Proceedings of Machine Learning Research **318** (2026)
The 39th Canadian Conference on Artificial Intelligence
Canadian AI 2026
25–29 May 2026, Vancouver, BC

TAG-BiLSTM: A Temporal Attention-guided BiLSTM for Real-time Rear-end Collision Prediction in Intelligent Transportation Systems

Ebenezer Penoo[†,*], Thangarajah Akilan[†]
[†] Department of Electrical & Computer Engineering, Lakehead University
[‡] Department of Software Engineering, Lakehead University

Abstract

Rear-end collision prediction is a critical component of modern Intelligent Transportation Systems (ITS), enabling the early detection of hazardous driving conditions and supporting advanced driver assistance functions. Although recent deep learning (DL) approaches have shown strong predictive capability, many architectures, especially those relying on heavy attention or graph-based computations, suffer from high inference cost, limiting their suitability for real-time deployment on edge devices. To address this challenge, we propose a light-weight temporal attention-guided bidirectional long short-term memory (TAG-BiLSTM) network that efficiently models historical vehicle-kinematic sequences while capturing long-range temporal dependencies. The BiLSTM backbone encodes forward and backward motion context, while the attention gate selectively emphasizes the most safety-critical frames associated with rapid deceleration, headway compression, or abrupt motion changes. Experimental evaluation on benchmark open-source datasets, including the Next Generation Simulation (NGSIM)[1] and highD[2], demonstrates the robustness of the proposed model, achieving a mean performance exceeding 94%.

Keywords: Attention mechanism, collision prediction, deep learning, time series.

1. Introduction

Road traffic accidents [1], particularly rear-end collisions, represent a significant global public safety concern. The emergence of ITS has facilitated the integration of sensor data and DL methods for proactive prediction and mitigation of such incidents. Although existing research has made substantial progress in vehicle trajectory forecasting and anomaly detection, the specific task of real-time rear-end collision prediction remains challenging. For instance, the vehicle dynamics are inherently nonlinear and temporally dependent, requiring models that can capture sequential relationships in time-series data. However, real-world driving signals often exhibit measurement uncertainty and modeling complexity, which can challenge accurate system identification and prediction. Hence, models designed for strong predictive performance often rely on computationally intensive architectures that are impractical for real-time inference on in-vehicle processing units, which are typically resource-constrained. To address these, this work introduces a computationally efficient an eight layer recurrent neural network using BiLSTM and attention gating modules. Thus, the main contributions of this work are as follows: (i) An TAG-BiLSTM that effectively models the temporal evolution of vehicle kinematics for accurate collision prediction in dynamic traffic environments; (ii) A computationally efficient model configuration, demonstrating that TAG-BiLSTM can operate with low input dimensionality and short sequence lengths, making it suitable for real-time deployment in resource-constrained ITS; and (iii) A thorough experimental analysis of widely used benchmark datasets to compare performance metrics as well as to demonstrate generalization of the proposed model.

[1] Federal Highway Administration Intelligent Transportation Systems DataHub
[2] Developed by the Institute for Automotive Engineering, RWTH Aachen University

[*] epenoo@lakeheadu.ca

The remainder of the paper is organized as follows: Section 2 summarizes related developments in collision prediction and related sequence modeling approaches. Section 3 elaborates on the proposed architectural design, while Section 4 details the experimental setup and results. Lastly, Section 5 concludes the paper and discusses potential future research directions.

2. Related Work

As summarized in Table 1, this study taxonomically grouped the collision prediction models into four: (i) parametric approaches, (ii) traditional machine learning approaches, (iii) deep learning approaches, and (iv) hybrid approaches that combine multiple methodologies.

2.1. Parametric Approaches

Parametric approaches represent some of the earliest methods for collision prediction, relying on physics-based thresholds and kinematic models. Typical examples include time-to-collision (TTC)-based methods with fixed thresholds, such as the 2.2 s criterion used in [2]. This formulation underpins industrial forward collision warning (FCW) systems, exemplified by Honda's TTC-based approach (named as the Honda FCW algorithm), which is widely adopted as a baseline in collision prediction studies. The Berkeley (PATH) distance-to-collision (DTC) method [3] is a rule-based approach that estimates collision risk using deterministic kinematic relationships and conservative stopping-distance margins, often incorporating assumptions related to driver perception–reaction time (PRT). However, its reliance on fixed hyperparameter settings limits adaptability to diverse driving conditions, which may lead to conservative or suboptimal warning behavior and reduced driver trust.

Method	Key Idea	Pros and Cons
Parametric Approaches		
Honda [2]	Time-to-collision threshold	Simple, computationally efficient; relies on fixed setup and lacks adaptability to varying traffic conditions.
Berkeley [3]	Distance-to-collision	Provides conservative safety margins; however, remains insensitive to environmental variations.
Machine Learning Approaches		
SVM [4–6]	Hand-crafted feature classification	Performs well with engineered features; however, limited in capturing temporal dependencies.
Fuzzy Logic [7–9]	Rule-based decision making	Well handles uncertainty, but requires complex rules.
MCWA [10, 11]	MLP-based collision warning	Nonlinear input–output mapping; but, relies on handcrafted kinematic features rather than raw trajectory data, lacks temporal dependency modeling.
Deep Learning Approaches		
RCPM [12]	CNN + imbalance sample handling strategy	Addresses class imbalance; however, requires a structured input representation suitable for CNNs.
LSTM [13, 14]	Sequence modeling (and attention-enhanced info.)	Captures temporal dynamics; however, standard LSTM variants operate in a unidirectional manner.
Hybrid Approaches		
Fuzzy-xAI [15]	Fuzzy rules with explainable AI (xAI)	Combines interpretability and learning capability, but results in increased architectural complexity.
CNN-LSTM [16, 17]	Hybrid/ ensemble modeling	Integrates spatial and temporal modeling; it introduces increased model complexity, requires careful design.

Table 1. Taxonomical Summary of the Relevant Studies.

2.2. **Machine Learning Approaches**

With increasing technological advancements and computational capability, the collision-prediction research shifted toward machine learning techniques, including decision trees [18, 19], support vector machines (SVM) [4–6], and fuzzy-logic systems [7–9]. The fuzzy logic-based collision avoidance systems utilize fuzzy inference frameworks that map kinematic inputs (e.g., distance and relative velocity) into linguistic variables using membership functions, evaluate them through expert-defined rule bases, and produce defuzzified control actions for collision mitigation [7–9, 20]. However, their reliance on expert-defined rules and membership function design may limit scalability and generalization under complex and dynamic traffic conditions.

In contrast, the MLP-based rear-end collision warning models (MCWA) [10, 11] adopt a fully data-driven approach, learning the mapping from kinematic variables such as relative distance and velocity directly to collision risk without relying on rule-based thresholds or predefined heuristic decision logic. By training on observed driving data, the model captures nonlinear interactions among input variables and improves prediction accuracy for imminent rear-end collision scenarios. Empirical evaluations in prior work show that this formulation can enhance warning performance and reduce false alarms compared to conventional threshold-based collision warning strategies.

2.3. **Deep Learning Approaches**

To improve predictive performance, recent collision prediction models have incorporated DL techniques. For example, the real-time collision prediction model (RCPM) [12] employs a convolutional neural network (CNN) combined with genetic oversampling to mitigate class imbalance in crash datasets. In such approaches, trajectory data are typically reformulated into structured spatial–temporal representations suitable for convolution operations. While effective in extracting local patterns, CNN-based methods may introduce additional modeling complexity and are generally less explicit in capturing long-range temporal dependencies compared to sequence-oriented architectures.

To model temporal dependencies in sequential driving data, recurrent architectures such as LSTM networks are widely used due to their ability to capture historical context over time. By accurately predicting future trajectories of surrounding vehicles, such models can support downstream decision-making processes, including motion planning and collision risk assessment, thereby contributing to safer driving strategies. For instance, Park *et al.* [13] proposed an LSTM encoder–decoder framework to learn temporal dependencies from past trajectories and generate multi-step future predictions. Similarly, Lin et al. [14] introduced a spatial–temporal attention-based LSTM model that captures both temporal dynamics and inter-vehicle interactions by focusing on the most relevant time steps and neighboring vehicles. However, standard LSTM models typically operate in a unidirectional manner in causal prediction settings, limiting their ability to incorporate future context during inference. This limitation can be mitigated by integrating BiLSTM modules, as explored in this work, enabling richer representations through bidirectional sequential processing when full sequence information is available.

2.4. **Hybrid and Emerging Approaches**

Hybrid approaches have been investigated to combine complementary learning paradigms for enhanced collision-risk prediction. For instance, [8] proposes a type-3 fuzzy logic system integrated with explainable AI techniques for dynamic collision avoidance in autonomous vehicles. Such approaches often involve complex architectural design and rule tuning, which can limit scalability and practical deployment. More recent CNN–LSTM frameworks exploit

Dataset	Sampling Rate (Hz)	Samples	Train	Test	Val
NGSIM [21]	10	1,048,575	634,964	203,375	210,236
highD [22]	25	1,017,235	712,135	151,560	153,540

NGSIM offers fixed-roadside freeway trajectories. highD provides drone-based highway data.

Table 2. Summary of the benchmark datasets used in this work.

spatial–temporal modeling by first extracting spatial features using CNNs and subsequently learning temporal evolution through recurrent layers. For instance, [16] proposes a hybrid CNN–LSTM network for real-time lane detection, which can support safer driving by improving scene understanding and reducing collision risk. On the other hand, Yang *et al.* [17] propose a multi-stream ensemble DL framework that integrates segment-level LSTM modeling with trajectory-level CNN learning for crash prediction under mixed traffic sensing conditions in connected vehicle environments.

In general, hybrid approaches combine spatial feature extraction and temporal sequence modeling within unified or separate learning pipelines, or employ multimodal ensemble strategies to achieve strong predictive performance. However, such hybrid architectures typically introduce additional computational complexity, which can increase inference cost and pose challenges for real-time deployment in safety-critical collision warning systems.

3. Methodology

3.1. The Benchmark Dataset and Data Curation

The main dataset used in this study is the NGSIM [21]. Table 2 summarizes the key details of it. We also conduct extended experiments on an additional dataset–highD [22], for model robustness analysis. The NGSIM provides naturalistic traffic observations sampled at 10 Hz from US Highway 101 and I-80. Passenger vehicles with complete kinematic records are retained, and key features, including velocity, acceleration, space and time headway, lane ID, and vehicle dimensions, were extracted. Raw trajectories are chronologically ordered by vehicle identifier and time index, and velocity and acceleration signals are smoothed to reduce noise and improve temporal consistency. The data are segmented into overlapping (stride of s) temporal sequences of T frames (in this case $s = 1,$ and $T = 10$), with each sequence labeled at its final time step to avoid future information leakage. All preprocessing, class balancing, and dataset partitioning are performed at the sequence level to ensure strict temporal separation with no overlap between subsets. The resulting data are stratified into training, validation, and test sets using a 70:15:15 split, as shown in Fig. 1, preserving class distributions and enabling unbiased evaluation across all splits.

3.1.1. Temporal Segmentation into Fixed-length Sequences

Trajectory data is segmented into fixed-length sequences of T time steps using a sliding-window approach, defined as

$$\mathbb{X}_i = [\mathbf{x}_t, \mathbf{x}_{t+1}, \ldots, \mathbf{x}_{t+T-1}] \in \mathbb{R}^{T\times 7}, \tag{3.1}$$

where each frame $\mathbf{x}$ includes seven kinematic features describing vehicle motion and spatial context. Each sequence inherits the label of its final frame $\mathbf{x}_{t+T-1}$, enabling the model to learn collision risk from the preceding temporal dynamics.

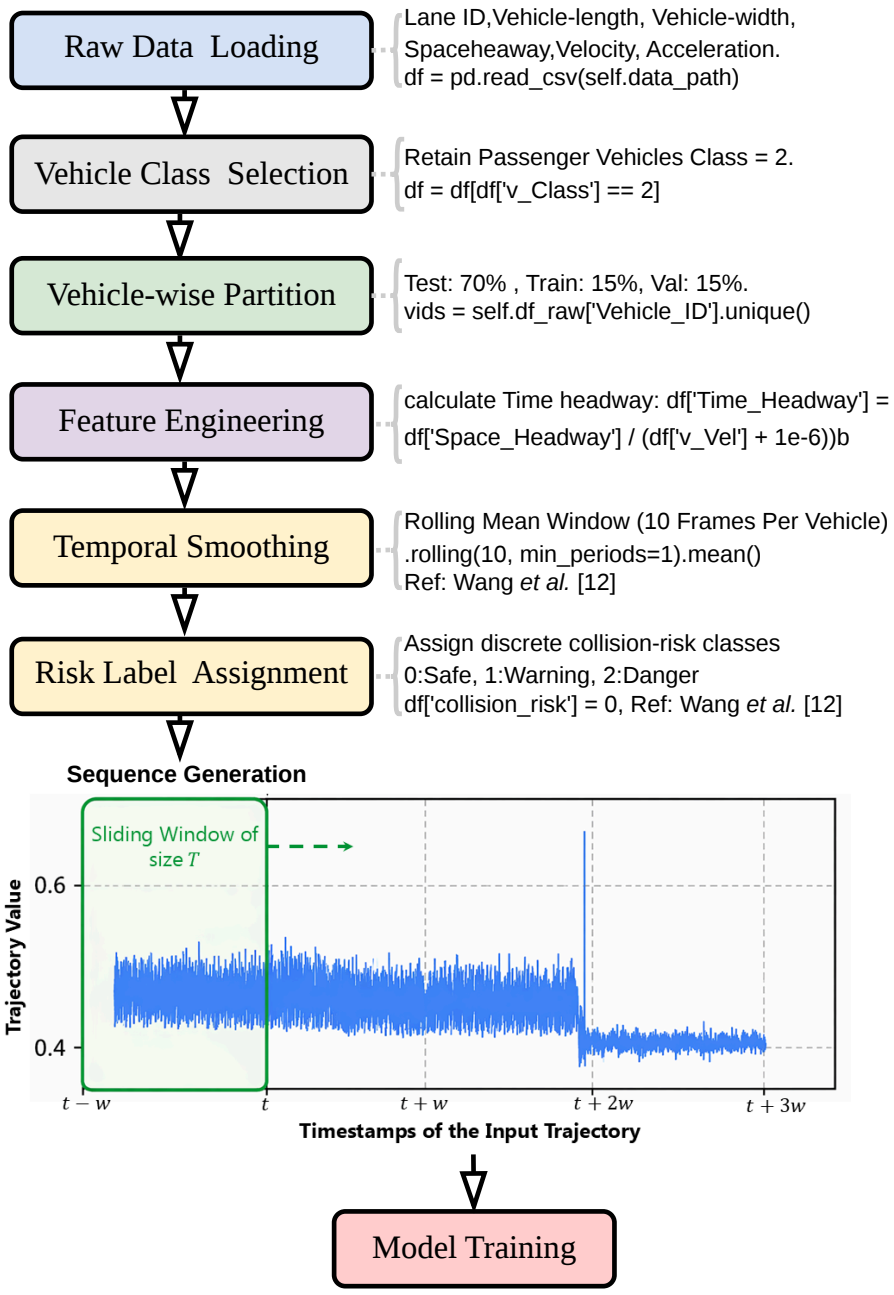

Figure 1. The NGSIM vehicle trajectory preprocessing, temporal sequence generation, and balanced dataset curation for TAG-BiLSTM model development.

3.2. **The End-to-End Architecture**

As depicted in Fig. 2, the proposed TAG-BiLSTM subsumes three key processing stages: a temporal feature extraction, temporal attention learning, and sequence aggregation and classification. Table 3 provides a detailed breakdown of the model architecture and connectivity pattern, including all dimensions and operations of the model.

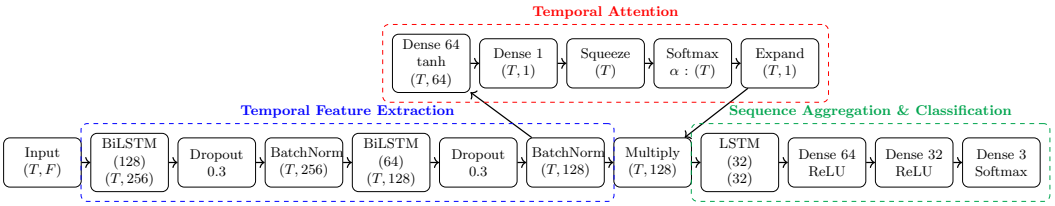

Figure 2. An illustration of the TAG-BiLSTM architecture, where T and F denote the input's sequence length and the number of attributes, respectively. It processes the input sequence through the BiLSTM feature-extraction stack, refined by an attention gate, and passed through dense layers to generate the final collision-risk classification output.

ID	Layer Type	Output Shape	Input
L1	Input Layer	$b \times 10 \times 7$	Mini-batch
Primitive Temporal Feature Extraction (BiLSTM Stack: L2–L3)			
L2	BiLSTM (128 units)	$b \times 10 \times 256$	L1
	Dropout ($p = 0.3$)	$b \times 10 \times 256$	L2
	BatchNormalization	$b \times 10 \times 256$	L2
L3	BiLSTM (64 units)	$b \times 10 \times 128$	L2
	Dropout ($p = 0.3$)	$b \times 10 \times 128$	L3
	BatchNormalization	$b \times 10 \times 128$	L3
Temporal Attention Learning Applied to L3 Output L4			
L4	Attention Gate (Additive Softmax)	$b \times 10 \times 128$	L3
	Weighted Context Fusion ($x \odot \alpha$)	$b \times 10 \times 128$	L4
High-Level Feature Refinement (L5–L7)			
L5	LSTM (32 units)	$b \times 32$	L4
	Dropout ($p = 0.2$)	$b \times 32$	L5
	BatchNormalization	$b \times 32$	L5
L6	Dense(64) $\rightarrow$ ReLU	$b \times 64$	L5
	Dropout ($p = 0.2$)	$b \times 64$	L6
L7	Dense(32) $\rightarrow$ ReLU	$b \times 32$	L6
	Dropout ($p = 0.2$)	$b \times 32$	L7
Prediction Head (L8)			
L8	Dense(3) $\rightarrow$ Softmax $\rightarrow \hat{\mathbf{y}}$	$b \times 3$	L7

`Total learnable parameters`- 346,923; `BiLSTM`— bidirectional long short-term memory layer; `Attention Gate`— Additive attention mechanism producing normalized weights α_t; `BatchNormalization`— Batch-wise normalization; `Dense(`n`)`— Fully connected layer with n neurons; `Dropout(`p`)`— Dropout with probability p; `ReLU`— Rectified linear unit.

Table 3. Layer-by-Layer Connectivity Pattern of the TAG-BiLSTM.

The temporal feature extraction stage receives an input sequence $\mathbf{X} \in \mathbb{R}^{b \times T \times F}$, where b, T, and F denote the batch size, sequence length, and number of kinematic features, respectively. The temporal encoding stage consists of two stacked bidirectional LSTM layers (BiLSTM-128 and BiLSTM-64, i.e., L2 and L3 as in Table 3), which extract forward and backward temporal dependencies as defined in (3.2), and (3.3):

$$\mathbf{H}^{L2} = \text{BiLSTM}_{128}(\mathbf{X}), \text{ and} \tag{3.2}$$

$$\mathbf{H}^{L3} = \text{BiLSTM}_{64}(\mathbf{H}^{L2}), \tag{3.3}$$

where $\mathbf{H}^{L2} \in \mathbb{R}^{b \times T \times 256}$ (128 hidden units in each direction), and $\mathbf{H}^{L3} \in \mathbb{R}^{b \times T \times 128}$ (64 hidden units in each direction). Thus, the bidirectional structure concatenates forward and backward hidden states, enabling each timestep to encode both past and future contextual information. The attention learning module operates on $\mathbf{H}^{L3} \in \mathbb{R}^{b \times T \times 128}$ and computes

temporal relevance scores using an additive attention mechanism:

$$\mathbf{Z} = \tanh\big(\mathbf{W}_a \mathbf{H}^{L3} + \mathbf{b}_a\big), \quad \mathbf{e} = \mathbf{v}^\top \mathbf{Z}, \tag{3.4}$$

$$\alpha = \frac{\exp(\mathbf{e})}{\sum_{k=1}^{T} \exp(\mathbf{e}_k)}, \tag{3.5}$$

where α represents the normalized attention weight for timestep, forming $\boldsymbol{\alpha} \in \mathbb{R}^{b \times T}$. The attention-weighted representation is obtained via element-wise gating:

$$\tilde{\mathbf{H}}^{L4} = \alpha \mathbf{H}^{L3}, \quad \tilde{\mathbf{H}}^{L4} \in \mathbb{R}^{b \times T \times 128}. \tag{3.6}$$

The sequence aggregation stage applies a unidirectional LSTM with 32 hidden units:

$$\mathbf{h}^{L5} = \mathrm{LSTM}_{32}(\tilde{\mathbf{H}}^{L4}), \quad \mathbf{h}^{L5} \in \mathbb{R}^{b \times 32}. \tag{3.7}$$

This is followed by fully connected layers:

$$\mathbf{h}^{L6} = \mathrm{ReLU}(\mathbf{W}_1 \mathbf{h}^{L5} + \mathbf{b}_1), \quad \mathbf{h}^{L6} \in \mathbb{R}^{b \times 64}, \tag{3.8}$$

$$\mathbf{h}^{L7} = \mathrm{ReLU}(\mathbf{W}_2 \mathbf{h}^{L6} + \mathbf{b}_2), \quad \mathbf{h}^{L7} \in \mathbb{R}^{b \times 32}. \tag{3.9}$$

Finally, the prediction head maps the refined representation into a probability distribution over three classes:

$$\hat{\mathbf{y}} = \mathrm{softmax}(\mathbf{W}_o \mathbf{h}^{L7} + \mathbf{b}_o), \quad \hat{\mathbf{y}} \in \mathbb{R}^{b \times 3}. \tag{3.10}$$

Hence, the output classes correspond to: `Safe`, `Warning`, and `Danger`.

4. Experimental Analysis

4.1. Implementation Environment

All experiments were implemented in Python 3.12.7 using TensorFlow 2.13.0, along with NumPy 1.23.5, Pandas 1.5.3, SciPy 1.10.1, scikit-learn 1.2.2, and Matplotlib 3.7.1 for data processing, model development, and evaluation on a system equipped with an NVIDIA RTX 5060 GPU and 32 GB of RAM. To ensure reproducibility, a fixed random seed of 42 was used across all stochastic operations throughout the experimental evaluation.

4.2. Training Strategy

The proposed model was trained using the Adam optimizer with a learning rate of 1×10^{-4} and a batch size of 64. Training stability and generalization were improved via `EarlyStopping` and `ReduceLROnPlateau` callbacks. Model parameters are optimized by minimizing the categorical cross-entropy (CE) loss between the predicted class distribution $\hat{\mathbf{y}} \in \mathbb{R}^{b \times 3}$ and the ground-truth labels $\mathbf{y} \in \mathbb{R}^{b \times 3}$. For a mini-batch of size b and $C = 3$ classes, the loss is defined as

$$\mathcal{L}_{\mathrm{CE}} = -\frac{1}{b} \sum_{i=1}^{b} \sum_{c=1}^{C} y_{i,c} \log \hat{y}_{i,c}, \tag{6}$$

where $y_{i,c}$ denotes the one-hot encoded ground-truth label and $\hat{y}_{i,c}$ represents the predicted probability for class c. Minimizing CE maximizes the likelihood of the correct collision-risk predictions. The mini-batch size of 64 is selected because it provides a balanced trade-off between gradient stability and computational efficiency, which is sufficiently large to produce smooth gradient estimates for recurrent layers, yet small enough to utilize GPU memory and maintain fast training iterations fully. Stable convergence is achieved, as shown in the fig. 3), confirming robust generalization and supporting the high-performance results achieved with these trained parameters.

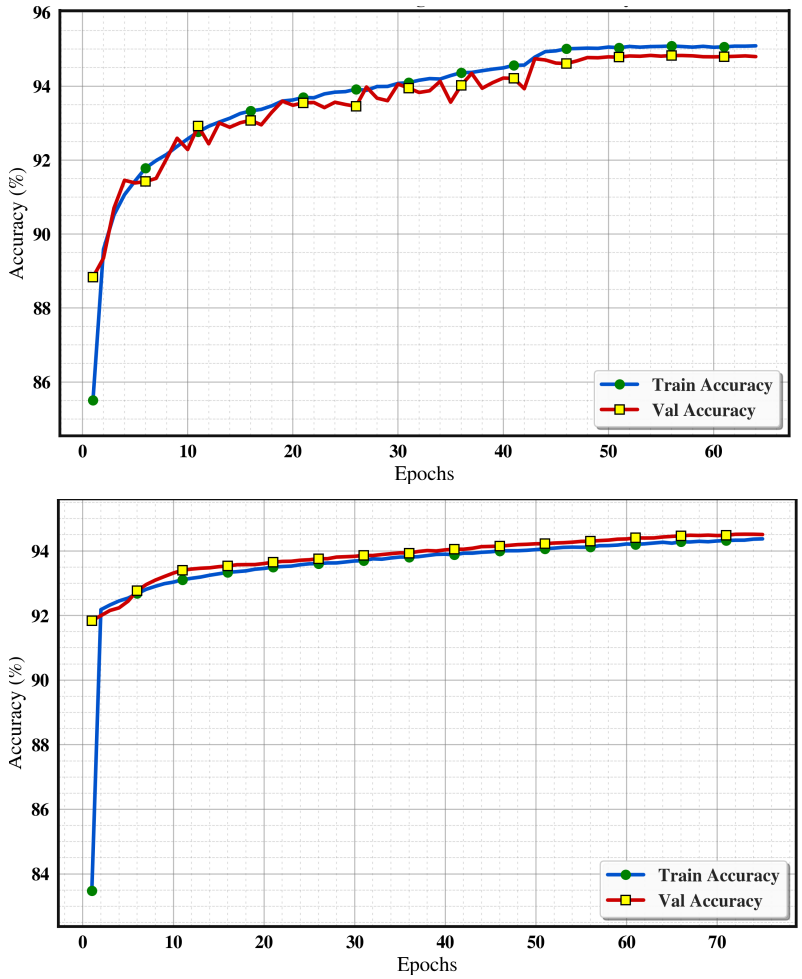

Figure 3. The training progress plots of the TAG-BiLSTM on the NGSIM (top) and highD (bottom) datasets.

4.3. **Evaluation Metrics**

The performance of the proposed model is evaluated on a held-out test set using overall accuracy (OA), the area under the curve (AUC) derived from the receiver operating characteristic (ROC), class-wise precision, recall, and F1-score, along with macro-averaged F1 to account for class imbalance. The OA is defined as

$$\text{OA} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(\hat{y}_i = y_i) \times 100\%, \tag{7}$$

where N denotes the total number of samples, $\hat{y}_i$ represents the predicted class label of the i-th sample, y_i is the corresponding ground-truth label, and $\mathbb{I}(\hat{y}_i = y_i)$ is an indicator function that equals 1 if the prediction is correct and 0 otherwise.

For each class $c \in \{\texttt{safe}, \texttt{warning}, \texttt{danger}\}$, the evaluation metrics are computed in a one-vs-rest manner. Thus, the class-wise precision, recall, and F1-score are defined as

$$P_c = \frac{TP_c}{TP_c + FP_c}, \quad R_c = \frac{TP_c}{TP_c + FN_c}, \text{and} \quad \text{F1}_c = 2 \cdot \frac{P_c \cdot R_c}{P_c + R_c}, \tag{8}$$

Method	OA (%)↑	Pre. (%)↑	Rec. (%)↑	AUC (%)↑	F1 (%)↑
Honda FCW Algorithm [2]*	–	–	–	63.60	58.30
Berkeley PATH Algorithm [3]*	–	–	–	71.00	70.28
MCWA [10]*	–	–	–	81.00	74.33
RCPM [12]	86.00	87.00	–	82.30	90.20
YOLOv3 [23]	86.00	88.00	–	89.20	84.70
YOLOv3-RCEP [23]	92.00	90.00	–	96.20	94.20
Multi-Layer BiLSTM (baseline)	94.22	96.24	94.20	99.13	94.86
TAG-BiLSTM (proposed)	**94.77**	**96.44**	**94.77**	**99.21**	**95.30**

Note: ↑ Higher values indicate better performance; * results are adopted from [12]; **boldface** denotes best performance.

Table 4. Performance Analysis on NGSIM Dataset.

where TP_c, FP_c, and FN_c denote the true positives, false positives, and false negatives for class c, respectively. Hence, the macro-averaged F1-score is computed as the arithmetic mean across all classes:

$$\text{Macro-F1} = \frac{1}{3}\sum_{i=1}^{3} \text{F1}_i, \tag{9}$$

The weighted F1-score accounts for class imbalance by incorporating class support:

$$\text{Weighted-F1} = \sum_{i=1}^{3} \left(\frac{\text{Support}_i}{N} \cdot \text{F1}_i \right), \tag{10}$$

where Support_i is the number of samples in class i, and N is the total number of samples in the evaluation set. Together, these metrics present a comprehensive benchmark comparison as discussed in Section 4.4.

4.4. **Performance Analyses**

Table 4 presents a comparative analysis of the proposed TAG-BiLSTM against established collision prediction methods on the NGSIM dataset. It also includes a baseline model with the same architecture as TAG-BiLSTM but without the temporal attention module, enabling an explicit evaluation of the contribution of the attention mechanism. Traditional algorithms, viz. Honda FCW, Berkeley PATH, and MCWA show limited discriminative capability, with AUC scores ranging from 63.60% to 81.00%. Vision-based deep learning methods deliver stronger results, with YOLOv3-RCEP achieving 92.00% accuracy and 96.20% AUC. The multi-layer BiLSTM baseline establishes a strong temporal foundation, achieving 94.22% accuracy, 96.24% precision, 94.20% recall, 99.13% AUC, and an F1-score of 94.86%.

The proposed TAG-BiLSTM achieves state-of-the-art performance across all metrics, improving accuracy to 94.77%, precision to 96.44%, recall to 94.75%, and maintaining 99.21% AUC, along with an F1-score of 95.30% compared to the baseline. This demonstrates the effectiveness of attention-gated mechanisms in capturing critical temporal patterns for collision risk assessment. The near-perfect AUC of 99.21% and balanced F1-score of 95.30% highlight the suitability of the model for real-time safety.

4.5. **Generalization Performance Analysis on highD Dataset**

To further validate the proposed approach, we experiment on the highD dataset [22]. A naturalistic vehicle trajectory dataset captured on six German highway locations using a drone-mounted camera from a bird's-eye aerial perspective at a sampling rate of 25 Hz, with

Method	OA (%)↑	Pre. (%)↑	Rec. (%)↑	AUC (%)↑	F1 (%)↑
Multi-Layer BiLSTM (baseline)	92.71	91.58	92.71	84.77	90.06
TAG-BiLSTM (proposed)	**94.75**	**94.38**	**94.75**	**95.03**	**94.51**

Note: ↑ Higher values indicate better performance; ↓ lower values are better; and **boldface** denotes the best-performing result.

Table 5. An Additional Analysis on highD Dataset.

trajectories extracted via computer vision techniques. Table 5 presents results on the highD dataset. TAG-BiLSTM outperforms the baseline across all key metrics: increasing accuracy from 92.71% to 94.75%, precision from 91.58% to 94.38%, recall from 92.71% to 94.77%, and F1-score from 90.06% to 94.51%. Notably, it also achieves a substantial improvement in AUC from 84.77% to 95.03%. These results confirm that TAG-BiLSTM not only maintains efficiency but also generalizes robustly to distinct traffic datasets, consistently demonstrating its practical value for diverse real-world ITS applications.

5. Conclusion

The proposed TAG-BiLSTM achieves 94.77% accuracy on NGSIM and 94.75% accuracy on highD for real-time collision risk assessment, Its attention-gating mechanism captures critical temporal patterns for reliable discrimination. However, performance under diverse weather conditions and varying road conditions remains unverified, as the model was evaluated primarily on clear-weather datasets. Future work will focus on extending prediction horizons to capture earlier precursors of collisions and rigorously validating model performance under adverse environmental conditions to ensure robust real-world operation.

References

[1] T. Stewart. "Overview of motor vehicle traffic crashes in 2021". In: *Traffic safety facts: research note* (2023).

[2] Y. Fujita, K. Akuzawa, and M. Sato. "Radar brake system". In: *Intelligent Transportation: Serving the User Through Deployment. Proceedings of the 1995 Annual Meeting of ITS America. ITS America*. 1995.

[3] S. E. Shladover, C. A. Desoer, J. K. Hedrick, M. Tomizuka, J. Walrand, W.-B. Zhang, D. H. McMahon, H. Peng, S. Sheikholeslam, and N. McKeown. "Automated vehicle control developments in the PATH program". In: *IEEE Transactions on vehicular technology* 40.1 (1991), pp. 114–130.

[4] X. Li, D. Lord, Y. Zhang, and Y. Xie. "Predicting motor vehicle crashes using support vector machine models". In: *Accident Analysis & Prevention* 40.4 (2008), pp. 1611–1618.

[5] X. Xiong, L. Chen, and J. Liang. "A new framework of vehicle collision prediction by combining SVM and HMM". In: *IEEE Transactions on Intelligent Transportation Systems* 19.3 (2017), pp. 699–710.

[6] L. G. Galvão and M. N. Huda. "Pedestrian and vehicle behaviour prediction in autonomous vehicle system—A review". In: *Expert Systems with Applications* 238 (2024), p. 121983.

[7] H. Elsayed, B. A. Abdullah, and G. Aly. "Fuzzy logic based collision avoidance system for autonomous navigation vehicle". In: *2018 13th international conference on computer engineering and systems (icces)*. IEEE. 2018, pp. 469–474.

[8] Y. Sabry, M. Aly, W. Oraby, and S. El-demerdash. "Fuzzy control of autonomous intelligent vehicles for collision avoidance using integrated dynamics". In: *SAE International Journal of Passenger Cars-Mechanical Systems* 11.06-11-01-0001 (2018), pp. 5–21.

[9] U. D. Agarwal, S. Sinha, R. Srivastava, S. Pathak, and S. Raushan. "Development of Collision Avoidance System Using Fuzzy Logic". In: *Advances in Engineering Design: Select Proceedings of FLAME 2018*. Springer, 2019, pp. 779–787.

[10] D. Lee and H. Yeo. "A study on the rear-end collision warning system by considering different perception-reaction time using multi-layer perceptron neural network". In: *2015 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2015, pp. 24–30.

[11] D. Lee and H. Yeo. "Real-time rear-end collision-warning system using a multilayer perceptron neural network". In: *IEEE Transactions on Intelligent Transportation Systems* 17.11 (2016), pp. 3087–3097.

[12] X. Wang, J. Liu, T. Qiu, C. Mu, C. Chen, and P. Zhou. "A real-time collision prediction mechanism with deep learning for intelligent transportation system". In: *IEEE transactions on vehicular technology* 69.9 (2020), pp. 9497–9508.

[13] S. H. Park, B. Kim, C. M. Kang, C. C. Chung, and J. W. Choi. "Sequence-to-sequence prediction of vehicle trajectory via LSTM encoder-decoder architecture". In: *2018 IEEE intelligent vehicles symposium (IV)*. IEEE. 2018, pp. 1672–1678.

[14] L. Lin, W. Li, H. Bi, and L. Qin. "Vehicle trajectory prediction using LSTMs with spatial–temporal attention mechanisms". In: *IEEE Intelligent Transportation Systems Magazine* 14.2 (2021), pp. 197–208.

[15] J. M. Mendel and P. P. Bonissone. "Critical thinking about explainable AI (XAI) for rule-based fuzzy systems". In: *IEEE Transactions on Fuzzy Systems* 29.12 (2021), pp. 3579–3593.

[16] Y. Kortli, S. Gabsi, L. F. Lew Yan Voon, M. Jridi, M. Merzougui, and M. Atri. "Deep embedded hybrid CNN-LSTM network for lane detection on NVIDIA Jetson Xavier NX". In: *Knowledge-Based Systems* 240 (2022), p. 107941.

[17] S. Yang, M. Abdel-Aty, and L. Han. "Crash prediction under limited CV coverage: an ensemble deep learning model integrating multi-source traffic data". In: *Transportation Research Part C: Emerging Technologies* 183 (2026), p. 105472.

[18] Z. Zheng, P. Lu, and D. Tolliver. "Decision tree approach to accident prediction for highway–rail grade crossings: Empirical analysis". In: *Transportation Research Record* 2545.1 (2016), pp. 115–122.

[19] M. T. Ashraf, K. Dey, S. Mishra, and M. T. Rahman. "Extracting rules from autonomous-vehicle-involved crashes by applying decision tree and association rule methods". In: *Transportation research record* 2675.11 (2021), pp. 522–533.

[20] A. K. Karmakar and M. Uddin. "Fuzzy Logic Based Automatic Vehicle Collision Prevention System". In: *Proceedings of the IEEE International Conference on Informatics, Electronics & Vision (ICIEV)*. Dhaka, Bangladesh, 2012, pp. 1015–1020.

[21] V. Punzo, M. T. Borzacchiello, and B. Ciuffo. "On the assessment of vehicle trajectory data accuracy and application to the Next Generation SIMulation (NGSIM) program data". In: *Transportation Research Part C: Emerging Technologies* 19.6 (2011), pp. 1243–1262.

[22] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein. "The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems". In: *2018 21st international conference on intelligent transportation systems (ITSC)*. IEEE. 2018, pp. 2118–2125.

[23] D. Punia and R. Kumar. "A YOLOv3-powered edge computing technique for real-time rear-end collision prediction in autonomous vehicles". In: *Applied Soft Computing* (2025), p. 113981.