Scientific Research Publishing

Journal of Transportation Technologies, 2023, 13, 158-172
https://www.scirp.org/journal/jtts
ISSN Online: 2160-0481
ISSN Print: 2160-0473

Application of LiDAR Data for Deep Learning Based Near Crash Prediction at Signalized Intersection

Jewel Rana Palit[1*], Osama A. Osman[2*#]

[1]Department of Civil and Chemical Engineering, University of Tennessee, Chattanooga, USA
[2]Center for Urban Informatics and Progress, University of Tennessee, Chattanooga, USA
Email: jnc411@mocs.utc.edu, #osama-osman@utc.edu

How to cite this paper: Palit, J.R. and Osman, O.A. (2023) Application of LiDAR Data for Deep Learning Based Near Crash Prediction at Signalized Intersection. *Journal of Transportation Technologies*, **13**, 158-172.
https://doi.org/10.4236/jtts.2023.132008

Received: January 13, 2023
Accepted: February 25, 2023
Published: February 28, 2023

Abstract

Near crash events are often regarded as an excellent surrogate measure for traffic safety research because they include abrupt changes in vehicle kinematics that can lead to deadly accident scenarios. In this paper, we introduced machine learning and deep learning algorithms for predicting near crash events using LiDAR data at a signalized intersection. To predict a near crash occurrence, we used essential vehicle kinematic variables such as lateral and longitudinal velocity, yaw, tracking status of LiDAR, etc. A deep learning hybrid model Convolutional Gated Recurrent Neural Network (CNN + GRU) was introduced, and comparative performances were evaluated with multiple machine learning classification models such as Logistic Regression, K Nearest Neighbor, Decision Tree, Random Forest, Adaptive Boost, and deep learning models like Long Short-Term Memory (LSTM). As vehicle kinematics changes occur after sudden brake, we considered average deceleration and kinematic energy drop as thresholds to identify near crashes after vehicle braking time t_b. We looked at the next 3 seconds of this braking time as our prediction horizon. All models work best in the next 1-second prediction horizon to braking time. The results also reveal that our hybrid model gathers the greatest near crash information while working flawlessly. In comparison to existing models for near crash prediction, our hybrid Convolutional Gated Recurrent Neural Network model has 100% recall, 100% precision, and 100% F1-score: accurately capturing all near crashes. This prediction performance outperforms previous baseline models in forecasting near crash events and provides opportunities for improving traffic safety via Intelligent Transportation Systems (ITS).

*Equally contributed to the work.
#Corresponding author.

Keywords

Near Crash Prediction, Machine Learning, Kinematics, Convolutional Gated Recurrent Neural Network, Recall

1. Introduction

The World Health Organization (WHO) estimates that almost 1.3 million individuals lose their lives in road accidents every year [1]. 42,915 people died in motor vehicle traffic crashes last year in the United States, up 10.5% from the 38,824 fatalities in 2020, according to the National Highway Traffic Safety Administration (NHTSA) [2]. In addition to being the biggest annual percentage increase in the history of the Fatality Analysis Reporting System, the anticipated death toll is also the highest since 2005. With the exponential increase in population, traffic crashes continue to increase and take a heavy toll on human lives and economy which necessitates the scope for investigating crashes and near crash scenarios [3]. While crashes imply the event of collision, a near crash is any situation where the participating vehicle, or any other vehicle, pedestrian, bicycle, or animal, must make a quick, evasive maneuver to escape a collision [4]. It can be quantified by time to collision, speed, acceleration, etc. Prior studies have concentrated on in-depth analysis of collision and near-collision occurrences to improve traffic safety systems and adapt to dynamic traffic changes on the road.

Numerous ways have been used in recent years to examine and comprehend traffic crash occurrence, risk factors, and injury severity. These methodologies include statistical, machine learning, and deep learning models. For instance, Chen *et al.* applied a logistic regression model to analyze crash severity and obtained seven risk factors including driver age, gender, traffic control type, etc. [5]. Rezapour *et al.* used ordinal logistic regression and multinomial regression to investigate the crucial factors for serious truck and automobile collisions [6]. In their analysis of accidents involving powered two-wheelers, Montella *et al.* used a decision tree model and association rules, and they found that curve alignment, rural regions, run-off-the-road wrecks, nighttime, and wet weather highly influence accident severity [7]. However, they usually relied on official accident reports, which are difficult to acquire and lack thorough driving data [8].

Wang *et al.* found that Artificial Neural Network outperforms the previous model while predicting driving risk based on real-world data incorporating features like speed, fuel consumption, etc. [9]. In 2020, Lie *et al.* merged Convolutional Neural Network (CNN) with Long Short-Term Memory (LSTM), a CNN-LSTM approach, to predict real-time crash risk [10]. Yu *et al.* proposed a CNN methodology with a refined loss function to analyze risk severity [11]. However, these studies mainly address factors behind the crash occurrence. To prevent traffic crash, it is important to investigate more critical safety event that triggers the crash such as a near crash. As traffic crash incidents usually are reported in official records, these near crash events go unnoticed which could have turned into fatal crashes. In recent times, the influence on safety is typically evaluated using a near crash as a popular surrogate measure as it is seen that using near crashes as a crash surrogate could provide definite benefits when data about enough crashes are not available [12]. In essence, analyzing and predicting near crash events can substantially assist in reducing the danger to drivers and enhancing overall safety [13].

Emerging data collection system like Naturalistic Driving Study (NDS) enables that scope to extract in-depth crash and near crash data [13]. In 2006, NHTSA took large-scale initiatives to collect naturalistic driving data through "100-Car Naturalistic Driving Study" which collected data regarding road behavior and performance, such as excessive tiredness, impairment, mistakes in judgment, carelessness, willingness to take risks, and eagerness to engage in secondary activities [14]. Numerous behavior-based studies, such as those that examined the validity of using near-collisions as crash surrogates, made use of the 100-Car study data [15]. Jovanis *et al.* used NDS data to estimate the probability of near crashes and crash events [16]. Seacrist *et al.* used SHRP2 NDS dataset to analyze near crash characteristics among risky drivers [17]. Perez *et al.* used naturalistic driving data to identify crash and near crash events on kinematic threshold [18]. Osman *et al.* used high resolution SHRP2 NDS dataset to test several models like adaptive boost (AdaBoost), K Nearest Neighbor (KNN), Support Vector Machine (SVM) etc. to predict near crash based on vehicle kinematic factors [13]. Nazi *et al.* predicted risk level near crash event using classification approach on NDS dataset [19]. With significant improvement in data collection and analysis, NDS data still suffer few disadvantages. NDS devices are costly and have low coverage. Only the NDS equipped vehicle and its surrounding environments are covered by NDS data. Near crash events won't be reported if there aren't enough vehicles using the required NDS devices for a certain section of the road [20].

1.1. Research Gap & Contribution

Existing literature investigates crashes and near crashes with datasets that have a wide range of limitations in terms of coverage, unreported events, visibility issues, etc. We now have access to a wide range of data sources thanks to intelligent transportation systems (ITS), which help us better understand travel behavior, traffic flow, and other factors that affect traffic safety, planning and policy making [21]. And with the advancement in data collection technology in ITS, newer and more precise methods have been adopted in contemporary research of traffic safety such as Light Detection and Ranging (LiDAR). Data collection advancement through LiDAR has surpassed previous methodologies in terms of coverage and data collection limitations. While NDS is only a single vehicle system, LiDAR accounts for multiple traffic agents. While in NDS, onboard sensors are prone to visibility obstruction, LiDAR is free from the issue. Due to its huge advantage over other methods, it has been used frequently to detect crash and near crash events. It can not only investigate vehicle-to-vehicle crashes but also can investigate vehicle-pedestrian conflicts. Lv *et al.* identified vehicle-pedestrian conflicts using road user trajectories from roadside LiDAR [22]. Wu *et al.* used LiDAR data for vehicle pedestrians near crashes [20]. Even though several recent research has utilized LiDAR datasets to identify surrogate metrics like near crashes, machine learning, and deep learning models have not been actively deployed to predict these measures. This paper aims to fill this gap by investigating the benefits of LiDARs as a powerful data source to predict vehicle to vehicle near crash events at signalized intersections. We develop and comparatively evaluate various advanced and traditional machine learning and deep learning models to predict near crashes using LiDAR data. This work could support numerous applications such as cooperative perception for cooperative driving automation technology.

2. Methodology

During near crash events, every vehicle goes through a sudden abrupt change in vehicle kinematics [13] such as speed, acceleration, kinetic energy drop etc. Several thresholds have been established on vehicle kinematics to filter out near crash events. Time To Collision (TTC) is one of the popular measures which is defined as the travel time difference between following and leading vehicles [23]. the TTC parameter presumes that objects move at a constant pace, which could not accurately reflect the scenario as it doesn't consider deceleration/acceleration [20]. To detect the incidence of near crashes, Wang *et al.* utilized a threshold of average deceleration after braking is less than −1.027 m/s^2 and more than 30% kinetic energy reduction due to brake as risky near crash conditions [8]. We use these measures as our threshold to filter out near crash incidents. **Figure 1** shows one such event where the vehicle is currently at A. When it crosses close to the near crash area B, it encounters another vehicle at C, then brakes and starts to decelerate. After a while, if the leading vehicle clears out, it increases speed again. This sudden brake at B creates abrupt changes in vehicle kinematics such as acceleration, speed, etc. based on which we can compare this with the threshold to decide if it is a near crash or not.

If time to reach at braking trigger point B = t_b,

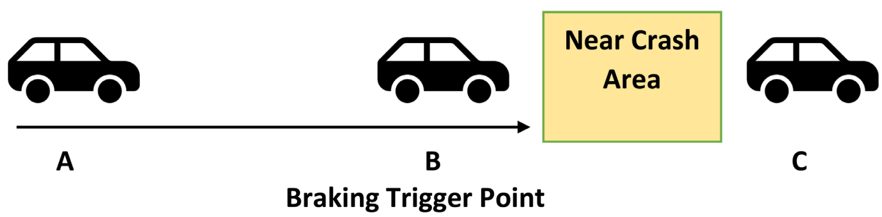


Figure 1. Near crash scenario.

velocity at that point = $v(t)_b$,
time to reach the point of maximum deceleration = t_1,
and car mass = m,
Average acceleration/deceleration,

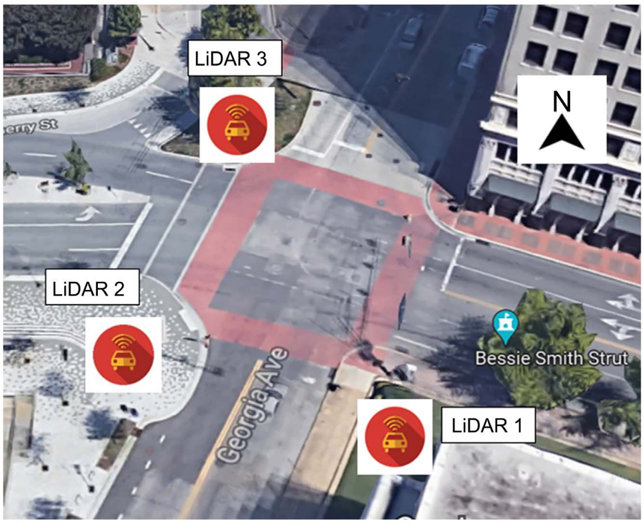
$$a_{avg} = \frac{1}{t_1 - t_b} \int_{t_b}^{t_1} a(t)\,\mathrm{d}t \tag{1}$$

% Kinetic energy reduction [8],

$$KE = \frac{\left(\frac{1}{2}\right) mv(t_b)^2 - \left(\frac{1}{2}\right) mv(t_1)^2}{\left(\frac{1}{2}\right) mv(t_b)^2} = 1 - \left[\frac{v(t_1)}{v(t_b)}\right]^2 \tag{2}$$

2.1. Study Area and Data Collection

Intersections are the most dangerous area with numerous conflict points. In this study, we investigate such a busy signalized intersection, Georgia Avenue at MLK corridor in Chattanooga, Tennessee. We use LiDAR dataset Transportation Forecasting Competition (TRANSFOR 22) collected at Georgia Ave. To record the real-time movement of all road users, including automobiles, pedestrians, cyclists, etc., three Ouster® OS1-128 LiDAR sensors were permanently mounted on light poles which have a rotation frequency of 10 HZ. Software developed by Seoul Robotics has pre-processed the raw LiDAR point cloud data. The LiDAR 2 was chosen as the origin after combining the point clouds from three LiDAR sensors, and the four corners of the highlighted crossing zones are A (13.505 m, 14.413 m), B (9.874 m, 5.121 m), C (17.368 m, 4.792 m), and D (4.242 m, 24.612 m). Figure 2(a) shows the position of these LiDARs on Georgia Avenue and Figure 2(b) shows LiDAR recorded data of the detected object. The detected road user information outputs are saved in CSV file including attributes shown in Table 1.



(a)

(b)

Figure 2. Data collection site (a) Location, (b) Trajectories in LiDAR.

Table 1. LiDAR detected attributes.

Attribute	Unit	Note
Timestamp	millisecond	Unix timestamp of LiDAR input message
ID		The ID of the object
Label		None (0), Car (1), Pedestrian (2), Cyclist (3), Miscellaneous (4)
Confidence		Confidence of tracking quality (0.0 - 1.0)
BBOX_position_x	meter	Center X co-ordinate of bounding box
BBOX_position_y	meter	Center Y co-ordinate of bounding box
BBOX_size_x	meter	Longitudinal length of the bounding box (relative to yaw)
BBOX_size_y	meter	Lateral length of the bounding box (relative to yaw)
BBOX_size_z	meter	Height of bounding box
Velocity_x	Meter/second	Velocity in longitudinal direction
Velocity_y	Meter/second	Velocity in lateral direction
BBOX_yaw	radian	Heading (0.0 - 2.0 Pi)
Tracking Status		None (0), Validating (1), Invalidating (2), Tracking (3), Drifting (4), Expired (5) **Validating:** checking validity in the early stage of tracking. **Invalidating:** short-term prediction when tracking is lost in Validating status. **Tracking:** stable tracking. **Drifting:** short-term prediction when tracking is lost in Tracking status. **Expired:** expired tracking.

2.2. Data Processing and Prediction Modeling

The competition dataset is collected on 8 October 2021 for 75 minutes from 3 PM to 4:15 PM. Figure 3 shows the trajectories of vehicles on Georgia Avenue. From the figure, we can see that huge clusters of thru movement and left movement from the southbound direction. It makes sense as during this time office hours come to an end and people tend to return from offices which are mainly

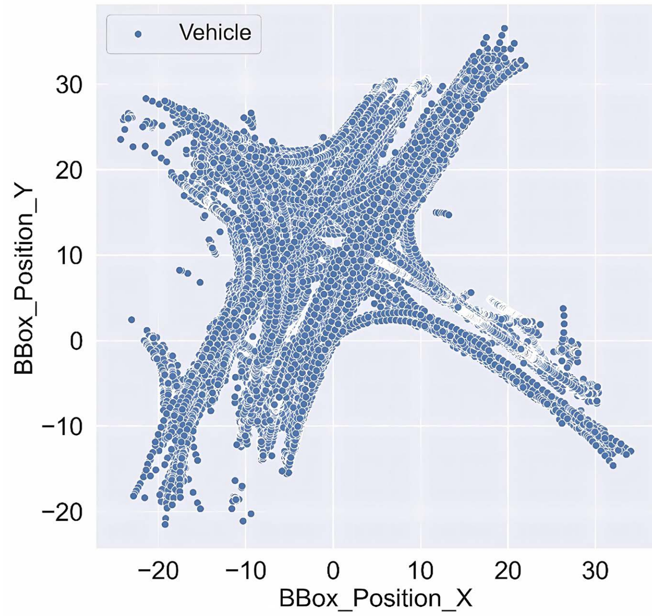


Figure 3. Object trajectories from LiDAR data.

in the downtown toward the southbound part. The "Label" attribute provides us with four different object trajectories: Car, Pedestrian, Cyclist, and Miscellaneous. For our purpose we only consider cars. We filter out other trajectories except for the car and consider this for the next processing. Then rest of the 12 attributes with 251,947 vehicle trajectories construct a 251,947 × 12 vector set to consider for further processing which is shown in **Figure 4** framework. After that, every single complete event/trajectory for every single unique vehicle given by their "ID" is stacked together according to their timestamp as a vector set of $E = \{Et_1, Et_2, \cdots, Et_n\}$, where $t_1, t_2, \cdots, t_n$ are the timestamp of each event start. Approximately 1500 such unique vehicle events are filtered with a threshold filtering process with 2 components: average deceleration after brake and reduction in kinetic energy. This filter is represented as $\{a_{avg} < -1.027, KE > 30\}$ which are calculated through Equation (1) and (2) for each event discussed avobe. Through this filter, every event entering the predictive models as labeled dataset of normal event (labeled as 0) and near crash (labeled as 1). And finally, the output layer provides binary classification prediction output of the same.

For our Convolutional Gated Recurrent Neural Network (CNN + GRU), we use conventional supervised methods of dataset *R*, where *R*: (x_1, y_1), (x_2, y_2) … (x_i, y_i). Here x_i are input parameters and y_i are labeled target data which take two values Y_1 or Y_2 as shown in **Figure 4** where $Y_1 \to 0$: Normal Event and $Y_2 \to 1$: Near Crash . While filtering out with threshold of $\{a_{avg} < -1.027, KE > 30\}$, we look at a short-term 3-second prediction horizon to build up target y_i. As we know for every single event Et_n, t_b is the braking time, and if this abrupt change crosses the threshold or not, we make this decision in the next 3 second: t_{b+1}, t_{b+2} and t_{b+3}. Event Labelling after filtering includes inputs at a particular time and its three-prediction horizon labeled dataset

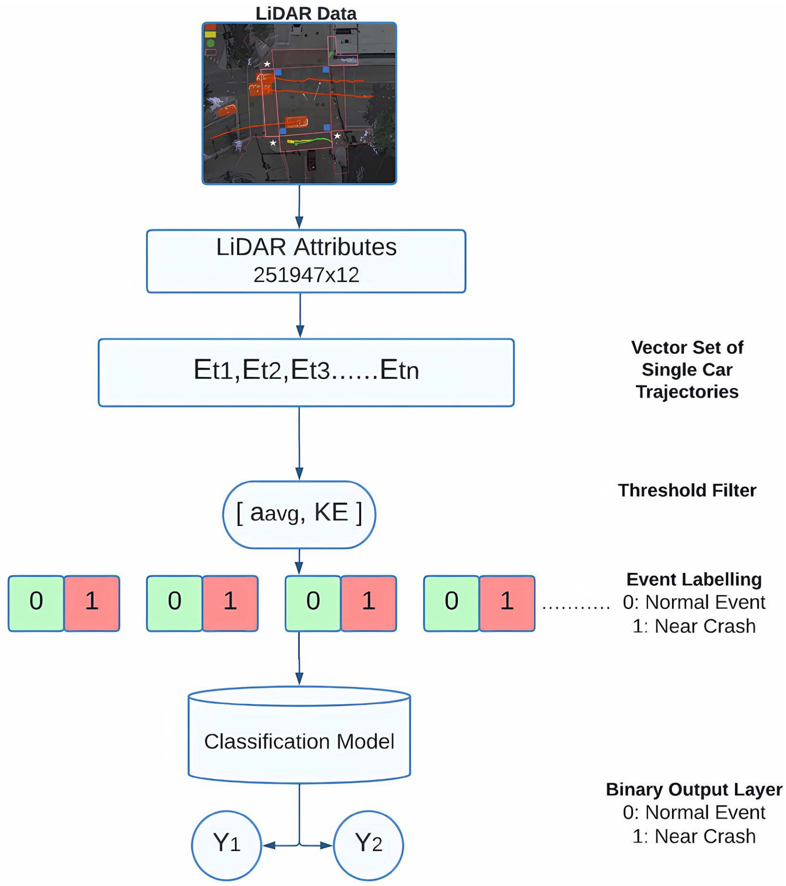


Figure 4. Framework of proposed methodology.

to construct the total readied data frame for prediction. It can be expressed as $R: R \rightarrow \{x_i, y_i(t_{b+1})\}$, $\{x_i, y_i(t_{b+2})\}$, $\{x_i, y_i(t_{b+3})\}$. We use similar methodology for other machine learning and deep learning models used to evaluate comparative performance with our hybrid CNN + GRU. For model inputs, all attributes in Table 1 are used except "Label" which gives us total 12 inputs for 1500 unique vehicle. Thus, the predictive model input dimension shapes at 1500 × 12. All the input features are normalized within a same scale between 1 and 0 to ensure all parameters are treated equally by the machine learning and deep learning models. For each input value x, normalized value x_n is given by,

$$x_n = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{3}$$

3. Experimental Analysis

We train several machine learning and deep learning models such as Logistic Regression, K Nearest Neighbor, Decision Tree, AdaBoost, and deep learning models like Long Short-Term Memory (LSTM) and compare performance with our hybrid Convolutional Gated Recurrent Neural Network (CNN + GRU) model shown in Figure 5. All models are trained and tested at 75% - 25% split. Model hyperparameters are summarized in Table 2 respectively. We mainly

Table 2. Hyperparameters of the model.

Logistic Regression	KNN	Decision Tree	AdaBoost	LSTM	CNN + GRU
Solver: Liblinear	Neighbor no. = 6	Max. Depth = 12	Estimator = 100, Learning rate = 0.005	Three layers: 2LSTM layer followed by 1 dense layer, Optimizer = Adam, Learning rate = 0.005, Batch size = 16.	Four layers: 1 1-D convolution layer followed by 1 dense layer, 1 GRU layer and 1 dense layer, Activation = softmax, Optimizer = Adam, Learning rate = 0.005, Batch size = 16.

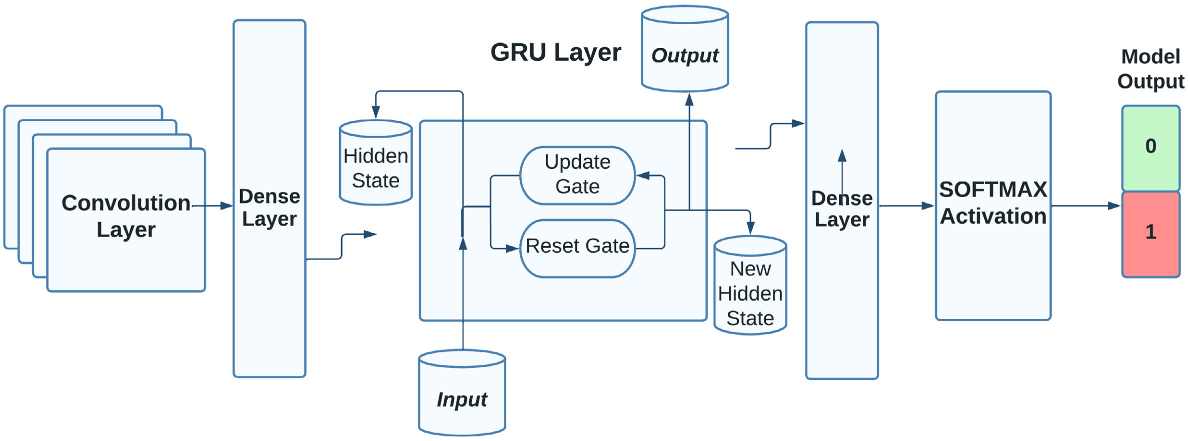


Figure 5. Framework of hybrid CNN + GRU.

consider Recall, Precision, and F-1 score for our evaluation measure. Recall indicates how many positive events are predicted correctly over all the all-positive events. And how many positive event predictions are correct are indicated by Precision. A model can have a higher Precision value but lower Recall. Also, it can be of high Recall value with lower Precision. However, a good model has a highly balanced Precision and Recall score. F1 score gives the same weight to the Precision and Recall while measuring the accuracy. Higher F-1 score represents better performing model. Following equations are used for these measures.

$$\text{Recall} = \frac{\text{No. of near crash predicted correctly}}{\text{No. of near crash predicted correctly} + \text{No. of wrongly predicted near crash which are normal event}} \tag{4}$$

$$\text{Precision} = \frac{\text{No. of near crash predicted correctly}}{\text{No. of near crash predicted correctly} + \text{No. of wrongly predicted normal event which are near crash}} \tag{5}$$

$$\text{F1 score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6}$$

Recall, Precision and F1 score are compared separately in Figures 6(a)-(c) consecutively for all six models. We can see clearly that logistic regression performs the poorest being the simplest model in case of Recall, Precision and F-1 score. Lowest Recall value for logistic regression in Figure 6(a) expresses that it misses many near crash events and predict them as normal events. With the increase of prediction horizon, the performance goes down a bit. Same trend is

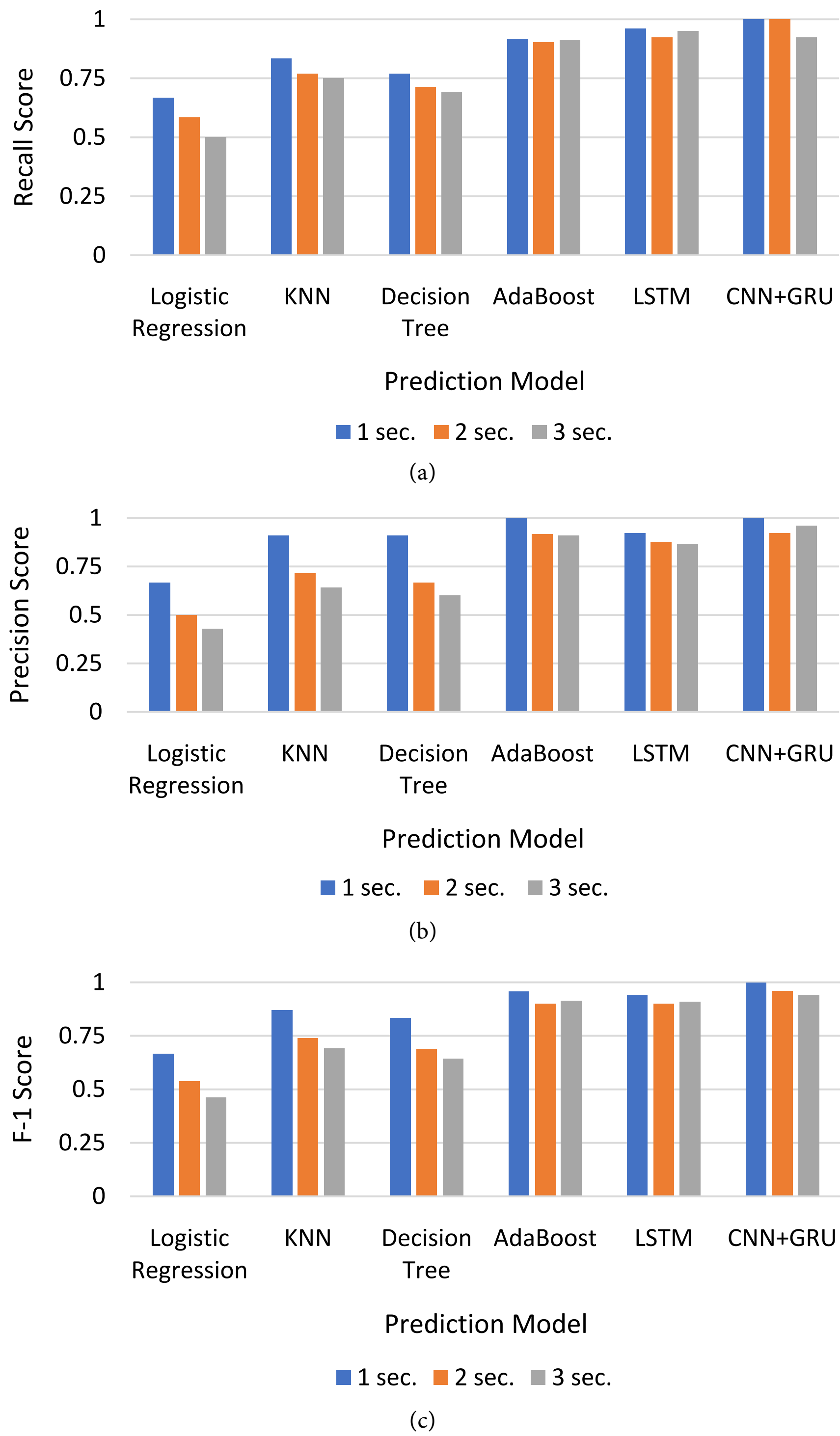


Figure 6. Recall (a), Precision (b) and F-1 Score (c) comparison.

noticed in almost all models. It makes sense as broader prediction horizon adds more dynamic non-linearity in the data hence making the prediction a bit difficult. The Recall value is higher in KNN in comparison to Decision Tree which expresses that KNN is better at capturing near crash events instead of mistaking them as normal events. AdaBoost, LSTM, and our hybrid CNN + GRU have higher Recall than the previous three models with a good gap in performance. It shows how good they are at classifying near crash and normal events. Model AdaBoost, LSTM and our hybrid model CNN + GRU keep consistent performance from lower to higher horizons showing little variation in the performance due to added traffic dynamics though a minimal variation is noticed in the 2-second horizon. Figure 6(b) shows the reduced Precision value for AdaBoost and LSTM after the 1-second horizon referring to the fact that in broader prediction horizon these models are predicting more normal events as near crash events increasing the false positive. Though LSTM had higher Recall than AdaBoost, we can see AdaBoost has higher Precision indicating that it has fewer wrong predictions of normal events which are near crash. At a prediction horizon of 1 second, CNN + GRU seems to have perfect 100 % precision, recall, and F-1 score proving 100% accuracy referring to the fact that it classifies all normal events and near crash accurately with 0 False positive which is very encouraging and shown in Figure 7 through confusion matrix. Even in broader horizons it shows nearly 95% recall and precision which surpasses previous models even in their best-performing prediction horizons. Figure 6(c) shows similar trend in F-1 score confirming the results of Recall and Precision have been balanced. We can also notice that there is a considerable gap in F-1 score from Prediction Horizon of 1 second to 3 seconds in case of Logistic Regression, Decision Tree, and KNN. On the other hand, it is more stable in the case of LSTM and our hybrid CNN + GRU. The F-1 score is the highest in our hybrid CNN + GRU which strengthens its superiority to other tested models in this study.

Overall results are expressed in Table 3 shows performance measures of all 6 models in all three prediction horizons in details. We see a huge margin of performance accuracy jump from Decision Tree to AdaBoost. It seems AdaBoost performs well for a machine-learning model in this classification problem. LSTM

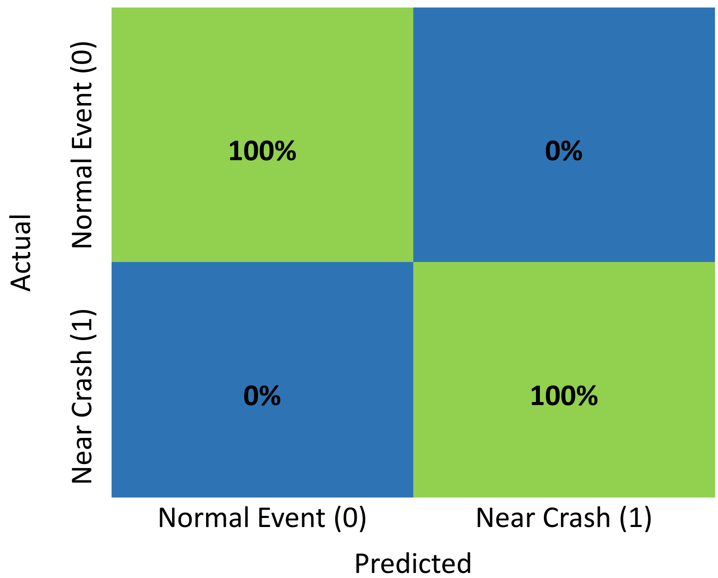


Figure 7. Confusion matrix of CNN + GRU.

Table 3. Performance measure comparison of all models.

Prediction Horizon	Measure	Log. Regression	KNN	Decision Tree	AdaBoost [13]	LSTM [19]	CNN + GRU
1 sec.	Precision	0.667	0.909	0.909	1.000	0.923	1.000
	Recall	0.667	0.833	0.769	0.917	0.960	1.000
	F1	0.667	0.870	0.833	0.957	0.941	1.000
2 sec.	Precision	0.500	0.714	0.667	0.917	0.877	0.923
	Recall	0.583	0.769	0.714	0.902	0.923	1.000
	F1	0.538	0.741	0.690	0.900	0.900	0.960
3 sec.	Precision	0.429	0.643	0.600	0.909	0.867	0.960
	Recall	0.500	0.750	0.692	0.913	0.950	0.923
	F1	0.462	0.692	0.643	0.915	0.910	0.941

and hybrid model CNN + GRU provide the best accuracy as expected from deep learning supremacy. Our hybrid model CNN + GRU seems to perform over 92% - 94% accuracy in all prediction horizons which shows stable higher performance capability with a minimal training time of 2 - 3 minutes while LSTM takes 6 - 7 min training time. AdaBoost and LSTM are the latest models that have been adopted in near crash prediction. AdaBoost requires for the best performance horizon: 1 second, AdaBoost has 100% precision same as hybrid CNN + GRU performing better than LSTM which is at 92.3% precision score. It indicates that AdaBoost doesn't have any False positives meaning it never considers any normal event as near crash. LSTM has far less precision of 92.3% and a higher recall of 96% which indicates that it classifies many near crash as normal events which should be avoided. Though in horizons 2 seconds and 3 seconds, LSTM proves to outperform AdaBoost in the recall, overall performance is close to each other. However, CNN + GRU outperforms all the models in all prediction horizons with the best accuracy on a horizon of 1 second. Our CNN + GRU model outperforms baseline models for near crash prediction as well as the latest models like AdaBoost and LSTM in all evaluation measures.

4. Conclusions

Traditional crash and near crash data collection approaches in the literature suffer from numerous constraints such as unreported events, visibility issues, excessive costs, etc. which have been addressed successfully through LiDAR technology. LiDAR has provided a paradigm shift in analyzing crucial surrogate measures, near crash events overcoming limitations of previous data collection approaches. In this study, we used LiDAR data collected on a signalized intersection of a downtown busy road to successfully predict near crash events based on vehicle kinematics with several machine learning and deep learning algorithms. We considered the sudden change in vehicle kinematics that occurred in near crash scenarios such as average deceleration from the braking time to the minimum deceleration and kinematic energy reduction as our thresholds for near crash identification. A wide range of vehicle kinematics data including lateral and longitudinal velocity, yaw rate, the confidence of LiDAR detection, tracking status of LiDAR, etc. was considered as model inputs. Machine Learning models like Logistic Regression, Decision Tree, KNN, AdaBoost, and deep learning models like LSTM and hybrid CNN + GRU were considered to predict near crash events. All models were trained and tested with the same evaluation measures: Recall, Precision and F-1 Score with a sensitivity analysis from 1-second to 3-second prediction horizons of braking event. It has been proved that typical machine learning models like Logistic Regression, Decision Tree and KNN perform worse than other deep learning methods except for AdaBoost. AdaBoost competes with LSTM on par with a slightly higher precision but less recall than LSTM. Our hybrid model CNN + GRU with minimal training time outperform not only LSTM but also other existing baseline methods in all prediction horizon including 100% recall, 100% precision, and 100% F-1 score in 1-second prediction horizon providing 100% accuracy. The CNN + GRU model performs best under future 1-second prediction horizon while outperforming other models in all prediction horizons with accuracy varying from 95% to 100%.

In further research, we will examine longer-term LiDAR data to identify the precise trend of vehicle trajectories. We will also investigate conflicts between vehicles and pedestrians by including other road users like cyclists and pedestrians. Overall, the results of this study inspire us and provide us with an interesting opportunity to examine near crash prediction from a variety of angles. For such near crash and crash prediction, using LiDAR data with high coverage will be helpful in integrating a real-time near crash detection system in traffic regulating devices and could be used in the autonomous system with great accuracy, opening new windows to develop safer traffic automation along the Intelligent Transportation System corridor and raising traffic safety in an affordable manner.

Acknowledgements

We would like to also acknowledge Dr. Mina Sartipi for her support in data accessing and Mr. Austin Harris for his support in the data extraction effort. We also would like to acknowledge Ms. Junxuan Zhao for her collaboration in constructive discussions in the early stages of the work.

Author Contributions

The authors confirm contribution to the paper as follows: study conception and design: O. A. Osman and J. R. Palit; data collection: O. A. Osman; analysis and interpretation of results: J. R. Palit and O. A. Osman; draft manuscript preparation: J. R. Palit and O. A. Osman. All authors reviewed the results and approved the final version of the manuscript.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] WHO (2022) Road Traffic Injuries. https://www.who.int/en/news-room/fact-sheets/detail/road-traffic-injuries

[2] NHTSA (2022) Newly Released Estimates Show Traffic Fatalities Reached a 16-Year High in 2021, in NHTSA Media.

[3] Palit, J.R. (2022) Application of Machine Learning and Deep Learning Approaches for Traffic Operation and Safety Assessment at Signalized Intersections. Masters Theses and Doctoral Dissertations, University of Tennessee at Chattanooga. https://scholar.utc.edu/theses/779

[4] Guo, F., *et al.* (2010) Evaluating the Relationship between Near-Crashes and Crashes: Can Near-Crashes Serve as a Surrogate Safety Metric for Crashes? Washington DC.

[5] Chen, H., Cao, L. and Logan, D.B. (2012) Analysis of Risk Factors Affecting the Severity of Intersection Crashes by Logistic Regression. *Traffic Injury Prevention*, **13**, 300-307. https://doi.org/10.1080/15389588.2011.653841

[6] Rezapour, M. and Ksaibati, K. (2018) Application of Multinomial and Ordinal Logistic Regression to Model Injury Severity of Truck Crashes, Using Violation and Crash Data. *Journal of Modern Transportation*, **26**, 268-277. https://doi.org/10.1007/s40534-018-0166-x

[7] Montella, A., *et al.* (2012) Analysis of Powered Two-Wheeler Crashes in Italy by Classification Trees and Rules Discovery. *Accident Analysis & Prevention*, **49**, 58-72. https://doi.org/10.1016/j.aap.2011.04.025

[8] Wang, J., *et al.* (2015) Driving Risk Assessment Using Near-Crash Database through Data Mining of Tree-Based Model. *Accident Analysis & Prevention*, **84**, 54-64. https://doi.org/10.1016/j.aap.2015.07.007

[9] Wang, Y., *et al.* (2017) Machine Learning Methods for Driving Risk Prediction. *Proceedings of the* 3*rd ACM SIGSPATIAL Workshop on Emergency Management Using*, Redondo Beach, 7 November 2017, Article No. 10. https://doi.org/10.1145/3152465.3152476

[10] Li, P., Abdel-Aty, M. and Yuan, J. (2020) Real-Time Crash Risk Prediction on Arterials Based on LSTM-CNN. *Accident Analysis & Prevention*, **135**, Article ID: 105371. https://doi.org/10.1016/j.aap.2019.105371

[11] Yu, R., *et al.* (2020) Convolutional Neural Networks with Refined Loss Functions for the Real-Time Crash Risk Analysis. *Transportation Research Part C: Emerging Technologies*, **119**, Article ID: 102740. https://doi.org/10.1016/j.trc.2020.102740

[12] Guo, F., *et al.* (2010) Near Crashes as Crash Surrogate for Naturalistic Driving Studies. *Transportation Research Record*, **2147**, 66-74. https://doi.org/10.3141/2147-09

[13] Osman, O.A., *et al.* (2019) Prediction of Near-Crashes from Observed Vehicle Kinematics Using Machine Learning. *Transportation Research Record*, **2673**, 463-473. https://doi.org/10.1177/0361198119862629

[14] Dingus, T.A., *et al.* (2006) The 100-Car Naturalistic Driving Study, Phase II-Results of the 100-Car Field Experiment. United States Department of Transportation. National Highway Traffic Safety.

[15] Jonasson, J.K. and Rootzén, H. (2014) Internal Validation of Near-Crashes in Naturalistic Driving Studies: A Continuous and Multivariate Approach. *Accident Analysis & Prevention*, **62**, 102-109. https://doi.org/10.1016/j.aap.2013.09.013

[16] Jovanis, P.P., *et al.* (2011) Analysis of Naturalistic Driving Event Data: Omitted-Variable Bias and Multilevel Modeling Approaches. *Transportation Research*

Record, **2236**, 49-57. https://doi.org/10.3141/2236-06

[17] Seacrist, T., *et al*. (2020) Near Crash Characteristics among Risky Drivers Using the SHRP2 Naturalistic Driving Study. *Journal of Safety Research*, **73**, 263-269. https://doi.org/10.1016/j.jsr.2020.03.012

[18] Perez, M.A., *et al*. (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. https://doi.org/10.1016/j.aap.2017.03.005

[19] Naji, H.A., *et al*. (2022) Risk Levels Classification of Near-Crashes in Naturalistic Driving Data. *Sustainability*, **14**, 6032. https://doi.org/10.3390/su14106032

[20] Wu, J., *et al*. (2018) A Novel Method of Vehicle-Pedestrian Near-Crash Identification with Roadside LiDAR Data. *Accident Analysis & Prevention*, **121**, 238-249. https://doi.org/10.1016/j.aap.2018.09.001

[21] Palit, J.R., Osman, O.A. and Rakha, H. (2023) Travel Patterns and Street Networks: A Novel Visualization Methodology of City-Level Traffic Flows and Network Usability. *Journal of Transportation Technologies*, **13**, 115-137. https://doi.org/10.4236/jtts.2023.131006

[22] Lv, B., *et al*. (2019) Automatic Vehicle-Pedestrian Conflict Identification with Trajectories of Road Users Extracted from Roadside LiDAR Sensors Using a Rule-Based Method. *IEEE Access*, **7**, 161594-161606. https://doi.org/10.1109/ACCESS.2019.2951763

[23] Smith, D.L., Najm, W.G. and Glassco, R.A. (2002) Feasibility of Driver Judgment as Basis for a Crash Avoidance Database. *Transportation Research Record*, **1784**, 9-16. https://doi.org/10.3141/1784-02