraries developed by the Google Brain Team (Chollet 2015, Abadi et al. 2016), which are widely used in the deep learning context by several studies (Baghbaderani and Hairong 2019, Nezafat et al. 2019, Baghbaderani et al. 2020). The model training and classification were run on a workstation computer with the TITAN RTX graphical processing unit (GPU) with 4608 cores, 1770 MHz clock speed, and 24 GB RAM.

Fig. 6 illustrates the accuracy and loss for the training and validation data compared to the training epoch for each model. Based on the results, the performance of the 1DCNN-LSTM model in terms of training and validation accuracy and loss is better than other models. The results will be discussed further in Section 5.3.

5.2. Evaluation metrics

To score the models performance, four common metrics are utilized: accuracy, precision, recall, and F-measure (F_1), which are defined as following:

1 Accuracy, which measures the overall performance of the model by quantifying the proportion of correct predictions over all predictions.

$$Accuracy = \frac{N_c}{N} \tag{17}$$

where N_c is total number of correct observations, and N is total number of observations.

2 Precision, which measures the number of true positives over total predicted positive. The precision of class c can be obtained by:

$$Precision_c = \frac{TP_c}{TP_c + FP_c} \tag{18}$$

where TP_c and FP_c are the number of true-positive and false-positive of class c (i.e. baseline and critical events), respectively.

3 Recall, which measures the number of corrected classified observations over the total observations in the class c.

$$Recall_c = \frac{TP_c}{TP_c + FN_c} \tag{19}$$

where FN_C is the number of false-negative of class c.

4 F_1 score, which applies weighted harmonic average to the precision and recall.

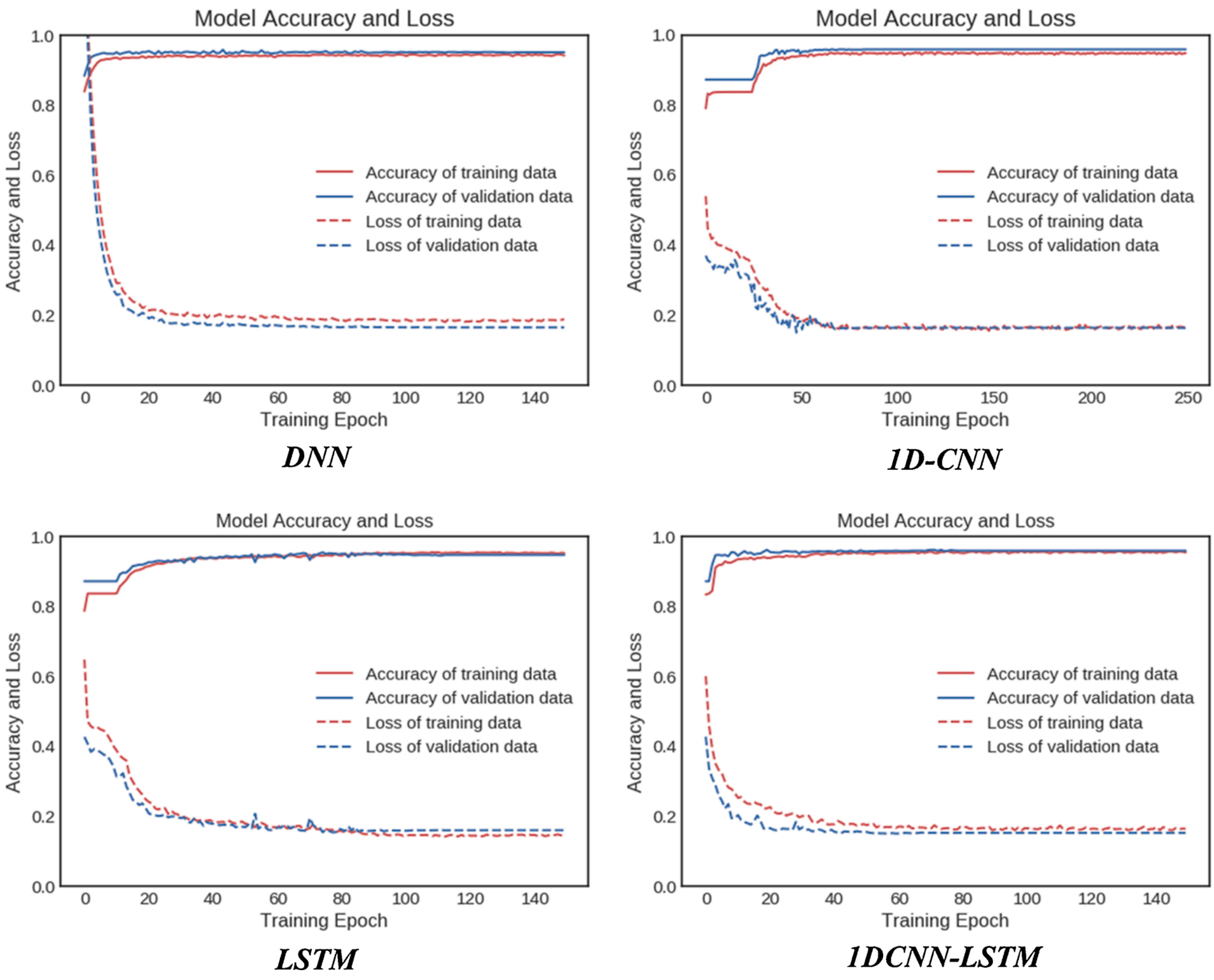

Fig. 6. Accuracy and loss for the training and validation datasets.

$$F_{1_c} = 2 * \frac{Precision_c * Recall_c}{Precision_c + Recall_c} \tag{20}$$

The advantage of F_1 is its suitability for imbalanced data where the proportion of one of the classes is lower comparing to others. This measure combines precision and recall metrics and weighting the classes based on the proportion.

5 Area under curve (AUC), is one of the well-known measures that measure the area under ROC curve to quantify performance of a binary classifier (Bradley 1997) where ROC can be obtained by plotting the true positive rate (recall) vs false positive rate.

5.3. Comparative results

In order to study the models' performance in real-world scenarios, several factors are considered. First, 30% of the data is used as a hold-out sample. This helps us to achieve an unbiased evaluation of the fitted models on the training dataset. The test data contains sampled data that spans both baseline and CNC events that our models would face once it is used in the real-world scenarios (James et al. 2013). Second, the model should perform well not only on prediction of safety critical events, but also generate a low number of false alarms on baseline events. Therefore, several metrics including precision, recall, F1-score, and AUC are used for both baseline and CNC events. This paper presents the performance measures for baseline and CNC separately since the model should not only predict reasonable proportion of critical events, but also predicts baseline events correctly and generates very few false alarms. Finally, the processing time of trained models on test sample events is measured by running it on a CPU to study whether it can perform predictions in real-time.

The comparative results of the DNN, 1D-CNN, LSTM, and 1DCNN-LSTM models are provided in Table 5. Focusing on overall performance on the test dataset, it can be inferred that 1DCNN-LSTM model performs the best compared to other models with the 95.45% accuracy, and 0.9603 AUC. Furthermore, it provides the best performance in terms of precision, recall, and F1-score for both baseline and CNC events. Based on the results, the model predicts 99.43% of baseline and 73.40% of CNC events correctly on the test dataset with the precision of 95.6% which is acceptable. The 1DCNN-LSTM model generates false CNC alarms at the rates of 0.13% and 0.57% on the train and test datasets, respectively. In terms of processing time of models on each event, the trained model is run on a CPU to study whether it can perform predictions in real-time. The average processing times on each test data are provided in Table 5. The 1DCNN-LSTM model can perform in almost real-time (0.345 millisecond) to predict the outcome of an event, which depicts the capability of model to perform real-time in real-world scenarios.

By comparing the results of four developed models, the accuracy of the DNN model is 92.10%, while 1D-CNN, LSTM, and 1DCNN-LSTM models have substantially improved accuracy to 94.54%, 94.32%, and 95.45%, respectively. It can be observed that the LSTM, 1D-CNN, and

Table 5
Model performance evaluation.

Performance		Train Data				Test Data			
		DNN	*1D-CNN*	*LSTM*	*1DCNN-LSTM*	*DNN*	*1D-CNN*	*LSTM*	*1DCNN-LSTM*
Test time (millisecond)		-	-	-	-	0.181	0.194	19.65	0.345
Overall	*Accuracy(%)*	94.46	95.02	94.62	**96.19**	92.10	94.54	94.32	**95.45**
	AUC	0.9472	0.95412	0.9536	**0.9836**	0.9085	0.9535	0.9371	**0.9603**
Baseline	*Precision*	0.9470	0.9489	0.9458	**0.9686**	0.9380	0.9440	0.9461	**0.9563**
	Recall	0.9956	**0.9992**	0.9949	0.9987	0.9917	0.9943	0.9899	**0.9943**
	F1-Score	0.9707	0.9734	0.9697	**0.9834**	0.9641	0.9685	0.9675	**0.9749**
CNC	*Precision*	0.9674	**0.9943**	0.9615	0.9606	0.9269	0.9517	0.9193	**0.9567**
	Recall	0.6988	0.7090	0.6915	**0.8107**	0.6164	0.6547	0.6701	**0.7340**
	F1-Score	0.8114	0.8278	0.8045	**0.8793**	0.7404	0.7758	0.7751	**0.8307**

1DCNN-LSTM models improved the fit through the incorporation of temporal dependency between observations. It highlights the hypothesis that there is a need to consider local dependency of time-series input in this context. The results also reveal that combining 1D-CNN and LSTM models can improve the fit by extracting additional features from the input data and incorporate those to a time-series analyzer model (i.e. LSTM model). Furthermore, the number of baseline and CNC events and the predicted values on the test dataset for the developed models are provided in the Fig. 7. By comparing the models, it can be inferred that the 1DCNN-LSTM model provides the best results in terms of prediction of both baseline and CNC events.

5.4. Discussion on feature importance

As showed in the previous section, the 1DCNN-LSTM model performed the best in terms of predicting extreme events occurrence comparing to the other discussed models. In this section we will discuss the importance of each block of data in the prediction performance. Three sets of features are considered: 1- Vehicle kinematics, 2- Driving volatility, and 3- Distraction profile. Initially, the model is trained with vehicle kinematics, next driving volatility features are added to the model, and finally distraction profile is added to the feature space. The results for the 1DCNN-LSTM model is summarized in the Table 6. According to the results, by feeding the model only with vehicle kinematics, the model accuracy on the test dataset will reach 90.31%. By adding the extracted volatility features to the model, the model accuracy reaches 91.12%. Finally, adding the distraction profile of the driver to the model will enhance the accuracy to 95.45%. Overall, it can be inferred that driving volatility and distraction profile substantially improve the prediction accuracy of the 1DCNN-LSTM model.

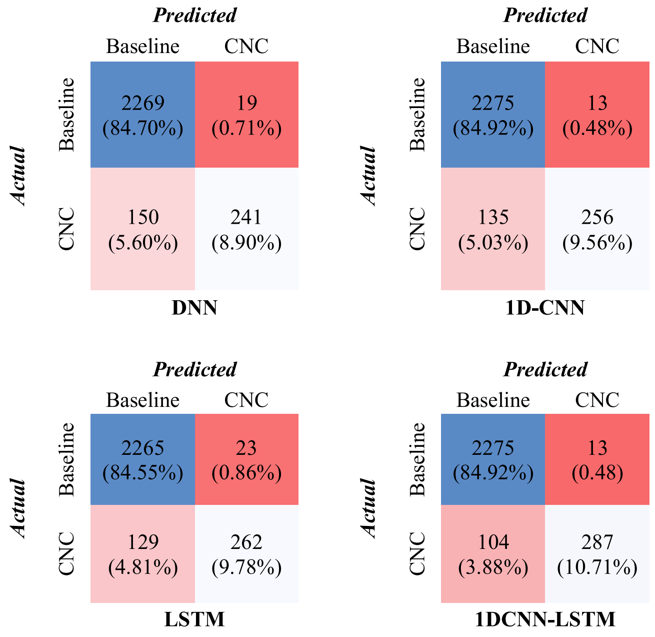

Fig. 7. Confusion matrix of models' prediction on test data.

6. Conclusion

Emergence of high-resolution big data generated by connected and automated vehicles and new sensors coupled with availability of high-performance computational resources, enabled the application of new concepts and methods. This study develops a framework which quantifies the crash occurrence risk by integrating multiple data sources and applying deep learning methods. The kinematic movement of the vehicle and information on driver impairment, in terms of distraction, is obtained from second Strategic Highway Research Program (SHRP2). Volatility functions are employed to extract additional features from vehicular movements to quantify variations in instantaneous driving decisions. The initial statistical analysis revealed that impaired driving, in terms of distraction, and instability in driving can serve as the leading indicator of crash occurrence.

Several deep learning models including 1D-CNN, LSTM, and 1DCNN-LSTM approaches are utilized to compare performance, with the DNN model as the baseline to perform real-time critical event prediction. Based on the results, by capturing the time dependency of the input data, the model performance can be improved significantly. The results reveal that extra features extracted by the CNN layers, when coupled with the LSTM model can help us to achieve 95.45% accuracy on the test data and can correctly predict 73.4% of the crashes and near-crashes with a 95.7% precision. Furthermore, the analysis revealed that by adding driving volatility features and driver distraction profile, the model accuracy can substantially be improved. Although the model is trained using high computational power computers, the analysis revealed that the model is capable when implemented in relatively low computational environment machines with CPU and it performs in real-time. The results showed that it only requires 0.345 milliseconds on average to process and predict an event, which is sufficient for safety critical events, as IEEE 1609.x standard for DSRC considers 100 milliseconds for safety applications (Harding et al. 2014). Furthermore, the trained model and weights can be used as a pre-trained model and be transferred to other datasets.

The developed model in this study can be used to proactively monitor the driving performance of the drivers in real-time and provide warnings at the times in which drivers exhibit volatile and distracted driving behavior. This study utilized driving instability, driver distraction, and vehicle kinematics as the inputs of the network. In future research, other streams of data, including roadway condition, traffic state, and information of the surrounding vehicles can be incorporated to improve the model performance by providing additional information regarding the surrounding environment.

Table 6
Evaluation of feature importance in the 1DCNN-LSTM model.

Performance		Vehicle Kinematics		Kinematics & Volatility		Kinematics & Volatility & Distraction	
		Train	Test	Train	Test	Train	Test
Overall	*Accuracy (%)*	90.80	90.31	91.61	91.12	**96.19**	**95.45**
	AUC	0.8903	0.8459	0.9404	0.8833	**0.9836**	**0.9603**
Baseline	*Precision*	0.9202	0.9174	0.9492	0.9247	**0.9686**	**0.9563**
	Recall	0.9886	0.9812	0.9886	0.9720	**0.9987**	**0.9943**
	F_1-Score	0.9532	0.9482	0.9685	0.9477	**0.9834**	**0.9749**
CNC	*Precision*	0.8823	0.8146	0.8084	0.7664	**0.9606**	**0.9567**
	Recall	0.4987	0.4833	0.6905	0.5371	**0.8107**	**0.7340**
	F_1-Score	0.6373	0.6067	0.7860	0.6315	**0.8793**	**0.8307**

CRediT authorship contribution statement

Ramin Arvin: Methodology, Writing - original draft. **Asad J. Khattak:** Supervision, Writing - review & editing. **Hairong Qi:** Methodology.

ACKNOWLEDGEMENT

This research is funded by the U.S. Department of Transportation (USDOT)under Project R-23 through the Collaborative Sciences Center for Road Safety (CSCRS), a consortium led by The University of North Carolina at Chapel Hill in partnership with The University of Tennessee. The data was provided by Dr. Tom Karnowski, Oak Ridge National Laboratory, for which the authors are very grateful. Any opinions, findings, conclusions and recommendations are those of the authors and do not necessarily reflect views of the sponsors. The authors thank Ms. Kinzee Clark and Mr. Zachary Jerome for editing the manuscript.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., 2016. Tensorflow: A system for large-scale machine learning. Proceedings of the 12th Symposium on Operating Systems Design and Implementation 265–283.

Highway statistics, 2016, 2018. Federal Highway Administration.

Anon, 2008. National highway traffic safety administration. National motor vehicle crash causation survey: Report to congress. National Highway Traffic Safety Administration Technical Report DOT HS 811 059.

Arvin, R., Kamrani, M., Khattak, A., 2019a. The role of pre-crash driving instability in contributing to crash intensity using naturalistic driving data. Accident Analysis & Prevention 132.

Arvin, R., Kamrani, M., Khattak, A.J., 2019b. Examining the role of speed and driving stability on crash severity using shrp2 naturalistic driving study data. In: Transportation Research Board 98th Annual Meeting. Washington DC.

Arvin, R., Kamrani, M., Khattak, A.J., 2019c. How instantaneous driving behavior contributes to crashes at intersections: Extracting useful information from connected vehicle message data. Accident Analysis & Prevention 127, 118–133.

Arvin, R., Khattak, A.J., 2020. Driving impairments and duration of distractions: Assessing crash risk by harnessing microscopic naturalistic driving data. Accident Analysis & Prevention 146, 105733.

Arvin, R., Khattak, A.J., Kamrani, M., Rio-Torres, J., 2020. Safety evaluation of connected and automated vehicles in mixed traffic with conventional vehicles at intersections. Journal of Intelligent Transportation Systems 1–18.

Baghbaderani, R.K., Hairong, Q., 2019. Incorporating spectral unmixing in satellite imagery semantic segmentation. IEEE International Conference on Image processing.

Baghbaderani, R.K., Qu, Y., Qi, H., Stutts, C., 2020. Representative-discriminative learning for open-set land cover classification of satellite imagery. Proceedings of the European Conference on Computer Vision 1–17.

Bao, J., Yu, H., Wu, J., 2019. Short-term ffbs demand prediction with multi-source data in a hybrid deep learning framework. IET Intelligent Transport Systems.

Bradley, A.P., 1997. The use of the area under the roc curve in the evaluation of machine learning algorithms. Pattern recognition 30 (7), 1145–1159.

Brownlee, J., 2018. Deep learning for time series forecasting: Predict the future with mlps, cnns and lstms in python Machine Learning Mastery.

Brownlee, J., 2020. Imbalanced classification with python: Better metrics, balance skewed classes, cost-sensitive learning Machine Learning Mastery.

Cai, Q., Abdel-Aty, M., Sun, Y., Lee, J., Yuan, J., 2019. Applying a deep learning approach for transportation safety planning by using high-resolution transportation and land use data. Transportation research part A: policy practice 127, 71–85.

Cai, Q., Abdel-Aty, M., Yuan, J., Lee, J., Wu, Y., 2020. Real-time crash prediction on expressways using deep generative models. Transportation research part C: emerging technologies 117, 102697.

Chollet, F., 2015. Keras.

Ciresan, D.C., Meier, U., Masci, J., Gambardella, L.M., Schmidhuber, J., 2011. Flexible, high performance convolutional neural networks for image classification. Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence.

Cui, H., Yuan, G., Liu, N., Xu, M., Song, H., 2020. Convolutional neural network for recognizing highway traffic congestion. Journal of Intelligent Transportation Systems 24 (3), 279–289.

De Naurois, C.J., Bourdin, C., Bougard, C., Vercher, J.-L., 2018. Adapting artificial neural networks to a specific driver enhances detection and prediction of drowsiness. Accident Analysis & Prevention 121, 118–128.

Ding, D., Tong, J., Kong, L., 2020. A deep learning approach for quality enhancement of surveillance video. Journal of Intelligent Transportation Systems 24 (3), 304–314.

Dingus, T.A., Hanowski, R.J., Klauer, S.G., 2011. Estimating crash risk. Ergonomics in Design 19 (4), 8–12.

Eren, L., Ince, T., Kiranyaz, S., 2019. A generic intelligent bearing fault diagnosis system using compact adaptive 1d cnn classifier. Journal of Signal Processing Systems 91 (2), 179–189.

Everitt, B., Skrondal, A., 2002. The cambridge dictionary of statistics. Cambridge University Press Cambridge.

Figlewski, S., 1994. Forecasting volatility using historical data.

Formosa, N., Quddus, M., Ison, S., Abdel-Aty, M., Yuan, J., 2020. Predicting real-time traffic conflicts using deep learning. Accident Analysis & Prevention 136, 105429.

Gao, Z., Liu, Y., Zheng, J., Yu, R., Wang, X., Sun, P., 2018. Predicting hazardous driving events using multi-modal deep learning based on video motion profile and kinematics data. Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC) 3352–3357.

Gers, F.A., Schraudolph, N.N., Schmidhuber, J., 2002. Learning precise timing with lstm recurrent networks. Journal of machine learning research 3 (Aug), 115–143.

Ghanim, M.S., Abu-Lebdeh, G., 2015. Real-time dynamic transit signal priority optimization for coordinated traffic networks using genetic algorithms and artificial neural networks. Journal of Intelligent Transportation Systems 19 (4), 327–338.

Gong, Y., Abdel-Aty, M., Yuan, J., Cai, Q., 2020. Multi-objective reinforcement learning approach for improving safety at intersections with adaptive traffic signal control. Accident Analysis & Prevention 144, 105655.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep learning. MIT press.

Hankey, J.M., Perez, M.A., Mcclafferty, J.A., 2016. Description of the shrp 2 naturalistic database and the crash, near-crash, and baseline data sets. Virginia Tech Transportation Institute.

Harding, J., Powell, G., Yoon, R., Fikentscher, J., Doyle, C., Sade, D., Lukuc, M., Simons, J., Wang, J., 2014. Vehicle-to-vehicle communications: Readiness of v2v technology for application. National Highway Traffic Safety Administration, United States.

Hauer, E., 2006. The frequency–severity indeterminacy. Accident Analysis & Prevention 38 (1), 78–83.

Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A.-R., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Kingsbury, B., 2012a. Deep neural networks for acoustic modeling in speech recognition. IEEE Signal processing magazine 29.

Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R., 2012b. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural computation 9 (8), 1735–1780.

Huber, P.J., 2005. Robust statistics. John Wiley & Sons.

James, G., Witten, D., Hastie, T., Tibshirani, R., 2013. An introduction to statistical learning. Springer.

Jeon, H., Lee, J., Sohn, K., 2018. Artificial intelligence for traffic signal control based solely on video images. Journal of Intelligent Transportation Systems 22 (5), 433–445.

Johnson, J.M., Khoshgoftaar, T.M., 2019. Survey on deep learning with class imbalance. Journal of Big Data 6 (1), 27.

Kamrani, M., Arvin, R., Khattak, A.J., 2019. The role of aggressive driving and speeding in road safety: Insights from shrp2 naturalistic driving study data. In: Transportation Research Board 98th Annual Meeting. Washington DC.

Khadhir, A., Anil Kumar, B., Vanajakshi, L.D., 2020. Analysis of global positioning system based bus travel time data and its use for advanced public transportation system applications. Journal of Intelligent Transportation Systems 1–19.

Khattak, A.J., Ahmad, N., Wali, B., Dumbaugh, E., 2021. A taxonomy of driving errors and violations: Evidence from the naturalistic driving study. Accident Analysis & Prevention 151, 105873.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv.

Kluger, R., Smith, B.L., Park, H., Dailey, D.J., 2016. Identification of safety-critical events using kinematic vehicle data and the discrete fourier transform. Accident Analysis & Prevention 96, 162–168.

Lecun, Y., Bottou, L., Bengio, Y., I. Haffner, P.J.P.O.T., 1998. Gradient-based learning applied to document recognition. 86 (11), 2278–2324.

Li, P., Abdel-Aty, M., Yuan, J., 2020. Real-time crash risk prediction on arterials based on lstm-cnn. Accident Analysis & Prevention 135, 105371.

Li, Y., Khoshelham, K., Sarvi, M., Haghani, M., 2019. Direct generation of level of service maps from images using convolutional and long short-term memory networks. Journal of Intelligent Transportation Systems 23 (3), 300–308.

Lin, L., He, Z., Peeta, S., 2018. Predicting station-level hourly demand in a large-scale bike-sharing network: A graph convolutional neural network approach. Transportation Research Part C: Emerging Technologies 97, 258–276.

Liu, J., Khattak, A.J., 2016. Delivering improved alerts, warnings, and control assistance using basic safety messages transmitted between connected vehicles. Transportation research part C: emerging technologies 68, 83–100.

Liu, M., Shi, J., 2019. A cellular automata traffic flow model combined with a bp neural network based microscopic lane changing decision model. Journal of Intelligent Transportation Systems 23 (4), 309–318.

Longerstaey, J., Spencer, M., 1996. Riskmetricstm—technical document, 51. Morgan Guaranty Trust Company of New York, New York, p. 54.

Ma, X., Tao, Z., Wang, Y., Yu, H., Wang, Y., 2015. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. Transportation Research Part C: Emerging Technologies 54, 187–197.

Mohammadnazar, A., Arvin, R., Khattak, A.J., 2021. Classifying travelers' driving style using basic safety messages generated by connected vehicles: Application of unsupervised machine learning. Transportation Research Part C. Emerging Technologies 122, 102917.

Nasr Esfahani, H., Song, Z., 2020. A Deep Neural Network Approach for Pedestrian Trajectory Prediction Considering Heterogeneity. In Transportation Research Board. 99thh Annual Annual Meeting.

Nezafat, R.V., Sahin, O., Cetin, M., 2019. Transfer learning using deep neural networks for classification of truck body types based on side-fire lidar data. Journal of Big Data Analytics in Transportation 1 (1), 71–82.

Nhtsa, 2009. Traffic safety facts: Motorcycles. National Highway Traffic Safety Association, p. 159.

Ordóñez, F., Roggen, D., 2016. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. Sensors 16 (1), 115.

Osman, O.A., Hajij, M., Karbalaieali, S., Ishak, S., 2018. Crash and near-crash prediction from vehicle kinematics data: A shrp2 naturalistic driving study.

Osman, O.A., Hajij, M., Karbalaieali, S., Ishak, S., 2019. A hierarchical machine learning classification approach for secondary task identification from observed driving behavior data. Accident Analysis & Prevention 123, 274–281.

Palaz, D., Collobert, R., 2015. Analysis of cnn-based speech recognition system using raw speech as input. Idiap.

Parsa, A.B., Taghipour, H., Derrible, S., Mohammadian, A.K., 2019. Real-time accident detection: Coping with imbalanced data. Accident Analysis Prevention 129, 202–210.

Perez, M.A., Sudweeks, J.D., Sears, E., Antin, J., Lee, S., Hankey, J.M., Dingus, T.A., 2017. Performance of basic kinematic thresholds in the identification of crash and near-crash events within naturalistic driving data. Accident Analysis & Prevention 103, 10–19.

Qian, Y., Liang, Y., Li, M., Feng, G., Shi, X., 2014. A resampling ensemble algorithm for classification of imbalance problems. Neurocomputing 143, 57–67.

Sermanet, P., Lecun, Y., 2011. Traffic sign recognition with multi-scale convolutional networks. Proceedings of the IJCNN 2809–2813.

Shi, X., Wong, Y.D., Li, M.Z.-F., Palanisamy, C., Chai, C., 2019. A feature learning approach based on xgboost for driving assessment and risk prediction. Accident Analysis & Prevention 129, 170–179.

Simard, P.Y., Steinkraus, D., Platt, J.C., 2003. Best practices for convolutional neural networks applied to visual document analysis. Proceedings of the Icdar.

Song, Y., Zhang, Z., Baghbaderani, R.K., Wang, F., Qu, Y., Stuttsy, C., Qi, H., 2019. Land cover classification for satellite images through 1d cnn. Proceedings of the 2019 10th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS) 1–5.

Tang, J., Yu, S., Liu, F., Chen, X., Huang, H., 2019a. A hierarchical prediction model for lane-changes based on combination of fuzzy c-means and adaptive neural network. Expert systems with applications 130, 265–275.

Tang, K., Chen, S., Khattak, A.J., Pan, Y., 2019b. Deep architecture for citywide travel time estimation incorporating contextual information. Journal of Intelligent Transportation Systems 1–17.

Wali, B., Khattak, A.J., Karnowski, T., 2019. Exploring microscopic driving volatility in naturalistic driving environment prior to involvement in safety critical events—concept of event-based driving volatility. Accident Analysis & Prevention 132, 105277.

Wali, B., Khattak, A.J., Karnowski, T., 2020. The relationship between driving volatility in time to collision and crash-injury severity in a naturalistic driving environment. Analytic Methods in Accident Research 28, 100136.

Wei, X., Yang, Z., Liu, Y., Wei, D., Jia, L., Li, Y., 2019. Railway track fastener defect detection based on image processing and deep learning techniques: A comparative study. Engineering Applications of Artificial Intelligence 80, 66–81.

Xu, C., Ji, J., Liu, P., 2018a. The station-free sharing bike demand forecasting with a deep learning approach and large-scale datasets. Transportation research part C: emerging technologies 95, 47–60.

Xu, K., Zeng, Y., Zhang, Q., Yin, Q., Sun, L., Xiao, K., 2019. Online probabilistic goal recognition and its application in dynamic shortest-path local network interdiction. Engineering Applications of Artificial Intelligence 85, 57–71.

Xu, M., Wu, J., Huang, L., Zhou, R., Wang, T., Hu, D., 2018b. Network-wide traffic signal control based on the discovery of critical nodes and deep reinforcement learning. Journal of Intelligent Transportation Systems 1–10.

Yang, J., Nguyen, M.N., San, P.P., Li, X.L., Krishnaswamy, S., 2015. Deep convolutional neural networks on multichannel time series for human activity recognition. Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence.

Ye, M., Osman, O.A., Ishak, S., Hashemi, B., 2017. Detection of driver engagement in secondary tasks from observed naturalistic driving behavior. Accident Analysis & Prevention 106, 385–391.

Yuan, J., Abdel-Aty, M., Gong, Y., Cai, Q., 2019. Real-time crash risk prediction using long short-term memory recurrent neural network. Transportation research record 2673 (4), 314–326.

Zeng, Q., Huang, H., Pei, X., Wong, S., Gao, M., 2016. Rule extraction from an optimized neural network for traffic crash frequency modeling. Accident Analysis & Prevention 97, 87–95.

Zhang, S., Abdel-Aty, M., Yuan, J., Li, P., 2020. Prediction of pedestrian crossing intentions at intersections based on long short-term memory recurrent neural network. Transportation research record, 0361198120912422.

Zhao, Z., Chen, W., Wu, X., Chen, P.C., Liu, J., 2017. Lstm network: A deep learning approach for short-term traffic forecast. IET Intelligent Transport Systems 11 (2), 68–75.