Journal of Network and Computer Applications

From LSTMs to Transformers and GNNs: A Comprehensive Analysis on Lane Change Trajectory Prediction

--Manuscript Draft--

Manuscript Number:	JNCA-D-25-02373
Article Type:	Research Paper
Keywords:	Autonomous vehicles, Lane change prediction, Deep learning, Transformer, Informer, NGSIM
Corresponding Author:	Jayaram Peggem, M.Tech National Institute of Technology Tiruchirappalli INDIA
First Author:	Jayaram Peggem, M.Tech
Order of Authors:	Jayaram Peggem, M.Tech
	Nithya B
	Harsha Vardhan Gogada
	Venkateswara Reddy S
Abstract:	Predicting lane changes is crucial for Autonomous Vehicle(AV) safety, but it's a difficult task. A primary challenge is that human driving behavior is complex and unpredictable. Most existing models are trained on ideal data, causing them to fail when faced with real-world scenarios with noisy sensor inputs or limited computing resources. The current solutions typically use single model types like Recurrent Neural Networks (RNNs) or Transformers, which are either not robust enough for imperfect data or too computationally expensive for real-time applications. To address these issues, this research provides a holistic evaluation of diverse deep learning models—including GRUs, Bi-LSTMs, Transformers (Informer), and novel hybrid architectures—for lanechange prediction on the NGSIM dataset. Moving beyond studies limited to a single model type, we rigorously assess performance under varied conditions, including clean, noisy, rich, and sparse feature sets. A key novelty of our work is the investigation of hybrid Transformer-RNN models, which synergistically combine self-attention mechanisms with sequential processing to achieve superior robustness and efficiency. These models are quantitatively benchmarked across all lane-change maneuvers (Left, Keep, Right), and their computational efficiency is analyzed via training and inference times. This study serves as both a technical benchmark and a practical guide, offering valuable insights into the trade-offs between performance and efficiency for future research and real-world autonomous vehicle applications.

From LSTMs to Transformers and GNNs: A Comprehensive Analysis on Lane Change Trajectory Prediction

Abstract

Predicting lane changes is crucial for Autonomous Vehicle(AV) safety, but it's a difficult task. A primary challenge is that human driving behavior is com plex and unpredictable. Most existing models are trained on ideal data, caus ing them to fail when faced with real-world scenarios with noisy sensor inputs or limited computing resources. The current solutions typically use single model types like Recurrent Neural Networks (RNNs) or Transformers, which are either not robust enough for imperfect data or too computationally expen sive for real-time applications. To address these issues, this research provides a holistic evaluation of diverse deep learning models—including GRUs, Bi LSTMs, Transformers (Informer), and novel hybrid architectures—for lane change prediction on the NGSIM dataset. Moving beyond studies limited to a single model type, we rigorously assess performance under varied condi tions, including clean, noisy, rich, and sparse feature sets. A key novelty of our work is the investigation of hybrid Transformer-RNN models, which syn ergistically combine self-attention mechanisms with sequential processing to achieve superior robustness and efficiency. These models are quantitatively benchmarked across all lane-change maneuvers (Left, Keep, Right), and their computational efficiency is analyzed via training and inference times. This study serves as both a technical benchmark and a practical guide, offering valuable insights into the trade-offs between performance and efficiency for future research and real-world autonomous vehicle applications.

Keywords: Autonomous vehicles, Lane change prediction, Deep learning, Transformer, Informer, NGSIM.

1. Introduction

The development of Autonomous Vehicles (AVs) is a significant advance ment in intelligent transportation, promising increased safety, reduced traffic, and improved mobility [1]. A crucial and intricate aspect of autonomous driv ing is executing safe and intelligent lane changes. This maneuver goes beyond simple mechanical control, requiring complex judgments regarding vehicle dynamics, spatial gaps, driver intent, and interactive decision-making with surrounding vehicles. Unlike simpler tasks like lane keeping, lane changes introduce significant uncertainty due to the need to predict and respond to the behavior of neighboring vehicles, especially in dense traffic.

For the foreseeable future, AVs will operate in mixed-traffic environments, coexisting with human-driven vehicles. This necessitates that autonomous systems not only operate safely on their own but also understand, adapt to, and predict human driving behavior, particularly during complex maneuvers like lane changes [2]. Human drivers rely on subtle communication cues, contextual awareness, and social driving norms—elements that AVs must learn to interpret and emulate. Integrating this behavioral intelligence is vital for building trust, ensuring traffic harmony, and achieving seamless integration into existing transportation systems.

Extensive research has focused on designing models that can accurately anticipate future maneuvers, particularly lateral decisions like lane changes, based on historical trajectory data and surrounding context. Classical ap proaches often relied on handcrafted features combined with rule-based de cision logic or shallow classifiers like Support Vector Machines (SVMs) and Random Forests [3]. While interpretable, these methods are rigid and lack the adaptability required for dynamic, real-world traffic scenarios. The rise of deep learning has significantly transformed this field. Temporal sequence models such as Recurrent Neural Networks (RNNs), specifically Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks, have become fundamental for many lane-change prediction pipelines. More recent approaches have introduced Bidirectional LSTMs to capture both forward and backward dependencies in vehicle trajectories, offering a richer temporal representation more robust to noisy inputs.

Concurrently, the development of Transformer-based models, inspired by advancements in Natural Language Processing (NLP), has shown promise for multi-agent trajectory modeling. Transformers utilize self-attention mecha nisms to model global interactions among vehicles across different time steps, without relying on recurrence. Variants like Informer and Autoformer extend this capability by introducing sparse attention patterns and seasonal-trend decomposition modules, enabling them to handle long-term dependencies and reduce computational burden. These architectures are particularly effective in multi-agent scenarios where lane-changing decisions depend on the inten tions and actions of surrounding traffic participants. Despite these advances, the current body of research still faces key limitations. Most models are trained on clean, dense data, while real-world inputs are often noisy or incom plete due to sensor limitations. The impact of label noise—especially from auto-generated lane-change annotations—is rarely studied. Additionally, few works report resource demands, limiting insights into real-time deployment feasibility in resource-constrained embedded systems typical of autonomous vehicles[4].

This research provides a comprehensive evaluation of diverse models for lane-change prediction, spanning GRUs,Bidirectional LSTM (Bi-LSTMs), Transformers like Informer, and novel hybrid architectures. Unlike prior work limited to single model types, we assess performance across clean, noisy, rich, and sparse feature settings. Our study addresses the underexplored potential of hybrid Transformer-RNN models, combining self-attention and sequential learning for better robustness and efficiency. These hybrids strike a balance between accuracy and computational cost. Results show superior perfor mance in challenging scenarios, highlighting the value of cross-architecture synergy.

The significance of this research lies in its comprehensive analysis and empirical evaluation of diverse modeling architectures for the critical task of lane-change prediction. Unlike prior work that often focuses on a sin gle model type, our study investigates a wide spectrum of learning archi tectures—ranging from traditional RNN-based models such as GRUs and Bi-LSTMs to advanced Transformer variants like Informer and strategic hy brid architectures combining them. This comparative framework enables a deeper understanding of how each model class behaves under realistic driving conditions.

The novelty of our approach lies in both its breadth and depth: eval uating standard and hybrid models under varying conditions, quantifying their trade-offs, and offering the first-of-its-kind comparative study that in tegrates performance, resilience, and computational footprint in the context of human-lane-change prediction. Our work therefore serves as both a tech nical benchmark and a practical guide for future research and real-world AV applications.

The paper is organized into five main sections. Section I establishes the context of autonomous driving and the importance of lane-change prediction. Section II analyzes the recent studies on lane-change modeling, discussing their methodologies and limitations. Comprehensive descriptions of the various models are discussed in Section III. These models span a range of architectures, from well-established baselines to novel hybrid configurations. The rationale behind these hybrid designs is explained, highlighting how they are intended to combine the strengths of both attention-based and recurrent components to improve performance and robustness. The performance of the models are compared in Section IV on metrics like accuracy, robustness, and computational efficiency. Finally, section V summarizes the findings and discusses future research directions for developing reliable lane-change prediction modules.

2. Related Work

A comprehensive review of existing research on lane change prediction was conducted, with a focus on advanced models aimed at enhancing prediction accuracy and real-time safety.

K. Gao et.al [5] proposed a dual Transformer-based model for jointly predicting lane change intentions and vehicle trajectories in mixed traffic. It uses spatial-temporal attention and Transformer modules for intention classification and trajectory prediction. Though it is noted better performance on Next Generation Simulation(NGSIM) and highD datasets, it may face challenges with noisy data. Lu, Yuhuan et.al [6] proposed KLEP, a knowledge-driven lane change prediction system that combines real-world driving insights with neural networks to improve safety in IoV. It models driver decisions using factors like speed, acceleration, and proximity, and uses a graph transformer to analyze structured data. It processes data faster and offers better interpretability. Due to region-specific training data, it lacks generalizability.

Kunsong Shi et.al[7] proposed an LSTM-based Intrusion Detection System (IDS) that predicts the next CAN-bus ID, using one-hot encoded inputs and SoftMax output. It detects anomalies like ID insertion, drops, and reordering using exact match and log-loss scoring. It is trained on 36 million real-world messages with synthetic attacks. Mingxing Peng et.al [8] introduced LC-LLM, which reframes lane change prediction as a language task using Large Language Models(LLM) to output decisions and explanations with structured prompts. LC-LLM outperformed traditional methods under highD dataset. However, high computational cost and potential information loss during text conversion limit real-time use.

Kequan Chen et.al[9] proposed a real-time crash prediction model that uses vehicle trajectory data to detect risky lane changes before collisions occur. It employs Generalized Extreme Value (GEV) distributions to distin guish crash vs. non-crash behavior. The model achieved high AUC scores (0.92–0.98) and outperformed traditional methods. However, its reliance on a small dataset and high-resolution data limits generalizability and real-world deployment. A. N. Qasemabadi et.al [10] presented a deep learning model using a multi-layer LSTM network to predict ego vehicle lane changes in Co operative Adaptive Cruise Control (CACC) systems. The model processes temporal features from the ego vehicle's motion along with traffic data from surrounding vehicles. The model is trained on the HighD dataset. Though it achieved better performance, real-time deployment may be limited by its computational demands.

Jongyong Do et.al [11] presented a model to predict lane change intentions and trajectories of nearby vehicles using real-world highway trajectory data. It models crash and non-crash behavior using GEV distributions. The sys tem estimates crash likelihood in real time. Its performance is optimized for structured highways and may face challenges in broader deployment. Yunjie Huang et.al [12] introduced TrafficTL, a transfer learning framework that improves traffic prediction in data-scarce cities by leveraging patterns from data-rich ones. It groups road segments with similar trends using mutual information and applies temporal clustering, graph reconstruction, and en semble learning to enhance prediction. Tested on datasets from three cities, it outperformed existing methods by 8–25%. However, it requires some quality data in the target city and may oversimplify complex traffic dynamics.

Hongrui Zhang et.al [13] presented an enhanced Spiking Neural Network (SNN) for predicting lane-change intentions using time-series. With the HighD and NGSIM datasets, it achieves accuracies of 98.28% and 94.26%, outperforming Echo State Networks(ESNs) and LSTM networks in speed and efficiency. Qingwen Xue et.al [14] proposed a dual-stage model that predicts both lane change intent and vehicle trajectory by incorporating density and vehicle type. It combines Extreme Gradient Boosting(XGBoost) and LSTM for trajectory forecasting using HighD and NGSIM datasets. It achieved 98.20% accuracy and reduced trajectory errors. However, the model relies on high-quality labeled data. It struggles to handle unpredictable driver be haviors without additional retraining, raising concerns about scalability and robustness in dynamic real-world conditions.

Jinbao Zhang et.al [15] proposed an improved Lane Change Risk Assess ment Index (LCRAI) for expressway weaving segments, capturing both fre quency and severity of vehicle conflicts. Drone-based trajectory data and spa tiotemporal check ups help to identify and localize high-risk zones. LCRAI is adaptable to various road types, offering practical value for traffic safety planning. Hongyu Guo et.al [16] introduced a smart system using connected vehicle data to detect and predict lane changes for improved driving safety.It combines an autoencoder for detecting anomalies with a transformer model to forecast lane-change intent up to two seconds ahead. With the Safety Pilot Model Deployment(SPMD) dataset, it achieved a PR-AUC above 0.98, outperforming traditional models. It depends on accurate sensor data and lacks the integration of human behavior cues.

Yuhuan Lu et.al [17] proposes a real-time incident detection system us ing a Spatio-Temporal Variational Digraph Auto-Encoder (ST-VDAE) with edge-cloud collaboration. The edge module analyzes live local traffic, while the cloud learns global patterns, enabling fast detection of traffic incidents. It relies on high-quality trajectory data and fixed edge-cloud fusion. Wenjian Sun et.al [18] proposed a model which enhanced autonomous lane change safety by predicting nearby vehicle movements. It used LSTM-based model with time-series driving data and vehicle-to-vehicle(V2V) communication. It generates smooth lane change paths via cubic polynomials and integrates an improved Gipps model within a Model Predictive Control (MPC) framework. This system relies on V2V communication and lacks real-world deployment validation.

Lin Li et.al [19] proposed a real-time lane-change intention model for making safer decisions on highways using RNNs. It improved autonomous responses in highway simulations. The model shows strong prediction ac curacy but depends on continuous high-quality data. Dongwei Xu [20] pre sented Multi-view Adaptive Hierarchical Spatial Graph Convolution Net work(MVHGN), a trajectory prediction model for heterogeneous traffic agents.

It used spatial graph convolutions and GRU-based temporal predictor to model agent interactions. Though it achieved better performance, it struggles with pedestrian predictions and sensitive to noisy data and computationally intensive.

Table 1: A Summary of Literature Review

Algorithm and Year of Publication	Objective	Methodology	Inputs	Requirements / Assumptions	Performance / Enhancements	Limitations
Dual Transformer-Based Prediction (IEEE, 2023)	Jointly predict lane change intention and trajectory	Two interconnected Transformers (Intention + Trajectory)	Position, velocity, heading, neighbor data	Clean data; coexistence of human and autonomous vehicles	High accuracy in NGSIM/highD; superior to existing methods	High computational cost; needs clean sensor data
KLEP: Knowledge-Driven Lane Change (IEEE, 2025)	Combine human knowledge + ML for lane change prediction	Driver decision model + Graph Transformer	Structured driving data (speed, acceleration, proximity)	Nearby vehicle info; real-world driving knowledge	6–7% higher accuracy; 50% better early detection	Limited long-range interaction; regional dataset bias
Integrated DNN-Based Trajectory + Intrusion Detection (Harvard, 2022)	Predict CAN ID for detecting cyber attacks	One-hot encoding + 2 LSTM layers + Softmax + log-loss scoring	CAN ID sequences	CAN-bus sequences follow consistent patterns	F1: 0.90 (insertion), 0.84 (drop), 1.0 (illegal ID)	Dataset not shared;data collection method unclear
LC-LLM: Explainable Lane Change Using LLMs (ScienceDirect, 2025)	Make predictions interpretable with natural language	LLM + Chain-of-Thought prompts from structured data	Motion, traffic context, map data	Ability to convert numeric to text prompts	Better accuracy + explainability (highD dataset)	High computation; limited generalization; data-to-text loss
Real-Time Crash Prediction (Elsevier, 2024)	Predict crashes during lane changes	GEV distribution modeling for crash likelihood	Relative speed, inter-vehicle distance, acceleration	High-res trajectory data	AUC: 0.92–0.98; 76% better than TTC	Trained on only 30 crash events; no live deployment
CACC-Based Lane Change Prediction (IEEE, 2023)	Predict ego vehicle lane changes in CACC systems	Multi-LSTM on CACC (8-vehicle data) vs ACC (3-vehicle)	Ego motion + surrounding vehicle data	Access to inter-vehicle communication (CACC)	Higher accuracy than ACC; HighD dataset	High sensor needs; lacks generalizability; computationally heavy

Lane Change Intention & Trajectory of Others (IEEE, 2023)	Predict surrounding vehicle intentions + crashes	GEV for crash vs non-crash; real-time alerts	Relative motion features	Structured highway settings	AUC: 0.92–0.98	Struggles in urban or noisy data; oversimplifies behavior
TrafficTL: Transfer Learning for Traffic (IEEE, 2023)	Improve predictions in low-data cities	Mutual info clustering + graph recon + ensemble learning	Traffic patterns from multiple cities	Some target city data; shared traffic traits	8–25% improvement over baselines	Relies on clustering; less effective in dynamic conditions
Spiking Neural Network for Intention (ScienceDirect, 2024)	Lightweight, fast lane change prediction	SNN on lateral position, velocity, acceleration	Time-series data	Structured highway data	98.28% (HighD), 94.26% (NGSIM); faster convergence	Biased by class imbalance; lacks sensory context
Context-Aware Integrated Prediction (ScienceGate, 2022)	Predict lane change + trajectory with context	XGBoost for intention + LSTM for trajectory	Traffic density, vehicle type, motion	High-quality labeled context data	97.02% → 98.20% accuracy; error reduction	Fails in urban/mixed; high compute; retraining needed
LCRAI for Weaving Segments (ScienceDirect, 2023)	Assess lane change risk in weaving segments	GTTC + ERI + SRI from drone-based data	Speed, conflict duration/intensity, GTTC	Conflict zone definition; drone data	Identifies high-risk zones for interventions	Depends on trajectory + drone data; limited adaptability
Connected Vehicle Lane Change Prediction (ScienceGate, 2022)	Detect & predict lane changes using CV data	Autoencoder + Transformer with attention	Speed, steering, SPMD CV data	Accurate sensors; label quality	PR-AUC > 0.98; better than classic ML	Lacks behavioral cues; limited to highway
Edge-Cloud Incident Detection (Springer, 2023)	Real-time traffic incident detection	ST-VDAE on edge + cloud fusion	Local traffic (edge) + historical (cloud)	Edge-cloud sync; high-res vehicle data	0.25s latency; 26.3% better accuracy	Only tested offline; fixed fusion; edge resource issues
LSTM-MPC Lane Change Safety (IEEE, 2024)	Enable smoother, safer AV lane changes	LSTM + cubic polynomial + Gipps model in MPC	Time-series data + vehicle communication	V2V availability; simulator setup	Better decisions in obstacle and brake tests	No real-world test; assumes communication

RNN-Based Intention Inference (IEEE, 2021)	Predict highway lane change intentions	RNN with motion sequence input	Speed, acceleration, positions	Continuous, high-quality data	Good prediction in simulation	Limited complexity; simulation only; outdated model
MVHGN for Multi-Agent Prediction (IEEE, 2023)	Forecast multi-agent heterogeneous traffic motion	Multi-view spatial graph + GRU temporal decoder	Agent type, motion, proximity	ApolloScape; heterogeneous agents	Strong in dense traffic, cyclists	Less accurate on pedestrians; high compute cost

2.1. Inferences and Contributions

From the aforementioned discussion, it can be inferred that various deep learning architectures—such as Transformers, LSTMs, RNNs, and Spiking Neural Networks—along with knowledge-driven and graph-based models, have been applied for lane change prediction and risk assessment. While these methods show significant improvements in prediction accuracy, early detection, and interpretability, they often encounter challenges such as high computational demands, data quality issues, limited generalizability across dynamic environments, and constraints in real-time deployment. Detailed summary provided in Table 1. To address these challenges, this paper con tributes the following:

- Spatial Temporal Interactions: Multiple models, ranging from clas sical recurrent neural networks to attention-enhanced architectures ap plied to capture interactions
- Long Sequence Forecasting: Informer and transformer based archi tecture designed to strengthen temporal sequence modelling
- Next Generation Simulation datasets: Utilized the publicly avail able NGSIM high-resolution vehicle trajectory dataset contain dynamic scenarios.
- Processing data: Our pipeline includes preprocessing the input data, generating meaningful labels, and structuring sequences to align with the requirements of time series learning models.

- Computational efficiency: Implemented Gated Recurrent Unit (GRU) based model which are computationally efficient and best suitable for trajecroty data.

- Generalization and Robust predictions: Employed Attention based Long Short-Term Memory (Attn-LSTM) architecture for selective focus on input data for better predictions.

3. Spatial-Temporal Modeling Approaches for Lane Change Pre diction

This section conducts a comprehensive analysis of multiple spatial - tem poral modeling approaches for lane change prediction. As shown in Figure 1, it ranges from classical recurrent neural networks such as GRUs and Bi LSTMs, to attention-enhanced architectures like Attention-based LSTMs. The power of self-attention mechanisms is further examined through the use of Transformer models, and extended by combining them with Graph Neu ral Networks (GNNs) to capture the spatial-temporal interactions between neighboring vehicles. The performance of Informer is also explored for long sequence forecasting, in both standalone and hybrid forms. Variants include Informer + GNN, Informer + Transformer, and Informer + LSTM, each tailored to leverage complementary strengths in temporal sequence model ing and interaction reasoning. By systematically evaluating these models on real-world trajectory datasets, the most effective architecture is identified for predicting lane change behavior in complex driving scenarios.

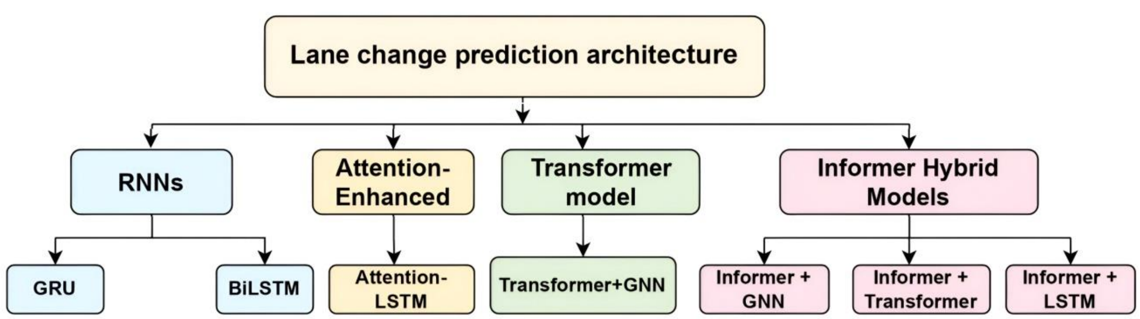

Figure 1: Proposed Lane Change Prediction Models

3.1. Dataset and Preprocessing

In this work, the NGSIM dataset [?] is utilized. It is a comprehensive, real-world collection of vehicle trajectory data captured on US highways. The data set records second-by-second vehicle movements and provides the granular details necessary to model driving behavior and trajectory-based prediction tasks, such as lane change prediction. This dataset provides de tailed information such as vehicle positions, velocities, accelerations, and Lane ID values across consecutive frames for each vehicle.

Effective preprocessing plays a vital role in preparing trajectory data for lane change prediction. Since raw vehicle data may contain noise, missing values, or require reformatting, careful preprocessing ensures the quality and consistency needed for temporal modeling. As shown in Figure 2, the pre processing pipeline involves cleaning the input data, engineering meaningful labels, and organizing sequences in a way that aligns with the requirements of time-series learning models.

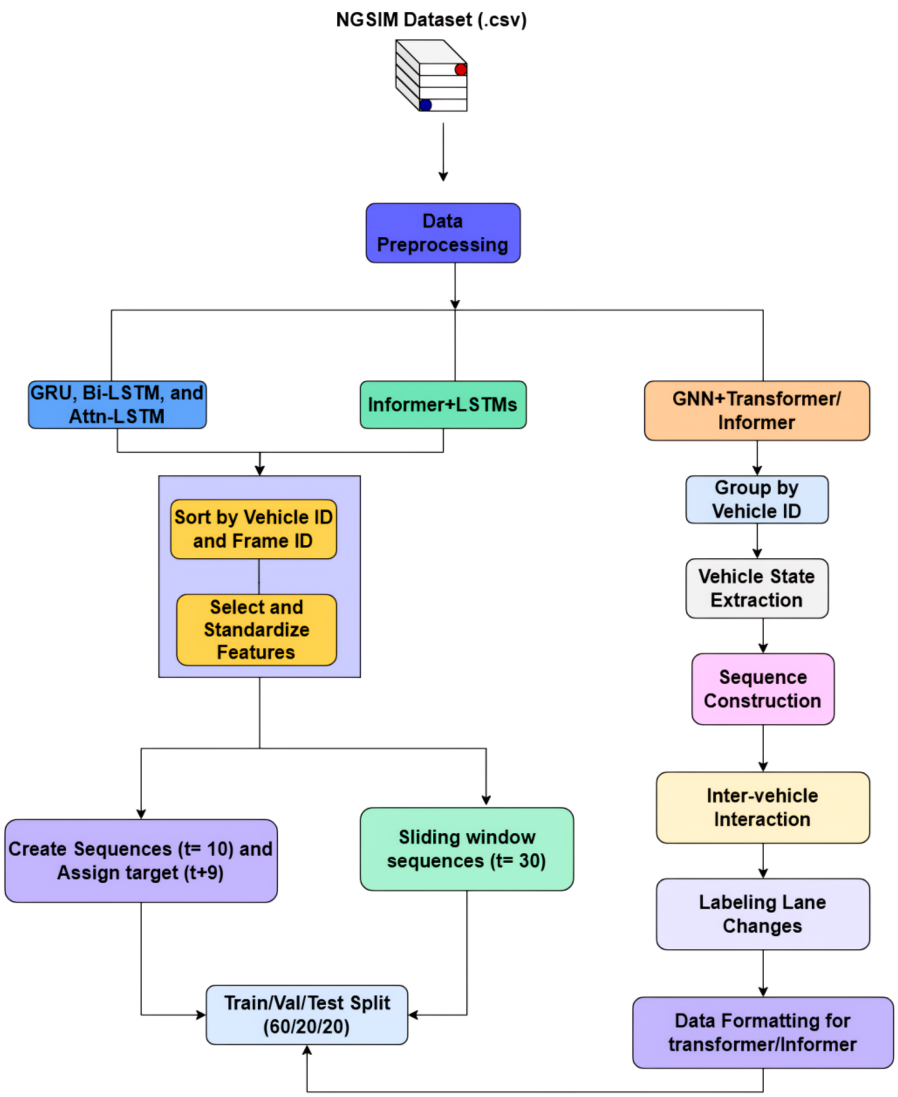

Figure 2: Data Preprocessing of various models.

A subset of features are selected to capture both vehicle dynamics and its interaction with surrounding traffic. The 14 input features that are included in the proposed model are:

- Local X, Local Y: Local 2D coordinates indicating the position of the vehicle in the camera frame.
- Global X, Global Y: Global position coordinates in the real-world coordinate system.
- v length, v Width: Physical dimensions of the vehicle, used to un

derstand its spatial footprint.

- v Class: Category of the vehicle (e.g., motorcycle, car, truck), helpful in modeling diverse driving behaviors.
- v Vel: Vehicle’s instantaneous velocity.
- v Acc: Instantaneous acceleration, important for modeling decelera tion, merging, or sudden maneuvers.
- Lane ID: Identifier of the lane the vehicle is occupying, essential for understanding lateral positioning.

- Preceding, Following: IDs of the leading and trailing vehicles in the same lane, providing contextual interaction.
- Space Headway, Time Headway: Represent spatial and temporal gaps with respect to the leading vehicle.

All continuous features are normalized using z-score standardization to bring them to a common scale. This step is critical for efficient gradient de scent convergence in neural networks and ensures no feature dominates due to scale differences. The output label used for classification is the lane change category. This label is encoded as 0,1, and 2, i.e, left lane change, lane keep ing, and right lane change, respectively. This label is computed by observing changes in the Lane ID of each vehicle over time. As shown in Figure 3, it is considered as follows:

- 0 – Lane change to the left (e.g., Lane 12 to Lane 11)
- 1 – No lane change
- 2 – Lane change to the right (e.g., Lane 3 to Lane 4)

By including both vehicle-centric dynamics and interaction-based fea tures, our feature set is designed to capture both local motion patterns and the contextual environment. This enables our models to learn complex tem poral dependencies and behavioral cues necessary for predicting lane change maneuvers effectively.

3.2. Recurrent Neural Network Models

To establish a strong recurrent baseline for lane change prediction, GRU and Bi-LSTM are implemented and their performance are compared with other models.

3.2.1. Gated Recurrent Unit Model

GRUs are a simplified variant of LSTM networks that are computationally more efficient while still capable of capturing temporal dependencies in sequential data. This makes them particularly attractive for problems involving time-series or trajectory data, such as predicting vehicle lane changes.

The input to the GRU consists of normalized vehicle trajectory features drawn from the NGSIM dataset. These include spatial coordinates (Local X, Local Y), vehicle dynamics (v Vel, v Acc), vehicle class, and interaction variables such as Preceding, Following, and headway measures. Each instance is treated as a sequence of a single timestep (shape: [1, feature dim]), making the model relatively lightweight. The architecture comprises a two layer GRU with 64 hidden units per layer, followed by a fully connected layer that outputs logits for the three lane change classes: left (0), no change (1), and right (2). The model is trained using the cross-entropy loss function, with class weights computed to mitigate the effects of dataset imbalance. The Adam optimizer is used and the model is tranined over 100 epochs.

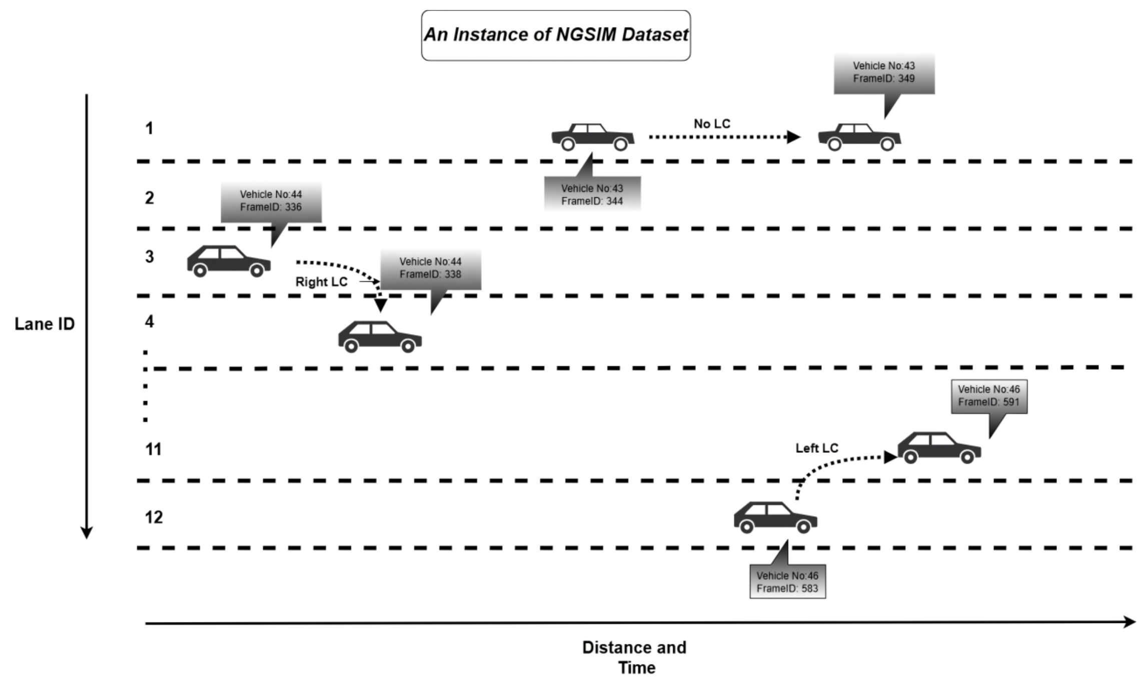

Figure 3: NGSIM Dataset

3.2.2. Bidirectional Long Short-Term Memory (Bi-LSTM) To further enhance the temporal modeling of vehicle trajectory data, the Bi-LSTM network is implemented. Unlike traditional LSTMs, Bi-LSTMs process input sequences in both forward and backward directions, thereby capturing past and future context simultaneously. This is particularly bene ficial for lane change prediction, where driver behavior is influenced not only by past motion but also by upcoming interactions and spatial constraints. The Bi-LSTM architecture consists of a single LSTM layer with 64 hid den units in each direction, followed by a fully connected layer that outputs class probabilities at each timestep. The model is trained using padded se quences generated by grouping trajectory data per vehicle ID. This approach preserves the temporal structure of each individual trajectory. Sequences are zero-padded using PyTorch utilities, and padding positions are masked during loss computation using the ignore index=-100 setting in the cross entropy loss function. It is trained with class-balanced cross-entropy loss using Adam optimization. The model achieved strong performance in both validation and test phases, demonstrating its capability to leverage bidi rectional temporal information for more accurate predictions compared to unidirectional recurrent models like GRUs.

3.3. Attention-based LSTM Model

An Attention-based Long Short-Term Memory (Attn-LSTM) architecture is adopted for lane change prediction to further enhance sequence modeling capabilities. This model augments the traditional bidirectional LSTM by integrating a trainable attention mechanism that allows the model to selec tively focus on relevant time steps in the input sequence, rather than treating all inputs equally.

The input trajectory sequences are constructed using a fixed-length slid ing window (length 10) over normalized vehicle data grouped by Vehicle ID and sorted by Frame ID. These sequences were used to predict the lane change behavior of a vehicle at the last timestep. The attention layer computes weights over each timestep in the LSTM output, producing a context vector that captures the most influential features across the sequence. This context is then passed through a fully connected classifier to predict one of the three lane change classes: left, no change, or right.

The model was trained using cross-entropy loss with class balancing and optimized using the Adam optimizer. Results show that the attention mecha nism significantly improved the model's ability to distinguish subtle temporal patterns, leading to better generalization and more robust predictions, par ticularly in edge cases where standard LSTM models tend to underperform.

3.4. Transformer Model

This model predicts future lane change behavior of a vehicle by combining a Transformer encoder-decoder with a pretrained Graph Neural Network (PCurveNet). The task is treated as a multi-class classification problem with outputs: left lane change (LLC), right lane change (RLC), and lane keeping (LK). The input to the model is a temporal sequence tensor as in Equation 1:

$$V \in \mathbb{R}^{N \times C \times T} \tag{1}$$

where N is the number of vehicles, C = 21 is the number of features per vehicle (e.g., position, speed), and T is the number of past time steps. The model predicts the future behavior of the ego vehicle as specified in Equation 2 :

$$\hat{y}_{ego}^{T+1} = f(V) \tag{2}$$

where $f(\cdot)$ represents the full prediction pipeline from input to output class scores. To capture interactions between vehicles, a pre-trained GNN (PCurveNet) generates high-level representations in terms of the encoded feature vector $Z^T{}_{ego}$ as specified in Equation 3 :

$$Z_{ego}^{T} = g_\theta(V) \tag{3}$$

where g_θ is the GNN with learned parameters θ. Since Transformers lack inherent temporal order, sinusoidal positional encodings are added to the inputs as in Equation 4:

$$PE_{t,d} = \begin{cases} \sin\left(\frac{t}{10000^{d/C'}}\right), & \text{if } d \text{ is even} \\ \cos\left(\frac{t}{10000^{d/C'}}\right), & \text{if } d \text{ is odd} \end{cases} \tag{4}$$

where t is the time index, d the feature dimension, and C' the embedding dimension. This helps the model identify the position of each time step in the sequence. The Transformer encoder uses multi-head attention to capture temporal dependencies and interactions as in Equation 5:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{5}$$

Here, Q, K, and V are the query, key, and value matrices, and d_k is the key dimension. This computes weighted relevance between inputs to focus on important context. The decoder processes the ego vehicle's state and produces class logits as specified in Equation 6 :

$$\hat{y}_{ego}^{T+1} = \text{softmax}(h_{ego}^{T+1}) \tag{6}$$

where h_{ego}^{T+1} is the final hidden state for the ego vehicle at the future time step, and the softmax converts it to class probabilities. As given in Equation 7, the model is trained using focal loss to manage class imbalance (e.g., more Lane Keep than Left Lane Change or Right Lane Change):

$$\mathcal{L} = -\frac{1}{B}\sum_{i=1}^{B} \alpha_i (1 - \hat{p}_i)^\gamma \log(\hat{p}_i) \tag{7}$$

Here, B is the batch size, $\hat{p}_i$ is the predicted probability of the true class for the i-th sample, α_i is a class-specific weighting factor, and γ reduces the loss contribution of well-classified examples. This helps the model focus more on hard-to-classify, minority class samples.

3.5. Informer based Hybrid Models

This study also investigates Informer's effectiveness in long-sequence fore casting in hybrid configurations that include Informer combined with GNN,

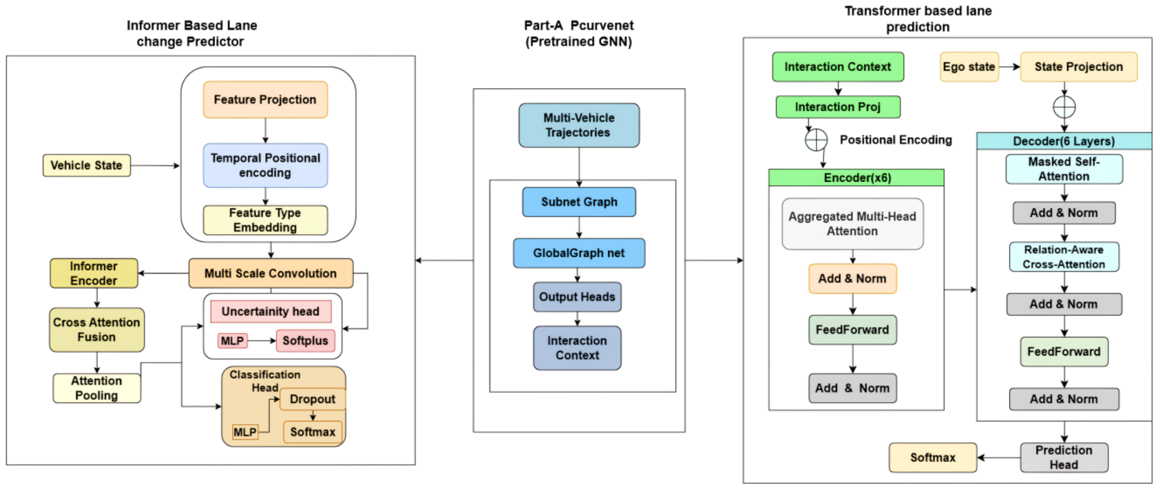

Figure 4: Proposed architecture combining GNN, Transformer, and Informer modules for lane change prediction

Transformer, and LSTM. Each combination is designed to facilitate complementary advantages in temporal sequence modeling and interaction reasoning.

3.5.1. Informer + GNN

This model improves computational efficiency without sacrificing prediction accuracy by replacing the standard Transformer with the Informer architecture and it is shown in Figure 4. Informer is specifically designed for long-range sequence modeling and leverages *ProbSparse attention*, which significantly reduces the computational cost of attention mechanisms. The model is trained and evaluated on the NGSIM dataset.

The input to the model is a historical vehicle trajectory tensor represented as in Equation 8:

$$\mathbf{S} \in \mathbb{R}^{M \times D \times T} \tag{8}$$

where *M* is the number of vehicles in the scene, *D* is the number of motion features per vehicle (such as position, velocity, and acceleration), and *T* is the number of historical time steps. The goal is to predict the lane-change maneuver of the ego vehicle at the next time step, formulated as in Equation 9:

$$\hat{l}^{\text{ego}}_{T+1} = \mathcal{F}(\mathbf{S}) \tag{9}$$

To capture vehicle-to-vehicle interactions, a pretrained PCurveNet GNN is used to extract high-level spatiotemporal features. This is defined as in Equation 10:

$$\mathbf{R}_T^{\text{ego}} = \mathcal{G}_\phi(\mathbf{S})$$

where G_ϕ is the GNN with learnable parameters ϕ, and R_T^{ego} is the ego vehicle's interaction-aware feature representation at the last observed time step. These features are further refined using multi-scale temporal convolu tions with kernel sizes 3, 5, and 7, enabling the model to capture patterns at different temporal resolutions. Learnable temporal positional encodings E_t and feature-type embeddings E_f are added to inject contextual and semantic information into the sequence.

For long-sequence modeling, the Informer encoder is employed, which re places traditional full attention with ProbSparse attention as in the Equation 11. Instead of computing attention over all query-key pairs, it focuses only on the most relevant top-*k* interactions:

$$\text{Attention}(\mathbf{Q}', \mathbf{K}', \mathbf{V}') \approx \text{Top}_k\left(\frac{\mathbf{Q}'\mathbf{K}'^\top}{\sqrt{d}}\right)\mathbf{V}' \tag{11}$$

Here, Q', K', V' are the query, key, and value matrices, and *d* is the key dimensionality. This sparse approximation reduces the time complexity from $O(L^2)$ to $O(L \log L)$, making it feasible to process long input sequences efficiently.

A cross-attention mechanism is used to integrate the ego vehicle's current temporal representation R_T^{ego} with the globally encoded interaction context C_{global}. The output is aggregated using an attention pooling operation as specified in Equation 12:

$$\mathbf{u}_{\text{scene}} = \text{AttentionPool}(\mathbf{C}_{\text{global}}) \tag{12}$$

The final classification output is generated from the decoder's hidden state for the ego vehicle as specified in Equation 13:

$$\hat{l}_{T+1}^{\text{ego}} = \text{softmax}(\mathbf{R}_T^{\text{ego}}) \tag{13}$$

To improve reliability, the model also includes an uncertainty head that estimates the confidence associated with each predicted class as specified in Equation 14:

$$\hat{\sigma}_j = \text{Softplus}(\mathbf{W}_\sigma \mathbf{z}_{\text{ego}} + \mathbf{b}_\sigma) \tag{14}$$

where W_σ and b_σ are learnable parameters, and $\hat{\sigma}_j$ is the predicted uncertainty for class *j*. This allows the system to quantify uncertainty in predictions, which is especially valuable in safety-critical scenarios such as autonomous driving.

To address the class imbalance often present in real-world trajectory datasets, the model employs the focal loss function as in Equation 15:

$$\mathcal{L}_{\text{focal}} = -\frac{1}{B}\sum_{i=1}^{B} \alpha_i (1-\hat{q}_i)^\gamma \log(\hat{q}_i) \tag{15}$$

Here, B is the batch size, $\hat{q}_i$ is the predicted probability of the true class for the *i*-th sample, α_i is a class balancing weight, and γ is a focusing parameter that reduces the impact of well-classified examples on the loss.

3.5.2. Informer Transformer Model

To capture long-range temporal dependencies in trajectory data for lane change prediction, Informer Transformer model is used that leverage the powerful attention mechanism to process sequential data. As shown in Figure 5 (a), the model begins with an encoder that projects 14-dimensional input features into a higher-dimensional space using a linear layer, followed by positional encoding to preserve the order of frames within each trajectory segment. This encoded representation is processed by stacked Transformer encoder layers that learn contextual representations of the input sequence using multi-head self-attention.

To decode the sequence representation into class predictions, a minimal istic decoder is used which consists of learnable positional input combined with transformer decoder layers. The final output is derived by applying a linear classification layer to the decoder's output, yielding logits over the three lane change classes.

This architecture is chosen for its ability to model both local and global interactions among trajectory features over time. Its non-recurrent struc ture allows for better parallelism and faster training compared to traditional

RNNs, while the self-attention mechanism provides interpretability and ro bustness in modeling sequential behaviors critical for understanding lane changes.

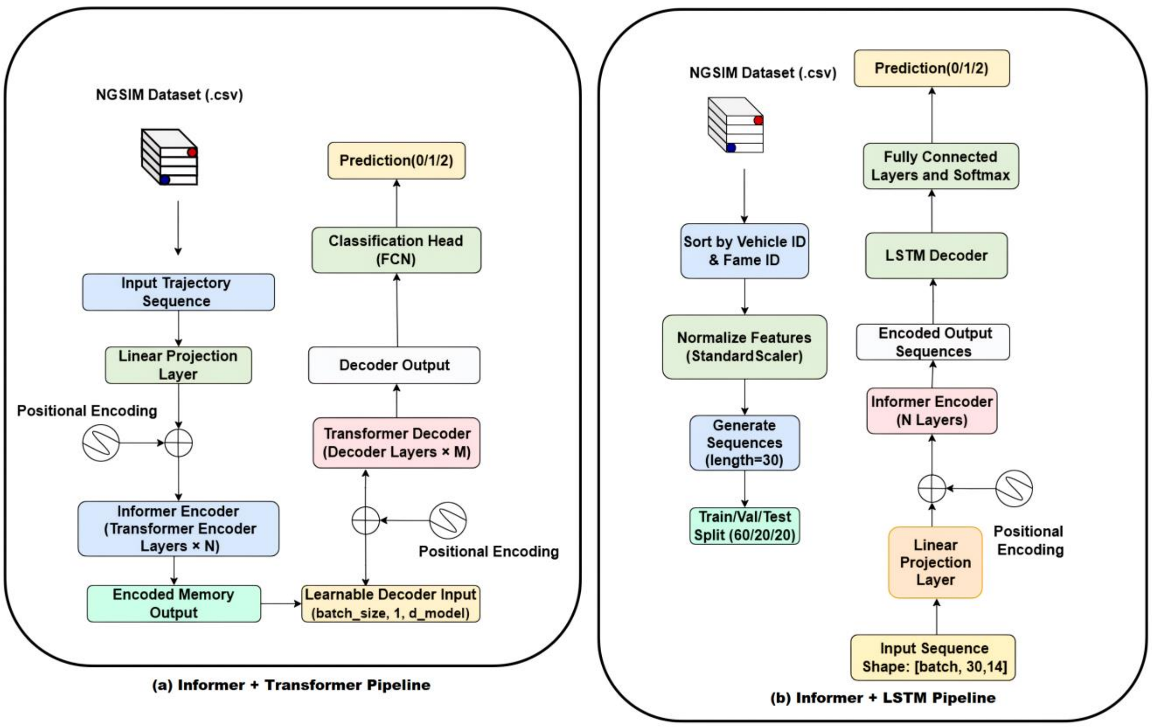

Figure 5: Informer based Hybrid Models

3.5.3. Informer + LSTM

This proposed model for lane change prediction is a hybrid architecture combining Informer-based transformer encoding with an LSTM-based decod ing mechanism and it is depicted in Figure 5 (b). The raw NGSIM dataset, provided in CSV format, is first preprocessed by sorting the vehicle trajec tory data by Vehicle ID and Frame ID to ensure temporal consistency. The relevant numerical features such as local and global positions, velocity, accel eration, lane ID, and headway measures are normalized using standard score normalization via StandardScaler to stabilize training. Following prepro cessing, the data is structured into fixed-length sequences of 30 time steps per vehicle. Each sequence is labeled based on the lane change value at the final time step of the sequence, representing the vehicle's intended maneu ver: left lane change (0), lane keeping (1), or right lane change (2). These sequences are then split into training and validation sets with an 80:20 ratio.

The model ingests input sequences of shape [batch size, 30, 14], where 14 denotes the number of standardized features. Each input sequence is first passed through a linear projection layer which maps the 14-dimensional input into a higher-dimensional space defined by a model dimension d model (e.g., 512). To encode the temporal order of the sequence, sinusoidal positional en coding is added to the projected features. The transformed sequence is then processed by the encoder component, which resembles the architecture of the Informer model. This encoder is built from multiple TransformerEncoder layers (typically 2), each consisting of multi-head self-attention mechanisms, feedforward layers, and layer normalization blocks. This component cap tures long-term temporal dependencies and inter-feature interactions within the sequence efficiently.

The output from the encoder, still maintaining the sequence shape [batch size, 30, d model], is then passed into an LSTM layer. The LSTM acts as a decoder that compresses the sequence information into a final hid den state vector of dimension hidden dim (e.g., 256), capturing temporal dynamics in a condensed form. The final hidden state from the LSTM is then fed into a fully connected (FC) layer which maps the representation to three output logits corresponding to the three lane change classes. The model outputs are passed through a softmax activation function to obtain class probabilities. The final predicted class is determined by selecting the class with the highest probability. The model is trained using a categorical cross-entropy loss function and optimized using the Adam optimizer with

a learning rate of 10^{-4}. The training loop involves forward propagation, loss computation, backpropagation, and parameter updates across multiple epochs. Model performance is evaluated using classification metrics including accuracy, confusion matrix, and precision-recall-based scores on the valida tion set.

4. Performance Analysis

This section presents a comprehensive analysis of the models developed for lane change prediction using the NGSIM dataset. Multiple recurrent and transformer-based architectures—including GRU, Bi-LSTM, Attention LSTM, and Informer-Transformer hybrids—were evaluated and compared. Each model was trained under varying hyperparameter settings to identify optimal configurations. The performance of the models was quantitatively as sessed using standard classification metrics such as Accuracy, Precision, Re call, and F1-Score, reported across all three lane change classes (Left, Keep, Right). These metrics are organized into comparative tables to facilitate model benchmarking. Additionally, computational efficiency was examined by measuring training time and model inference overhead, which are illus trated through bar charts for visual clarity. The evaluation was conducted in a controlled simulation environment, as previously described, ensuring reproducibility and fair comparison across all experiments.

4.1. Simulation Environment

All models were developed and evaluated using a combination of cloud based and local computing environments. The primary training and exper imentation were conducted on Google Colaboratory, utilizing an NVIDIA Tesla T4 GPU with 16GB of VRAM. This setup provided the necessary acceleration for training deep learning models such as Bi-LSTM, Attention LSTM, and Informer-based architectures on sequential NGSIM trajectory data. PyTorch served as the main deep learning framework, with support ing libraries including NumPy, scikit-learn, and Matplotlib for preprocessing, metric computation, and visualization.

For baseline comparisons and performance profiling, selected models were also tested on a local machine equipped with a 12th Gen Intel(R) Core(TM) i7-1255U CPU (1.70GHz) and 16GB RAM. This allowed for analysis of com putational overhead, memory efficiency, and training time under resource constrained environments. The use of both GPU and CPU environments

ensured a robust evaluation across different hardware configurations and re inforced the practical applicability of the models.

4.2. Performance Analysis of RNN models and Attn-LSTM Models Table 2 presents a comparative analysis of the GRU, Bi-LSTM, and Attention-based

LSTM (Attn-LSTM) models under varying hyperparame ter settings. Each model was evaluated based on accuracy, precision, recall, and F1 score to assess its effectiveness in lane change prediction using the NGSIM dataset. The results reflect the impact of learning rate (α) and hidden size (n) on model performance.

Attn-LSTM demonstrates superior performance, with Bi-LSTM trailing behind and GRU being the least effective. Attn-LSTM leverages a selec tive attention mechanism, enabling it to prioritize key temporal patterns and subtle trajectory shifts crucial for lane change decisions. Bi-LSTM ben efits from bidirectional context aggregation, enhancing its sequence com prehension compared to unidirectional GRU. GRU, while computationally lightweight, exhibits limited contextual depth, which constrains its ability to model intricate temporal transitions in vehicle behavior.

Table 2: Performance of GRU, Bi-LSTM, and Attn-LSTM

Model	Hyper-param	Acc.	Prec.	Rec.	F1
GRU	α=5e−4, n=64	0.6671	0.6344	0.6351	0.6334
	α=1e−4, n=128	0.6582	0.6229	0.6235	0.6225
Bi-LSTM	α=1e−3, n=64	0.7853	0.7852	0.7768	0.7804
	α=1e−3, n=128	0.7641	0.7638	0.7573	0.7602
Attn-LSTM	α=1e−3, n=64	0.8944	0.8922	0.8891	0.8909
	α=5e−3, n=96	0.8847	0.8784	0.8918	0.8823

4.3. Performance Analysis of Hybrid Informer Models

To assess the effectiveness of the proposed Informer-based models for lane change prediction, we present a detailed comparison of two architectures: In former+Transformer and Informer+LSTM. Each model was trained on the NGSIM dataset using varying hyperparameter configurations, including the learning rate (α), model dimension (d_{model}), and the number of attention heads (n_{heads}). The performance of each configuration is reported in terms of standard classification metrics: *Accuracy*, *Precision*, *Recall*, and *F1 Score*. These metrics reflect the model's ability to correctly classify lane change behavior across left, keep, and right transitions.

Table 3 below demonstrates how tuning these parameters influences the prediction quality, with Informer+LSTM achieving higher scores in multiple setups. *Informer+LSTM* consistently outperforms *Informer+Transformer* across all metrics, especially at lower learning rates and higher model dimensions. *Informer+LSTM* integrates Informer's efficient long-range temporal encoding with LSTM's strong sequential decoding, enabling better retention of motion trends and transitions in vehicle behavior. *Informer+Transformer*, while effective, lacks the temporal granularity of LSTM decoders, leading to slightly lower recall and F1 in complex lane transitions. The hybrid *Informer+LSTM* architecture benefits from combining sparse attention with recurrent temporal refinement, making it more suited for fine-grained trajectory prediction.

Table 3: Performance of Hybrid Informer Models

Model	Hyper-param	Accuracy	Precision	Recall	F1 score
Informer + Transformer	$\alpha = 5 \times 10^{-4}$ $d_{\text{model}} = 256$ $n_{\text{heads}} = 8$	0.9496	0.9492	0.9458	0.9474
	$\alpha = 1 \times 10^{-4}$ $d_{\text{model}} = 128$ $n_{\text{heads}} = 4$	0.9725	0.9702	0.9724	0.9713
	$\alpha = 1 \times 10^{-4}$ $d_{\text{model}} = 256$ $n_{\text{heads}} = 4$	0.9805	0.9807	0.9796	0.9802
Informer + LSTM	$\alpha = 5 \times 10^{-4}$ $d_{\text{model}} = 256$ $n_{\text{heads}} = 8$	0.9552	0.9551	0.9552	0.9550
	$\alpha = 1 \times 10^{-4}$ $d_{\text{model}} = 128$ $n_{\text{heads}} = 4$	0.9828	0.9828	0.9818	0.9808
	$\alpha = 1 \times 10^{-4}$ $d_{\text{model}} = 256$ $n_{\text{heads}} = 4$	**0.9854**	**0.9848**	**0.9852**	**0.9836**

4.4. Performance Analysis of Transformer and Informer GNN variants To evaluate the effectiveness of integrating graph-based spatial interac tions with sequence modeling, we conducted experiments using two hybrid architectures: Transformer + GNN and Informer + GNN. Both models were trained using various hyperparameter settings, including a fixed learning rate of $1 * 10^{-4}$ for stable convergence, batch sizes of 32 and 64 to observe effects on generalization, and focal loss parameters to handle class imbalance. Specif ically, alpha was set to 0.25 to emphasize minority classes, and gamma was varied between 1.5 and 2.0 to focus more on hard-to-classify samples. Across all configurations, Informer + GNN consistently outperformed Transformer + GNN in terms of accuracy, precision, recall, and F1 score as depicted in Table 4. Informer's use of probabilistic attention allowed it to better capture long-range temporal patterns—crucial for predicting lane changes. When paired with GNN's ability to model spatial interactions, it offered stronger contextual understanding and generalization. In contrast, Transformer + GNN was less effective in modeling long-term dependencies, leading to lower overall performance. Thus, Informer + GNN proved to be the more accurate and robust choice for lane change prediction.

Table 4: Performance of Transformer and Informer Variants

Model	Hyper-parameter	Accuracy	Precision	Recall	F1 Score
Transformer + GNN	learning rate $= 1 \times 10^{-4}$ batch size $= 32$ focal alpha $= 0.25$ focal gamma $= 2.0$	0.9266	0.9310	0.9201	0.9181
	learning rate $= 1 \times 10^{-4}$ batch size $= 64$ focal alpha $= 0.25$ focal gamma $= 1.5$	0.9008	0.8954	0.8986	0.8882
Informer + GNN	learning rate $= 1 \times 10^{-4}$ batch size $= 64$ focal alpha $= 0.25$ focal gamma $= 1.5$	0.9401	0.9281	0.9590	0.9368
	learning rate $= 1 \times 10^{-4}$ batch size $= 32$ focal alpha $= 0.25$ focal gamma $= 2.0$	**0.9561**	**0.9572**	**0.9611**	**0.9516**

4.5. Best Model's Metrics

To highlight the most effective models from each major category in our lane change prediction study, we selected the top-performing architecture from (i) traditional sequence models such as GRU, Bi-LSTM, and Attn LSTM, (ii) spatial interaction models including Transformer + GNN and Informer + GNN, and (iii) hybrid temporal models such as Transformer + Informer and Informer + LSTM. As summarized in Table 5, the best model in the traditional category is Attn-LSTM, which performs well but shows lower accuracy in predicting lane changes compared to others. In the spatial interaction group, Informer + GNN achieves strong accuracy across all classes, especially for Left and Right Lane Change. However, the best overall performance comes from the Informer + LSTM model, which achieves the highest accuracy in all three behavior classes and an overall accuracy of 0.9854. This superior performance is due to the synergy between Informer's long-range attention mechanism and LSTM's ability to model sequential patterns, enabling a deeper understanding of both short- and long term vehicle behavior.

Table 5: Model Accuracies Grouped by Class Type

Model	Accuracies			
	Left LC	**Keep**	**Right LC**	**Overall**
Attn-LSTM	0.8317	0.9228	0.8536	0.8944
Informer + GNN	**0.9812**	0.9385	0.9657	0.9561
Informer + LSTM	0.9708	**0.9862**	**0.9739**	**0.9854**

4.6. Training Time Comparison

To better understand the computational efficiency of each architecture, we recorded the total training duration for all models over a fixed number of epochs. This comparison provides insights into how model complexity, archi tectural design (such as recurrent layers, attention mechanisms, and hybrid combinations like *Informer+LSTM* or *Transformer+GNN*), and parameter count influence the overall training time.

The bar graph in Figure 6 highlights the time that each model took to complete training, enabling an objective assessment of the trade-off between performance and computational cost. GRU, Bi-LSTM, and Attn-LSTM offer the fastest training times, while Transformer+GNN is the slowest, followed by Informer+GNN. GRU and Bi-LSTM train quickly due to their lightweight structure and fewer parameters. Attn-LSTM adds minimal overhead from attention but retains efficient recurrent processing. Informer+LSTM and In former+Transformer take longer due to the Informer's sparse attention and transformer-based encoding. Transformer+GNN and Informer+GNN are the most time-consuming due to added graph-based computation, message pass ing, and large parameter count.

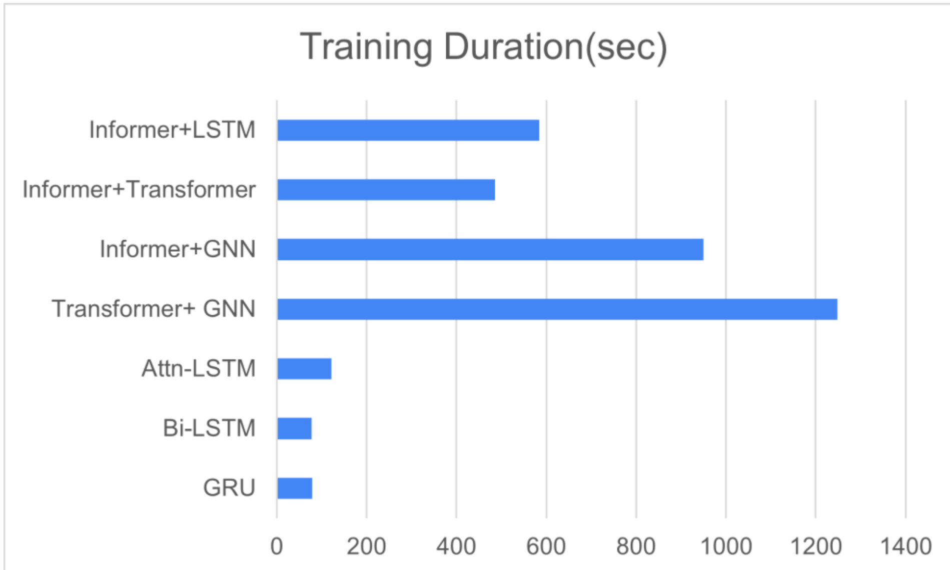

Figure 6: Training Time Vs Models

4.7. Memory Overhead Comparison

In addition to evaluating the performance and training time of each model, we also compare their memory consumption to assess their compu tational demands. This includes both GPU and CPU memory utilization during training. By analyzing the memory overhead, we gain a deeper un derstanding of how each model scales in terms of hardware resource require ments.

Memory usage was recorded at the end of the training phase for each model, capturing both GPU memory allocation (for CUDA-enabled mod els) and CPU RAM usage. The resulting bar graph in Figure 7 illustrates the comparative memory footprint of each architecture.GRU and Bi-LSTM

exhibit the lowest memory overhead, making them suitable for resource constrained environments. Informer+LSTM and Attn-LSTM, while deliver ing superior performance, consume significantly more CPU memory. GRU is computationally efficient due to its simpler architecture, resulting in minimal GPU and CPU usage. Informer+LSTM and Attn-LSTM offer high predic tive accuracy but at the cost of increased memory usage, especially on the CPU side, due to the combination of attention mechanisms and temporal modeling. Informer+Transformer balances performance and memory bet ter than Informer+LSTM but still requires considerable resources. Hybrid models like Informer+GNN and Transformer+GNN incur moderate memory usage, attributed to added complexity from graph-based interactions.

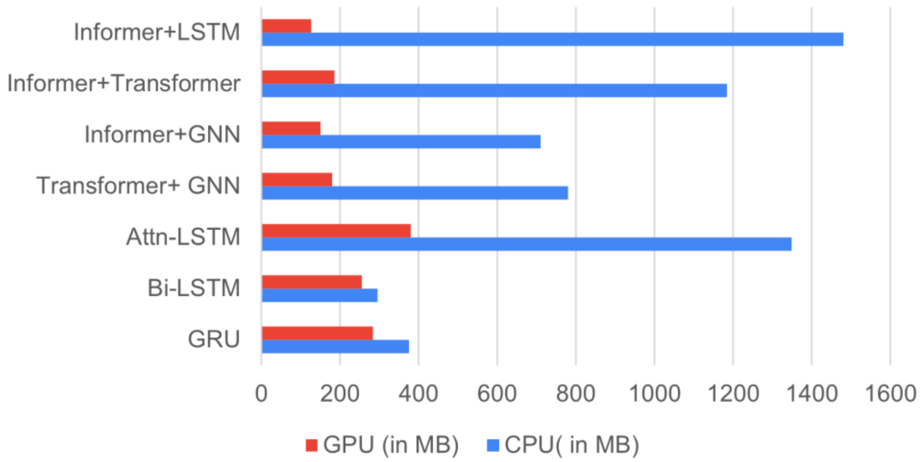

Figure 7: Memory Overhead Vs Models

5. Conclusion

This study presents a comprehensive analysis of hybrid deep learning models for predicting human lane change behavior using the NGSIM dataset. We evaluate models ranging from GRU and Bi-LSTM baselines to advanced hybrid architectures like Informer+LSTM and Transformer+GNN. The In former+LSTM model, which combines the strengths of long-range attention and sequential modeling, achieved the highest accuracy across all lane change types. While computationally intensive, this architecture proved superior at capturing the subtle spatial and temporal patterns critical for robust, real time prediction in dynamic traffic environments. Our work highlights the effectiveness of hybrid models and their trade-offs between performance and computational efficiency for autonomous vehicle applications.

References

[1] Ali, Hamza and Zheng, Jianfeng and Alrashidi, Raid S. and Alfawzan, Mohammed S. and Ullah, Irfan and Jamal, Arshad, *Simulating the Adoption of Autonomous Vehicles in CPEC Logistics and Transporta tion: A Case Study of a Scenario Approach* , IEEE Access, 1-1, 2025, doi: 10.1109/ACCESS.2025.3590148.

[2] Hu,Dong and Huang, Chao and Zhao, Jing and Zhao, Yifan and

Wu,Jingda, *Autonomous Driving Economic Car-following Motion Strategy based on Adaptive Rollout Model-based Policy Optimization* , IEEE Transactions on Transportation Electrification, 1-1, 2025, doi:10.1109/TTE.2025.3590199

[3] Abid, Nesrine and Ouni, Tarek and Abid and Mohamed, *Vehicle de tection for intelligent traffic surveillance system*, 2020 5th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP),1-5, 2020, doi:10.1109/ATSIP49331.2020.9231936.

[4] Segu, Girish Sai Pavan Kumar and Sivannarayana and Annam Devi Satya Naga and Ramesh, *Real Time Road Lane Detec tion and Vehicle Detection on YOLOv8 with Interactive Deploy ment*, 2024 IEEE 16th International Conference on Computational Intelligence and Communication Networks (CICN), 267-272, 2024, doi:10.1109/CICN63059.2024.10847549.

[5] Gao, Kai and Li, Xunhao and Chen, Bin and Hu, Lin and Liu, Jian and Du, Ronghua and Li and Yongfu, *Dual Transformer Based Prediction for Lane Change Intentions and Trajectories in Mixed Traffic Environment*, IEEE Transactions on Intelligent Transportation Systems, 24,6,6203- 6216,2023, doi:10.1109/TITS.2023.3248842.

[6] Lu, Yuhuan and Zhang, Zhen and Wang, Wei and Zhu, Yiting and Chen, Tiantian and Al-Otaibi, Yasser D. and Bashir, Ali Kashif and Hu, Xip ing *Knowledge-Driven Lane Change Prediction for Secure and Reliable Internet of Vehicles*, IEEE Transactions on Intelligent Transportation Systems, 1-12,2025, doi:10.1109/TITS.2025.3526341.

[7] Kunsong Shi and Yuankai Wu and Haotian Shi and Yang Zhou and Bin Ran, *An integrated car-following and lane changing vehicle trajec tory prediction algorithm based on a deep neural network*, Physica A: Statistical Mechanics and its Applications,599,127303,0378-4371,2022, doi:https://doi.org/10.1016/j.physa.2022.127303.

[8] Mingxing Peng and Xusen Guo and Xianda Chen and Kehua Chen and Meixin Zhu and Long Chen and Fei-Yue Wang, *LC-LLM: Explainable lane-change intention and trajectory predictions with Large Language Models*, Communications in Transportation Research, 5,100170,2772-4247,2025, doi:https://doi.org/10.1016/j.commtr.2025.100170.

[9] Chen, Kequan and Li, Zhibin and Liu, Pan and Xu, Chengcheng and

Wang, Yuxuan, *Real-Time Lane-Changing Crash Prediction Model at the Individual Vehicle Level Using Real-World Trajectories Prior to Crashes*, 2024, doi:http://dx.doi.org/10.2139/ssrn.4829767

[10] Qasemabadi, Armin Nejadhossein and Mozaffari, Saeed and Rezaei, Mahdi and Ahmadi, Majid and Alirezaee, Shahpour, *A Novel Model for Driver Lane Change Prediction in Cooperative Adaptive Cruise Control Systems*, 2023 International Symposium on Signals, Circuits and Sys tems (ISSCS), 1-4,2023, doi:10.1109/ISSCS58449.2023.10190867.

[11] Do, Jongyong and Han, Kyoungseok and Choi, Seibum B. *Lane Change–Intention Inference and Trajectory Prediction of Surround ing Vehicles on Highways*, IEEE Transactions on Intelligent Vehicles, 8,7,3813-3825,2023, doi:10.1109/TIV.2023.3266102.

[12] Huang, Yunjie and Song, Xiaozhuang and Zhu, Yuanshao and Zhang, Shiyao and Yu, James J. Q., *Traffic Prediction With Trans fer Learning: A Mutual Information- Based Approach*, IEEE Trans actions on Intelligent Transportation Systems, 24,8,8236-8252,2023, doi:10.1109/TITS.2023.3266398.

[13] Hongrui Zhang and Yonggang Wang and Shengrui Zhang and Jingtao Li and Qushun Wang and Bei Zhou, *Improved time series models for the prediction of lane-change intention*, Transportation Letters,Taylor & Francis, 17,4, 747-761, 2025, doi:10.1080/19427867.2024.2379702.

[14] Qingwen Xue ,Yingying Xing , Jian Lu, *An integrated lane change prediction model incorporating traffic con text based on trajectory data*, Transportation Research

Part C: Emerging Technologies, 141,103738,0968-090X,2022 doi:https://doi.org/10.1016/j.trc.2022.103738.

[15] Jinbao Zhang and Jaeyoung Lee and Mohamed Abdel-Aty and Ou Zheng and Guiming Xiao, *Enhanced index of risk assessment of lane change on expressway weaving segments: A case study of an expressway in China*, Accident Analysis & Prevention, 180,106909,0001-4575,2023, doi:https://doi.org/10.1016/j.aap.2022.106909.

[16] Hongyu Guo and Mehdi Keyvan-Ekbatani and Kun Xie, *Lane change detection and prediction using real-world connected vehicle data*, Trans

portation Research Part C: Emerging Technologies, 142,103785,0968-090X,2022 doi:https://doi.org/10.1016/j.trc.2022.103785.

[17] Yuhuan Lu, Qinghai Lin, Haiyang Chi and Jin-Yong Chen, *Auto matic incident detection using edge-cloud collaboration based deep learn ing scheme for intelligent transportation systems*, Applied Intelligence, 53,1573-7497,24864 - 24875, 2023, doi:10.1007/s10489-023-04673-7.

[18] Wenjian Sun , Linying Pan , Jingyu Xu , Weixiang Wan and Yong Wang, *Automatic driving lane change safety prediction model based on LSTM*, arXiv,2403.06993,2024, doi:https://arxiv.org/abs/2403.06993.

[19] Li, Lin and Zhao, Wanzhong and Xu, Can and Wang, Chun yan and Chen, Qingyun and Dai, Shijuan, *Lane-Change Inten tion Inference Based on RNN for Autonomous Driving on High ways*, IEEE Transactions on Vehicular Technology, 70,6,5499-5510,2021, doi:10.1109/TVT.2021.3079263.

[20] Xu, Dongwei and Shang, Xuetian and Peng, Hang and Li, Haijian, *MVHGN: Multi-View Adaptive Hierarchical Spatial Graph Convolution Network Based Trajectory Prediction for Heterogeneous Traffic-Agents*, IEEE Transactions on Intelligent Transportation Systems, 24,6,6217-6226,2023, doi:10.1109/TITS.2023.3248090.

Authors:

1. Jayaram Peggem
2. Dr. Nithya B
3. Gogada Harsha Vardhan
4. S.Venkateswara Reddy

Declaration of Interest Statement

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐ The author is an Editorial Board Member/Editor-in-Chief/Associate Editor/Guest Editor for this journal and was not involved in the editorial review or the decision to publish this article.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: