

Article

A CNN-LSTM-Attention Model for Near-Crash Event Identification on Mountainous Roads

Jing Zhao ¹, Wenchen Yang ²  and Feng Zhu ^{3,*}

¹ College of Traffic & Transportation, Chongqing Jiaotong University, Chongqing 400074, China; jjiin1220@163.com

² National Engineering Research Center for Efficient Maintenance, Safety and Durability of Roads and Bridges, Broadvision Engineering Consultants Co., Ltd., Kunming 650200, China; tongjiywc@163.com

³ School of Civil and Environmental Engineering, Nanyang Technological University, Singapore 639798, Singapore

* Correspondence: zhufeng@ntu.edu.sg

Abstract: To enhance traffic safety on mountainous roads, this study proposes an innovative CNN-LSTM-Attention model designed for the identification of near-crash events, utilizing naturalistic driving data from the challenging terrains in Yunnan, China. A combination of a threshold method complemented by manual verification is used to label and annotate near-crash events within the dataset. The importance of vehicle motion features is evaluated using the random forest algorithm, revealing that specific variables, including x-axis acceleration, y-axis acceleration, y-axis angular velocity, heading angle, and vehicle speed, are particularly crucial for identifying near-crash events. Addressing the limitations of existing models in accurately detecting near-crash scenarios, this study combines the strengths of convolutional neural networks (CNN), long short-term memory (LSTM) networks, and an attention mechanism to enhance model sensitivity to crucial temporal and spatial features in naturalistic driving data. Specifically, the CNN-LSTM-Attention model leverages CNN to extract local features from the driving data, employs LSTM to track temporal dependencies among feature variables, and uses the attention mechanism to dynamically fine-tune the network weights of feature parameters. The efficacy of the proposed model is extensively evaluated against six comparative models: CNN, LSTM, Attention, CNN-LSTM, CNN-Attention, and LSTM-Attention. In comparison to the benchmark models, the CNN-LSTM-Attention model achieves superior overall accuracy at 98.8%. Moreover, it reaches a precision rate of 90.1% in detecting near-crash events, marking an improvement of 31.6%, 14.8%, 63.5%, 8%, 23.5%, and 22.6% compared to the other six comparative models, respectively.

Keywords: traffic safety; near-crash events; deep learning; naturalistic driving data; self-attention; mountainous road



Citation: Zhao, J.; Yang, W.; Zhu, F. A CNN-LSTM-Attention Model for Near-Crash Event Identification on Mountainous Roads. *Appl. Sci.* **2024**, *14*, 4934. <https://doi.org/10.3390/app14114934>

Academic Editor: Suchao Xie

Received: 19 April 2024

Revised: 1 June 2024

Accepted: 3 June 2024

Published: 6 June 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Mountainous roads, characterized by their complex terrain and unpredictable weather conditions, consistently emerge as hotspots for traffic accidents [1]. From 2012 to 2016, over 15% of all traffic accidents in China occurred in such areas, with approximately 68% of these accidents resulted in severe injuries [2], highlighting the urgent need for enhanced road safety in mountainous areas. Similarly, in the United States, frequent traffic accidents have been observed in mountainous highway sections such as Colorado's I-70 [3]. To develop targeted improvements in traffic safety, researchers often rely on mountainous road traffic accident data for causation studies, examining the impact of driver behavior, surrounding traffic, and environmental conditions on accident occurrence [4,5]. However, relying solely on accident data limits our understanding of the transition from normal driving conditions to accident-prone situations. While virtual simulation methods, such as micro-traffic simulations and driving simulators, offer insights into traffic dynamics, their

effectiveness is curtailed by the authenticity of the simulation environments. There are notable discrepancies between simulated and real driving conditions, leading to potential inaccuracies in safety assessments for mountainous roads. In contrast, naturalistic driving data [6] stands out for its ability to provide complete records of accidents, presenting a pivotal research direction for analyzing and preemptively addressing accidents on mountainous roads. Unlike other studies, naturalistic driving studies offer unobtrusive insights into drivers' interactions with complex road conditions, making them a critical tool for advancing road safety in challenging terrains [7].

Given the rarity of actual traffic accidents and the ethical concerns of detailed accident data analysis, existing studies often leverage near-crash events from naturalistic driving data as a viable alternative to real accidents [8–10]. The relationships between the influencing factors and traffic accidents can be inferred from their connections with the near-crash events. Dingus et al. [10] defined near-crash events as the ones that necessitate emergency maneuvers by drivers. They found that the events could occur 10–15 times more frequently than actual crashes. This higher frequency renders near-crash events more suitable for causation studies of traffic accidents, which demand substantial sample sizes for robust conclusions.

The identification of near-crash events in naturalistic driving data are a critical step in this line of research. Current studies mainly focus on the correlations among variables in naturalistic driving data but often overlook their temporal dynamics. This gap is significant as simple machine learning or neural network models, which are commonly employed for identifying near-crash events, may not effectively capture crucial temporal information or fully understand variable interrelations. Consequently, this oversight can compromise the model's ability to accurately detect near-crash events. To fill this gap, this paper introduces an integrated framework for identifying near-crash events on mountainous roads using naturalistic driving data. The main contributions of this paper are summarized as follows.

1. A near-crash event identification process is designed using vehicle motion features such as velocity, longitudinal acceleration, and lateral acceleration. The identification process consists of multiple steps, including data noise filtering, near-crash event extraction and labeling, feature variable selection, and the construction and evaluation of the event identification model.
2. A multilevel deep learning-based near-crash event identification model is proposed. This model integrates the convolutional neural network (CNN), the long short-term memory (LSTM) network, and the attention mechanism. The CNN-LSTM-Attention model utilizes the CNN layer to extract feature variables, captures the temporal correlation between feature variables through the LSTM layer, and employs the attention mechanism to assign different weights to each feature variable.
3. A naturalistic driving dataset collected on mountainous roads is used to evaluate the proposed method. It is demonstrated that the proposed process accurately and efficiently identifies near-crash events from the raw data. The proposed multilevel deep learning-based identification model demonstrates significantly better accuracy and response speed compared to benchmark models.

The remainder of this paper is organized as follows. Section 2 summarizes the state of the art of near-crash event identification methods and outlines the overall framework of the proposed methodology. Section 3 introduces the details of collecting and processing naturalistic driving data. Section 4 presents the architecture and different components of the CNN-LSTM-Attention model. Section 5 reports the results of data processing and near-crash event identifications. Section 6 concludes the paper by summarizing the main findings and providing directions for future research.

2. Preliminary of Near-Crash Event Identification

2.1. Related Work

Existing methods of near-crash event identification can be divided into three categories, i.e., statistical methods, threshold methods, and data-driven approaches. Statistical

methods treat the results of near-crash event identification as a categorical dependent variable and construct statistical models to establish relationships between this dependent variable and observable independent variables. The data from near-crash events is used to calibrate the parameters in these statistical models, enabling the calibrated models to identify potential near-crash events. In addition to identifying near-crash events, statistical models can also reveal the relationships between each independent variable and the near-crash events. For example, Arvin et al. [10] utilized statistical methods to analyze the micro fluctuations in vehicle motion states before a collision, suggesting that nine types of pre-collision driving fluctuations, such as speed and deceleration, can be used to measure the severity of a collision. Arvin and Khattak [11] constructed a logistic regression model to analyze the impact of distracted driving duration on the probability of collision/near-collision events, concluding that the duration of various types of distraction is one of the main indicators for the occurrence of collision/near-collision events, with longer durations correlating with higher collision risk. Khattak et al. [12] used a mixed logistic regression model to analyze the relationship between fluctuations and the occurrence of hazardous events, finding a positive correlation between fluctuations and hazardous events, with rear-end collisions and lane departure incidents being most affected by fluctuations. Statistical methods focus on analyzing the impact of specific factors on event occurrence. However, when it comes to identifying hazardous events, relying solely on these factors often results in underreporting the number of hazardous events.

Threshold methods involve researchers setting specific indicators that represent the occurrence of a hazardous event and defining corresponding threshold values. When an indicator exceeds its threshold, it signifies a near-crash event. Various strategies based on naturalistic driving data have been devised for this purpose. Dingus et al. [13] observed that abnormal longitudinal vehicle acceleration, often resulting from emergency braking, effectively contributes to detecting various near-crash events in the 100-Car naturalistic driving study. Some studies have adapted Dingus's thresholds for identifying near-crash events [14,15]. Based on Dingus's study, Sudweeks [16] developed a functional yaw rate classifier capable of identifying 92% of crash events and 81% of near-crash events, while reducing 42% of invalid or erroneous event identifications. Perez et al. [17] evaluated multi-kinematic parameter thresholds to detect near-crash events from naturalistic driving data in the US and Canada, suggesting that some established thresholds in the literature may be overly sensitive. Wu et al. [18] proposed threshold methods and Zou's test to filter near-crash events, which employed survival analysis and ROC curves to fine-tune the optimal detection thresholds. Although threshold methods might generate false positives, they are suitable for preliminary screening due to their efficiency in filtering a significant portion of near-crash events, thereby facilitating the early stages of causation analysis.

To address the high false-positive rate issue of threshold methods, researchers have turned to machine learning techniques, which excel at identifying complex patterns, for identifying safety-critical events in naturalistic driving data. Osman et al. [19] experimented with various machine learning methods, including K-nearest neighbors (KNN), random forest, and support vector machine (SVM), to predict safety-critical events based on vehicular kinematic data. Shi et al. [20] introduced a feature extraction framework based on XGBoost and the Fuzzy C-Means algorithms, achieving 89% accuracy in identifying near-crash events within the NGSIM dataset. Further, Kluger et al. [21] applied discrete Fourier transforms and K-means clustering to the longitudinal acceleration data from naturalistically driven vehicles, marking distinctive patterns in the acceleration time series that indicate imminent or near crashes, and reducing the false-positive rate to 22%. Shi et al. [22] proposed a classification model based on Extreme Gradient Boosting (XGBoost), utilizing Convolutional Neural Networks (CNN) and Gated Recurrent Units (GRU) for feature engineering, achieving an accuracy of 84.7% and a recall rate of 71.3% in accident identification. However, these models did not account for the temporal characteristics of natural driving data, resulting in suboptimal performance.

Upon reviewing the literature, several limitations are identified. Firstly, statistical methods are often used to study the impact of specific factors on the occurrence of near-crash events but lack the ability to effectively identify such events. Secondly, threshold methods require significant human and time resources, and the threshold settings vary across different datasets, making it difficult to establish a unified, standardized framework. Lastly, previous studies have overlooked the time-series characteristics of natural driving datasets using simple machine learning or neural network models. Considering the characteristics of natural driving datasets, this paper proposes a near-crash event identification method based on the CNN-LSTM-Attention model.

2.2. Overall Framework

The overall framework for identifying near-crash events based on naturalistic driving data are illustrated in Figure 1, comprising the following four main steps.

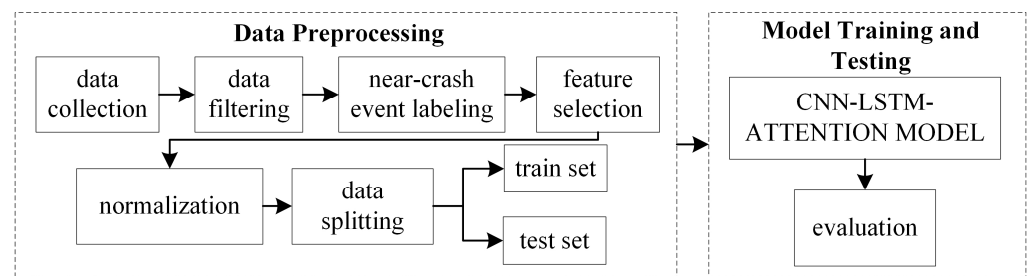


Figure 1. The overall framework for near-crash event identification.

Step 1: Data Collection (with more details introduced in Section 3.1)

Using onboard sensors of a sport utility vehicle specialized for naturalistic driving, a total of 150 h and approximately 8000 km of raw data were collected on mountainous roads in Yunnan Province, China.

Step 2: Data Preprocessing (with more details introduced in Section 3)

- (1) **Data Filtering.** An adaptive filtering algorithm is applied to the collected naturalistic driving data for noise reduction, enhancing data quality for subsequent analysis.
- (2) **Data Labeling.** A dual approach, combining the threshold method and manual video verification, is applied to label the raw data accurately and generate a subset of near-crash events.
- (3) **Feature Selection.** The random forest method is employed to assess the importance of each feature in the near-crash event dataset, with those features deemed most significant chosen as inputs for the identification model.
- (4) **Normalization.** Prior to model training, the dataset is normalized to eliminate discrepancies in scale among features. At the same time, it accelerates the convergence speed of the model and can help avoid overfitting to some extent. The normalization function is:

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

where x_{max} and x_{min} represent the maximum and minimum values of the sample data, respectively.

- (5) **Dataset Division.** The data are randomly divided into 60%, 20%, and 20% for training, validation, and test purposes, respectively. The training set is used for model parameter optimization, the test set for the final assessment of model performance, and the validation set for model tuning.

Step 3: Model Construction and Training (with more details introduced in Section 4)

The CNN-LSTM-Attention model for near-crash event identification is developed using the Keras deep learning framework. The Adam optimizer [23] is employed for

updating model weights, and binary cross-entropy is used as the loss function for model optimization. Binary classification accuracy is utilized as the evaluation metric for assessing model performance.

Step 4: Model Performance Evaluation (with more details introduced in Section 5)

The CNN-LSTM-Attention model's effectiveness is assessed against various models, including CNN, LSTM, Attention, CNN-LSTM, CNN-Attention, and LSTM-Attention. It provides a comprehensive evaluation of the CNN-LSTM-Attention model's accuracy in identifying near-crash events.

3. Data Collection and Processing

3.1. Data Collection

This study utilizes naturalistic driving data collected from 20 drivers, accumulating over 8000 km and more than 150 h of driving on mountainous roads in Yunnan Province, China. The chosen experimental routes include the G78 National Expressway and the G324 National Highway, among others. The naturalistic driving data collection system, as shown in Figure 2, includes the following components:

- (1) Four high-definition cameras. Operating at a 25 Hz frame rate, these cameras are strategically positioned to capture both frontal and rear views of the vehicle, along with the upper body and foot movements of the driver. The image data serves primarily for manual verification and calibration of near-crash events.
- (2) A nine-axis inertial sensor. This sensor is deployed to collect data on x-axis, y-axis, and z-axis acceleration, along with angular velocity and angular data at a frequency of 20 Hz. The orientation of these axes is detailed in Figure 2.
- (3) A GNSS positioning system. It provides detailed information on longitudinal velocity, latitude, longitude, and altitude with a sampling rate of 1 Hz.
- (4) Two industrial computers. These units are used for processing and storing the voluminous data collected by the aforementioned devices.



Figure 2. Demonstration of the Naturalistic Driving Data Collection System.

Previous studies [13,24] have found that in emergency situations, over 90% of drivers would make emergency evasive reactions, which were prominently reflected in the changes observed in vehicular dynamics parameters. Consequently, this study primarily utilizes vehicle dynamics data collected via the inertial sensor to identify near-crash events. Additionally, the video footage from the onboard cameras are also used in labeling near-crash events.

3.2. Data Filtering Process

During the data collection process by onboard sensors, the external environment is complex and variable, with many irrelevant factors causing interference. These interferences can contaminate the collected data, resulting in redundant and erroneous points. Although these interferences may seem minor, they can significantly impact subsequent modeling work, sometimes leading to substantial deviations between the model's predictions and actual values. Therefore, to obtain more accurate natural driving data and construct precise models, it is necessary to filter the data to eliminate noise pollution. To efficiently handle outliers in naturalistic driving data from vehicle sensors, an adaptive mean-standard deviation filter is used for noise attenuation. This filter dynamically adjusts the size of the data filtering window, utilizing the Z-score of the data within the window to identify outliers. Based on this, it then smooths these anomalous values to ensure data integrity. The Z-score is defined as:

$$Z = \frac{|x - \bar{x}|}{s} \quad (2)$$

where x is the value of the sample point, \bar{x} is the mean, and s is the standard deviation of the data. The mean \bar{x} and standard deviation s are defined as:

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n} \quad (3)$$

$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}} \quad (4)$$

where x_i represents the value of the i sample point within the window, and n is the size of the window.

3.3. Near-Crash Event Labeling

To identify and label near-crash events from the collected naturalistic driving data, inspired by Hankey et al. [25], we utilized a dual approach combining the threshold approach and manual video verification. The threshold approach not only aids in the extraction of relevant events but also supports the subsequent training of event identification models. The process of identifying near-crash events is illustrated in Figure 3, with initial threshold parameters during extraction as set by Dingus et al. [13]. Whenever the data at a specific moment align with any predefined threshold settings, a 5 s video record, capturing both before and after that moment, is further subjected to manual verification. Note that this dual labeling approach has its limitations, such as potential subjectivity in parameter selection and the possibility of missing some near-crash events. If a near-crash event is confirmed, it is labeled as 1; otherwise, it is labeled as 0. If the initial extraction yields too few events, the threshold would be relaxed to ensure a robust dataset. Table 1 outlines the initial threshold settings used for event selection.

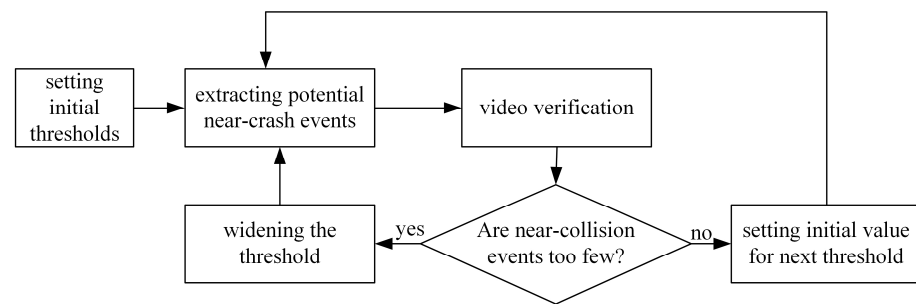


Figure 3. Process of near-crash event screening.

Table 1. Initial thresholds of near-crash event screening.

Number	Threshold Setting
1	Lateral acceleration greater than or equal to 0.7 g
2	Longitudinal acceleration greater than or equal to 0.58 g
3	Longitudinal deceleration less than -0.75 g
4	Emergency event button triggered

3.4. Feature Selection

To mitigate potential issues such as overfitting, escalating model complexity, and low model generalizability, this study employs the random forest algorithm for selecting features from the naturalistic driving data. The random forest algorithm selects high-scoring features as input variables for the near-crash event identification model by calculating the importance of each feature parameter within the data. In the field of naturalistic driving research, the random forest algorithm is widely used for feature importance ranking due to its ability to provide high-accuracy results, insensitivity to noise, and ability to overcome overfitting problems [26,27]. Based on empirical observations, random forest may be effective in managing high-dimensional and nonlinear data, although results can vary based on specific data characteristics and contexts. Therefore, this paper uses the random forest algorithm to analyze the importance of high-dimensional data features and then selects the most important features as input variables for the near-crash event identification model.

4. Near-Crash Event Identification Model

4.1. Model Architecture

The architecture of the near-crash event identification model, i.e., CNN-LSTM-Attention, is shown in Figure 4. For the spatiotemporal relationships in time-series data that previous studies have often overlooked, the model uses the CNN layer to perform convolution operations on the time series to capture features adjacent in time. This is followed by processing long-term dependencies in the sequence through the LSTM layer to capture global features. This design enables the model to better understand the spatiotemporal structure of the input data. Furthermore, we introduced an attention layer embedded within the LSTM layer. This attention mechanism is used to capture global dependencies in the time-series data, allowing the network to dynamically adjust the weights of each time step during the learning process. This design helps better capture long-term dependencies and global patterns. Through the collaborative operation of these three layers, the entire model can more quickly and accurately capture key information related to near-crash events in naturalistic driving data. We also added a Dropout layer to the model to randomly drop some neurons, reducing inter-neuron dependencies and preventing overfitting. Additionally, we used regularization techniques by adding a penalty term for the sum of the squared weight parameters in the loss function to limit the complexity of the model.

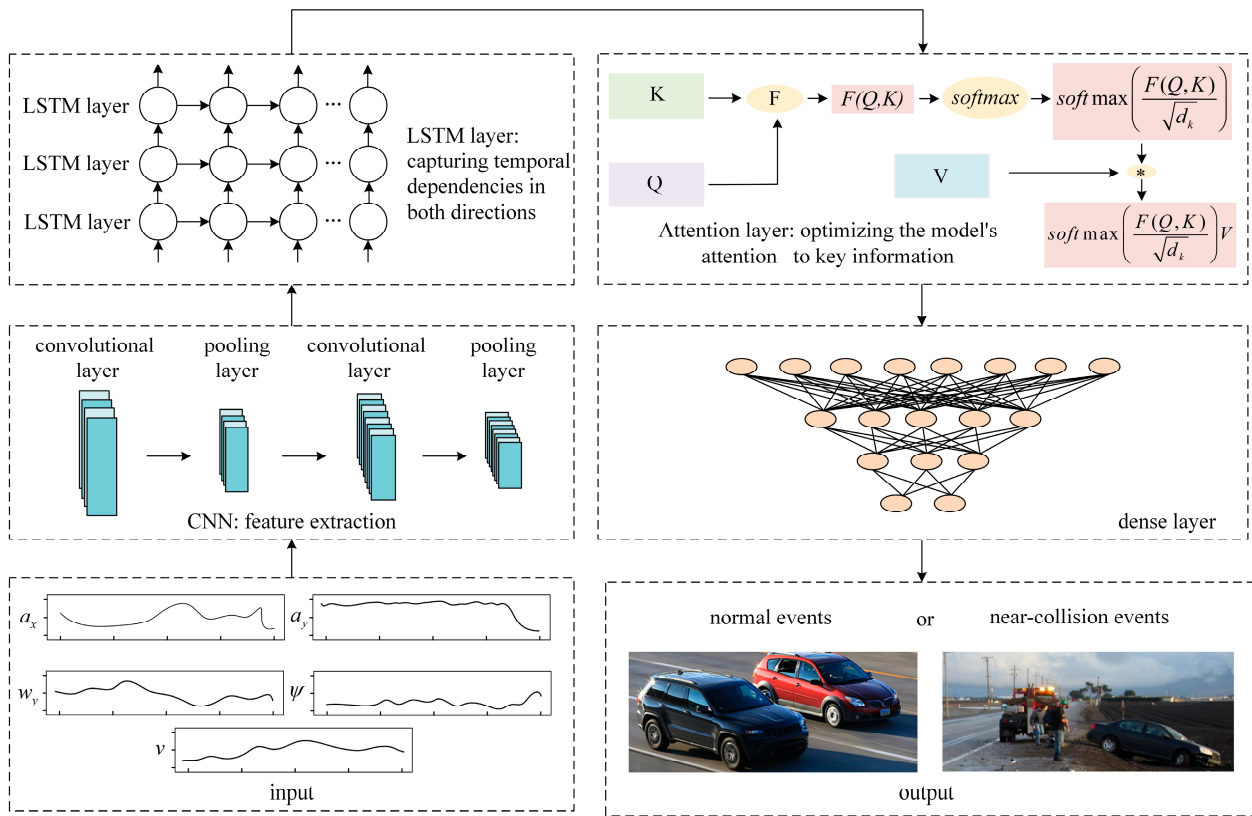


Figure 4. Architecture of the CNN-LSTM-Attention model (* is the multiplication sign).

4.2. CNN Layer

The CNN model is recognized in the field of deep learning for its distinctive capabilities of weight sharing and local connectivity [28]. This architecture excels at performing feature extraction and learning efficient data representations through convolution operations paired with nonlinear activation functions. As shown in Figure 5, the used CNN architecture comprises convolutional layers, pooling layers, and fully connected layers [29]. These components work in concert to diminish the volume of network parameters while preserve the deep features of multi-dimensional input data, thereby enhance the model’s efficiency and effectiveness.

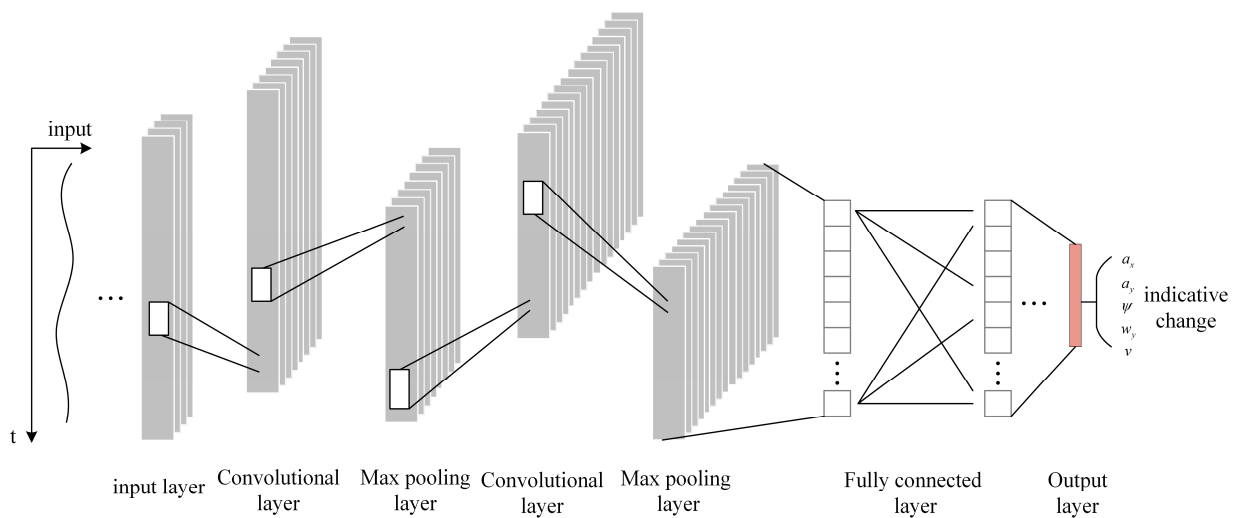


Figure 5. CNN structure.

(1) Convolutional Layer

The convolutional layer is primarily composed of kernels tasked layer that stands as the cornerstone of a CNN, setting it apart from traditional fully connected neural networks. It executes dot product calculations across the input data according to specified size and dilation rates, facilitating the extraction and mapping of features, effectively condensing the data while retaining essential information. The convolutional layer extracts features from input data through convolution operations and reduces the number of parameters, thus streamlining the network for better performance and efficiency. The output of the convolutional layer [30], y_{conv} , is calculated as follows:

$$y_{conv} = \sum_{e=1}^m \sum_{f=1}^n \left((w_r^l)_{ef} d_{ef} \right) \tag{5}$$

where m and n represent the two dimensions of the filter, d_{ef} is the data value of the input matrix at positions e and f , and $(w_r^l)_{ef}$ is the coefficient of the convolution kernel at positions e, f .

(2) Pooling Layer

The pooling layer, also known as the subsampling layer, is one of the critical components of a convolutional neural network. Pooling refers to extracting a certain attribute (such as the maximum, average, or L2-norm) from the corresponding sampling window as a lower-dimensional output, thus sparsely processing the data to reduce computational load. Moreover, pooling operations enhance the robustness and noise resistance of the extracted features while preserve the main characteristics of the data. The output of max pooling is represented by:

$$y_{pool} = \max(d_{ef}), e \in [1 \cdots p], f \in [1 \cdots q] \tag{6}$$

where p and q represent the two dimensions of the pooling window size, d_{ef} is the data value of the input matrix at positions e and f , and y_{pool} is the pooling output.

4.3. LSTM Layer

Following the CNN layer, the extracted features are processed by the LSTM layer, which plays a pivotal role in capturing temporal dependencies. The LSTM network, a variant of recurrent neural network, incorporates gating mechanism [31] to regulate the flow of information. It consists of three control gates: (1) the input gate, which controls the information entering the memory cell; (2) the forget gate, which manages the portions of historical data to discard, and (3) the output gate, which determines the inclusion of current memory in the final output. This design effectively counters the challenges of gradient vanishing and exploding that are prevalent in regular recurrent neural networks [32].

As shown in Figure 6, the used LSTM network typically features three control gates (forget gate, input gate, and output gate) along with a conveyor belt for carrying long-term memory. The operations of these gates are governed by Equations (7)–(12), ensuring control over the information’s entry, retention, and output within the network. Through these gating mechanisms, the LSTM network controls the extraction, discarding, and updating of historical information in a refined manner, thus maintains the network’s sensitivity to both short-term and long-term dependencies without succumbing to gradient issues [33].

$$f_t = \sigma(W_f \cdot [x_t, h_{t-1}] + b_f) \tag{7}$$

$$i_t = \sigma(W_i \cdot [x_t, h_{t-1}] + b_i) \tag{8}$$

$$\tilde{C}_t = \tanh(W_C \cdot [x_t, h_{t-1}] + b_C) \tag{9}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{10}$$

$$o_t = \sigma(W_o \cdot [x_t, h_{t-1}] + b_o) \tag{11}$$

$$h_t = o_t \odot \tanh(C_t) \tag{12}$$

where the sigmoid function (σ) and hyperbolic tangent (\tanh) are activation functions; (W_f, b_f), (W_i, b_i), (W_C, b_C), and (W_o, b_o) are respectively the weight matrices and bias terms for the forget gate, input gate, cell state, and output gate; x_t represents the input vector, h_{t-1} the hidden layer state sensitive to short-term conditions, C_t the cell state that stores long-term conditions, f_t the output of the forget gate, i_t the output of the input gate, \tilde{C}_t the candidate value of the input gate's output, o_t the output of the output gate, and h_t the output of the LSTM model.

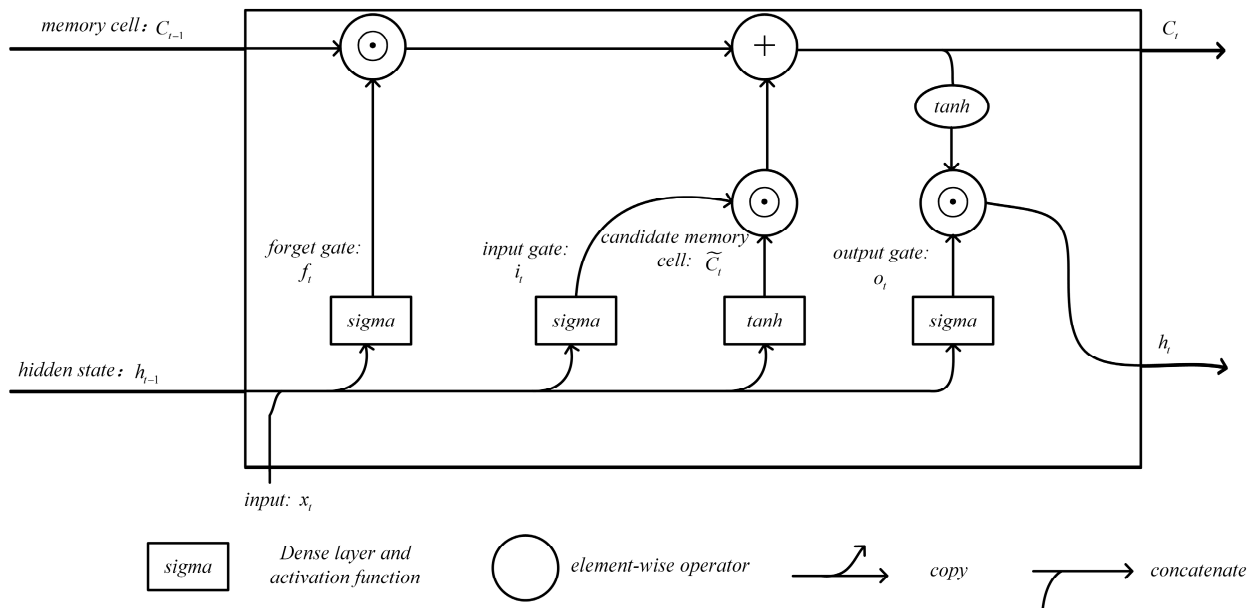


Figure 6. LSTM neural network basic unit.

4.4. Attention Layer

During the detection of near-crash events from naturalistic driving data, the dynamic features of various variables play a critical role. For example, rapid changes in vehicle acceleration and heading angle often signal the occurrence of near-crash events. However, as the LSTM compresses the entire sequence into a singular, fixed-length hidden vector and assigns equal weights to information at different times, it might overlook key information at certain moments, such as drastic changes in acceleration, thereby reduce the predictive accuracy of the model. To address this limitation, this study introduces an attention mechanism [34]. Its inspiration comes from the human biological system, where humans tend to focus on distinct parts when processing large amounts of information. With the development of deep neural networks, the introduction of attention mechanisms helps networks capture more important information, thereby optimizing the computational performance of the entire network. The attention mechanism enhances prediction accuracy and improves computational efficiency by selectively focusing on important input elements.

The attention mechanism is divided into hard attention and soft attention. The hard attention applies attention directly to a specific position in the input sequence. It is akin to a selection operation, and only inputs the features of that position into the model to obtain precise attention information. The soft attention distributes attention in the form of weighted vectors to the features at each position in the input sequence. By considering the entire sequence, it makes the model more fault-tolerant and suitable for longer sequence data.

The attention mechanism is widely used in fields such as drone controlling and naturalistic driving. Shan et al. [35] added a self-attention mechanism to the Transformer

network to minimize drift issues caused by drone movement. Yang et al. [36] proposed a multi-instance learning approach based on the attention mechanism to accurately locate anomalous events in drone flight data. Li et al. [37] proposed a Transformer Encoder model with an attention mechanism to predict vehicle lane change intentions.

This paper adopts the soft attention mechanism, which allocates variable levels of attention to different segments of the data based on a probability-weighted distribution. This approach ensures that significant moments, especially those indicative of near-crash events, are accentuated, thereby enhancing the model's capacity to capture essential temporal information more accurately.

The implementation of this mechanism within this study employs the scaled dot-product attention technique. This method linearly maps the input sequence to produce three different interpretation vectors: the query vector Q , the key vector K , and the value vector V . Following this, an attention score for each sample is derived through a similarity function $F(Q, K)$. These scores are then normalized through the softmax function, establishing the weight coefficients for each temporal segment. These coefficients finally multiply with the value vector V to produce the output of the attention mechanism $Attention(Q, K, V)$.

$$Attention(Q, K, V) = \text{softmax}\left(\frac{F(Q, K)}{\sqrt{d_k}}\right)V \quad (13)$$

where d_k represents the dimension of Q , K , and V . It refers to the dimension of the input vector, which is also the dimension of Q , K , and V vectors. $F(Q, K)$ is the similarity function:

$$F(Q, K) = QK^T \quad (14)$$

5. Results

5.1. Results of Data Processing

In this study, an adaptive mean-standard deviation filter is first used to denoise the naturalistic driving data. Figure 7 shows a comparison between the SG (Savitzky-Golay) filter [38] and the adaptive mean-standard deviation filter. From the figure, it can be seen that the two filters can effectively filter out abrupt anomalous points, such as the upward and downward anomalies near sample point 8000. However, from an overall trend perspective, the SG filter adheres too closely to the original curve, only removing clearly abnormal points and being less effective for smaller anomalies. In contrast, the adaptive mean-standard deviation filter used in this study can effectively remove most of the anomalous points while smooth the data and preserve the characteristic information of the original signal.

For the subsequent model training, we extract normal events and near-crash events from naturalistic driving data based on the threshold method combined with manual verification. Eventually, a dataset containing 3111 normal events and 301 near-crash events was constructed, with each event containing 200 frames and having 8 feature variables.

Next, the random forest algorithm is used for feature selection. The process begins by determining the optimal number of feature variables for splitting nodes in the random forest model based on the out-of-bag (OOB) error produced by the bootstrap samples that are not used in the model's training. As shown in Figure 8, the out-of-bag error reaches its minimum value at 0.0862 when the number of feature variables is 5. Therefore, the number of node feature variables is set to 5. Hence, the number of decision trees in the random forest model is determined. As shown in Figure 9, the out-of-bag error gradually decreases as more decision trees are added, eventually plateauing beyond 300 trees. The random forest model is configured with 300 decision trees to ensure efficiency and accuracy in feature selection, which can strike a balance between computational resource use and predictive performance.

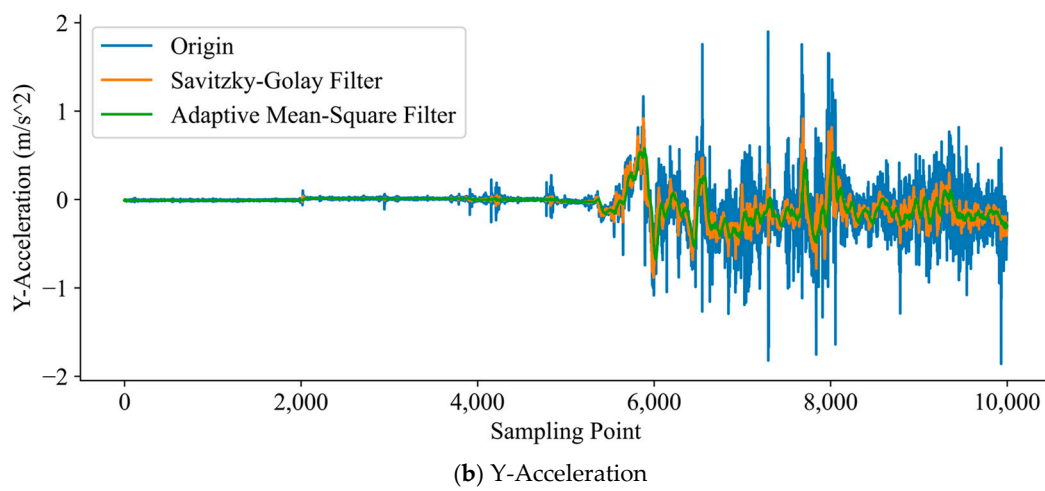
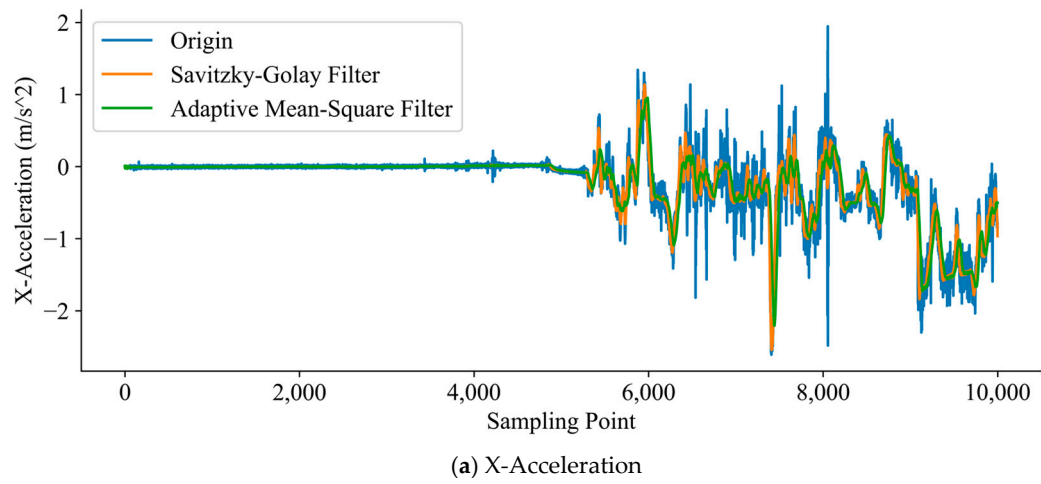


Figure 7. Comparison of the effects of mean filtering and adaptive mean-standard deviation filtering.

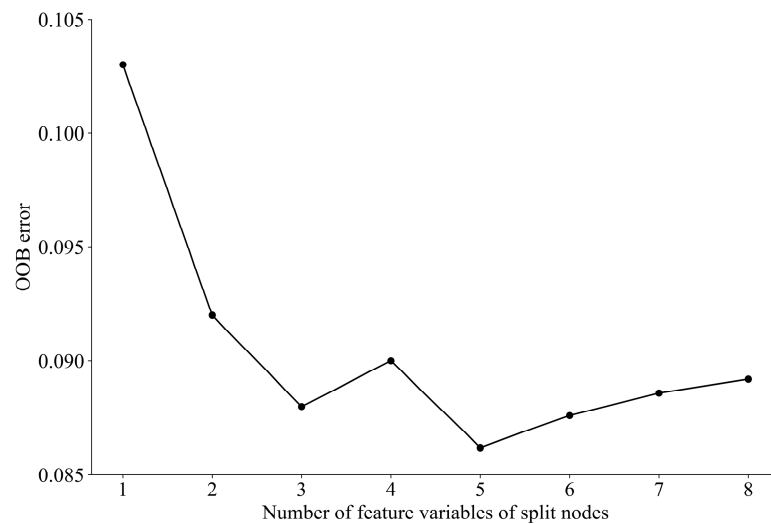


Figure 8. Determination of node feature variables based on OOB error.

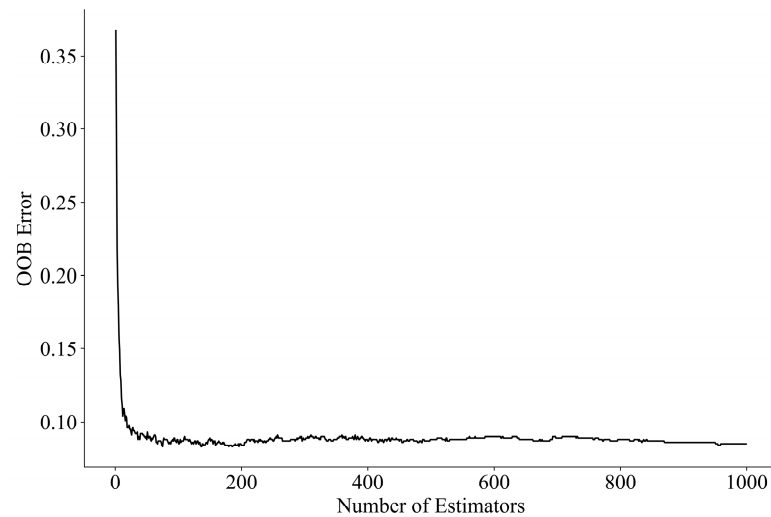


Figure 9. Determination of decision trees in the random forest model.

Finally, feature selection is conducted based on the determined number of decision trees, and the importance of each feature variable is assessed, as shown in Table 2. This evaluation results in the selection of the top five motion features based on their importance, which are subsequently used as input variables for the near-crash event identification model. These critical features include x-axis acceleration, y-axis acceleration, y-axis angular velocity, heading angle, and vehicle speed, which offer a robust foundation for accurately detecting near-crash events in naturalistic driving scenarios.

Table 2. Importance of vehicle motion features.

Feature Variable	Symbol	Definition	Importance
Vehicle speed	v	Rate of change in distance over time, reflecting changes in the state of the vehicle	20%
Heading angle	ψ	The angle between the direction the vehicle's front is pointing and a reference direction (usually geographic north)	8%
Pitch angle	θ	The angle of the vehicle's front tilting up or down	1%
Roll angle	α	The angle of the vehicle's front tilting to one side	2%
X-axis acceleration	a_x	Rate of change in speed in the x-axis direction over time	28%
Y-axis acceleration	a_y	Rate of change in speed in the y-axis direction over time	12%
X-axis angular velocity	w_x	Angular velocity of the vehicle rotating around its x-axis	3%
Y-axis angular velocity	w_y	Angular velocity of the vehicle rotating around its y-axis	26%

5.2. Evaluation of Model Performance

5.2.1. Evaluation Metrics

The effectiveness of the near-crash event identification model is evaluated through five key metrics, ensuring a comprehensive assessment of its performance:

(1) Accuracy (A). This metric measures the proportion of correctly predicted observations by the model out of the total number of samples:

$$A = \frac{N_c}{N} \quad (15)$$

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

(2) Precision (P). This metric measures the proportion of actual positive samples among all samples predicted as positive:

$$P = \frac{TP}{TP + FP} \quad (16)$$

where TP and FP are the numbers of true positive and false positive predictions, respectively.

(3) Recall (R). This metric measures the ratio of the number of correctly predicted positive samples to the total number of positive samples:

$$R = \frac{TP}{TP + FN} \quad (17)$$

where FN is the number of false negatives.

(4) F1 Score (F_1). This metric is the harmonic mean of precision and recall metrics, which is calculated by:

$$F_1 = \frac{2 * P * R}{P + R} \quad (18)$$

(5) Area Under the Curve (AUC). The performance of the classifier is assessed by plotting the true positive rate against the false positive rate (ROC curve).

5.2.2. Experimental Results

To evaluate the performance of the proposed CNN-LSTM-Attention model, the model is benchmarked against various models, including CNN, LSTM, Attention, CNN-LSTM, CNN-Attention, and LSTM-Attention. The parameter settings of the models are detailed in Table 3. The number of model parameters in Table 3 is output by the TensorFlow. Keras used in the study. For the sake of space, only the structure of the proposed CNN-LSTM-Attention model is provided in Table 4. Figure 10 shows the performance of each model on the validation set, where it is observed that the proposed CNN-LSTM-Attention model outperforms the other models in terms of accuracy and loss. The ROC curves for all models are shown in Figure 11. It indicates that the AUC value of the CNN-LSTM-Attention model is higher than that of the other models, demonstrating the highest overall prediction accuracy. To further evaluate the performance of the proposed model in the classification tasks of normal events and near-crash events, we use confusion matrices to compare the classification results of the CNN-LSTM-Attention model and other comparison models.

Table 3. Model parameters.

Model Parameters	Number of CNN Layers	Number of Filters	Width of Convolution Kernel	Number of LSTM Layers	Number of Units	Attention Layers	Number of Model Parameters
CNN	2	64	3	0	0	0	54,465
LSTM	0	0	0	3	32	0	100,865
CNN-LSTM	2	64	3	3	32	0	153,537
Attention	0	0	0	0	0	1	1913
CNN-Attention	2	64	3	0	0	1	120,833
LSTM-Attention	0	0	0	3	32	1	117,505
CNN-LSTM-Attention	2	64	3	3	32	1	170,177

Table 4. Structure of CNN-LSTM-Attention model.

Layer	Output Shape
Input layer	(height, width, channels) = (200, 5, 1)
Convolutional layer (with Relu)	(height, width, channels) = (200, 5, 64)
Maxpooling layer	(height, width, channels) = (100, 2, 64)
Convolutional layer (with Relu)	(height, width, channels) = (100, 2, 64)
Maxpooling layer	(height, width, channels) = (50, 1, 64)
Reshape layer	(length, number of features) = (50, 64)
LSTM layer	(length, number of hidden units) = (50, 64)
LSTM layer	(length, number of hidden units) = (50, 64)
LSTM layer	(length, number of hidden units) = (50, 64)
Attention layer	(length, number of hidden units) = (50, 64)
Global Average Pooling layer	(length, number of hidden units) = (64)
Dense layer (with Relu)	(number of units) = (256)
Dropout layer	(number of units) = (256)
Dense layer (with Sigmoid)	(number of units) = (1)

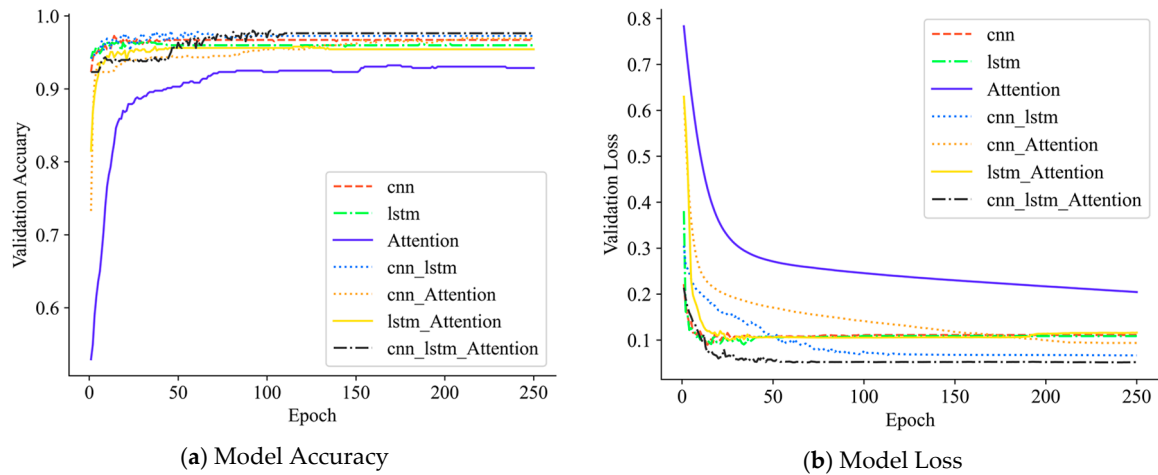


Figure 10. Accuracy and loss of each model on the validation datasets.

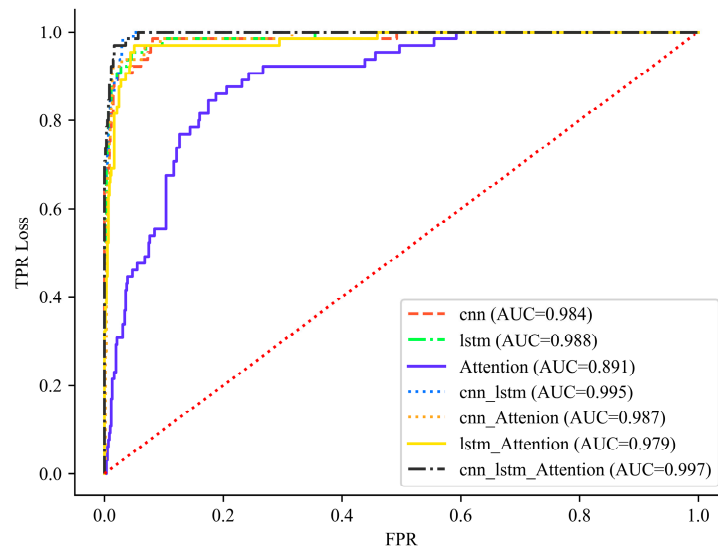


Figure 11. ROC curves of various models.

Table 5 shows the performance of each model in identifying normal events and near-crash events. The four columns of data in the table represent the number of correctly and incorrectly identified normal events, as well as the number of incorrectly and correctly identified near-crash events.

Table 5. Confusion matrix for model comparison.

Model	Counts			
	Correctly Identified Normal Events	Incorrectly Identified Normal Events	Correctly Identified Near-Crash Events	Incorrectly Identified Near-Crash Events
CNN	594	24	3	62
LSTM	603	15	2	63
Attention	510	108	12	53
CNN-LSTM	605	13	2	63
CNN-Attention	590	28	3	62
LSTM-Attention	591	27	3	62
CNN-LSTM-Attention	611	7	1	64

It can be observed that the CNN model performs well in identifying normal events but poorly in identifying near-crash events, with 38 normal events being incorrectly identified

as near-crash events. The LSTM model performs slightly better than the CNN model in identifying normal events, with 19 normal events being incorrectly identified as near-crash events, though some errors still exist. However, the Attention model performs the worst among all models, incorrectly identifying 108 normal events as near-crash events and misjudging 12 near-crash events as normal events.

When examining other Attention-related models in Table 5, it is found that adding an Attention layer to CNN (CNN-Attention model) improves performance, whereas adding it to LSTM (LSTM-Attention model) results in the opposite effect. This indicates that, for the data used in this study, the Attention mechanism does not enhance all deep learning models.

In comparison, the CNN-LSTM model performs better, with improvements in various aspects. The CNN-LSTM-Attention model correctly identified 611 normal events, with only seven normal events being incorrectly identified as near-crash events and one near-crash event misjudged as a normal event, making it the best-performing model overall. In some cases, this model may identify normal events as near-crash events, possibly due to the driver being an aggressive driver who pursues high speed and high acceleration, which could lead to misjudgments by the model.

Table 6 shows the identification results of the CNN-LSTM-Attention model and the comparative models. Among them, the CNN-LSTM-Attention model achieves the highest overall accuracy (*A*) of 98.8%, and the AUC value reached 0.997. All models achieve over 95% accuracy in identifying normal events. However, in identifying near-crash events, the CNN-LSTM-Attention model performs best, with a precision of 90.1%, which is significantly higher than the 61.6% of the CNN model, 76.8% of the LSTM model, and 82.9% of the CNN-LSTM model. The recall rate (*R*) of the CNN-LSTM-Attention model is 98.5%, which is also superior to the performance of the other models. In all, the CNN-LSTM-Attention model achieve the best results in overall performance in identification of normal events and near-crash events. The model can identify more near-crash events with a lower false rate than the comparative models.

Table 6. Performance of each model.

Models	Overall		Normal Events			Near-Crash Events		
	<i>A</i>	AUC	<i>P</i>	<i>R</i>	<i>F</i> ₁	<i>P</i>	<i>R</i>	<i>F</i> ₁
CNN	93.9%	0.984	99.3%	93.9%	0.965	61.6%	93.8%	0.744
LSTM	96.9%	0.988	99.7%	96.9%	0.983	76.8%	96.9%	0.857
Attention	82.4%	0.891	97.7%	82.5%	0.895	32.9%	81.5%	0.469
CNN-LSTM	97.8%	0.995	99.7%	97.9%	0.988	82.9%	96.9%	0.894
CNN-Attention	95.5%	0.987	99.5%	95.5%	0.975	68.9%	95.4%	0.800
LSTM-Attention	95.6%	0.979	99.5%	95.6%	0.975	69.7%	95.4%	0.805
CNN-LSTM-Attention	98.8%	0.997	99.8%	98.9%	0.993	90.1%	98.5%	0.941

In the comparison of CNN and CNN-Attention Models, it is observed that the CNN model is effective in extracting local features, while the advantage of the Attention mechanism lies in focusing on key features. However, given that the data also relies on time-series features, the inclusion of the Attention mechanism only provides marginally additional significant information, resulting in limited performance improvement.

In the comparison of LSTM and LSTM-Attention models, it is observed that the LSTM model excels at processing temporal relationships within time-series data; however, due to the lack of local feature inputs of the CNNs, the introduction of the Attention mechanism may, in some instances, provide redundant information, thus the enhancement is not substantial. In the context of this study, near-crash events are the result of the combined effect of multidimensional, temporal information influenced by nearby time-series variables. While the LSTM model is well-suited for handling relationships in time-series data, the LSTM-Attention model may suffer from overfitting in the dataset used in this study. This could be attributed to the absence of dimensionality reduction in local feature inputs from the CNNs and the introduction of the Attention mechanism, which increases model

complexity. Consequently, the LSTM-Attention model presents lower prediction accuracy compared to the LSTM model.

In the comparison of the CNN-LSTM and CNN-LSTM-Attention models, it is observed that the combination of the CNN, LSTM, and Attention mechanisms allows for feature extraction and information processing at different levels, and creates a complementary effect. The CNN layer extracts local features, the LSTM layer recognizes temporal relationships, and the Attention mechanism focuses on key information. This tripartite combination maximizes the utilization of data features, and enhances the model's performance.

The analysis above indicates that when applying machine learning models in specific scenarios, it is crucial to tailor the approach to the study subject and data conditions through targeted research on feature selection and model structure, and to continually validate and optimize as data enriches to address the uncertainties in machine learning model performance.

6. Conclusions

This study presents a novel approach to identify near-crash events on mountainous roads, employing naturalistic driving data from Yunnan. It introduces a CNN-LSTM-Attention model, which stands out for its innovative integration of the convolutional neural networks (CNN), the long short-term memory networks (LSTM), and the attention mechanism. This model demonstrates unparalleled accuracy and efficiency in detecting near-crash scenarios compared to traditional models.

In the feature analysis, we employ the random forest algorithm for feature selection of the most important vehicle motion features, including x-axis acceleration, y-axis acceleration, y-axis angular velocity, heading angle, and speed. These features are instrumental in the superior performance of the CNN-LSTM-Attention model, which achieved impressive metrics: a recall rate of 98.5% and an F1 score of 0.941 for near-crash events, and for normal events, an even higher recall rate of 98.9% and an F1 score of 0.993. Comparative analyses underscore the CNN-LSTM-Attention model's supremacy, which achieves an overall accuracy of 98.8% and an AUC of 0.997, significantly outperforming the CNN, LSTM, Attention, CNN-LSTM, CNN-Attention, and LSTM-Attention models. Its precision rate of 90.1% in identifying near-crash events represented a remarkable improvement over the other models. It highlights the critical importance of the CNN, LSTM, and attention mechanisms in achieving high performance.

In the future, this study could be significantly enhanced by incorporating a broader spectrum of input variables into the model, including driver physiological data, leading vehicle dynamics obtained from in-vehicle cameras, and millimeter-wave radar. Additionally, an sensitivity analysis of the model performance would be valuable. Such enhancements hold promise to further refine the model's precision on mountainous roads with complex driving environments.

Author Contributions: Conceptualization, J.Z., W.Y. and F.Z.; Methodology, J.Z., W.Y. and F.Z.; Validation, J.Z.; Formal analysis, J.Z., W.Y. and F.Z.; Investigation, J.Z., W.Y. and F.Z.; Resources, W.Y.; Data curation, J.Z. and W.Y.; Writing—original draft, J.Z.; Writing—review & editing, W.Y. and F.Z.; Visualization, J.Z. and F.Z.; Supervision, W.Y. and F.Z.; Project administration, W.Y.; Funding acquisition, W.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was jointly funded by the National Key Research and Development Program of China (2022YFC3002601), and the Science and Technology Innovation Program of the Department of Transportation, Yunnan Province, China (2022-107).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The code of the study could be accessed at: <https://1drv.ms/f/s!AijNnfVhfkalrnIaw30xsGIU4Jn?e=YfGyM3> (accessed on 2 June 2024).

Acknowledgments: The authors would like to thank Li Li for the invaluable assistance with data processing in this study.

Conflicts of Interest: Author Wenchen Yang was employed by the company Broadvision Engineering Consultants Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Wang, Y.; Prato, C.G. Determinants of injury severity for truck crashes on mountain expressways in China: A case-study with a partial proportional odds model. *Saf. Sci.* **2019**, *117*, 100–107. [[CrossRef](#)]
2. Traffic Administration Bureau of Police Ministry. *Road Traffic Accident Annual Census Report of China*; Traffic Management Bureau: Beijing, China, 2017.
3. Yu, R.; Xiong, Y.; Abdel-Aty, M. A correlated random parameter approach to investigate the effects of weather conditions on crash risk for a mountainous freeway. *Transp. Res. Part C Emerg. Technol.* **2015**, *50*, 68–77. [[CrossRef](#)]
4. Khattak, A.J.; Ahmad, N.; Wali, B.; Dumbaugh, E. A taxonomy of driving errors and violations: Evidence from the naturalistic driving study. *Accid. Anal. Prev.* **2021**, *151*, 105873. [[CrossRef](#)] [[PubMed](#)]
5. Arvin, R.; Khattak, A.J.; Qi, H. Safety critical event prediction through unified analysis of driver and vehicle volatilities: Application of deep learning methods. *Accid. Anal. Prev.* **2021**, *151*, 105949. [[CrossRef](#)] [[PubMed](#)]
6. Fitch, G.M.; Hanowski, R.J. Using naturalistic driving research to design, test and evaluate driver assistance systems. In *Handbook of Intelligent Vehicles*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 559–580.
7. Singh, H.; Kathuria, A. Analyzing driver behavior under naturalistic driving conditions: A review. *Accid. Anal. Prev.* **2021**, *150*, 105908. [[CrossRef](#)] [[PubMed](#)]
8. Guo, F.; Klauer, S.G.; Hankey, J.M.; Dingus, T.A. Near crashes as crash surrogate for naturalistic driving studies. *Transp. Res. Rec.* **2010**, *2147*, 66–74. [[CrossRef](#)]
9. Wu, K.F.; Aguero-Valverde, J.; Jovanis, P.P. Using naturalistic driving data to explore the association between traffic safety-related events and crash risk at driver level. *Accid. Anal. Prev.* **2014**, *72*, 210–218. [[CrossRef](#)] [[PubMed](#)]
10. Arvin, R.; Kamrani, M.; Khattak, A.J. Examining the role of speed and driving stability on crash severity using shrp2 naturalistic driving study data. In Proceedings of the Transportation Research Board 98th Annual Meeting, 2019, Washington, DC, USA, 13–17 January 2019.
11. Arvin, R.; Khattak, A.J. Driving impairments and duration of distractions: Assessing crash risk by harnessing microscopic naturalistic driving data. *Accid. Anal. Prev.* **2020**, *146*, 105733. [[CrossRef](#)] [[PubMed](#)]
12. Khattak, Z.H.; Fontaine, M.D.; Li, W.; Khattak, A.J.; Karnowski, T. Investigating the relation between instantaneous driving decisions and safety critical events in naturalistic driving environment. *Accid. Anal. Prev.* **2021**, *156*, 106086. [[CrossRef](#)]
13. Dingus, T.A.; Klauer, S.G.; Neale, V.L.; Petersen, A.; Lee, S.E.; Sudweeks, J.; Knipling, R.R. The 100-Car naturalistic driving study. In *Phase 2, Results of the 100-Car Field Experiment*; Department of Transportation, National Highway Traffic Safety Administration: Washington, DC, USA, 2006.
14. Bagdadi, O. Assessing safety critical braking events in naturalistic driving studies. *Transp. Res. Part F Traffic Psychol. Behav.* **2013**, *16*, 117–126. [[CrossRef](#)]
15. Abbas, M.; Higgs, B.; Medina, A.; Yang, C. Identification of warning signs in truck driving behavior before safety-critical events. In Proceedings of the 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), Washington, DC, USA, 5–7 October 2011; pp. 558–563.
16. Sudweeks, J.D. Using Functional Classification to Enhance Naturalistic Driving Data Crash/Near Crash Algorithms. In *National Surface Transportation Safety Center for Excellence*; Report 15-UT-030; Virginia Tech Transportation Institute: Blacksburg, VA, USA, 2015.
17. Perez, M.A.; Sudweeks, J.D.; Sears, E.; Antin, J.; Lee, S.; Hankey, J.M.; Dingus, T.A. Performance of basic kinematic thresholds in the identification of crash and near-crash events within naturalistic driving data. *Accid. Anal. Prev.* **2017**, *103*, 10–19. [[CrossRef](#)]
18. Wu, K.F.; Jovanis, P. Screening naturalistic driving study data for safety-critical events. *Transp. Res. Rec. J. Transp. Res. Board* **2013**, *2386*, 137. [[CrossRef](#)]
19. Osman, O.A.; Hajj, M.; Karbalaieali, S.; Ishak, S. Crash and near-crash prediction from vehicle kinematics data: A SHRP2 naturalistic driving study. In Proceedings of the Transportation Research Board 97th Annual Meeting 2018, Washington, DC, USA, 7–11 January 2018. No. 18-03927.
20. Shi, X.; Wong, Y.D.; Li, M.Z.-F.; Palanisamy, C.; Chai, C. A feature learning approach based on XGBoost for driving assessment and risk prediction. *Accid. Anal. Prev.* **2019**, *129*, 170–179. [[CrossRef](#)]
21. Kluger, R.; Smith, B.L.; Park, H.; Dailey, D.J. Identification of safety-critical events using kinematic vehicle data and the discrete Fourier transform. *Accid. Anal. Prev.* **2016**, *96*, 162–168. [[CrossRef](#)] [[PubMed](#)]
22. Shi, L.; Qian, C.; Guo, F. Real-time driving risk assessment using deep learning with XGBoost. *Accid. Anal. Prev.* **2022**, *178*, 106836. [[CrossRef](#)] [[PubMed](#)]
23. Soydaner, D. A comparison of optimization algorithms for deep learning. *Int. J. Pattern Recognit. Artif. Intell.* **2020**, *34*, 2052013. [[CrossRef](#)]

24. Papazikou, E.; Quddus, M.; Thomas, P.; Kidd, D. What came before the crash? An investigation through SHRP2 NDS data. *Saf. Sci.* **2019**, *119*, 150–161. [[CrossRef](#)]
25. Hankey, J.M.; McClafferty, J.A.; Perez, M.A. *Description of the SHRP 2 Naturalistic Database and the Crash, Near-Crash, and Baseline Data Sets*; Virginia Tech Transportation Institute: Blacksburg, Virginia, 2016.
26. Shangquan, Q.; Fu, T.; Wang, J.; Luo, T.; Fang, S. An integrated methodology for real-time driving risk status prediction using naturalistic driving data. *Accid. Anal. Prev.* **2021**, *156*, 106122. [[CrossRef](#)]
27. Yu, B.; Bao, S.; Chen, Y.; LeBlanc, D.J. Effects of an integrated collision warning system on risk compensation behavior: An examination under naturalistic driving conditions. *Accid. Anal. Prev.* **2021**, *163*, 106450. [[CrossRef](#)]
28. Li, Z.; Liu, F.; Yang, W.; Peng, S.; Zhou, J. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 6999–7019. [[CrossRef](#)]
29. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [[CrossRef](#)] [[PubMed](#)]
30. Ma, X.; Dai, Z.; He, Z.; Ma, J.; Wang, Y.; Wang, Y. Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction. *Sensors* **2017**, *17*, 818. [[CrossRef](#)] [[PubMed](#)]
31. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
32. Yu, Y.; Si, X.; Hu, C.; Zhang, J. A review of recurrent neural networks: LSTM cells and network architectures. *Neural Comput.* **2019**, *31*, 1235–1270. [[CrossRef](#)] [[PubMed](#)]
33. Wang, Y.; Liu, J.; Qian, G. Hierarchical FFT-LSTM-GCN based model for nuclear power plant fault diagnosis considering spatio-temporal features fusion. *Prog. Nucl. Energy* **2024**, *171*, 105178. [[CrossRef](#)]
34. Niu, Z.; Zhong, G.; Yu, H. A review on the attention mechanism of deep learning. *Neurocomputing* **2021**, *452*, 48–62. [[CrossRef](#)]
35. Shan, J.; Huang, P.; Loong, C.N.; Liu, M. Rapid full-field deformation measurements of tall buildings using UAV videos and deep learning. *Eng. Struct.* **2024**, *305*, 117741. [[CrossRef](#)]
36. Yang, J.; Tang, D.; Yu, J.; Zhang, J.; Liu, H. Explaining anomalous events in flight data of UAV with deep attention-based multi-instance learning. *IEEE Trans. Veh. Technol.* **2024**, *73*, 107–119. [[CrossRef](#)]
37. Li, J.; Jiang, T.; Liu, H.; Sun, Y.; Lv, C.; Li, Q.; Liu, Y. Lane changing maneuver prediction by using driver's spatio-temporal gaze attention inputs for naturalistic driving. *Adv. Eng. Inform.* **2024**, *61*, 102529. [[CrossRef](#)]
38. Lyu, N.; Wang, Y.; Wu, C.; Peng, L.; Thomas, A.F. Using naturalistic driving data to identify driving style based on longitudinal driving operation conditions. *J. Intell. Connect. Veh.* **2022**, *5*, 17–35. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.