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

Michael Roberg, *Student Member, IEEE,* Tibault Reveyrand, *Member, IEEE,*
Ignacio Ramos, *Student Member, IEEE,* Erez Falkenstein, *Student Member, IEEE,* and Zoya Popović, *Fellow, IEEE*

***Abstract*—This paper reviews sixteen recent research works on lane change intention and trajectory prediction, risk assessment, and safety modeling in intelligent transportation systems. The surveyed approaches span deep learning architectures—including Transformers, LSTM, RNNs, and Spiking Neural Networks—as well as knowledge-driven and graph-based models. Techniques address both individual and multi-agent scenarios, leveraging real-world datasets such as NGSIM and HighD, and incorporate contextual information like traffic density, vehicle interactions, and risk indices. While these methods demonstrate notable improvements in prediction accuracy, early detection, and interpretability, they commonly face challenges related to computational demands, data quality, generalizability to diverse environments, and real-time deployment. A comparative analysis highlights the advances and limitations across these state-of-the-art models. Index Terms—Lane change prediction, Trajectory forecasting, Deep learning, Transformer, LSTM, Risk assessment, Autonomous vehicles, Intelligent transportation systems, Real-time safety.**

***Index Terms*—Lane change prediction, Trajectory forecasting, Deep learning, Transformer, LSTM, Risk assessment, Autonomous vehicles, Intelligent transportation systems, Real-time safety.**

I. Introduction

The development of autonomous vehicles (AVs) is a significant advancement in intelligent transportation, promising increased safety,[1] reduced traffic, and improved mobility. A crucial and intricate aspect of autonomous driving 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 [2] 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. 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.

To bridge this behavioral gap, researchers increasingly utilize naturalistic driving datasets like the NGSIM dataset,[3] which provides fine-grained, real-world vehicle trajectory data across diverse traffic scenarios. These datasets serve as training grounds for advanced machine learning models, particularly deep learning architectures such as Bidirectional LSTMs, Transformers, [4] and hybrid Informer-based models. These models are adept at capturing spatio-temporal dependencies and interaction dynamics. By learning patterns in human lane change behavior from such datasets, these models enable AVs to anticipate and react in a manner consistent with human expectations. This paper focuses on developing and comparatively analyzing such models to improve lane change prediction, leveraging weak features, incorporating noise resilience, and evaluating real-world computational efficiency, thus contributing to more robust, interpretable, and deployment-ready autonomous decision-making 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 approaches often relied on handcrafted features (e.g., relative vehicle positions, speeds, accelerations, lane occupancy) combined with rule-based decision logic or shallow classifiers like Support Vector Machines (SVMs) and Random Forests[5]. 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. These models excel at capturing temporal dependencies in driving behavior, learning from continuous sensor data streams to predict upcoming maneuvers. More recent approaches have introduced Bidirectional LSTMs (Bi-LSTMs) [4] [5] 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 mechanisms 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 intentions and actions of surrounding traffic participants in addition to the ego vehicle.

Despite these advances, the current body of research still faces key limitations. Firstly, most models are trained and evaluated under ideal conditions with clean, dense, and highly informative feature sets. However, in real-world deployments, data can be noisy, weakly correlated, or sparsely populated due to sensor limitations, occlusions, or loss of contextual cues. Secondly, few studies systematically explore the effect of label noise, which is common in trajectory annotation, especially when lane-change labels are algorithmically generated without human verification. Thirdly, while state-of-the-art models often achieve competitive accuracy, their training time, GPU/CPU memory usage, and inference latency are rarely reported. This limits understanding of their practicality for real-time deployment in resource-constrained embedded systems typical of autonomous vehicles [6].

Furthermore, lane-change prediction research has largely remained compartmentalized across modeling paradigms. RNN-based architectures (LSTMs or GRUs) and Transformer-based approaches are typically explored in isolation. Few studies attempt to combine the complementary strengths of these models into hybrid architectures—for instance, by using Transformers to model long-range temporal dependencies and then applying LSTM/GRU modules for fine-grained temporal decoding. This separation overlooks the potential of hybrid models to achieve a balance between global context modeling, local sequential smoothness, and computational efficiency. Similarly, combinations like Informer encoders with RNN decoders, or Transformer encoders followed by MLP classification heads, remain underexplored despite their intuitive appeal and potential gains in robustness and generalization.

Finally, little emphasis is placed on understanding how these models perform under degraded conditions, such as when only minimal features are available (e.g., vehicle class, length, and width) or when training labels are partially corrupted. This creates a gap between academic experimental setups and the challenges encountered in real-world autonomous driving scenarios. Without systematically benchmarking how different models behave under constrained and noisy inputs, it's difficult to assess their readiness for real-time AV deployment or edge inference [1] [6].

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 in autonomous driving. Unlike prior work that often focuses on a single model type, our study investigates a wide spectrum of learning architectures—ranging from traditional RNN-based models such as GRUs and Bi-LSTMs to advanced Transformer variants like Informer and strategic hybrid architectures combining them. This comparative framework enables a deeper understanding of how each model class behaves under realistic driving conditions, across clean, noisy, rich, and minimal feature settings.

Theoretically, our work addresses a notable gap in the existing literature: the lack of hybridization in model design. We explore hybrid Transformer-RNN models that leverage the long-term dependency modeling capabilities of self-attention mechanisms and the sequential learning strengths of recurrent networks. These combinations not only improve robustness and generalization but also provide a middle ground between complexity and computational cost—an important aspect often overlooked in lane-change prediction research. Our results demonstrate that hybrid architectures outperform standalone models, particularly in scenarios involving noisy labels or weak input features, thus showcasing the benefit of cross-architecture synergy.

Practically, this research has direct implications for deploying autonomous systems in mixed-traffic environments. By benchmarking multiple architectures using real-world vehicle trajectory data from the NGSIM dataset and simulating [2] conditions such as label noise and limited sensor input, we offer a realistic evaluation pipeline that mirrors challenges encountered in on-road autonomous navigation. Furthermore, we report detailed computational overheads—including CPU and GPU memory usage, runtime performance, and training cost—to help guide practitioners in selecting models that are not only accurate but also resource-efficient and deployable in real-time AV systems.

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

This study specifically investigates whether Informer-based architectures and their hybrid extensions offer a performance advantage over standard Transformer and recurrent neural network (RNN) models for human lane-change prediction. The research is motivated by limitations in prior studies that often rely on a single class of models (either RNN-based or pure attention-based) without leveraging potential synergies between these paradigms.

Given the sequential nature of vehicle trajectory data and the long-range dependencies involved in lane-changing behavior, Informers are a compelling candidate due to their use of ProbSparse self-attention [7] and series distillation techniques, designed to handle long sequences more efficiently than traditional Transformers. However, their standalone use in trajectory prediction tasks remains underexplored, especially with dense, real-world driving datasets like NGSIM.

To bridge this gap, this work poses the following research question: To what extent do Informer models, and their hybrid combinations with RNNs (e.g., Informer-LSTM, Transformer-GRU), improve the prediction of human lane-change behavior compared to standalone Transformer or RNN models, both in terms of accuracy and computational efficiency?

This inquiry is addressed by designing a systematic evaluation pipeline where each model class is trained and tested under consistent conditions. The pipeline incorporates experiments across multiple data regimes, including full-feature vs. minimal-feature settings and clean vs. noisy labels. Performance is measured not only by prediction accuracy and classification metrics but also through quantitative profiling of training time, GPU/CPU memory consumption, and scalability characteristics.

In doing so, the study aims to establish whether hybrid architectures that fuse the strengths of both attention-based and recurrent components can better capture the temporal and contextual intricacies of human lane-change decisions, thereby contributing to the design of more robust and interpretable autonomous driving systems.

The remainder of this paper is organized into five main sections. Section I presents the Introduction, establishing the broader context of autonomous driving and highlighting the critical role of lane-change prediction in mixed-traffic environments. Section II delves into Related Work, offering an in-depth discussion of sixteen prior research studies focused on lane-change modeling. Each of these works is individually analyzed in terms of methodology, dataset, model architecture, and reported performance, followed by a tabular summary that synthesizes the key insights and limitations observed across the literature. Section III describes the Methodology employed in this study, including details of the data preprocessing pipeline, the structure of the NGSIM dataset, and comprehensive descriptions of the models implemented—ranging from baseline architectures like GRU and Transformer to hybrid configurations such as Informer-LSTM and Transformer-GRU. Following this, Section IV presents the Results and Evaluation, where the models are compared across multiple metrics including classification accuracy, robustness to weak features and noise, training time, and memory overhead on both CPU and GPU. Visualizations such as bar graphs and confusion matrices are used to illustrate the comparative performance and computational efficiency of each approach. Finally, Section V offers the Conclusion, summarizing the core findings of the study and discussing the broader implications for designing reliable lane-change prediction modules in autonomous vehicles. The paper concludes by outlining possible future research directions, especially in extending hybrid modeling frameworks and testing under more diverse traffic datasets.

II. Related Work

K. Gao et.al [8] proposed a dual Transformer-based model for jointly predicting lane change intentions and vehicle trajectories in mixed traffic, improving autonomous driving safety. It uses spatial-temporal attention to model interactions between human-driven and autonomous vehicles. The model includes separate Transformer modules for intention classification and trajectory prediction. Tested on NGSIM and highD datasets, it outperforms existing methods but may face challenges with noisy data and real-time processing.

Lu, Yuhuan et.al [9] proposed KLEP, a knowledge-driven lane change prediction system that combines real-world driving insights with neural networks to improve safety in the Internet of Vehicles. It models driver decisions using factors like speed, acceleration, and proximity, and uses a graph transformer to analyze structured data. KLEP achieved 6–7% higher accuracy and over 50% better early detection than nine baseline models. It also processes data faster and offers better interpretability. However, its limited focus on nearby vehicles and region-specific training data may affect generalizability.

Kunsong Shi et.al[10] proposed an LSTM-based Intrusion Detection System (IDS) that predicts the next CAN-bus ID from the past 20, using one-hot encoded inputs and SoftMax output. It detects anomalies like ID insertion, drops, and reordering using exact match and log-loss scoring. Trained on 36 million real-world messages with synthetic attacks and it achieved F1 scores of 0.90 for insertion, 0.84 for drop and 1.00 for illegal ID attacks. A key limitation is the unavailability of the real-world dataset used.

Mingxing Peng et.al [11] introduced LC-LLM, which reframes lane change prediction as a language task using Large Language Models(LLM) to output both decisions and natural language explanations with structured prompts from vehicle motion, traffic, and map data using Chain-of-Thought reasoning. Tested on the highD dataset, LC-LLM outperformed traditional models in both accuracy and interpretability. However, high computational cost and potential information loss during text conversion limit real-time use and also struggles with generalization and integration into existing vehicle systems.

Kequan Chen et.al[12] 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 distinguish crash vs. non-crash behavior using features like relative speed, distance, and acceleration. The model achieved high AUC scores (0.92–0.98) and outperformed traditional methods by up to 76%. However, its reliance on a small dataset and high-resolution data limits generalizability and real-world deployment. Further validation with diverse, large-scale data is needed for practical use.

A. N. Qasemabadi et.al [13] presented a novel deep learning model using a multi-layer Long Short-Term Memory (multi-LSTM) network to predict ego vehicle lane changes in Cooperative Adaptive Cruise Control (CACC) systems. The model processes temporal features from the ego vehicle's motion along with traffic data from surrounding vehicles. It compared performance with traditional Adaptive Cruise Control (ACC). Trained on the HighD dataset. However, it depends on high-quality sensor and communication data and lacks generalizability beyond the dataset. Real-time deployment may be limited by its computational demands.

Jongyong Do et.al [14] 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 Generalized Extreme Value (GEV) distributions based on features like speed, distance, and acceleration. The system estimates crash likelihood in real time, outperforming traditional methods with AUC scores of 0.92–0.98.

However, it simplifies complex behaviors, depends on high-quality data, and may struggle in urban or noisy environments. Its performance is optimized for structured highways and may face challenges in broader deployment.

Yunjie Huang et.al [15] 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 ensemble 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. Its effectiveness may drop in highly unpredictable traffic conditions.

Hongrui Zhang et.al [16] presented an enhanced Spiking Neural Network (SNN) for predicting lane-change intentions using time-series. Trained on HighD and NGSIM datasets, and achieved accuracies of 98.28% and 94.26%, outperforming Echo State Networks(ESNs) and Long Short-Term Memory(LSTM) networks in speed and efficiency. However, reliance on structured highway data and class imbalance limits generalization to complex urban scenarios. Like many neural models, the SNN lacks interpretability and real-world deployment validation.

Qingwen Xue et.al [17] proposed a dual-stage model that predicts both lane change intent and vehicle trajectory by incorporating traffic context like density and vehicle type. It combines XGBoost and LSTM for trajectory forecasting using HighD and NGSIM data, improving prediction accuracy to 98.20% and reducing trajectory errors. However, the model relies on high-quality labeled data. It struggles to handle unpredictable driver behaviors without additional retraining, raising concerns about scalability and robustness in dynamic real-world conditions.

Jinbao Zhang et.al [18] proposed an improved Lane Change Risk Assessment Index (LCRAI) for expressway weaving segments, capturing both frequency and severity of vehicle conflicts. Exposure-to-Risk (ERI) and Severity Risk Indices(SRI) are evaluated. Drone-based trajectory data and spatiotemporal checks help identify and localize high-risk zones. This model found the middle section of road is most dangerous, guiding targeted safety interventions. LCRAI is adaptable to various road types, offering practical value for traffic safety planning such as lane change guidance systems or structural modifications.

Hongyu Guo et.al [19] 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. Tested using Safety Pilot Model Deployment(SPMD) dataset, it achieved a PR-AUC above 0.98, outperforming traditional models. While effective, it depends on accurate sensor data and was validated mainly on highways, limiting urban applicability. It also lacks the integration of human behavior cues like turn signals or mirror checks,head movements which will reduce intent recognition depth.

Yuhuan Lu et.al [20] proposes a real-time incident detection system using 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 with minimal delay of around 0.25s and a 26.3% accuracy boost over existing models. It addresses latency, privacy, and data imbalance issues common in centralized systems. However, it relies on high-quality trajectory data, fixed edge-cloud fusion, and lacks real-world deployment. It also struggles with false positives and overlooks edge device limits and environmental factors like weather.

Wenjian Sun et.al [21] proposed a model which enhanced autonomous lane change safety by predicting nearby vehicle movements using an 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. A Risk Assessment Index evaluates maneuver safety. Simulation results showed superior accuracy, smoother paths, and safer decisions compared to baseline methods. However, the system relies on V2V communication and lacks real-world deployment validation.

Lin Li et.al [22] proposed a real-time lane-change intention model to help autonomous vehicles make safer decisions on highways using Recurrent Neural Networks(RNN)s to analyze driving data and predict whether a vehicle will change lanes or keep its lane. It improved autonomous responses in highway simulations. The model shows strong prediction accuracy but depends on continuous high-quality data.The model only predicts basic lane-change intentions. Additionally, it lacks handling of complex behaviors and doesn't explore impacts of prediction errors on safety.

Dongwei Xu [23] presented Multi-view Adaptive Hierarchical Spatial Graph Convolution Network(MVHGN), a trajectory prediction model for heterogeneous traffic agents (cars, bikes, pedestrians) to improve autonomous driving safety. It used spatial graph convolutions and GRU-based temporal predictor to model agent interactions.It captures both localized and group-level motion patterns effectively with adaptive clustering.Trained on the ApolloScape dataset. However, it struggled with pedestrian predictions and is sensitive to noisy data and computationally intensive. Still, it marks a key step toward more accurate, multi-agent trajectory forecasting.

A. *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

TABLE I
A SUMMARY OF LITERATURE REVIEW

Algorithm & Year of Publication	Objective	Methodology	Input Metrics for Making a Decision	Requirements / Assumptions	Achieved Enhancements	Simulator / ML Concept	Limitations
LCCDE: A Decision-Based Ensemble Framework for Intrusion Detection in the Internet of Vehicle (2022)	Improve attack detection in IoV using ensemble learning	Uses XGBoost, LightGBM, and CatBoost; leader model chosen per class; decision by confidence or best F1-score model	Features from CICIDS2017 & Car-Hacking dataset	Different models excel for different classes; labeled data is available	99.9997% (Car-Hacking) and 99.811% (CICIDS2017) F1-score	Ensemble of GBDTs (XGBoost, LightGBM, CatBoost)	Higher execution time due to model complexity
Intrusion Detection Method for Internet of Vehicles Based on Parallel Analysis of Spatio-Temporal Features (2023)	Enhance intrusion detection using parallel spatio-temporal feature analysis	Uses TCN and LSTM in parallel with self-attention and MLP classifier	Selected network traffic features after correlation filtering	Spatio-temporal features are important; parallel learning is better	98.68% accuracy (NSL-KDD), 96.34% (UNSW-NB15)	TCN + LSTM + MLP with attention	High memory usage; drop in performance for rare attacks
Enhancing Cyber Security in Autonomous Vehicles: A Hybrid XGBoost-Deep Learning Approach for Intrusion Detection in the CAN Bus (2024)	Detect CAN-bus attacks using hybrid feature selection and deep learning	XGBoost for feature selection + Deep Neural Network for classification	Cleaned and labeled CAN message data (e.g., flooding, spoofing)	Real-world CAN data available; labeled samples sufficient	99.90% accuracy, high precision and recall	TensorFlow DNN + XGBoost	High data and compute demands; less robust to unknown vehicles
IIDS: Intelligent Intrusion Detection System for Sustainable Development in Autonomous Vehicles (2023)	Detect cyber-attacks in AVs using images and CNNs over 5G-V2X	Converts network data to images, applies CNN, isolates vehicles over 5G network	CAN and external network features converted to images	Traffic patterns can be visualized; assumes 5G-V2X availability	98% accuracy; low latency with V2X alerts	CNN + SUMO + NS2	Complex image preprocessing; struggles with zero-day attacks
Transforming Transportation: Safe and Secure Vehicular Communication and Anomaly Detection with Intelligent Cyber–Physical System and Deep Learning (2024)	Improve vehicle anomaly detection using hybrid CNN + LSTM	Bidirectional GRU + LSTM + CNN → Dense layers + Softmax classifier	Vehicle sensor data + temporal patterns	Hybrid models are more accurate; deep learning generalizes patterns	100% (car-hacking), 99% (UNSW-NB15) accuracy	CNN + LSTM + GRU using Keras	Long training times; hard for real-time use
Tree-Based Intelligent Intrusion Detection System in Internet of Vehicles (2019)	Use tree models for real-time IoV attack detection	Stacked Decision Tree, RF, Extra Trees, XGBoost with feature selection	CAN-intrusion & CICIDS2017 dataset features; top 90% selected	Tree models can run fast; SMOTE helps with imbalance	100% (CAN), 99.86% (CICIDS2017)	Tree ensemble with SMOTE & stacking	Training time rises without feature pruning
MTH-IDS: A Multitiered Hybrid Intrusion Detection System for Internet of Vehicles (2022)	Detect known and unknown IoV attacks in real-time	k-means + feature filters + stacked trees + biased classifiers	Balanced and reduced features from CAN + CICIDS2017 datasets	Hybrid multi-tier model handles zero-day threats; low-resource friendly	99.99% (CAN), 99.88% (CICIDS2017); fast: <0.6 ms/packet	Tree + clustering + Raspberry Pi deployment	Weak on fuzzy/XSS; needs better adaptation
Federated Learning-Assisted Distributed Intrusion Detection Using Mesh Satellite Nets for Autonomous Vehicle Protection (2024)	Use federated learning over satellites for AV IDS	Each satellite trains DNN locally, shares model updates asynchronously	Intrusion data from NSL-KDD, TON-IoT, X-IIoTID; no raw data	Assumes satellite nodes are secure; no central server	99.99% accuracy; works with non-IID data	DNN + Federated learning + PyTorch	Trust assumption in satellite nodes; needs model optimization
Detecting In-vehicle Intrusion via semi-supervised Learning-based Convolutional Adversarial Autoencoders (2022)	Semi-supervised intrusion detection	CAE + GAN to learn from 29x29 CAN ID images with limited labels	CAN bus dataset with unsupervised manifold structure	CAN data has enough patterns for efficient attack learning	99% F1 score, real-time feasible	GAN + CAE on Car-Hacking dataset	Needs at least 40% labeled data
A context-aware on -board intrusion detection system for smart vehicles (2024)	Context-aware IDS with minimal delay	Random Forest with gain-ratio feature selection	Speed, brake, position, yaw, distance, fuel, etc.	MetaDrive simulates near real-world driving data	99.9% recall; 29s latency	MetaDrive + Random Forest	Data collection method shared, but not the actual data
ID Sequence Analysis for Intrusion Detection in the CAN bus using Long Short Term Memory Networks (2020)	Sequence analysis using LSTM	LSTM + log loss on ID sequences; 20×42 matrices from CAN IDs	Encoded arbitration ID sequences from raw CAN-bus traffic	Patterns observable in ID sequences; some delays not included in training	0.9–1.0 F1 scores, real-time performance	LSTM; real data from first 100 hrs of driving	Real-world data not shared; collection is challenging
CANShield: Deep-Learning-Based Intrusion Detection Framework for Controller Area Networks at the Signal Level (2023)	Design an IDS at the signal level using CNN autoencoders	Multiple CNN-AEs + structured loss + ensemble detection + transfer learning	Signal value time series, inter-arrival time (IAT) series	Assumes access to DBC or reverse tools; trained on diverse, dynamic data	99.95% accuracy, sub-10ms detection	CNN-AEs + dual LSTM + transfer learning	Needs OEM DBC or high-fidelity dynamic data
Simulating Malicious Attacks on VANETs for Connected and Autonomous Vehicle Cybersecurity: A Machine Learning Dataset (2022)	Simulate VANET cyberattacks	Uses SUMO + OMNeT++ + Veins to simulate VANET attacks	Message type, GPS, signal strength for each event	Focus on dataset generation for ML benchmarking	Provides dataset for ML algorithm testing	SUMO, OMNeT++, Veins	Does not propose an IDS; focuses on dataset creation
Intrusion Detection System for Autonomous Vehicles Using Non-Tree Based Machine Learning Algorithm (2024)	Test KNN/SVM for CAN IDS	SVM, KNN, Logistic Regression on CAN dataset	Preprocessed CAN payloads	Evaluated on standard CAN datasets	98%+ accuracy with SVM	KNN, SVM, Logistic Regression	Underperforms in evolving complex attacks
X-CANIDS: Signal-Aware Explainable Intrusion Detection System for Controller Area Network -Based In-Vehicle Network (2024)	Explainable signal-aware IDS	1D CNN with SHAP for signal-level interpretation	Signal segments with SHAP values	Requires DBC file and labeled signal data	97%+ accuracy with explainability	1D CNN + SHAP, signal datasets	Needs signal-level labeled data; DBC extraction required
Intrusion Detection System Using Deep Neural Network for In-Vehicle Network Security (2016)	Deep learning for CAN anomaly detection	Deep MLP with dropout on labeled CAN vectors	Numeric CAN message features	Normal & attack labeled data available in public dataset	99% accuracy, low false alarms	DNN, public CAN dataset	No timestamps; poor detection of time-based attacks

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. To address these challenges,this methodology proposes

- **Spatial Temporal Interactions:** Multiple models, ranging from classical recurrent neural networks to attention-enhanced architectures applied to capture interactions
- **Long Sequence Forecasting:** Informer and transformer based architecture designed to strengthen temporal sequence modelling
- **Next Generation Simulation datasets:** Utilized the publicly available 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.

III. PROPOSED HYBRID FORECASTING SYSTEMS

This study conducts a comprehensive analysis of multiple models as shown in Figure, ranging from classical recurrent neural networks such as Gated Recurrent Units (GRUs) and Bidirectional Long Short-Term Memory (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 Neural 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 modeling 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.

A. Dataset and Preprocessing

In this work, the Next Generation Simulation(NGSIM) dataset [ref] 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 detailed 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 1, the preprocessing 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.

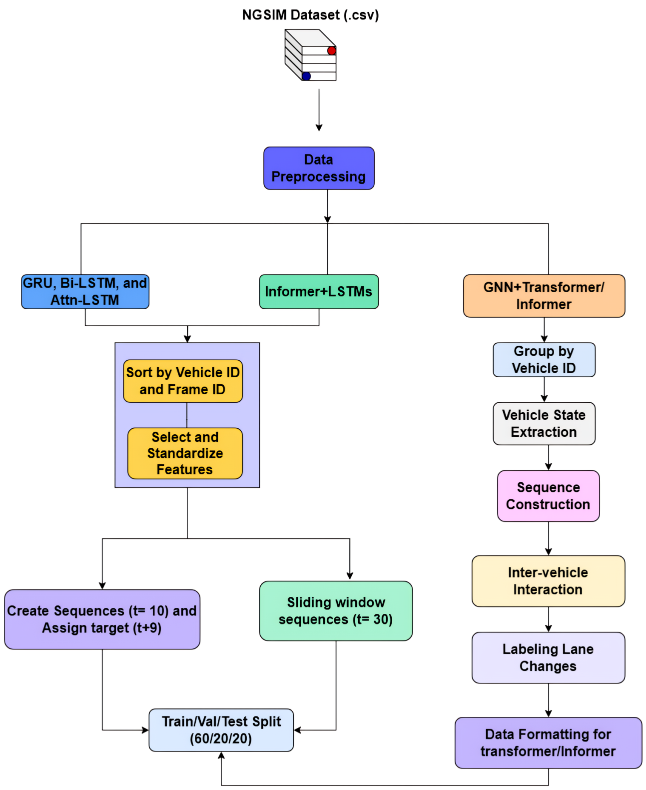


Fig. 1. Data Preprocessing of various models.

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 understand 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 deceleration, 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 descent 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 keeping, 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 2, 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 features, our feature set is designed to capture both local motion patterns and the contextual environment. This enables our models to learn complex temporal dependencies and behavioral cues necessary for predicting lane change maneuvers effectively.

B. 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.

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.

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 beneficial for lane change prediction, where driver behavior is influenced not only by past motion but also by upcoming interactions and spatial constraints.

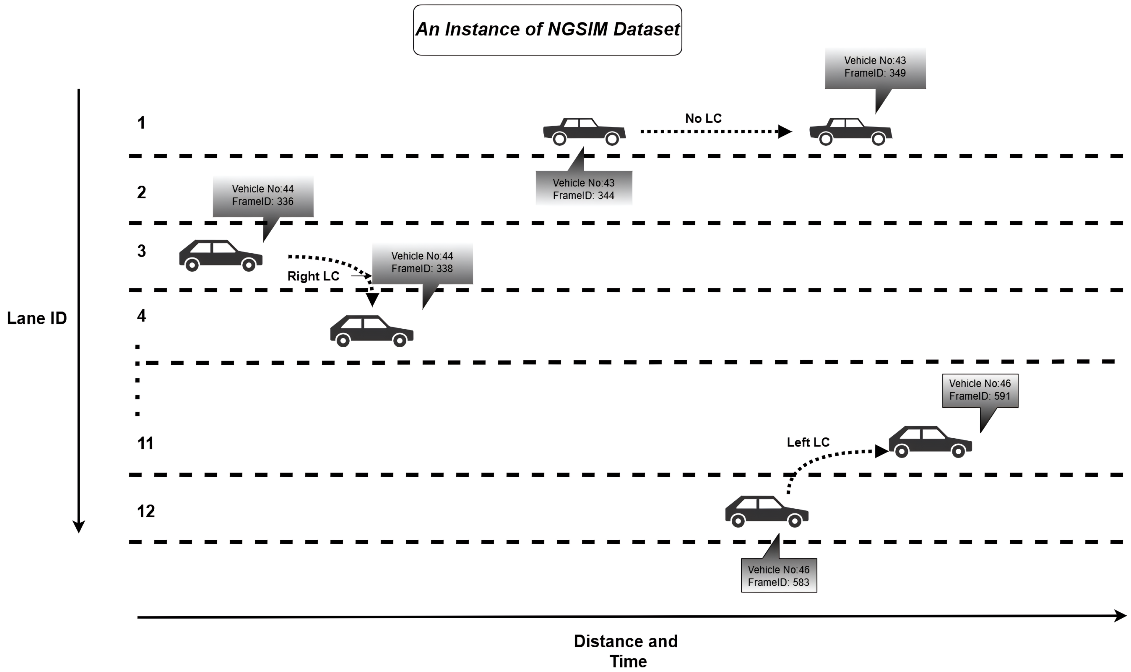


Fig. 2. NGSIM Dataset

The Bi-LSTM architecture consists of a single LSTM layer with 64 hidden units in each direction, followed by a fully connected layer that outputs class probabilities at each timestep. The model is trained using padded sequences 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 bidirectional temporal information for more accurate predictions compared to unidirectional recurrent models like GRUs.

C. 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 selectively 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 sliding 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 mechanism significantly improved the model's ability to distinguish subtle temporal patterns, leading to better generalization and more robust predictions, particularly in edge cases where standard LSTM models tend to underperform.

D. Transformer Model

This model predicts a vehicle's future lane change behavior by combining a Transformer architecture with a pretrained GNN(PCurvenet). A multi-class classification problem—left lane change, right lane change, or lane keeping—is used to model the task. The input is a temporal history of states for every vehicle that has been observed, organized as follows:

$$\mathbf{V} \in R^{N \times C \times T}$$

where T is the number of time steps, C is the number of features(C=21), and N is the number of vehicles. According to this model,

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

Explain this equation and the following equations also

The pretrained PCurveNet extracts high-level interaction features:

$$\mathbf{Z}_T^{\text{ego}} = g_\theta(\mathbf{V})$$

These features are combined with sinusoidal positional encodings:

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

explain this equation, mention what each variable means

and passed into a Transformer encoder with multi-head attention:

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

explain this equation, mention what each variable means

The decoder processes the ego vehicle's current state and outputs the predicted logits:

$$\hat{y}^{\text{ego}}_{T+1} = \text{softmax}(\mathbf{h}^{\text{ego}}_{T+1})$$

explain this equation, mention what each variable means

To handle class imbalance, the model is trained using the focal loss:

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

explain this equation, mention what each variable means

Performance is evaluated using standard metrics such as precision and recall.

E. Informer based Hybrid Models

This study also investigates Informer's effectiveness in long-sequence forecasting in hybrid configurations that include Informer combined with GNN, Transformer, and LSTM. Each combination is designed to facilitate complementary advantages in temporal sequence modeling and interaction reasoning.

1) Informer + GNN : This model maintains high accuracy while increasing computational efficiency by replacing the Transformer with the **Informer architecture**, which is optimized for long-term sequence modeling using *ProbSparse attention*. This model is trained and evaluated with the NGSIM dataset, just like the prior method.

The input trajectory tensor is defined as:

$$\mathbf{V} \in R^{N\times C\times T}$$

where N is the number of vehicles, T is the number of time steps, and $C = 21$ is the number of motion features per vehicle (e.g., position, velocity, acceleration, etc.).

The prediction task is defined as:

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

explain this equation, mention what each variable means

The interaction context is extracted using a pretrained PCurveNet module:

$$\mathbf{Z}^{\text{ego}}_T = g_\theta(\mathbf{V})$$

explain this equation, mention what each variable means

These features are then improved using both learnable temporal positional encodings and feature-type embeddings after passing through multi-scale temporal convolutions with kernel sizes 3, 5, and 7. The Informer encoder, which makes use of ProbSparse attention, processes the improved sequence:

$$\text{Attention}(Q, K, V) \approx \text{Top}_k\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

explain this equation, mention what each variable means

The current state of the ego vehicle is fused with the context of the global interaction using a **cross-attention mechanism**. **attention pooling** is then used to aggregate the output sequence, producing a global scene representation.

The class probabilities are obtained by:

$$\hat{y}^{\text{ego}}_{T+1} = \text{softmax}(\mathbf{h}^{\text{ego}}_{T+1})$$

explain this equation, mention what each variable means

In parallel, an **uncertainty head** produces per-class confidence values:

$$\hat{u}_j = \text{Softplus}(W\mathbf{h} + b)$$

explain this equation, mention what each variable means

The model is trained using the **focal loss** function to address class imbalance and emphasize harder samples:

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

explain this equation, mention what each variable means

This Informer + GNN combination offers accurate and efficient lane change prediction, with uncertainty estimates that enhance interpretability in real-world autonomous driving scenarios.

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 4 (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, we use a minimalistic decoder consisting 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 (left, keep, right).

This architecture is chosen for its ability to model both local and global interactions among trajectory features over time. Its non-recurrent structure allows for better parallelism and faster training compared to traditional RNNs, while the self-attention mechanism provides interpretability and robustness in modeling sequential behaviors critical for understanding lane changes.

3) Informer + LSTM: This proposed model for lane change prediction is a hybrid architecture combining Informer-based transformer encoding with an LSTM-based decoding mechanism. The raw NGSIM dataset, provided in CSV format, is first preprocessed by sorting the vehicle trajectory data by `Vehicle_ID` and `Frame_ID` to ensure temporal consistency. The relevant numerical features such as local and global positions, velocity, acceleration, lane ID, and headway

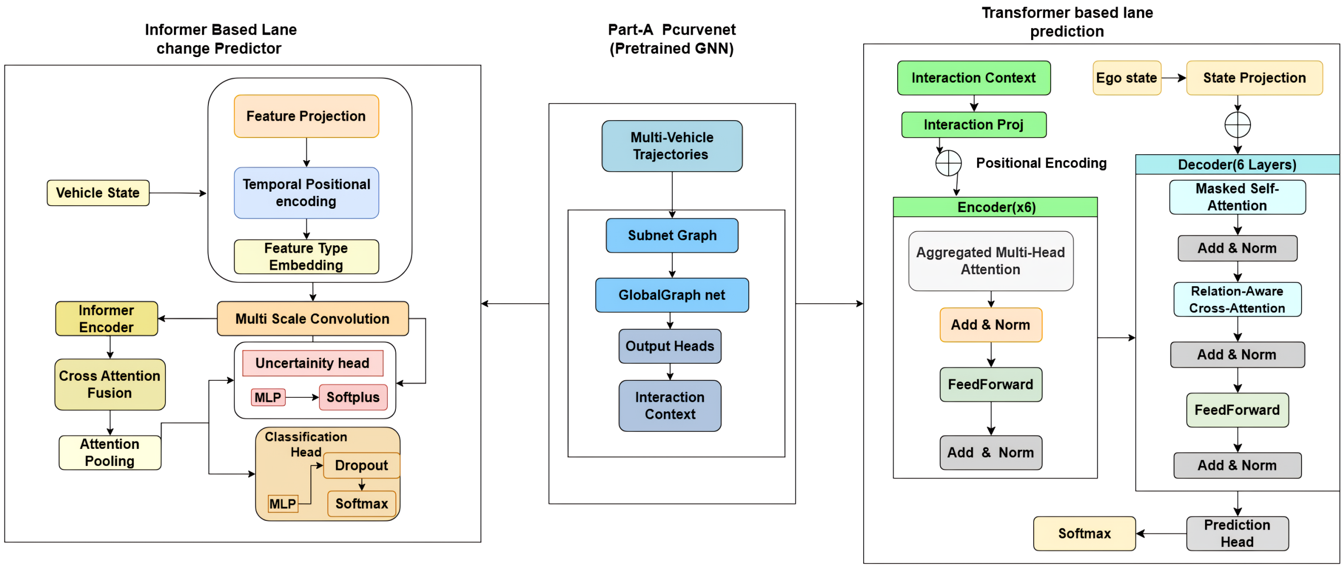


Fig. 3. Proposed architecture combining GNN, Transformer, and Informer modules for lane change prediction Where to cite this in th content

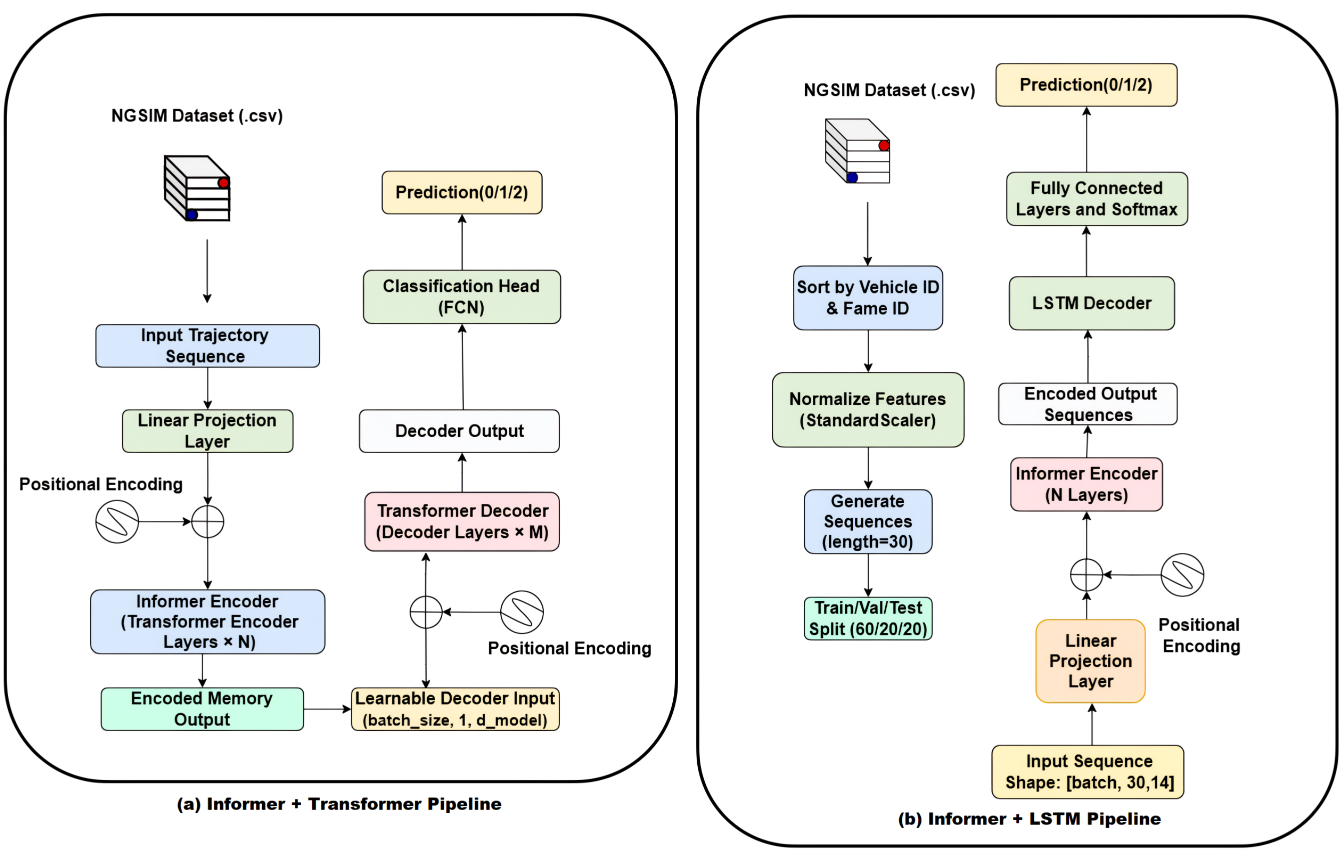


Fig. 4. Informer based Hybrid Models

measures are normalized using standard score normalization via `StandardScaler` to stabilize training.

Following preprocessing, 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 maneuver: 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 encoding 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 captures 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 hidden 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 validation set.

IV.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 assessed using standard classification metrics such as Accuracy, Precision, Recall, 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 illustrated through bar charts for visual clarity. A schematic diagram of a sample input sequence from the NGSIM dataset is included to provide context for the model input structure. 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 experimentation 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 supporting 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 computational 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 reinforced the practical applicability of the models.

4.2 Recurrent Models Analysis

Table II presents a comparative analysis of the GRU, Bi-LSTM, and Attention-based LSTM (Attn-LSTM) models under varying hyperparameter 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.

booktabs multirow

TABLE II
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 Informer's Hybrids Model Analysis

To assess the effectiveness of the proposed Informer-based models for lane change prediction, we present a detailed comparison of two architectures: **Informer+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. The table below demonstrates how tuning these parameters influences the prediction quality, with Informer+LSTM achieving higher scores in multiple setups. This comparison highlights the potential of hybrid Informer-based architectures in modeling temporal dependencies and spatial interactions effectively.

TABLE III

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 Transformer and Informer GNN variants

To evaluate the effectiveness of integrating graph-based spatial interactions with sequence modeling, we conducted experiments using two hybrid architectures: Transformer + GNN and Informer + GNN. These models were trained with varying learning rates, batch sizes, and focal loss parameters to explore their impact on lane change prediction performance. The table below summarizes the classification metrics—Accuracy, Precision, Recall, and F1 Score—achieved under different configurations. The best-performing setup is highlighted in bold for each metric.

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. The table below summarizes the classification accuracies of these best-performing models across three behavior classes—Left Lane Change, Lane Keep, and Right Lane Change—along with the overall accuracy. The highest accuracy in each category is highlighted in bold.

TABLE IV

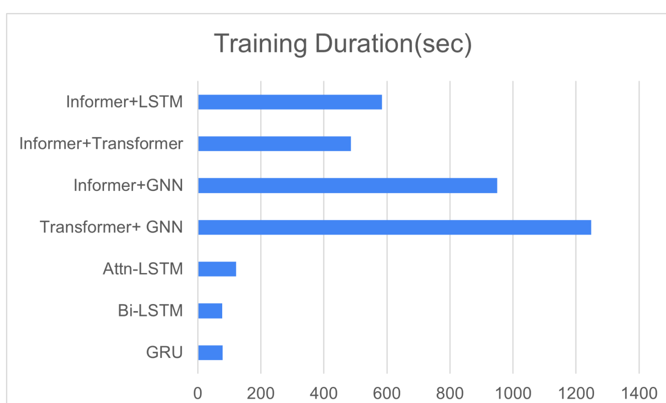
Model	Hyper-parameter	Accuracy	Precision	Recall	F1 Score
Transformer + GNN	learning rate = 1×10^{-4}, batch size = 32, focal alpha = 0.25, focal gamma = 2.0	0.9266	0.9310	0.9201	0.9181
	learning rate = 1×10^{-4}, batch size = 64, focal alpha = 0.25, focal gamma = 1.5	0.9008	0.8954	0.8986	0.8882
Informer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

TABLE V
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

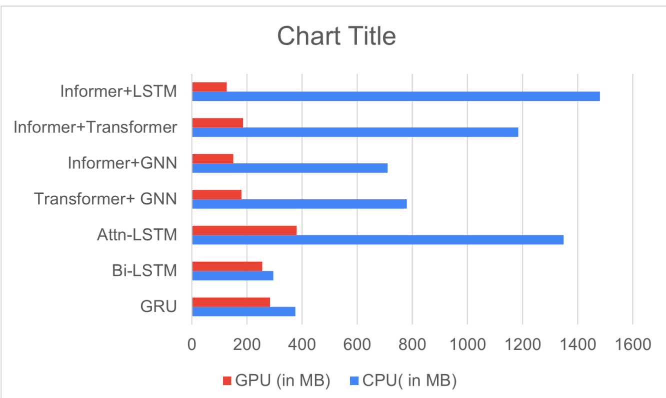
To better understand the computational efficiency of each architecture, we recorded the total training duration for all models over a fixed number of epochs. This comparison provides insights into how model complexity, architectural design (such as recurrent layers, attention mechanisms, and hybrid combinations like *Informer+LSTM* or *Transformer+GNN*), and parameter count influence the overall training time. The bar graph below highlights the time each model took to complete training, enabling an objective assessment of the trade-off between performance and computational cost.



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 computational demands. This includes both GPU and CPU memory utilization during training. By analyzing the memory overhead, we gain a deeper understanding of how each model scales in terms of hardware resource requirements.

The memory usage was recorded at the end of the training phase for each model, capturing both GPU memory allocation (for CUDA-enabled models) and CPU RAM usage. The resulting bar graph illustrates the comparative memory footprint of each architecture, helping to identify models that offer an optimal trade-off between performance, training speed, and memory efficiency.



V. Conclusion

References

[1] H. Ali, J. Zheng, R. S. Alrashidi, M. S. Alfawzan, I. Ullah, and A. Jamal, "Simulating the adoption of autonomous vehicles in cpec logistics and transportation: A case study of a scenario approach," *IEEE Access*, pp. 1–1, 2025.

[2] D. Hu, C. Huang, J. Zhao, Y. Zhao, and J. Wu, "Autonomous driving economic car-following motion strategy based on adaptive rollout model-based policy optimization," *IEEE Transactions on Transportation Electrification*, pp. 1–1, 2025.

[3] Z. LI, X. HUANG, J. WANG, and T. MU, "Lane change behavior research based on ngsim vehicle trajectory data," in *2020 Chinese Control And Decision Conference (CCDC)*, 2020, pp. 1865–1870.

[4] M. Amaris, M. A. Morais, and R. Y. De Camargo, "Efficient prediction of region-wide traffic states in public bus networks using lstms," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021, pp. 2215–2220.

[5] N. Abid, T. Ouni, and M. Abid, "Vehicle detection for intelligent traffic surveillance system," in *2020 5th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, 2020, pp. 1–5.

[6] G. S. P. K. Segu, A. D. S. N. Sivannarayana, and S. Ramesh, "Real time road lane detection and vehicle detection on yolov8 with interactive deployment," in *2024 IEEE 16th International Conference on Computational Intelligence and Communication Networks (CICN)*, 2024, pp. 267–272.

[7] J. Lu, D. Wang, and Z. Wang, "Stpsformer: Spatial-temporal probsparse transformer for long-term traffic flow forecasting," in *2024 International Joint Conference on Neural Networks (IJCNN)*, 2024, pp. 1–8.

[8] K. Gao, X. Li, B. Chen, L. Hu, J. Liu, R. Du, and Y. Li, "Dual transformer based prediction for lane change intentions and trajectories in mixed traffic environment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 6, pp. 6203–6216, 2023.

[9] Y. Lu, Z. Zhang, W. Wang, Y. Zhu, T. Chen, Y. D. Al-Otaibi, A. K. Bashir, and X. Hu, "Knowledge-driven lane change prediction for secure and reliable internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2025.

[10] K. Shi, Y. Wu, H. Shi, Y. Zhou, and B. Ran, "An integrated car-following and lane changing vehicle trajectory prediction algorithm based on a deep neural network," *Physica A: Statistical Mechanics and its Applications*, vol. 599, p. 127303, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378437122002503

[11] M. Peng, X. Guo, X. Chen, K. Chen, M. Zhu, L. Chen, and F.-Y. Wang, "Lc-llm: Explainable lane-change intention and trajectory predictions with large language models," *Communications in Transportation Research*, vol. 5, p. 100170, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2772424725000101

[12] K. Chen, Z. Li, P. Liu, C. Xu, and Y. Wang, "Real-time lane-changing crash prediction model at the individual vehicle level using real-world trajectories prior to crashes," 2024. [Online]. Available: http://dx.doi.org/10.2139/ssrn.4829767

[13] A. N. Qasemabadi, S. Mozaffari, M. Rezaei, M. Ahmadi, and S. Alirezaee, "A novel model for driver lane change prediction in cooperative adaptive cruise control systems," in *2023 International Symposium on Signals, Circuits and Systems (ISSCS)*, 2023, pp. 1–4.

[14] J. Do, K. Han, and S. B. Choi, "Lane change–intention inference and trajectory prediction of surrounding vehicles on highways," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 7, pp. 3813–3825, 2023.

[15] Y. Huang, X. Song, Y. Zhu, S. Zhang, and J. J. Q. Yu, "Traffic prediction with transfer learning: A mutual information-based approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 8, pp. 8236–8252, 2023.

[16] H. Zhang, Y. Wang, S. Zhang, J. Li, Q. Wang, and B. Zhou, "Improved time series models for the prediction of lane-change intention," *Transportation Letters*, vol. 17, no. 4, pp. 747–761, 2025. [Online]. Available: https://doi.org/10.1080/19427867.2024.2379702

[17] Q. Xue, Y. Xing, and J. Lu, "An integrated lane change prediction model incorporating traffic context based on trajectory data," *Transportation Research Part C: Emerging Technologies*, vol. 141, p. 103738, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X22001723

[18] J. Zhang, J. Lee, M. Abdel-Aty, O. Zheng, and G. Xiao, "Enhanced index of risk assessment of lane change on expressway weaving segments: A case study of an expressway in china," *Accident Analysis Prevention*, vol. 180, p. 106909, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S000145752200344X

[19] H. Guo, M. Keyvan-Ekbatani, and K. Xie, "Lane change detection and prediction using real-world connected vehicle data," *Transportation Research Part C: Emerging Technologies*, vol. 142, p. 103785, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X22002133

[20] H. C. J.-Y. C. Yuhuan Lu, Qinghai Lin, "Automatic incident detection using edge-cloud collaboration based deep learning scheme for intelligent transportation systems," *Applied Intelligence*, vol. 53, no. 1573-7497, pp. 24 864 – 24 875, 2023.

[21] W. Sun, L. Pan, J. Xu, W. Wan, and Y. Wang, "Automatic driving lane change safety prediction model based on lstm," 2024. [Online]. Available: https://arxiv.org/abs/2403.06993

[22] L. Li, W. Zhao, C. Xu, C. Wang, Q. Chen, and S. Dai, "Lane-change intention inference based on rnn for autonomous driving on highways," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 5499–5510, 2021.

[23] D. Xu, X. Shang, H. Peng, and H. Li, "Mvhgn: Multi-view adaptive hierarchical spatial graph convolution network based trajectory prediction for heterogeneous traffic-agents," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 6, pp. 6217–6226, 2023.

Michael Roberg (S'09) received the B.S.E.E degree from Bucknell University, Lewisburg, PA, in 2003, the M.S.E.E. degree from the University of Pennsylvania, Philadelphia, in 2006, and the Ph.D. degree from the University of Colorado at Boulder in 2012. From 2003 to 2009, he was an Engineer with Lockheed Martin–MS2, Moorestown, NJ, where he was involved with advanced phased-array radar systems. His current research interests include high efficiency microwave PA theory and design, microwave power rectifiers, MMIC design, and high-efficiency radar and communication system transmitters. He is currently employed by TriQuint Semiconductor - Defense Products and Foundry Services in Richardson, TX working on wideband high efficiency GaN MMIC PA design.

Tibault Reveyrand (M'07) received the Ph.D. degree from the University of Limoges, France, in 2002. From 2002 to 2004, he was a Post-Doctoral Scientist with CNES (French Space Agency). In 2005, he became a CNRS engineer at XLIM. His research interests include the characterization and modeling of RF and microwave nonlinear components and devices. Dr. Reveyrand was the recipient of the 2002 European GaAs Best Paper Award and is a member of the IEEE MTT-11 ”Microwave Measurements” Technical Committee.

Ignacio Ramos (S'12) received the B.S. degree in electrical engineering from the University of Illinois at Chicago in 2009, and is currently working toward the Ph.D. degree at the University of Colorado at Boulder. From 2009 to 2011, he was with the Power and Electronic Systems Department at Raytheon IDS, Sudbury, MA. His research interests include high-efficiency microwave power amplifiers, microwave DC/DC converters, radar systems, and wireless power transmission.

Erez Avigdor Falkenstein (S'07), Haifa, Israel in 1979. He earned a "Handesaie" degree (associate degree) in electronics from Amal Handesaim School Hadera, Israel in 1999. From 1999 to 2003 he served in the Israel Defense Force as part of a technological unit. He has been at the University of Colorado at Boulder 2004 – 2012. He received concurrent MS/BS degrees in Electrical engineering 2010 and a Ph.D 2012 from the University of Colorado at Boulder. Since 2007 he has been involved with research as part of the active antenna group. Research emphasis: far field wireless powering for low power densities. Interests include Antenna design and characterization, modeling and measurement of nonlinear devices at microwave frequencies and power management. He is currently employed at Qualcomm, Incorporated, Boulder, CO.

Zoya Popović (S'86-M'90-SM'99-F'02) received the Dipl.Ing. degree from the University of Belgrade, Belgrade, Serbia, Yugoslavia, in 1985, and the Ph.D. degree from the California Institute of Technology, Pasadena, in 1990. Since 1990, she has been with the University of Colorado at Boulder, where she is currently a Distinguished Professor and holds the Hudson Moore Jr. Chair with the Department of Electrical, Computer and Energy Engineering. In 2001, she was a Visiting Professor with the Technical University of Munich, Munich, Germany. Since 1991, she has graduated 44 Ph.D. students. Her research interests include high-efficiency, low-noise, and broadband microwave and millimeter-wave circuits, quasi-optical millimeter-wave techniques, active antenna arrays, and wireless powering for batteryless sensors. Prof. Popović was the recipient of the 1993 and 2006 Microwave Prizes presented by the IEEE Microwave Theory and Techniques Society (IEEE MTT-S) for the best journal papers and the 1996 URSI Issac Koga Gold Medal. In 1997, Eta Kappa Nu students chose her as a Professor of the Year. She was the recipient of a 2000 Humboldt Research Award for Senior U.S. Scientists of the German Alexander von Humboldt Stiftung. She was elected a Foreign Member of the Serbian Academy of Sciences and Arts in 2006. She was also the recipient of the 2001 Hewlett-Packard (HP)/American Society for Engineering Education (ASEE) Terman Medal for combined teaching and research excellence.