Security and privacy of communication protocols for collaborative and fully distributed Vehicular Ad-hoc Networks (VANETs)

Tafsir Moussa Sakho

université PARIS-SACLAY

Security and privacy of communication protocols for collaborative and fully distributed Vehicular Ad-hoc Networks (VANETs)

Sécurité et confidentialité des protocoles de communication dans les réseaux véhiculaires ad-hoc collaboratifs et entièrement distribués

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 580, Sciences et Technologies de l'Information et de la Communication (STIC)
Spécialité de doctorat: Sciences des Réseaux, de l'information et de la Communication
Graduate School : Sciences de l'Ingénierie et des Systèmes.
Référent : Faculté des sciences d'Orsay

Thèse préparée dans l'unité de recherche **Laboratoire des Signaux et Systèmes** (Université Paris-Saclay, CNRS, CentraleSupélec) sous la direction de **Jalel BEN OTHMAN**, Professeur

Thèse soutenue à Paris-Saclay, le 12 mai 2026, par

Tafsir Moussa SAKHO

Composition du jury

Membres du jury avec voix délibérative

Marco DI RENZO Directeur de recherche, CNRS/CentraleSupélec - Université Paris-Saclay, France	Président
Pascal LORENZ Professeur, Université de Haute-Alsace, France	Rapporteur & Examinateur
Selma BOUMERDASSI Professeure, Université Paris 8, France	Rapporteure & Examinatrice
Boubaker DAACHI Professeur, Université Paris 8, France	Examinateur
Jaafar GABER Professeur, Université de Technologie de Belfort Montbéliard, France	Examinateur
Hacène FOUCHAL Professeur, Université de Reims Champagne-Ardenne, France	Examinateur

THESE DE DOCTORAT
NNT : 2026UPAST048

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre: Sécurité et confidentialité des protocoles de communication dans les réseaux véhiculaires ad-hoc collaboratifs et entièrement distribués

Mots clés: Apprentissage automatique fédéré décentralisé, VANET, prédiction de trajectoires, algorithmes de communication, Sécurité et confidentialité, détection d'anomalies

Résumé: Les systèmes de transport intelligents (ITS), composantes essentielles des villes intelligentes, reposent sur l'intégration de technologies avancées visant à améliorer l'efficacité, la sécurité, la durabilité et le confort du transport. L'émergence des véhicules autonomes et connectés a profondément transformé ce paysage, en introduisant des capacités de perception, de décision autonome et de communication en temps réel. Au sein des réseaux véhiculaires de type VANET (Vehicular Ad-hoc Networks), les véhicules peuvent communiquer entre eux, interagir avec l'infrastructure routière, ou avec tout autre objet connecté du système ITS, formant ainsi un environnement distribué, dynamique et hautement interopérable.

Cependant, cette connectivité accrue expose les véhicules à un large éventail de vulnérabilités. La sécurité des protocoles de communication devient alors un enjeu majeur, notamment face à des menaces telles que l'usurpation d'identité, les attaques Sybil, les attaques par déni de service ou encore les attaques par empoisonnement.

Ainsi, l'objectif de cette thèse est d'analyser ces menaces spécifiques et de proposer des mécanismes de sécurisation adaptés aux contraintes propres aux véhicules connectés : forte mobilité, latence faible, ressources limitées et nature collaborative des échanges. Toutefois, comprendre, savoir imiter et anticiper les mouvements des véhicules est essentiel pour mieux appréhender les défis liés à la sécurité et à la préservation de la vie privée dans ce type d'environnement. Pour ces raisons, nous avons d'abord étudié les modèles de mobilité, puis conçu un framework de simulation collaboratif reposant sur l'apprentissage automatique fédéré. Un modèle de prédiction de trajectoire, basé sur des réseaux de neurones artificiels de type Transformer, a été intégré à ce framework. Par la suite, nous avons développé une approche de prédiction de mobilité entièrement décentralisée, combinant une architecture personnalisée du Transformer, pour la prédiction de trajectoires, et un algorithme de communication peer-to-peer de type gossip. Les véhicules échangent uniquement les paramètres de leurs modèles locaux, préservant ainsi la confidentialité des données brutes tout en favorisant l'apprentissage collaboratif. Une solution de sécurité et de préservation de la vie privée a été intégrée à ce système. Cette solution utilise un système de détection et de riposte contre les anomalies reposant sur des vérifications de cohérence et de plausibilité basées sur des erreurs d'innovation calculées par un filtre de Kalman étendu. Ce système permet de détecter les comportements malveillants et d'exclure les véhicules suspects en temps réel. Il est renforcé par un mécanisme léger de chiffrement et de signature unifiée, garantissant l'authenticité, la confidentialité et l'intégrité des messages échangés. L'approche de sécurité proposée a été validée expérimentalement à travers la simulation de plusieurs types d'attaques, notamment des attaques par empoisonnement, des attaques Sybil et des attaques DoS. Les résultats obtenus démontrent une efficacité très élevée en matière de détection d'anomalies et de résilience, tout en ne nécessitant qu'un coût réduit en termes de calcul et de communication, la rendant ainsi compatible avec les contraintes des systèmes VANETs embarqués.

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Title: Security and privacy of communication protocols for collaborative and fully distributed Vehicular Ad-hoc Networks (VANETs)

Keywords: VANET, decentralized federated learning (DFL), trajectory prediction, communication algorithms, security and privacy, anomaly detection

Abstract:

Intelligent Transportation Systems (ITS), key components of smart cities, rely on integrating advanced technologies designed to enhance transportation systems' efficiency, safety, sustainability, and comfort. The emergence of autonomous and connected vehicles has profoundly reshaped this landscape by introducing real-time perception, autonomous decision-making, and communication capabilities. Within Vehicular Ad-hoc Networks (VANETs), vehicles can communicate with each other, interact with road infrastructure, and exchange data with other thing connected to the ITS system, thus forming a distributed, dynamic, and highly interoperable environment. However, this increased connectivity also exposes vehicles to a wide range of vulnerabilities. Securing communication protocols has therefore become a critical issue, particularly against threats such as identity spoofing, Sybil attacks, denial-of-service (DoS) attacks, and data poisoning. This thesis aims to analyze these specific threats and propose security mechanisms tailored to the constraints inherent to connected vehicles: high mobility, low latency, limited computational resources, and a collaborative communication model. To better address the safety and privacy challenges in this environment, it is crucial to understand, mimic, and anticipate vehicle movements. To this end, we first studied mobility models and then developed a collaborative simulation environment based using federated learning (FL) techniques. A trajectory prediction model, using long short-term memory a customized Transformer architecture, was integrated into this framework. Subsequently, we designed a fully decentralized mobility model that combines the customized Transformer-based trajectory prediction architecture with a gossip-type communication algorithm. In the proposed approach, vehicles exchange only the parameters of their local neural network models, thereby preserving the privacy of raw data while enabling collaborative learning. A lightweight security solution was integrated into this system, combining encryption and digital signature in a unified step to ensure message authenticity, confidentiality, and integrity. In parallel, an anomaly detection and response mechanism was implemented, relying on consistency and plausibility checks based on innovation errors calculated using an extended Kalman filter. This system enables the real-time identification and isolation of malicious behavior within the network. The proposed security method was experimentally validated by simulating various types of attacks, including poisoning attacks, Sybil attacks, and DoS attacks. The results demonstrate very high effectiveness in anomaly detection and resilience, while maintaining low computational and communication overhead, making it suitable for embedded VANETs systems.

Acknowledgment

This thesis was made possible thanks to the support, guidance, and trust of many persons to whom I wish to express my sincere gratitude.

First and foremost, I extend my deepest gratitude to my thesis advisor, Prof. Jalel BEN OTHMAN, for his supervision, availability, support, and the trust he placed in me throughout these years. His advice, scientific rigor, and patience were invaluable in completing this work. I have learned a great deal by his side, both on a scientific, professional, and personal level.

I would also like to thank the jury members, Prof. Marco DI RENZO, Prof. Pascal LORENZ, Prof. Selma BOUMERDASSI, Prof Boubaker DAACHI, Prof. Jaafar GABER, and Prof. Hacène FOUCHAL, for agreeing to review this thesis and participate in my defense. I am grateful for the time they devoted to reading this manuscript and for their interest in this work.

My thanks also go to the Guinean Ministry of higher education and scientific research (MESRS - Guinée), whose financial support made this thesis possible.

I would like to thank all the members of the L2S laboratory for their warm welcome, kindness, and the stimulating research environment I enjoyed during these years. Special thanks also go to CNRS, CentraleSupélec and University Paris-Saclay for providing the excellent scientific and academic framework in which this thesis was conducted.

I am also grateful to everyone with whom I had the opportunity to exchange ideas, collaborate, or share friendly moments during this doctoral journey. These encounters greatly enriched and made these years memorable.

On a more personal note, I would like to express my special gratitude to my elder brother Alpha Oumar, who played a pivotal role in my journey since my early years of study. He is the one who set me on the path to education and who never ceased to support and encourage me until the completion of this thesis.

I also deeply thank my other older brother, Ibrahima, for his valuable advice and encouragement, as well as my entire extended family.

I would like to thank my wife Rouguiatou and my children Alpha Oumar, Fatoumata Bintou, and Mohamed Cellou for their patience, courage, and understanding regarding my absence during these years of research, as well as for the joy and strength they bring me every day.

To everyone who contributed directly or indirectly to the completion of this work, I offer my sincere thanks.

Contents

List of Figures

List of Tables

List of Abbreviations

ADE Average Displacement Error. 71, 75

AI Artificial Intelligence. 15, 16, 19, 20, 23, 33, 39, 40, 54, 59, 68, 91, 110

CFL Centralized Federated Learning. 40, 41

CNN Convolutional Neural Network. 35–37, 39, 62

CO2 Carbon Dioxide. 30

DFL Decentralized Federated Learning. 20, 23, 40, 41, 43, 44, 47, 49–52, 54, 79, 91, 93, 102, 104

DL Deep Learning. 19, 33, 34, 39, 59

DoS Denial-of-Service. 18, 23–25, 50, 77, 101, 104, 105, 111

DP Differential Privacy. 53, 54, 57

ECDH Elliptic Curve Diffie-Hellman. 22, 94, 95

ECDSA Elliptic Curve Digital Signature Algorithm. 22, 94

EKF Extended Kalman Filter. 22, 25, 95, 96, 98, 99

FDE Final Displacement Error. 71, 72, 75

FedAvg Federated Averaging. 21, 40, 43, 44, 69, 70, 72, 89–91

FIFO First In, First Out. 31

FL Federated Learning. 16, 19, 20, 23, 24, 40–43, 50–52, 54, 55, 57, 59–61, 68, 70, 72, 77, 87, 91, 93, 110, 111

GNN Graph Neural Network. 37, 39

GPS Global Positioning system. 32

HE Homomorphic Encryption. 53, 57

IoV Internet of Vehicles. 41

IP Internet Protocol. 81

ITS Intelligent Transportation Systems. 15, 19, 27, 44, 109

JSON JavaScript Object Notation. 86

LSTM Long Short-Term Memory. 34–37, 62

LWR Lighthill-Whitham-Richards. 29, 30

MANETs Mobile Ad-hoc Networks. 15

MDS Misbehavior Detection System. 22, 25, 57, 94, 95, 102, 105, 107, 112

ML Machine Learning. 19, 33, 34, 39, 59

MSE Mean Squared Error. 71, 73, 74, 86

NGSIM Next Generation Simulation. 20, 24, 70, 75, 76, 86, 87

OBUs On-Board Units. 15, 80

P2P Peer-to-Peer. 43–45, 47

RDM Random Direction Model. 29

RL Reinforcement Learning. 43

RMSE Root Mean Squared Error. 24, 72–75, 77, 86, 87, 104

RNN Recurrent Neural Network. 34, 36, 37, 39, 62

RSU Road Side Unit. 52, 79

RWalk Random Walk. 29

RWP Random Way-Point. 29

SMPC Secure Multi-Party Computation. 53, 54

TCP Transmission Control Protocol. 86

UDel University of Delaware. 31

UDP User Datagram Protocol. 81, 86

VANET Vehicular Ad hoc Network. 15–21, 23–25, 27, 33, 39, 40, 44, 45, 47, 49–54, 57, 59, 60, 77, 79, 80, 84, 91, 93, 94, 107, 109, 110, 112

1 - General Introduction

1.1 . Context

The emergence of smart cities integrating Intelligent Transportation Systems (ITS) [2] imposes strict technological requirements on autonomous vehicles, which must operate under wide area and ultra-reliable connectivity. These vehicles continuously exchange information with roadside infrastructure, cloud or edge servers, and neighboring vehicles through Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), and Vehicle-to-Everything (V2X) communications. This inter-vehicular communication ecosystem is enabled by Vehicular Ad hoc Network (VANET) [2], a subclass of Mobile Ad-hoc Networks (MANETs) [3], in which vehicles are equipped with wireless sensors and On-Board Units (OBUs) and form self-organizing networks without relying on fixed infrastructure [4]. VANETs constitute a fundamental component of ITS systems, by enabling both safety-related applications such as collision avoidance and driver assistance, as well as non-safety services, including traffic optimization and passenger infotainment [4, 5].

Unlike conventional wireless networks, VANETs are characterized by high vehicle mobility, rapidly changing network topology, intermittent connectivity, heterogeneous vehicular capabilities, and strict real-time constraints [5, 4]. These intrinsic properties make VANETs a highly dynamic and safety-critical system [2]. Decisions derived from exchanged data directly influence the vehicle behavior, its trajectory planning process, and cooperative safety mechanisms.

At the same time, the large volume of mobility-related data generated and exchanged within VANETs significantly expands the system's attack surface. VANET networks are inherently decentralized, are increasingly using Artificial Intelligence (AI)-based algorithms, and are increasingly evolving in collaborative processes. A prominent example is distributed learning, where vehicles cooperate to train a shared neural network model to enhance the efficiency of individual trajectory planning. In such scenarios, adversaries may exploit open wireless channels to inject false information, such as corrupted model updates, specifically designed to prevent global model convergence or induce erroneous trajectory predictions. Beyond manipulating model updates, attackers may impersonate legitimate vehicles or disrupt communications [6].

Threats targeting authentication, confidentiality, integrity, availability, and non-repudiation can compromise not only data privacy or system availability but also the physical security of passengers and their properties. In such contexts, inaccurate or maliciously altered information may directly impact vehicular decision-making processes, potentially leading to hazardous situa-

tions such as collisions or traffic disruptions.

Therefore, ensuring data integrity, availability, and trustworthiness in distributed vehicular environments is a fundamental prerequisite for deploying reliable intelligent transportation systems. In the following section, we examine both the opportunities enabled by collaborative vehicular systems and the fundamental technical and security challenges that accompany its deployment in highly dynamic VANET environments.

1.2 . Opportunities and Challenges

The integration of collaborative learning into VANETs offers significant opportunities. Distributed learning mechanisms, such as Federated Learning (FL) [7], enable vehicles to collectively improve prediction accuracy and anticipate traffic evolution without centralizing raw data [8]. Such collaboration improves scalability and enhances privacy by keeping sensitive data on board the vehicle, thereby reducing dependence on centralized infrastructure [9].

However, these opportunities come with substantial technical and security challenges such as:

- **Scalability and Privacy Limitations of Centralized Architectures**: Centralized approaches requiring the transmission of raw vehicular data to aggregation servers generate excessive communication overhead and impose high computational burdens on central entities [7]. These solutions fail to scale in large-scale deployments and introduce single points of failure. Moreover, transmitting raw mobility traces raises critical privacy concerns regarding vehicle and user sensitive data.

  FL [7] reduces privacy risks by keeping raw data local and exchanging only model updates (weights and biases) . However, it still presents important limitations such as: reliance on a central aggregation server, vulnerability to gradient leakage and inference attacks, exposure to poisoning and backdoor attacks, and communication overhead due to iterative update exchanges. This is particularly problematic in dynamic VANETs environments [10].

- **Computational Overhead and Latency Constraints**: Many existing security solutions are computationally expensive and communication-intensive. Heavy cryptographic protocols or complex multi-stage validation schemes may introduce unacceptable latency in safety-critical scenarios [11]. In VANETs, where millisecond-level responsiveness is required, even slight delays can lead to catastrophic failures.

- **Security Threats and Adversarial Attacks**: Collaborative AI-based approaches often prioritize predictive accuracy while overlooking the pri-

vacy and integrity of exchanged model updates. In fully decentralized settings, malicious nodes may inject poisoned gradients, launch backdoor attacks, or execute Sybil attacks to degrade system availability [12], or prediction accuracy. The dynamic topology and intermittent connectivity of VANETs further complicate the detection of such sophisticated Byzantine behaviors.

- **Dynamic Topology and Ephemeral Trust Management**: Due to high vehicular mobility, trust relationships remain highly dynamic. Traditional static trust approaches are no longer effective for these constantly evolving environments [13]. Effective solutions must be adaptive, lightweight, and compatible with decentralized configuration.

These challenges highlight the need for an integrated framework capable of simultaneously addressing prediction performance, communication efficiency, and robust security under stringent VANET constraints. The motivations underlying the solutions proposed in this thesis are presented in the following section.

1.3 . Motivations

The rapid evolution of VANETs toward highly autonomous and cooperative vehicular ecosystems raises a fundamental question: *how can a decentralized vehicular network achieve collaborative learning and precise trajectory prediction while maintaining real-time decision-making capabilities, even in the presence of adversarial threats and hardware limitations*?

This thesis is motivated by the observation that existing solutions typically address prediction accuracy, communication efficiency, or security in isolation. However, in highly dynamic vehicular environments, these three dimensions are strongly interdependent. Improving prediction performance without ensuring updates or parameters integrity may expose the system to model poisoning, denial-of-service attacks, and other threats. Strengthening security through heavy cryptographic or complex validation mechanisms may degrade latency and scalability. Reducing communication overhead without adapted aggregation strategies may compromise convergence and accuracy. Therefore, it is necessary to adopt a systemic approach based on a jointly design strategy.

This thesis is built on the core principle that understanding, mimicking, and accurately predicting vehicle mobility serves as the foundation of a secure and smart vehicular network. In VANETs, mobility is not merely an application-level parameter; it defines the network topology, connectivity duration, trust relationships, and data distribution patterns. The highly dynamic nature of vehicular movement directly influences the following characteristics:

- The set of neighboring vehicles participating in collaboration at each time step;
- The stability and reliability of communication links;
- The temporal consistency of exchanged information;
- The detectability of anomalous or malicious behaviors.

For those reason, this thesis begins by studying mobility models in VANETs. Accurate mobility prediction enhances cooperative driving performance, enabling proactive security mechanisms. By modeling expected vehicular dynamics, the system can compare predicted and observed behaviors, detect inconsistencies in exchanged updates, and identify abnormal patterns that may indicate adversarial manipulation. In this sense, mobility prediction becomes an enabler of adaptive anomaly detection and trust analysis.

Another major motivation arises from the limitations of centralized architectures. Centralized data aggregation requires transmitting large volumes of raw mobility data, generating excessive bandwidth consumption and exposing sensitive location information. Moreover, the computational burden imposed on aggregation servers limits scalability and introduces single points of failure. These constraints are incompatible with large-scale smart vehicular networks deployments.

While federated and fully decentralized learning paradigms alleviate some of these issues, they introduce new challenges. Fully decentralized collaboration requires robust peer-to-peer coordination mechanisms that can operate without global supervision. In such settings, malicious nodes may attempt to:

- Inject poisoned gradients or manipulated weights;
- Launch Sybil attacks by assuming multiple identities;
- Introduce backdoors into collaboratively trained models;
- Disrupt availability through Denial-of-Service (DoS) strategies.

Simultaneously, vehicular on-board units possess limited computational and storage resources compared to traditional cloud infrastructures. Security solutions must therefore remain lightweight in terms of processing and communication overhead to preserve real-time responsiveness.

Consequently, the motivation of this thesis is structured around three key areas:

- **To enhance predictive accuracy** in VANETs through collaborative learning mechanisms that exploit distributed vehicular data to improve prediction accuracy while preserving privacy.

- **To design a fully decentralized and efficient communication algorithm** compatible with dynamic topologies, bandwidth constraints, and real-time requirements.
- **To integrate lightweight, privacy-preserving, and data-centric security mechanisms** capable of ensuring integrity, authenticity, and availability without imposing excessive computational or communication overhead.

By addressing these objectives, this research seeks to bridge the gap between high-performance collaborative trajectory prediction and practical, security-aware deployment in real-world vehicular networks. The main goal is to improve the development of Intelligent Transportation Systems (ITS) that are not only accurate and efficient, but also resilient, trustworthy, and scalable within highly dynamic VANETs environments. Grounded in this idea, and aiming to overcome the aforementioned challenges while relying on identified opportunities, this thesis presents, in the next section, a structured set of progressive contributions designed to enhance vehicle prediction accuracy, communication efficiency, and security efficiency and robustness in fully decentralized vehicular environments.

1.4 . Main contributions of the thesis

To address the scientific and technical objectives of this thesis, a coherent and progressive global architecture is designed and structured around a set of complementary and interrelated contributions that presented as follows:

- **A comprehensive and comparative study of the state of the art**: the first contribution of this thesis consists of a comparative literature review of mobility models and security and privacy methods in Vehicular Ad Hoc Networks (VANETs). This analysis enabled us to classify existing mobility approaches into two main categories: (i) classical mobility models, based on mathematical principles, statistical approaches, and laws of physics; and (ii) models relying on Artificial Intelligence (AI) techniques, particularly Machine Learning (ML) and Deep Learning (DL) algorithms.

  Then AI-based models were studied. These models offer superior flexibility and, under specific conditions, achieve higher predictive performance. These were further subdivided into two groups: centralized models, which aggregate data from all participating vehicles onto a single infrastructure where the AI model is implemented and trained; and distributed models. The latter comprises two subcategories: traditional FL-based models, which require an aggregation server to coordinate

the training process, and fully Decentralized Federated Learning (DFL)-based models, in which vehicles communicate directly with one another without a centralized coordination infrastructure. This study facilitated a critical examination of the modeling assumptions, predictive performance, computational complexity, and the inherent strengths and limitations of each family of mobility models, highlighting their respective advantages and drawbacks.

Subsequently, existing security approaches designed for vehicular networks were analyzed. This study enabled us to identify the main threats affecting such systems, particularly those targeting communication integrity, availability, confidentiality, and authenticity. It also highlighted the strengths and limitations of current security and privacy-preserving mechanisms, particularly their partial inadequacy in fully decentralized, highly dynamic VANET environments.

This dual analysis—mobility modeling, and security and privacy mechanisms—constitutes the conceptual foundation upon which the contributions proposed in this thesis are built.

- **FedVANET-TP: A Federated Trajectory Prediction Model for VANETs**: This contribution presents an AI-based trajectory prediction model for VANETs built on an FL architecture. FedVANET-TP uses a customized Transformer [1] optimized for high-accuracy predictions while remaining lightweight to reduce computation and communication overhead between vehicles and the aggregation server. Prediction accuracy is improved through a sliding time-window generation algorithm that preprocesses spatio-temporal mobility data for more effective learning. Computational and communication costs are further reduced by removing unnecessary operations for trajectory tasks (e.g., tokenization) and by reducing the number of layers, attention heads, and embedding dimensions in the used architecture.

  The optimized model is deployed on each vehicle, while the FL process is implemented using the Flower framework [14]. Each vehicle trains its local model copy using its own mobility data and transmits only the learned model weights to the aggregation server. The server iteratively aggregates the received updates at each communication round and redistributes the aggregated global model to all participating vehicles. This allows vehicles to benefit from collective knowledge without exchanging raw, sensitive data, preserving privacy and communication efficiency.

  Experiments on the Next Generation Simulation (NGSIM) US-101 and I-80 datasets show that FedVANET-TP achieves lower RMSE than state-of-the-art models while maintaining efficient communication within the

federated framework.

- **FDG-VTP: A Fully Decentralized Gossip Vehicular Trajectory Prediction Model**: This contribution extends the collaborative federated paradigm, FedVANET-TP, by removing the central aggregation server to enable direct vehicle-to-vehicle (V2V) communication. It introduces an optimized peer-to-peer communication framework specifically tailored for the highly dynamic conditions of VANETs. The core features of FDG-VTP include:

  - **Enhanced Prediction Model**: The framework incorporates practical neighbor-selection algorithms to strengthen the existing time-window creation mechanism, specifically optimized for the customized Transformer-based prediction model to enhance trajectory prediction accuracy.
  - **Asynchronous Gossip-Based Communication Algorithm**: an asynchronous communication algorithm is introduced, along with a novel weighted aggregation method. Unlike traditional approaches, such as Federated Averaging (FedAvg) algorithm [7], that rely solely on dataset sizes, this method also evaluates the reliability of incoming weights by computing a reliability factor for each update received from each peer. The receiver assesses each peer's contribution at each communication round using the reliability factor; if the received weights do not yield a measurable improvement to the local model, they are excluded from the aggregation process to prevent model degradation.
  - **Adaptive Transmission Control**: a specialized parameter is introduced to adaptively control the updates sending frequency on each target vehicle. By synchronizing transmission with the evolution of communication rounds, the system prevents vehicles from prematurely broadcasting insufficiently trained weights to their peers.

    Scalability through Limited Dissemination: The Gossip protocol [15] ensures that each target vehicle communicates only with a subset of peers rather than the entire network. This approach allows vehicles to collaboratively train the global prediction model without sharing sensitive raw data, while maintaining low computational complexity and minimal communication overhead.

  Experimental evaluations demonstrate that FDG-VTP outperforms both centralized and decentralized baselines in trajectory prediction accuracy while maintaining significantly lower communication and computational costs.

- **PPIR: A Privacy-Preserving and Intrusion-Resilient decentralized gossip-based trajectory prediction framework for VANETs**: This contribution introduces security and privacy layers integrated into the fully decentralized trajectory prediction framework. PPIR is a lightweight and robust framework designed for detecting and responding to malicious intrusions, operational failures, and transmission errors. It consists of a data-centric Misbehavior Detection System (MDS) reinforced by a privacy-preservation mechanism.

  The features of PPIR are as follows:

  - **Privacy Preservation**: PPIR employs a signcryption scheme based on the Elliptic Curve Diffie-Hellman (ECDH) [16] key agreement and the Elliptic Curve Digital Signature Algorithm (ECDSA) [17]. Within this scheme, every message is encrypted and signed by the source vehicle prior to transmission. The receiving vehicle then performs the inverse operations using its private key and a shared public key.
  - **Data-Centric MDS**: The proposed MDS relies on consistency and plausibility checks performed on the weights received from each peer during every communication round.
    * *Plausibility Checks:* These checks examine the likelihood of a received update by comparing it with the last three updates received from the same vehicle to identify statistical anomalies.
    * *Consistency Checks*: These checks rely innovation errors—the difference between the weights predicted for each peer by an Extended Kalman Filter (EKF) [18] (representing the target vehicle's expectations) and the weights actually received. To ensure real-time feasibility, the EKF utilizes sparse matrices and diagonalized Jacobians, significantly reducing computational complexity.
  - **Multi-Stage Filtering**: Prior to the core behavioral checks, the system performs a suite of lightweight verification filters to serve as a first line of defense: (i) numerical integrity validation (to detect NaN or Inf values), and (ii) tensor shape conformity. These filters immediately isolate and discard clearly corrupted or malformed updates.

  This dual-layer approach ensures both cryptographic protection (confidentiality and authenticity) and behavioral validation of the exchanged updates.

Experimental evaluations demonstrate that PPIR outperforms state-of-the-art methods in detecting several types of attacks, including backdoor and Sybil attacks, while achieving performance comparable to existing approaches under DoS scenarios. Crucially, these security enhancements are achieved without introducing prohibitive communication or computational overhead, thereby maintaining compatibility with the stringent real-time constraints of VANETs.

1.5 . Outline of the thesis

The remainder of this thesis is organized as follows:

- **Chapter 2: Literature Review**

  This chapter establishes the theoretical and technical foundations of the thesis.

  First, it provides a detailed review of vehicular mobility modeling approaches. Classical mobility models are analyzed in terms of assumptions, complexity, scalability, and predictive performance. Their limitations in highly dynamic environments are discussed. Second, AI-based mobility prediction models are examined, including recurrent neural networks, convolutional models, and Transformer-based architectures. Particular attention is given to their ability to capture spatio-temporal dependencies and their computational requirements in vehicular contexts.

  Third, learning paradigms are analyzed, including centralized machine learning, classical Federated Learning (FL), and fully DFL. The chapter discusses their respective tradeoffs in terms of scalability, privacy preservation, communication overhead, convergence rapidity, and robustness.

  Finally, existing security and privacy-preserving mechanisms for VANETs are reviewed. The chapter analyzes authentication schemes, data integrity mechanisms, trust management systems, and defenses against various types of attacks, including poisoning, backdoor, Sybil, and DoS attacks. Their computational and communication costs are critically examined, highlighting open research gaps that motivate the proposed contributions.

- **Chapter 3 FedVANET-TP: A Federated Trajectory Prediction Model for VANETs**

  This chapter presents FedVANET-TP framework. The chapter starts by describing the system architecture and problem formulation for collaborative trajectory prediction in a Federated Learning (FL) setting. The

design of a customized lightweight Transformer model is detailed, including architectural simplifications introduced to reduce computational complexity and communication overhead while maintaining high predictive accuracy. The FL workflow is described, including local training procedures, aggregation mechanisms, communication process, and implementation using the Flower framework. Experiments conducted on the NGSIM publicly available vehicular datasets are presented. Performance is evaluated in terms of Root Mean Squared Error (RMSE), convergence rapidity, communication overhead, and computational efficiency. Comparisons with state-of-the-art centralized and FL-based baselines demonstrate the effectiveness of the proposed FedVANET-TP approach.

- **Chapter 4 FDG-VTP: A Fully Decentralized Gossip Vehicular Trajectory Prediction Model**

  This chapter extends the FL-based paradigm toward a fully decentralized architecture specifically adapted to the stringent constraints of VANETs environments. It starts by establishing the need to eliminate the central aggregation server to mitigate single points of failure and coordination bottlenecks. The core of this method lies in an asynchronous gossip-based communication algorithm, paired with a neighbor-selection mechanism that adapts to dynamic vehicular topologies. To ensure the quality of the shared updates, the framework employs a dynamic weighted aggregation strategy that accounting for local dataset sizes of the involved vehicles and a reliability factor computed by the receiver vehicle for each received update. This is complemented by an adaptive update-scheduling mechanism that prevents the premature transmission of insufficiently trained weights. The proposed framework was compared against both centralized and decentralized baselines. The results demonstrate superior trajectory prediction accuracy while maintaining low communication overhead and controlled computational costs.

- **Chapter 5 PPIR: Privacy-preserving and Intrusion-Resilient framework for Gossip-based Communication Algorithm in VANETs**

  Building on the proposed fully decentralized framework, the method introduced in this chapter comprises robust security and confidentiality layers integrated into the framework. The chapter provides a comprehensive analysis of the threat model, specifically addressing poisoning, backdoor, Sybil, and DoS attacks within collaborative vehicular learning environments. To mitigate those threats, the proposed approach combines cryptographic protection with anomalous detection process. Specifically, it implements an elliptic-curve Diffie–Hellman-based signcryption mechanism to ensure the confidentiality, integrity, authenti-

cation, and non-repudiation of exchanged model updates. This mechanism is reinforced by a misbehavior detection and resilient system (MDS) that relies on EKF consistency checks, using innovation errors to identify anomalous updates. Additionally, a plausibility verification mechanism analyzes the historical evolution of received weights to assess the trustworthiness of participating vehicles. The chapter concludes by experiencing the system's resilience against various attacks and comparing its detection performance with that of state-of-the-art methods. The results highlight superior detection capabilities for byzantine behavior such as backdoor and Sybil attacks and comparable performance under DoS type attacks, all while maintaining low computational and communication overhead.

- **Chapter 6: General Conclusion and Perspectives**

  The final chapter synthesizes the main contributions of the thesis and discusses their overall impact on a secure, collaborative, and fully decentralized mobility forecasting framework for VANETs. It highlights how the proposed frameworks collectively address prediction accuracy, decentralization, communication efficiency, and security robustness under strict vehicular constraints. The chapter also outlines future research directions, including large-scale cluster-based collaboration mechanisms, adaptive trust management models, enhanced adversarial robustness against sophisticated attacks, and scalability analysis in dense urban deployments. This concluding chapter positions the thesis contributions within the broader evolution of intelligent, decentralized, and secure transportation systems.

2 - Literature Review

2.1 . Introduction

VANETs are increasingly emerging as the cornerstone of ITS. They hold significant potential to enhance road safety, improve traffic efficiency and driving comfort [2]. VANETs notably ensure the transmission of alert messages between vehicles in case of accidents [6], the reporting and avoidance of traffic jams [2], and the calculation of alternative routes in case of traffic congestion [19]. They also provide onboard Internet access, support entertainment services, and facilitate the exchange of information about points of interest such as gas stations, and more [6].

One of the main characteristics of VANETs is their highly dynamic topology. The rapid movement of vehicles causes frequent changes in network topology, with connections forming and breaking at very high rates. Addressing security challenges in such a dynamic context requires a thorough understanding and anticipation of vehicle behavior on the road. For this reason, the literature review presented in this chapter begins with a detailed study of mobility models. The chapter then analyzes and compares distributed and collaborative, centralized and fully decentralized, mobility forecasting approaches leading to a classification of mobility models summarized in Figure 2.1. Furthermore, studies addressing the performance, security, and privacy of communication algorithms in distributed vehicular systems are reviewed and compared. Finally, the respective limitations of these approaches are examined, thereby motivating the development of the proposed methods.

The review of existing mobility models is presented in the following section.

2.2 . VANETs mobility models

Vehicle mobility models play a crucial role in the design, simulation, and optimization of modern transportation systems. They enable the reproduction of vehicle movements and behaviors in various contexts including urban traffic, highway scenarios, and the management of autonomous vehicle fleets. These models are essential to figure out traffic flow dynamics, predicting road congestion, and supporting urban planning. In the context of vehicular communication networks such as VANET, they also contribute to enhancing road safety, traffic efficiency, and driving comfort. In the literature, mobility models are generally classified according to the level of detail they provide in describing vehicle movements. This classification leads to three

main categories [20]:

- Microscopic models

Microscopic models consider individual vehicle characteristics, such as acceleration or deceleration needed to maintain a safe distance or headway to mimic or forecast vehicles movements. These models offer a high level of detail and accuracy but demand significant computational resources [21].

- Macroscopic models

Macroscopic models rely on hydrodynamic phenomena to describe the overall characteristics of traffic movement, such as flow, speed, and density, for a given road segment at a specific point in time [22]. While they are more straightforward and computationally more efficient, their realism and accuracy are often lower.

- Mesoscopic models

Mesoscopic models aim to combine the advantages of the two previous approaches, offering a trade-off between accuracy and computational complexity. They describe both the raw characteristics of traffic flow and individual interactions between vehicles [23]. This type of model has notably been used to study adaptive cruise control and autonomous driving [24]. It allows for modeling the distribution of progression and density for a group of vehicles at a given time and in a specific space, while also enabling the control of individual vehicle behavior.

For a more in-depth classification, mobility models can also be distinguished based on the methods or techniques used for their implementation. Our literature review has thus identified two main categories: classical mobility models and machine learning-based mobility models. These two approaches are detailed in the following subsections.

2.2.1 . Classical mobility models

Classical mobility models are based on mathematical foundations, statistical approaches, and laws of physics to describe and predict vehicle movement on road networks. This category of models is generally divided into four subcategories [25]: synthetic models, survey-based models, trace-based models, and simulator-based models. The following subsections discuss these mobility model subcategories.

Synthetic mobility models

Synthetic models describe vehicle movement using mathematical approaches. They encompass various categories, including stochastic models, traffic flow models, car-following models, behavioral models, and queuing models [26].

- Random mobility models

Random models leverage stochastic properties to describe entity movement. They characterize node displacement where both speed and distance traveled are randomly defined. These models aim to simulate mobility for specific applications while maintaining the integrity of their stochastic properties. Among the most frequently employed models in this category, the Random Way-Point (RWP) model [27], is prominent. In this model, each node randomly selects a destination (waypoint), a constant speed, and a pause time. These values are independent of previous choices, as the lack of historical data storage prevents nodes from modeling their speed. This inevitably leads to abrupt stops and sharp turns, and the assumption of constant speed between waypoints is often deemed unrealistic. The Random Walk (RWalk) model [26] was developed to address some of these limitations by removing pause times. Furthermore, it randomly generates azimuth and travel time instead of sampling a fixed destination. Once this time has elapsed, each node receives new time and direction values. This model reduces the frequency of abrupt turns, the lack of a preferred direction, and unexpected pauses. The Random Direction Model (RDM) [26] allows nodes to move at random speeds and in different directions until they reach a boundary line. After stopping, each node selects new values. While this model ensures a uniform node distribution,, its major drawback lies in decreasing the nodes' average speed over time. The authors in [28] proposed an improvement of the RDM in which vehicles choose new destinations and speeds based on their current direction and speed, thus avoiding sharp turns and sudden braking. The study in [29] introduces a random mobility model tailored to urban environments. It considers road topology, vehicle behavior at intersections, speed variation based on neighboring vehicles, traffic lights, and attraction points, making it more realistic than traditional RDM.

Due to their simplicity and ease of deployment, stochastic mobility models are widely used in various applications. However, they capture only a limited range of real-world vehicle behaviors.

- Traffic flow mobility models

Traffic flow models conceptualize mobility as a continuous process, analyzed at either the macroscopic or mesoscopic scale. Drawing inspiration from fluid dynamics, these models generally aim to represent the overall behavior of traffic. They are based on aggregated variables such as vehicle density, flow rate, and average speed, as well as structural characteristics of the road network, including the number of lanes, the presence of intersections, and speed limits. The Lighthill-Whitham-Richards (LWR) model [22, 30] is the most widely

used in this category. It relies on kinematic wave theory to model traffic phenomena such as congestion and acceleration waves and allows for the estimation of traffic speed and density. The authors in [31] proposed an extension of the LWR model that separates vehicles into two groups: those changing lanes and those remaining in their current lane. This model also considers vehicle types, car-following effects, lane-changing behavior, and road geometry. The study in [32] introduced a traffic flow evacuation model based on Markov chains to reduce vehicle Carbon Dioxide (CO_2) emissions through a speed control algorithm. This model enables the traffic control center to recommend an optimized speed to drivers. The results show a 31.68% reduction in CO_2 emissions. However, the model does not capture driver behavior or possible communication failures, particularly between vehicles and the traffic control center.

Traffic flow models are particularly effective for analyzing the overall dynamics of road traffic. However, they provide only a limited level of detail, and their applicability may be constrained when traffic conditions continuously evolve over time.

- Car-following mobility models

Car-following models replicate the behavior of real drivers by considering the vehicles directly ahead of them. They rely on variables such as the position and speed of surrounding vehicles to model the individual behavior of each vehicle. The Krauss model [33] is a car-following model that realistically reproduces traffic behavior such as acceleration, deceleration, and congestion. An extension of the Krauss model incorporating lane-changing behavior on a two-lane road is proposed in [34]. The model presented in [35] is among the first to introduce cooperation between drivers within a car-following framework. It realistically simulates traffic behavior and allows the assessment of the impact of various connected autonomous vehicle scenarios on traffic flow. However, while car-following models are realistic, they tend to be complex and require significant computational resources.

- Behavior-based mobility models

Behavior-based models consider the driver's reactions using behavioral rules and stimulus–response schemes. For example, Legendre et al. [36] proposed an approach that decomposes mobility into individual behaviors to simulate realistic movement patterns in critical scenarios such as traffic incidents and emergencies. Gipps [37] introduced a model based on the assumption that each driver sets limits on their braking rates and desired acceleration thresholds. The author demonstrated that the model can replicate real-world traffic flow characteristics when realistic parameter values are used in simulations. Kharrazi et al. [38] extended behavior-based models by incorporating

mission-specific information such as vehicle type, route, and driving environment. This enhancement allows the generated driving cycles to be more realistic and better aligned with the actual conditions the vehicle will encounter.

As with car-following models, behavior-based models are highly realistic but require considerable computational resources, especially for large-scale traffic scenarios.

- Queuing mobility models

Queuing models describe roads as First In, First Out (FIFO) queues, where vehicles are treated as customers. Liu et al. [39] introduced a shockwave theory model derived from the LWR model to estimate queue lengths in real time at signalized intersections, even under congestion and long queues. Mirchandani and Zou [40] developed another queuing model for a simplified adaptive control strategy and presented a numerical algorithm to compute steady-state performance metrics.

This category of models is relatively easy to understand and implement and can be applied to a variety of conditions and complex queuing scenarios. However, they fail to consider vehicle coordination at intersections when multiple vehicles cross at the same time, and they do not model lane-changing behavior.

Survey-based mobility models

Survey-based models are based on large sets of statistical data collected through surveys. These data sets may include information related to road users' habits, such as travel time, lunch breaks, distance traveled, or even preferred meal types. Such information can be used to calibrate mobility models and support the development of a generic model capable of reproducing the pseudo-random or deterministic behaviors observed in real urban traffic [25]. For example, the University of Delaware (UDel) model [41], is used to simulate urban traffic. It considers various factors such as arrival times at work, lunch breaks, pedestrian and vehicle dynamics, and time usage throughout the workday. The agenda-based mobility model, proposed in [42], is a microscopic mobility model that uses survey data to generate individual schedules for mobile nodes. It can reproduce various traffic phenomena, making it suitable for simulating urban traffic. Nevertheless, these models present limitations as they lack a detailed representation of node movements, thereby limiting their suitability in numerous high-precision mobility scenarios.

Trace-based mobility models

Trace-based mobility models rely on empirical trajectory data collected from real-world agents such as vehicles, pedestrians, or mobile devices. Unlike syn-

thetic models, which generate artificial motion patterns based on stochastic assumptions, trace-based models aim to replicate realistic mobility behaviors by learning directly from observed movement traces. These traces consist of temporally ordered location samples, typically gathered from Global Positioning system (GPS) sensors, vehicle on-board systems, or mobile communication networks. By leveraging these datasets, trace-based mobility models can capture the spatial heterogeneity and temporal dynamics of real-world motion, providing a high-fidelity representation of how agents move, interact, and respond to their environment. For instance, the study presented in [43] introduces a trace-based mobility model designed to simulate the movements of first responders, such as firefighters and rescue teams, in emergency scenarios. The model is constructed from GPS traces and reproduces context-dependent movement patterns, including group coordination, points of interest, topographical constraints, and physical obstacles. Similarly, the authors of the study in [44] propose the TRAILS model, which extracts a mobility graph from real-world data and uses it to generate realistic, flexible, and scalable simulation scenarios. The study in [18] presents a model that enhances human mobility prediction across various spatial scales by utilizing proxy data collected from multiple digital platforms and geolocation services.

Trace-based models are relatively easy to calibrate and enable the reproduction of realistic behaviors, providing an accurate representation of physical and spatial constraints as well as high scalability. However, the quality of such models strongly depends on the quality and quantity of the collected traces. Moreover, these models generally exhibit low generalizability and limited flexibility, and the trace data are often large, noisy, and incomplete, requiring extensive preprocessing and cleaning steps.

Simulator-based mobility models

Simulator-based mobility models enable realistic reproduction of the dynamic behavior of vehicles within virtual environments, whether urban or highway. Unlike mathematical or statistical approaches, simulators incorporate physical, behavioral, and contextual parameters, thereby offering a more faithful representation of agents movements. These simulation tools operate at various levels of granularity. At the microscopic level, each mobile entity is individually modeled considering specific movement characteristics such as acceleration, deceleration, and lane changes. Fine-grained simulators such as SUMO [45], CORSIM [46], VanetMobiSim [47], PARAMICS [48], and VISSIM [49], or TRANSIMS [50] originally developed for urban traffic engineering, are capable of accurately modeling urban traffic while also integrating factors such as energy consumption, pollutant emissions, and noise levels [25]. At a more aggregated scale, mesoscopic or macroscopic simulators, including AIMSUN [51], VISUM [52], MATSim [53], TransModeler [54] and the one proposed in [55],

model the behavior of vehicle groups or overall traffic flows. These tools, often hybrid, are particularly well-suited to large-scale transportation planning scenarios. In the context of VANET simulations, microscopic mobility simulators such as SUMO are frequently coupled with network simulators such as OMNeT++ [56], NS-3 [57], or Veins [58], to jointly represent both vehicle mobility dynamics and communication exchanges, between vehicles or between vehicles and road infrastructure. Simulator-based mobility models thus offer the ability to evaluate, in advance, various mobility policies, traffic management algorithms, or intelligent transportation infrastructures. However, they also present certain limitations: some simulators require the purchase of commercial licenses, integration with communication simulators can be technically complex, simulation outcomes heavily depend on the quality of input data, and computational costs may become significant, especially for large-scale simulations.

Vehicle behavior on the road is inherently uncertain, nonlinear, and complex, making it difficult to predict with sufficient accuracy using traditional mathematical or physics-based models. In contrast, methods based on AI techniques, such as ML [59] and DL [60], are comparatively easier to implement and achieve more accurate predictions [61]. As such, these approaches are widely employed to address the limitations of classical mobility models, including their implementation complexity and lack of realism or precision. This category of mobility models will be discussed in the following subsections.

2.2.2 . AI-based mobility models

The rapid advancement of increasingly robust, efficient, and resource-aware AI algorithms, facilitated by the abundant availability of vast datasets, has significantly contributed to the development of mobility models with enhanced accuracy, efficiency, and generalizability. AI-based mobility models employ either ML or DL techniques to model and predict vehicle behavior on the road. These models are trained on historical vehicle behavior data to reproduce observed movement patterns and forecast future trajectories [62, 63]. The training process involves optimization procedures aimed at minimizing the discrepancy between predicted outputs and ground-truth data.

Unlike physics based models that rely in mathematical laws, ML-based approaches utilize large-scale historical datasets to capture complex mobility patterns and temporal dependencies. These data-driven methods employ a range of learning algorithms, such as Support Vector Machines (SVM) [64], Dynamic Bayesian Networks (DBN) [65], K-Nearest Neighbors (KNN) [66], and Decision Trees [67] to extract latent features that govern vehicular dynamics. This shift from rule-based modeling to statistical learning enables more adaptive and context-aware mobility predictions, particularly in highly dynamic or uncertain traffic environments. Thus, the work in [68] proposes using an arti-

ficial neural network (ANN) [69] to predict vehicle trajectories, combined with an SVM to estimate the likelihood of a lane change. The SVM classifier is employed to determine the most probable action to be taken. The resulting model is capable of predicting a lane change up to three seconds before it occurs. To improve prediction performance, the study in [70] proposes a method for anticipating driver lane change intentions by combining SVM with bayesian filtering. This approach produces multiclass probabilistic outputs, which are subsequently refined by incorporating uncertainty and temporal dynamics. In [61], the authors proposed a method based on the KNN model to provide accurate short-term traffic flow predictions on urban expressways. Decision trees, for their part, were employed by the researchers in [71] to develop a trajectory prediction approach for autonomous vehicles based on two machine learning algorithms: decision tree and naive Bayes [72]. The method also integrates a random forest [73] regressor for feature selection during the data preprocessing phase. The experimental results demonstrated that the highest prediction accuracy was achieved using the decision tree algorithm without prior selection of characteristics. While the performance of ML-based mobility models can be enhanced by incorporating additional variables, these methods remain primarily effective for short-term and relatively simple prediction tasks, and tend to perform poorly when extended to complex or long-term forecasting scenarios. To address these limitations, researchers have explored DL-based approaches.

In contrast to ML-based models, DL-based mobility prediction methods have been widely adopted to address the challenges posed by dynamic environments, particularly those involving multi-agent interactions [74]. Among DL-based models, Recurrent Neural Network (RNN) [75], specifically Long Short-Term Memory (LSTM) networks [75, 76, 77], are well known for capturing temporal dependencies [78]. LSTM architectures are particularly advantageous due to their internal memory mechanisms, which allow them to retain and utilize information from previous time steps. Consequently, these models are well-suited for modeling sequential data and achieving highly accurate trajectory forecasting. Thus, in order to improve the ability of advanced driver-assistance systems (ADAS), embedded in autonomous vehicles, to anticipate the intentions of surrounding vehicles and predict complex motion patterns, Altché and De La Fortelle [79] employed an LSTM network to develop a consistent highway trajectory prediction model taking into account both longitudinal movements (acceleration and deceleration) and lateral movements (lane changes). Similarly, the model in [80] employs an LSTM network to predict a probabilistic distribution of future vehicle locations on a grid-based map. The article in [81] introduces an LSTM-based encoder–decoder architecture for vehicle trajectory prediction. The LSTM encoder processes the historical trajectory data, while the LSTM decoder generates the future sequence of po-

sitions. The model outputs the k most probable trajectory candidates using a beam-search [82] mechanism. In [83], where the trajectories of surrounding vehicles are predicted considering the multimodality of driving behaviors (e.g., lane keeping or lane changing), LSTM is used to identify the most likely action with a certain level of confidence, and then to generate a distribution over future trajectories conditioned on that action.

Hence, the results presented in the literature show that LSTM networks can achieve adequate performance in various mobility prediction scenarios. However, LSTM only captures temporal dependencies in historical vehicle movement sequences, consequently ignoring spatial interactions between each vehicle and its environment. As a result, predictions generated by LSTM often lack realism. To mitigate this drawback, researchers have proposed integrating Convolutional Neural Network (CNN) [84, 85, 86] with LSTM. Initially developed for image recognition tasks, CNN are particularly effective at processing spatial information [87]. Therefore, the combination of CNN and LSTM networks enables the models to efficiently capture both spatial interactions and temporal dependencies. Therefore, the study presented in [88] introduces a method that integrates CNN and LSTM architectures to improve the accuracy of surrounding vehicle trajectory predictions by leveraging both spatial and temporal features. The approach in [89] uses LSTM encoders per vehicle to encode past trajectories and then arranges the LSTM hidden states into an occupancy-like grid; applies convolutional layers (CNN) over that grid to extract spatial interaction features; an LSTM decoder and dense layers predict future trajectories. The TraPHic algorithm, presented in [90], is designed for short-term trajectory prediction in a heterogeneous and dense traffic environment (including buses, cars, scooters, bicycles, or pedestrians). TraPHic utilizes LSTM to capture temporal dependencies in the target agent's movement, and CNN for spatial interactions between the agent and its immediate surroundings, implicitly accounting for neighboring agents' varying shapes, dynamics, and behaviors.

Nevertheless, although LSTM networks are specifically designed to capture temporal dependencies, they often struggle to model long-term relationships within complex sequences. As the sequence length increases, they tend to suffer from error accumulation and a loss of contextual coherence [91]. Conversely, CNN focus primarily on local interactions within their convolutional windows, which limits their capacity to capture long-range dependencies between vehicles [92]. To overcome these limitations, recent research [93, 94] have proposed the integration of attention mechanisms [95, 96] with LSTM, CNN, or both. Often embedded within LSTM-based encoder–decoder frameworks, attention mechanisms enhance the model's ability to identify and extract the most salient information from large-scale input data through the computation of attention scores. As an example, the authors in [97] em-

ployed a two-stage LSTM encoder-decoder with an attention mechanism to correct cumulative errors caused by the iterative behavior of the LSTM, thus improving the accuracy of their trajectory prediction model. The spatial-temporal attention LSTM (STA-LSTM) model, proposed in [98], incorporates an attention mechanism to enhance both the accuracy and interpretability of trajectory predictions. Similarly, Jiang et al. [78] introduced the spatial-temporal attentive LSTM encoder–decoder (STAM-LSTM) model. This model combines an LSTM encoder-decoder architecture with a spatiotemporal attention mechanism. The spatial attention mechanism captures relationships between nearby vehicles to extract a global spatial feature, while the temporal attention component handles long-term dependencies. All of this information, enriched by the dynamic data of the surrounding vehicles, is fused at each time step to form a comprehensive representation of the target vehicle's environment, leading to more accurate trajectory predictions. The paper in [99] introduces an LSTM encoder-decoder model for trajectory prediction, integrating an attention mechanism to weigh adjacent traffic flows and the target vehicle's states. Furthermore, a geometric linearization method for curved roads is introduced to generalize the model's application, thereby significantly reducing prediction errors. The contribution in [100] introduces the Residual Attention LSTM (RA-LSTM) model, an LSTM encoder-decoder equipped with a residual attention mechanism. This key module modulates the influence of surrounding vehicles within an interaction tensor to filter out irrelevant information, thus improving the accuracy of future position distribution prediction. Integration of attention mechanisms to enhance the performance of hybrid CNN-LSTM approaches is also widespread. For example, Yang et al. [101] integrated an attention mechanism with an LSTM encoder-decoder architecture, complemented by a CNN network, to extract relevant contextual information from the surrounding environment. This approach aims to optimize the accuracy of trajectory predictions in a complex traffic environment. Similarly, in [102], the authors propose a multi-module CNN-LSTM hybrid model coupled with a spatio-temporal attention mechanism and a decomposition into interaction regions around the target vehicle. Here, the CNN extracts the spatial features, the LSTM captures the temporal dynamics, and the attention weights these spatial and temporal contributions to enhance trajectory prediction. A combination of CNN, BiLSTMs [103], and an attention mechanism is also employed in a data-driven car-following model to predict the trajectory of of the vehicle ahead [104], and in [105] to improve the influence of the relevant neighbors.

The integration of attention modules into RNN and CNN architectures has improved the accuracy of mobility predictions. However, these conventional attention mechanisms struggle to effectively weight the temporal and spatial interactions captured by the underlying networks. Moreover, sequence processing remains sequential, as each new state depends on the previous

one, thereby limiting parallel processing. To address these limitations, some studies have explored the use of Graph Neural Network (GNN) [106], which is particularly effective at modeling complex spatial relationships among nodes, thereby improving prediction accuracy. GNN is often combined with RNN, or CNN, or both, and integrate attention mechanism to capture spatial and temporal dependencies comprehensively. For instance, the study in [107] introduces a Graph-based Spatiotemporal Convolutional Network (GSTCN), which integrates graph structures with convolutional layers to estimate the most likely future trajectories of nearby vehicles. Similarly, the Graph-based Interaction-aware Trajectory Prediction (GRIP) model [108] represents spatial and dynamic dependencies among surrounding vehicles and other road users through graph structures. It employs graph convolutional blocks to capture intricate inter-object relationships and utilizes an LSTM encoder-decoder architecture to generate trajectory predictions. Chang and Wang [109] proposed a multimodal, dynamic-aware (MODA) trajectory prediction model. MODA uses an LSTM encoder-decoder augmented with a Graph Attention Network (GAT) to model complex interactions. It incorporates a Conditional Variational Autoencoder (CVAE), guided by ground-truth data, to generate multiple realistic trajectories. GNN has emerged as a powerful paradigm for modeling spatio-temporal dependencies in mobility and traffic data [110, 111, 112]. They naturally capture multi-agent interactions within complex road networks [113]; however, their effectiveness strongly relies on graph construction quality and remains limited by computational cost and scalability issues [114].

To overcome these limitations, the Transformer architecture [1] has been widely adopted. Entirely based on the attention mechanism, it places this mechanism at the core of its design through self-attention, which links each element of a sequence to all others, and multi-head attention, which enables the model to simultaneously learn various types of relationships (syntactic, semantic, spatial, temporal, etc.). These properties allow the Transformer to process sequences in parallel while effectively capturing long-range dependencies. Originally designed for natural language processing (NLP), the Transformer has been successfully adapted for trajectory prediction tasks [115, 116, 117]. Its main advantage lies in its ability to model long-term dependencies and complex interactions among agents while allowing for parallel processing. Furthermore, it outperforms LSTM and CNN in terms of both accuracy and long-term prediction horizon [118]. Thus, in [118] the authors propose the SAT (Situation-Aware Transformer) for long-term trajectory prediction in urban environments. This model combines the Transformer for sequential modeling with a state encoder for external context. Its main innovation is link projection to correct error accumulation and ensure on-road compliance. A new metric, the 'area-between-curves', is also introduced to evaluate the overall trajectory similarity. In the study proposed in [119], the authors

developed the TrajectoFormer model with the aim to improve the accuracy and robustness of trajectory forecasting. To adapt the Transformer to the trajectory prediction task, they modified its blocks and bypassed the tokenization process, which also optimizes computation times. They also integrated preprocessing algorithms (neighbor selecting, temporal windows creation, etc.) to improve the model's accuracy. Similarly, Wang et al. [120] introduced the Lane Transformer model, which leverages the Transformer's efficient attention mechanisms to streamline computations while improving both the accuracy and efficiency of trajectory predictions. Cheng et al. [121] also proposed a Transformer-based model for predicting surrounding vehicles' trajectories. This model integrates driving styles, uses utility theory to infer intentions, and generates candidate trajectories through polynomial clustering. The trajectory evaluation relies on a cost function weighted by risk assessment and driving styles, prioritizing safety, efficiency, and comfort to select the optimal path. Transformer-based mobility models can achieve excellent results, even with a simple architecture and without complex interaction mechanisms [116]. In addition to its ability to model long-range spatio-temporal dependencies and process sequences in parallel, the Transformer excels at efficiently combining heterogeneous inputs (HD maps, sensors, intentions, etc.) via multi-head attention, thus offering better generalization. However, its high computational cost, its need for diverse and abundant data, and its limited interpretability [122] may restrict its use in critical practical scenarios.

Although every method aimed at improving the efficiency, safety, and comfort of road traffic offers valuable advantages, it often comes with drawbacks such as computational complexity, accumulation errors, and limited scalability. This constantly drives researchers to propose new methods for optimizing the performance, robustness, and realism of mobility models, while managing the inherent constraints of high-mobility environments and the highly dynamic topology of vehicular networks. For example, inspired by the study in [123] on perceptual and psychological processes, which claims that a driver's visual sector, defined by its radius and angle, dynamically adjusts with speed (it narrows at high speeds and widens at low speeds), the authors in [117] introduced a Human-Like Trajectory Prediction (HLTP) model. This model is designed to improve decision-making in autonomous vehicles by mimicking human cognitive processes. The HLTP model is built on a teacher–student knowledge distillation architecture. The teacher model consists of two encoders that simulate human visual perception and a Transformer-based decoder responsible for capturing spatiotemporal interactions and contextual features. The student model, on the other hand, employs a GRU-based architecture [124] to generate a multimodal prediction distribution. Through knowledge distillation, the student model learns the complex behaviors of the teacher model, thereby achieving comparable performance while signifi-

cantly reducing computational requirements, making it more suitable for deployment in resource-constrained environments.

To implement these mobility models, historical vehicle data can be aggregated on a single device where the prediction model is also deployed. In this case, the model is trained using all centralized data. Alternatively, data can be maintained locally by each vehicle, which then holds a copy of the prediction model and trains it locally using its own data solely. This latter setup constitutes a distributed environment. These two AI-based mobility model configurations are discussed in the following subsections.

2.2.3 . AI-based centralized mobility models

Centralized mobility models based on ML or DL gather historical vehicle behavior data into a single centralized infrastructure, such as a server or data center. In this centralized setting, the full dataset is used to train the model, allowing it to learn and extract underlying mobility patterns. Once training is completed, predictions are distributed to vehicles to support their decision-making processes. For instance, in [125], the authors use a deep architecture (a stacked autoencoder) trained on large-scale traffic data in a centralized manner to capture spatial and temporal correlations essential for accurate traffic flow prediction. Similarly, the DeepTransport model in [126] uses CNN to extract spatial features from the road network, and RNNs reinforced with an attention mechanism, to capture temporal dependencies, all trained together in a centralized infrastructure to forecast traffic conditions. The trajectory prediction approach in [127] is also centralized. The model (based on CNN plus explicit encoding of vehicle kinematics) is trained on datasets collected and stored centrally, so that the resulting predictions are realistic and physically feasible for deployment. In the ReCoG framework [128], past motion sequences and spatial features (via RNN and CNN) are used to build a heterogeneous graph, and the GNN applied to this complete graph is trained in one centralized process on the full dataset to capture interactions among all nodes. Another example is DeepTrack in [63] that uses Temporal Convolutional Networks (TCNs) and depthwise convolutions, optimized using centrally stored data, allowing efficient real-time trajectory prediction and traffic monitoring with low computational overhead.

Although AI-based centralized mobility models demonstrate high accuracy and efficiency, they nevertheless pose significant challenges regarding the vulnerability of sensitive data and the increased use of communication resources and computation within VANET. To address these limitations, distributed AI-based mobility models have been developed. These models will be presented in the following sub-sections.

2.2.4 . AI-based distributed mobility models

Distributed mobility models eliminate the need to transmit and store all data generated by vehicles in the network on a centralized device or to use a single machine to train the AI model using this large amount of data. In distributed approaches, each vehicle utilizes its locally collected data to train a local instance of the AI model. Subsequently, the vehicle may share either the final predictions [129] or, often, the parameters of its local model [130] with other vehicles. These models generally rely on FL [7] techniques. Consequently, distributed mobility architectures allow for the distribution of computational and communication loads across the various vehicles in the network. They also provide several additional advantages, including improved data privacy, as the data is no longer centralized; reduced communication latency, since raw data, often large in size, is no longer transmitted across the network; and better scalability, as the system can more easily adapt to large-scale networks composed of numerous connected vehicles. Distributed mobility models can be divided into two main categories: models based on Centralized Federated Learning (CFL) [131], and fully decentralized models, referred to as DFL [15]. These two approaches will be presented in the following subsections.

2.2.5 . Centralized FL (CFL)-based mobility models

This approach is based on the collaborative training of an AI model while ensuring that raw data remains stored locally on each vehicle. A central server coordinates the overall training process. Within this framework, each vehicle independently trains a local model on its own dataset and transmits only the model parameters, typically the weights and biases, to the server. During each communication round, the server selects a subset of participating vehicles, collects their locally updated parameters, and aggregates them to update the global model, commonly using an algorithm such as FedAvg [7]. The updated global model is then redistributed to the vehicles, serving as the starting point for the next round of local training. This iterative process continues until a predefined stopping criterion, generally indicating the convergence of the global model, is met. By design, this architecture preserves data privacy, as sensitive information never leaves the vehicles, and reduces communication overhead, since only model parameters, rather than raw data, are exchanged. Moreover, aggregating knowledge across multiple vehicles enhances the accuracy of the resulting models. These advantages have contributed to the growing adoption of FL for mobility modeling in autonomous vehicles. For instance, Zhou et al. [132] proposed an FL-based Transformer model for vehicle trajectory prediction and data mining. To address challenges such as non-independent and non-identically distributed (non-IID) data and high communication costs in VANET networks, Li et al. [133] proposed the FedVANET model. This approach uses a structured intra-cluster FL architecture along

with a weighted cluster-to-cluster cycling algorithm.

In the FL technique, the primary security measure is decentralization; however, decentralization alone does not guarantee optimal security. Therefore, various methods have been proposed to enhance privacy or to detect and mitigate cyberattacks targeting FL-based mobility models. In this vein, Han et al. [134] introduced a mobility prediction method based on FL and homomorphic encryption [135] to reduce the risk of privacy breaches. Similarly, to counter cybersecurity threats in the Internet of Vehicles (IoV), Wang and Yan [136] proposed the FL-TP model. This algorithm allows a vehicle to detect cyberattacks and predict the trajectories of surrounding vehicles even under varying attack levels. Nonetheless, despite these efforts to improve privacy, performance, and efficiency of CFL-based mobility models using centralized aggregation, this architecture still faces several limitations. A significant issue is the single point of failure or bottlenecks at the central server. Moreover, during each communication round, the server must wait to receive parameters from all selected vehicles (or clusters), which introduces potential delays, especially when some vehicles have limited computational or communication resources. Federated Learning also requires frequent communications between vehicles (or local clients) and the central server, which may increase communication costs. Additionally, security-enhancing algorithms (such as homomorphic encryption, secure multiparty computation, differential privacy, etc.) can be computationally costly and sometimes increase communication overhead or reduce prediction accuracy. To mitigate these limitations, some researches have proposed removing the central server entirely and adopting a fully decentralized architecture, which will be discussed in the following subsection.

2.2.6 . Decentralized FL (DFL)-based mobility models

Introduced in 2018, in the study in [137], serverless or distributed federated learning (DFL) decentralizes the aggregation of model parameters by distributing the process among neighboring participants in the FL network [15], thereby eliminating the need for a central coordination server. Unlike CFL, where a central server aggregates parameters received from participating nodes, the DFL paradigm enables each node to transmit its locally computed updates, derived from its own data, along with relevant metadata directly to neighboring nodes via peer-to-peer communication protocols, rather than forwarding them to a central entity. As a result, DFL addresses several inherent limitations of CFL architectures, including the single point of failure, the dependency on a central trusted server, and performance bottlenecks at the server level [15], or due to low capacity participants at the nodes level [138]. Furthermore, DFL provides several advantages that make it well-suited to vehicular environments, including a distributed architecture, peer-to-peer

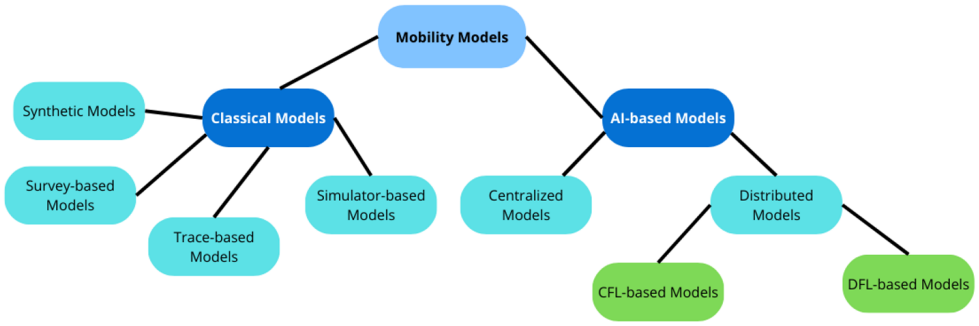

Figure 2.1: Mobility models classification

communication, enhanced privacy, and improved resilience and scalability.

However, this decentralized architecture also introduces new challenges:

- The lack of a central coordinator complicates synchronization and can slow the convergence of the global model.
- Unreliable communication, which is typical in vehicular networks, may lead to the partial or corrupted reception of exchanged models.
- Peer selection, adaptive model aggregation, and management of node heterogeneity, in terms of computational capabilities, data quality, or mobility, remain open research issues.

To address these challenges, several works have explored fully decentralized federated learning (i.e. no server or central aggregator) strategies tailored to the constraints of vehicular networks. For example, the fully decentralized trajectory prediction scheme based on gossip learning, in [139], introduces three peer-to-peer aggregation strategies, leveraging a performance estimator to adaptively weight the contributions received directly from neighboring nodes. Adaptive Model Aggregation (AMA) [140] also operates in a fully decentralized manner. Vehicles exchange updates directly with neighbors, using a performance estimator to weight contributions and ensure accurate trajectory prediction. Similarly, Soft-DSGD [141] also adopts a fully decentralized architecture. Instead of relying on a server to repair missing parameters, each vehicle locally replaces lost parameters with its own model values and dynamically adjusts parameter weights according to link reliability, helping to avoid system blockages and slowdowns in convergence due to packet losses. In the Committee Mechanism based FL (CMFL) framework [142], a peer-elected committee validates contributions each round, maintaining accuracy and stability without server, or super node coordination. Additionally, the mobility-aware decentralized FL (MDFL) framework [143] further demonstrates the effectiveness of the fully decentralized paradigm. In this approach,

vehicles elect temporary local leaders and optimize training iterations using multi-agent Reinforcement Learning (RL), improving convergence and communication efficiency. Heterogeneity in node capabilities and data distributions [144] has been tackled using lightweight local regularization methods embedded directly into decentralized optimization algorithms, such as in the decentralized parallel stochastic gradient descent (D-PSGD) [145] model, allowing nodes to balance their learning contributions without central control. Similarly, many studies have focused on jointly optimizing communication and computation costs in DFL-based approaches. For example, the C-DFL framework [146] leverages model compression techniques to reduce communication overhead, while enabling each node to perform multiple local update iterations followed by several communication phases within a single global iteration cycle. This approach differs from the conventional FedAvg algorithm, which performs multiple local updates within each communication round.

The overall effectiveness of security mechanisms and the optimization of convergence, communication, and computation costs within DFL systems fundamentally rely on the local aggregation strategy chosen by each vehicle to integrate the parameters received from neighboring nodes into its local model. The most widely employed aggregation algorithms in fully decentralized FL architectures are presented in the following subsection.

2.2.7 . Decentralized Aggregation Algorithms in DFL Systems

In centralized FL architectures, the FedAvg has established itself as the benchmark for aggregation process due to its simplicity and proven practical effectiveness [15]. In contrast, the situation is more complex in fully decentralized (DFL) systems. Although decentralized adaptations of FedAvg, such as in [147, 148], and novel specialized approaches have been proposed, no single aggregation algorithm has yet emerged as a universal solution [15]. The main difference lies in the absence of a central server in DFL. Consequently, nodes must coordinate model updates themselves, placing aggregation mechanisms at the core of key challenges:

- Ensuring convergence despite decentralization,
- Maintaining efficient communication among nodes,
- Preserving robustness, scalability, and privacy.

In contrast to FedAvg's reliance on a centralized aggregator, DFL approaches leverage Peer-to-Peer (P2P) protocols to facilitate the collaborative exchange and merging of local models. The effectiveness of these communication protocols is critically dependent on the underlying aggregation algorithm. A survey of the most prevalent aggregation algorithms in DFL systems is provided in Table 2.1. Each entry in this table outlines the algorithm's category, descrip-

tion, aggregation technique, communication scheme, strengths, and limitations. While the table groups algorithms into primary categories, several algorithms exhibit hybrid features that span multiple classes. For example, Decentralized Federated Averaging with Momentum (DFedAvgM) [147] extends the classical FedAvg framework by incorporating a momentum term to accelerate convergence and using a consensus matrix to assign parameters weights based on node connectivity. Similarly, the D^2 framework [148] combines decentralized weighted averaging with a connectivity-aware weighting scheme and distributed gradient descent, ensuring convergence for convex and non-convex functions while adapting to dynamic network topologies. Given its hybrid design, D^2 is classified as a FedAvg extension and a decentralized optimization method. The distributed subgradient method [149], achieves global consensus through local updates and neighborhood exchanges without centralized coordination. Its aggregation relies on weighting coefficients determined by communication links or random assignment, situating it within the consensus-based and distributed subgradient categories.

These aggregation methods are the key components that determine the effectiveness of communication protocols in collaborative vehicular networks based on DFL. The most commonly used communication protocols are presented and discussed in the following subsection.

2.2.8 . Communication Algorithms in DFL-based mobility models

VANETs are designed for environments characterized by high node mobility and highly dynamic network topology. Equipped with intelligent onboard communication systems, VANETs constitute a key component of Intelligent Transportation Systems (ITS). They enable various types of communications, including internal vehicle-to-sensor (V2S) communications, such as radars, cameras, and lidars, [150]. direct vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications, as well as extensions such as vehicle-to-pedestrian (V2P) and vehicle-to-network (V2N) [151]. More broadly, this entire spectrum of exchanges is often referred to as vehicle-to-everything (V2X) communications [152]. Protocol and algorithm design in this context must address pivotal challenges such as intermittent connectivity, high vehicle mobility, network heterogeneity, and security [153] to ensure low latency, reliability, scalability, safety, privacy, and robustness required by the VANET environment. In DFL-based distributed and collaborative trajectory prediction approaches, various communication schemes and mechanisms are employed. Among them, the two fundamental schemes are traditional peer-to-peer (P2P) and gossip-type communication protocols [15]. These protocols are described as follows:

Peer-to-Peer (P2P) Protocols

A P2P protocol constitutes a logical layer built on top of IP-based networks [154]. Within VANET, direct vehicle-to-vehicle (V2V) communication is enabled, eliminating the reliance on a central server. This design reduces latency by avoiding round-trips through infrastructure devices. The single point of failure is removed, since the network can continue functioning even if individual vehicles leave or fail, with other peers relaying information. The decentralized nature of P2P communication further enhances robustness and scalability, as each new vehicle joining the system increases the network's overall capacity to disseminate information across a wider area. However, the high mobility of vehicles results in a highly dynamic topology, where connections are frequently created and disrupted.

According to the adopted approach, a P2P protocol may rely either on a structured and controlled topology or one formed through random or opportunistic node connections. These categories of P2P topologies are generally characterized as follows:

- *Unstructured P2P Networks [155]:* Peers establish connections randomly or opportunistically and maintain knowledge only of their immediate neighbors. Such networks are highly resilient to node churn but generally inefficient, as search operations often rely on flooding or random walks, which generate significant redundant traffic.

- *Structured P2P Networks [155]:* Structured P2P networks are built on a well-organized and controlled topology. In such networks, data are stored at specific locations to enable efficient lookup and retrieval, typically using a distributed hash table (DHT), a decentralized indexing structure that maps keys (representing resources or data) to unique node identifiers across the network. Each resource is deterministically associated with a specific address through the DHT, ensuring predictable data placement, efficient routing, and resource-optimized communication among peers [154]. However, these networks are less tolerant to disconnections, as the departure of a peer may disrupt the structure and necessitate costly reorganization process.

- *Hybrid P2P Architectures [156, 155]:* Hybrid P2P architectures combine decentralized peer-to-peer communication with centralized coordination mechanisms. A common design pattern involves using super-peers [157], nodes endowed with higher computational capacity and bandwidth, that manage and coordinate clusters of ordinary peers. Such architectures balance scalability and control, offering improved management and lookup efficiency compared to fully unstructured systems, while

maintaining greater resilience and fault tolerance than purely centralized protocols.

Algorithm, paper	Category	Description	Aggregation method	Comm. type	Pros	Cons
SGP [158]	Fully decentralized SGD	Each node sends a portion of its parameters along with its weight to its neighbors	Dynamic weighted averaging	Asynchronous, gossip-based P2P	Guaranteed convergence for convex and non-convex functions, resilient to failures	Complexity of weight management and normalization, latency, convergence can be slower
A^2CiD^2 [159]	Fully decentralized SGD	Parallel asynchronous gradient computation and P2P communication with exponential correction	Pairwise weighted averaging	Asynchronous, Gossip-based P2P	Removes synchronization bottlenecks; reduces straggler effect; better wall-clock time in unstable networks	Complex implementation; staleness can affect accuracy; theoretical analysis challenging
DFedAvgM [147]	Fully decentralized FedAvg	Each client runs local SGD (Stochastic Gradient Descent) [160, 161] with momentum, exchanges its updates with neighbors defined by an undirected graph.	Weighted averaging	Synchronous, P2P	No single point of failure, privacy, scalability, fault tolerant, low latency, convergence under convex assumption	Topology-dependent, non-IID data issues, slow convergence, communication overhead, complexity
D-PSGD [145]	Fully decentralized parallel SGD	Each nodes performs SGD on mini-batches and communicates with neighbors only	Weighted averaging	Synchronous, P2P	Scalable efficient in high-latency or low-bandwidth settings; performs comparably to centralized SGD under IID data	Slow Convergence depends on graph connectivity; synchronous nature may suffer from stragglers
D^2 [148]	Fully decentralized SGD	Improves D-PSGD by introducing a variance-reduction mechanism to mitigate the effect of non-IID data and improve convergence speed	Local weighted averaging	Synchronous, P2P	More robust to data heterogeneity, faster convergence than D-PSGD	Dependent on graph connectivity, sensitive to non-IID data, computational overhead
FedPGA [162]	Fully decentralized FedAvg	Extends FL to decentralized settings, combines gradient slicing with adaptive step-size to reduce latency	Weighted averaging	Synchronous, P2P	Efficient bandwidth use; up to 14× training time reduction without accuracy loss	Requires careful slicing mechanism; complexity of adaptive update implementation
P2PFL [163]	Fully decentralized FL	Each node randomly selects one or more neighbors at irregular intervals to exchange model updates	Random weighted averaging	Asynchronous, gossip-based P2P	Resilience to failures, low overhead, highly scalability, support for node arrivals and departures, enhanced privacy	Possible slow convergence risk of unnecessary packet duplication, poorly connected nodes
DSM [149]	Fully decentralized SGD	Each agent minimizes its local objective function and aggregate its updates with those received from its neighbors	Weighted averaging	Asynchronous, P2P	Guaranteed convergence good balance between accuracy and iteration number, large scalability, tolerates unreliable communication	Slower convergence strong dependency on the quality of the weights (averaging matrix), high sensitivity to network topology variations
Boyd et al. [164]	Fully decentralized	At each iteration, a node randomly selects one neighbor, and then each node updates its state to the average of the two	Pairwise randomized averaging	Asynchronous, Gossip-based P2P	Easy to deploy in large and dynamic topologies, robust to partial failures, low overhead	Slower convergence, intensive communication, not calibrated for nodes with heterogeneous data
D-DistADMM [165]	Fully decentralized ADMM [166]	Linear mixing of variables via a push-sum type protocol on a directed graph	Weighted aggregation	Semi-synchronous, P2P	Guaranteed convergence, good management of local constraints and couplings, robust	High computational, communication, and storage costs, complex tuning

Table 2.1: Decentralized aggregation algorithms comparison

Gossip protocols

The gossip communication protocol (also known as epidemic) is a method for disseminating information in a distributed peer-to-peer network, inspired by

how a rumor or virus spreads within a population. In this approach, each node holds local information (such as the state of a service, an update, or the parameters of an artificial neural network model). It randomly selects a small set of neighbors and exchanges part or all of this information with them. The neighbors then update their state and, in turn, propagate the data to other nodes. By repeating this process, the information gradually spreads throughout the network without the need for central coordination. In this protocol, the same message may circulate multiple times along different paths, increasing robustness but increasing overhead [167]. The gossip approach tolerates node failures and high mobility particularly well, since propagation never depends on a single path. The gossip protocol offers several key advantages: it is simple to implement, resilient to node failures, message losses, and topological changes, and highly scalable, enabling efficient operation across large networks without central coordination [168]. These properties make gossip protocols well-suited to highly dynamic environments such as VANETs. However, it also presents limitations: redundant message lead to high communication overhead, slower and more probabilistic convergence than centralized or structured P2P protocols. Moreover, the mechanism remains vulnerable to attacks. Indeed, false or malicious information can spread rapidly when robust verification and authentication mechanisms is not implemented.

Therefore, gossip-based algorithms are better suited to highly dynamic environments with highly mobile nodes, such as VANETs. However, the lack of centralized control complicates source and data integrity verification. This creates opportunities for malicious nodes to inject false updates, creating significant security risks and necessitating the use of robust cryptographic techniques, strong authentication, and mechanisms for detecting and mitigating adversarial behaviors. In the following subsection, it is discussed the most prevalent threats targeting DFL-based VANET systems, which commonly exploit the classical P2P and gossip communication protocols. The corresponding countermeasures are presented as well.

2.2.9 . Security and Privacy in DFL-based mobility models

In DFL-based VANETs, collaborative trajectory prediction entails significant security challenges. Although DFL provides scalability, fault tolerance, and privacy preservation, its decentralized architecture also increases exposure to various attacks [169]. These attacks are generally divided into three broad categories: (i) those that aim to degrade the performance of the models of honest participants, (ii) those that compromise the confidentiality and integrity of parameters exchanged among vehicles, and (iii) those that affect the availability of VANET systems. These different attacks are discussed in the following subsections.

Attacks on Model Performance

The most common performance-related attacks include data poisoning, model poisoning, Sybil attacks, evasion attacks, and backdoor attacks [170, 171].

- *Data Poisoning Attacks:* Data poisoning attacks involve injecting misleading or malicious samples into the training dataset to degrade the performance of the model [172], for example, by modifying the labels of the training examples (e.g., changing the label "dog" to "cat"), an attack known as label flipping, the precision and recall of the global model can drop significantly even when the fraction of malicious nodes is small [173]. The effects of this attack can be amplified if it occurs in the later rounds [174].

- *Model Poisoning Attacks:* The aim of this type of attacks is to directly manipulate local model parameters before their transmission to degrade the global model performance. Such attacks can remain effective even against robust aggregation methods [175] such as Krum [176], Trimmed Mean [177, 178], or Median [178]. In decentralized settings without a central server, these attacks are more damaging, while designing effective defenses is highly complex [179].

- *Sybil Attacks:* A Sybil attack occurs when a single adversary creates and uses multiple virtual identities to simulate the existence of many distinct nodes in a decentralized system. This allows the attacker to bias redundancy, quorum, reputation, or election mechanisms and subvert the network's fault tolerance [169, 180].

- *Evasion Attacks:* This type of attacks is carried out at the time of model inference, where the adversary carefully alters the input data to compromise the machine learning model's predictions and cause wrong prediction [181].

- *Backdoor Attacks:* Backdoor attacks consist of poisoning the training data by adding a trigger (e.g., a visual pattern in an image, a word in a sentence, etc.). The model therefore operates normally on clean inputs, but when an input containing the trigger is encountered, it produces an output controlled by the attacker [182].

Attacks on Data Privacy and Integrity

This category of attacks exploits model updates or gradients exchanged between participants to extract sensitive information about the training data. The most common attacks in this category are as follows:

- *Model Inversion Attack:* Consists of a post-training attack in which an adversary uses the outputs of the AI model to reconstruct sensitive training data. For example, facial images could be reconstructed from confidence scores from a facial recognition model [183], and genetic attributes can be revealed from a medical dosing model [184]. This illustrates how even well-performing models can unintentionally disclose private information.

- *Membership Inference Attack:* Consists of a privacy attack in which an adversary aims to determine whether a particular data sample was part of the training set of a given AI model. This type of attack was first formalized by Shokri et al. [185], who demonstrated that shadow models can be used to train an attack model capable of inferring membership from the output probabilities of a target model. Moreover, these attacks are feasible not only in centralized learning, but also in collaborative and federated learning, where adversaries can exploit shared model updates to infer the participation of individual vehicles in the training process [186].

- *Man-in-the-Middle (MitM):* MitM attacks occur when adversaries intercept and manipulate communications between vehicles, RSUs, or other infrastructure. In DFL-enabled VANETs, attackers can inject poisoned gradients, alter model parameters, steal private information, or delay the global model's convergence. MitM is a key vulnerability in VANET authentication [6], and enables adversaries to bypass consensus security guarantees in blockchain-based vehicular systems if message integrity is not strictly enforced [187].

- *Hijacking Private Keys:* Private keys are critical for authentication and securing model update transmission. If adversaries succeed in hijacking or stealing these keys, they can impersonate legitimate clients, inject malicious updates, or even take control of consensus processes. Key management is a significant security weakness in blockchain systems, making hijacking a realistic threat [188]. Compromised private keys allow attackers to submit poisoned gradients under valid identities, severely degrading model integrity [188] and performance.

- *Vulnerabilities of Smart Contracts:* In blockchain-integrated VANETs, flaws in smart contracts may be exploited to disrupt transactions or extract sensitive information [169].

- *Gradient Leakage:* Gradient leakage attacks exploit the fact that gradients shared during the collaborative training process may encode information about the underlying data. The work in [189] showed high-fidelity reconstructions using gradient inversion with realistic training

setups. Such attacks reveal that gradients intended to protect raw data privacy may expose sensitive information. This makes gradient leakage one of the most critical threats in DFL.

- *Property Inference Attacks:* Property inference attacks aim to extract global properties of the training dataset [190]. When targeting a traffic prediction model, such an attack could determine whether most vehicles in the dataset are located in a rural or urban area, or if the average speed exceeds a specific limit.

- *Collusion Attacks:* Collusion attacks occur when multiple adversarial participants collaborate in FL or DFL to amplify their impact. Unlike a single malicious client, colluding adversaries can share information, coordinate poisoning strategies, and evade detection systems. Studies have shown that collusion enables powerful poisoning attacks that bypass Byzantine-resilient aggregation rules [191].

Attacks on System Availability

These attacks never directly target the model or the data, but aim to disrupt the overall functioning of VANETs, or the DFL protocol. This category covers the following attacks:

- *Replay Attacks:* Replay attacks involve capturing legitimate messages and retransmitting them later to delay the convergence of the model or disrupt the learning process. In VANETs, replayed safety beacons can create false traffic conditions [192]. In DFL setting, these attacks can trick the system into double-using stale model updates in the aggregation process.

- *Blackhole Attack:* The blackhole attack involves a malicious node presenting itself as offering the best route (or the best relay) to a destination, and then intercepting and ignoring (dropping) the packets transmitted to it. This often occurs in ad-hoc routing protocols [193].

- *Flooding / Message Spam Attack:* Flooding is a network over-provisioning attack: one or more nodes inundate the system with messages (requests, advertisements, redundant data) to overwhelm the bandwidth, queues, or CPU of the receiving peers. When carried out by a single sender, it is a local denial-of-service (DoS) attack (flooding); when conducted by many coordinated sources, it becomes a distributed denial-of-service (DDoS) attack [194].

- *Malformed Messages:* Sending intentionally malformed, corrupted, or specially constructed messages aims to exploit parsing bugs, memory

management errors, or protocol assumptions to cause crashes, memory leaks, or indeterminate states [195].

Ensuring the security, privacy, and reliability of VANETs systems requires careful attention to network topology and protection against attacks and vulnerabilities [15], while maintaining tolerance to frequent connection interruptions caused by high mobility [172]. These security measures should pursue two main objectives: (i) mitigating external threats and (ii) mitigating internal threats.

Security Against External Threats

External threats originate from entities that do not have legitimate access to the system and aim to disrupt, spy on, or sabotage it from the outside. Various mechanisms can be implemented to reduce their impact and improve the resilience of the system against such attacks. The most frequent and most effective among these countermeasures are the following:

- *Digital Signatures:* In FL- or DFL-based VANETs, digital signatures ensure the authenticity, integrity, and non-repudiation of local updates exchanged between vehicles. Each vehicle digitally signs its gradients or parameters before transmission, and peers verify the provenance before aggregation. Schemes such as the Elliptic Curve Digital Signature Algorithm (ECDSA), which conforms to the IEEE 1609.2 standard for V2V communication security, or variants better suited to massive parameter exchanges like BLS (Boneh–Lynn–Shacham) aggregated signatures [196, 197], offer a good trade-off between security and efficiency, thus strengthening resilience against poisoning or model replacement attacks [196, 198].

- *Cryptographic Hash Functions and MACs:* Hash functions (SHA-256 [199], SHA-3 [200]) and Keyed-Hash Message Authentication Codes (HMACs) [201] ensure the integrity and lightweight authenticity of exchanges. A vehicle can generate a hash or a MAC to quickly detect altered updates before aggregation process [202]. These mechanisms can be combined with asymmetric signatures for enhanced verification [198].

- *PKI and Certificate Management:* Public Key Infrastructure (PKI) and certificate management enable the assignment of verifiable identities to the participants vehicles and the effective handling of revocations. Certificates, issued by an authority or a distributed ledger, bind a public key to a pseudonym, facilitating signature verification and the exclusion of compromised nodes [203, 198].

- *Identity-Based Cryptography Schemes:* Identity-Based Cryptography (IBC) approaches use an identity as the public key, removing the need for explicit certificates and simplifying peer-to-peer exchanges in a highly distributed environment [204].
- *Key Establishment (ECDH):* Elliptic Curve Diffie-Hellman (ECDH) key exchange allows two vehicles to derive a shared symmetric key, which is then used to encrypt communications or generate MACs [198]. Being lightweight and efficient, it is suitable for the constraints of onboard units.
- *Changing Pseudonyms:* Changing (or rotating) pseudonyms ensures privacy and prevents tracking vehicles while retaining a verifiable link between identity and behavior. It enables local updates to be signed anonymously while maintaining controlled traceability [198, 203].
- *Secure Routing Protocols:* Secure routing protocols ensure the integrity and reliability of multi-hop exchanges, preventing route manipulation or the injection of fraudulent messages. They contribute to the network's resilience against wormhole [205], blackhole [206], or Sybil attacks [207].

Security Against Insider Threats

An insider threat comes from an authenticated but malicious participant, such as a vehicle, Road Side Unit (RSU), or aggregator node, whose goal is to disrupt the learning process, manipulate routing, or falsify the disseminated AI model parameters. The following solutions are used to counter these threats:

- *Robust Aggregation:* Securing the aggregation process is critical in the context of FL, or DFL applied to VANETs. Robust aggregation methods aim to make aggregation resilient against abnormal or byzantine behaviors, where some nodes send corrupted or poisoned updates to degrade the global model performance. The principle is to filter, ignore, or reduce the influence of suspicious contributions. Among the most common methods, Krum [176] selects the most consistent update with its $n - f - 2$ nearest neighbors (where n is the total number of nodes and f is the maximum number of byzantine nodes). Multi-Krum [176] applies the same criterion to several updates and averages them. Median [178] computes the coordinate-wise median, Trimmed Mean [177, 178] removes extreme values before averaging, while FoolsGold [191] detects coordinated attackers by identifying similar gradients among them. However, these approaches rely on an upper bound on the fraction of adversarial nodes and may lose effectiveness if this assumption is violated. In addition, strongly non-IID data distributions can

weaken their robustness [208], and their implementation often incurs higher computational and communication costs and potentially slowing convergence.

- *Reputation-Based Security Methods:* Reputation systems assign each vehicle a trustworthiness score based on its past contributions. The updates from low-reputation nodes are assigned lower weights during the aggregation process, allowing the system to mitigate the effect of malicious nodes [196].

- *Differential Privacy (DP):* This approach adds calibrated noise either to model parameters (or gradients) [209] or to the training data [210] to prevent the reconstruction of the data used to train the shared models between vehicles. Its decentralized variant, Local Differential Privacy (LDP) [211], is widely adopted in practice (e.g., by Google and Apple). LDP allows each node to add noise locally to its data or model updates, ensuring privacy without relying on a central aggregator [210]. Thus, DP guarantees strong protection for individual data. However, adding noise degrades model accuracy, and the trade-off between privacy and utility is often complex to tune. Furthermore, DP may increase computational and communication costs, particularly for tracking the privacy budget [212].

- *Homomorphic Encryption (HE):* HE secures local model training by allowing operations such as addition, multiplication, and aggregation to be performed directly on encrypted data, without the decryption process. It is commonly applied in VANETs to protect model updates [213], or inter-vehicle communications, even under frequent pseudonym changes, where re-encryption mechanisms refresh pseudonyms without additional key exchanges [214]. Additive HE, which supports only addition operations, is often used to reduce computational costs. For instance, in [215], the authors employ additive HE to preserve parameter confidentiality without relying on trust scores that might reveal vehicle trajectory or identity. Therefore, while HE provides strong privacy guarantees, it incurs significant computational overhead.

- *Secure Multi-Party Computation (SMPC):* SMPC is a cryptographic technique that enables several entities to jointly perform a computation (e.g., training a machine learning model) without disclosing their respective data [216, 217]. Each party randomly divides its data into secret shares, which it then distributes to all other participants. Original data can only be reconstructed by combining a sufficient number of shares. Thus, no single participant can recover the initial information from the sole fragment it has received [218, 217]. Once the collaborative

computation is completed, each participant only combines the shares they received to reconstruct the final result. [169]. This approach is relevant when collaboration is required, but direct data sharing is undesirable (owing to legal, ethical, competitive, or security constraints). However, it incurs high communication costs (due to the large number of exchanges between participants) and computational overhead (fragmentation, random value generation, recombination), which limits its scalability and makes it poorly suited to large-scale scenarios [217]. Furthermore, SMPC protocols often prevent verification of the validity of transmitted updates and need to be combined with DP to strengthen the privacy [169].

- *Anomaly Detection:* Anomaly or intrusion detection systems [219, 220] aim to identify malicious updates or nodes directly. For example, FLDetector [221] relies on temporal inconsistencies between successive updates to flag suspicious clients, whereas LA-DETECTS [222] is based on plausibility and consistency checks for mobility data such as position, speed, and acceleration. These methods can neutralize adversarial behavior before it impacts the global model. However, they often require storing update histories or metadata, with risks of false positives and additional computational overhead.

- *Blockchain:* Blockchain-based Federated Learning (BFL) approaches [223, 224] rely on distributed ledger technologies (DLT) [225] to ensure trust, transparency, and traceability throughout the collaborative learning process. Each model update can be recorded on the blockchain, providing an immutable audit trail that prevents tampering or injection of falsified updates [224]. Smart contracts may be used to automate aggregation policies, validate participant legitimacy, and implement incentive reputation mechanisms, thereby reducing reliance on a central server [213]. In vehicular settings, such decentralization enhances model reliability among mobile participants; however, blockchain integration introduces significant computation and communication overhead, which can affect scalability and latency [226].

The table 2.2 summarizes the study on the privacy and security algorithms in FL and DFL -based frameworks for VANETs.

2.3 . Conclusion

This literature review highlights the growing interest in VANETs. Recent studies consistently demonstrate that AI-driven mobility models outperform traditional physics-based approaches in terms of accuracy, flexibility, and scalability, while remaining easier to implement. Among these advances, FL-based

models stand out for their ability to improve local models through peer collaboration without sharing raw data, thus enhancing accuracy, privacy, and communication efficiency. The fully decentralized FL, further strengthens these benefits by eliminating the need for a central coordinator. In addition, gossip-based peer-to-peer communication protocols provide greater resilience to high mobility, intermittent connectivity, and scalability challenges, while balancing computational load and reducing communication overhead. However, this openness also expands the attack surface, exposing the network to both internal and external threats.

Method	Mitigated threat	Threat group	Protected assets	Counter measure	Comm. type	Pros	Cons
Chen et al. [227]	Byzantine and model-poisoning	Internal	Convergence and model integrity	Randomness generation and encrypted secrets sharing	V2V	Resilience to Byzantine nodes, fully distributed collaborative learning	Computation and communication costs, possible slower convergence
Ali et al. [228]	Eavesdropping, spoofing, forgery, message tampering	External	Privacy, authentication, and integrity of messages	Unified Signature and encryption scheme based on bilinear pairings	V2V/V2I	Single primitive for auth+confidentiality; provable security properties	Computation and communication costs
Wang et al. [229]	Privacy leakage and model poisoning	Internal and external	Model privacy and reliability,peers trustworthiness	Secret-sharing; encryption mechanisms and trust; reputation scheme	V2V	Improved privacy and trust; robustness to malicious peers	Computation and communication costs, possible slower convergence, complexity; overhead for reputation maintenance
Tang et al. [230]	Data and model poisoning; leakage during aggregation	Internal and external	Confidentiality of local data and model updates	Functional encryption	V2I	Strong confidentiality; secure computations on ciphertext	High computation cost latency, and complexity; SPF risks
Lv et al. [209]	Data poisoning , privacy leakage	Internal	Model integrity; participant accountability; audit trail	FL and blockchain-based data-centric MDS; Differential Privacy	V2V	Traceability, tamper-evidence, privacy protection; no-dependent to a central authority	Blockchain consensus computation, communication and storage overhead; latency and scalability limits
Gao et al [210]	Model poisoning	Internal	Privacy of updates; reliability and robustness of global model	Encryption, Differential privacy	V2V	Balances privacy and reliability; resilient aggregation	Communication and computation costs; reduced accuracy
Sultana et al. [222]	Data poisoning, replay, DoS, Sybil attacks	Internal	Integrity, consistency and plausibility of received update	Local, adaptive data-centric MDS	V2V/V2I	Lightweight, local decision, fast detection; less reliance on global trust	Sensitive to non-IID data; higher false positives rate; may need calibration
NK Prema [214]	Eavesdropping and inference from aggregated updates; honest-but-curious servers	Internal and external	Confidentiality of model updates and aggregated results	Secure aggregation via Fully Homomorphic Encryption (FHE)	V2I	Strong confidentiality (compute on encrypted data), raw data privacy	Very high computation cost, large ciphertext sizes, latency; scalability limits
Sharma and Liu [231]	Data poisoning and position forgering attacks	Internal	Plausibility and Integrity of received messages	ML-based data-centric MDS	V2V/V2X	Autonomous, real-time response MDS, improved accuracy	Stealthy attacks evade the MDS
Gyawali et al. [215]	privacy (feedbacks scores, and messages) leakage	Internal	Integrity of alerts messages; privacy of vehicle data	Additive Homomorphic Encryption and node centric MDS	V2V/V2I	Protects privacy while enabling detection; reduces raw data exposure	Added cryptographic process overhead; reliance on aggregators
Ghaleb et al. [232]	Mobility (position, speed) data forgering, data poisoning	Internal	Integrity and reliability of safety messages	Local, context-aware data-centric + node features MDS	V2V	Improved detection accuracy, multi-feature detection MDS	More complex feature extraction; may be computationally heavier; tuning needed
Zhang et al. [233]	Messages authenticity and integrity, vehicles anonymity	Internal	Vehicles indentities, authencity of exchanged messages	k-anonymity, cryptographic signatures, RSU-aided message authentication scheme	V2V/V2I	Strong authentication with RSU support; efficient verification, reduced message loss ratio, reduced latency	Requires infrastructure (RSUs); certificate and key management overhead, attack evasion risk in RSU-less mode. No misbehaving vehicles revocation
Asad et al. [213]	Model inversion and membership inference attacks, data privacy. Model poisoning, tampering, non-repudiation issues, trust deficits	Internal and external	Vehicles data privacy; integrity of model updates; traceability (non-repudiation)	Blockchain FL, and Homomorphic Encryption	V2V/V2I	Immutable audit trail, decentralized trust, incentive enforcement	High consensus computation and communication overhead; latency; storage costs
Mansouri et al. [220]	Intrusion, misbehavior, data tampering	Internat and external	Model updates integrity	Blockchain and FL based IDS	V2I	Distributed detection, tamper-evident logging, privacy via FL	Blockchain and FL overhead; synchronization and real-time constraints
Liu et al. [234]	Model poisoning attacks, hitchhiking, session key attacks; impersonation, privacy leakage	Internal and external	Local model updates privacy, reliability and integrity; authenticity	Authentication scheme and Blockchain; Chebyshev polynomials, hash functions, cryptography	V2V/V2I	Privacy, authenticity and traceability	May weaken cryptographic strength for efficiency; still some overhead and complexity

Table 2.2: Security and privacy algorithms comparison

To mitigate these issues, researchers have explored several privacy-preserving mechanisms, including cryptographic methods such as HE and digital signatures, as well as privacy-enhancing techniques like DP and Blockchain. Despite their benefits, these approaches often introduce significant computational and communication overhead or reduce model accuracy, making them less suitable for latency-sensitive vehicular environments. Misbehavior Detection Systems (MDS) are also widely employed to protect vehicular networks. MDS solutions are generally classified as node-centric or data-centric. Node-centric approaches assess the trustworthiness of nodes based on behavioral patterns, communication history, or reputation scores, whereas data-centric approaches evaluate message consistency and plausibility. Some studies suggest combining both paradigms into hybrid detection frameworks. Integrating MDS with blockchain improves traceability and non-repudiation, while coupling them with model auditing further enhances aggregation reliability. Nevertheless, these methods are often resource-intensive and face scalability limitations, highlighting the need for lightweight, adaptive, and fully distributed detection strategies suited to the dynamic and real-time nature of VANETs.

The following chapters present the solutions we propose to enhance the accuracy of collaborative mobility prediction models while minimizing computational and communication costs. We first introduce an FL-based collaborative trajectory prediction model, followed by a fully decentralized FL-based framework for mobility prediction with an optimized communication algorithm. Finally, a dedicated chapter details the proposed security and privacy method to strengthen confidentiality, integrity, availability, and efficiency in VANETs.

3 - FedVANET-TP: A Federated Trajectory Prediction model for VANETs

3.1 . Introduction

Recent advances in Artificial Intelligence (AI), particularly in Machine Learning (ML) and Deep Learning (DL), have opened new perspectives for designing intelligent, collaborative, highly effective, and adaptive mobility models. These models can learn directly from environmental and traffic data while adapting to local variations in context and network conditions.

In this chapter, a mobility model, termed FedVANET-TP (A Federated Trajectory Prediction Model for VANETs), is introduced. This model utilizes these AI-based approaches and is deployed on a distributed architecture based on Federated Learning (FL). Specifically, an advanced Deep Learning architecture, the Transformer, was selected for the trajectory prediction tasks of the proposed framework. This choice is motivated by the Transformer's efficiency in handling long-term spatial and temporal dependencies in the training data in a parallel manner, without the significant and cumulative errors observed with LSTM networks as sequence length increases. Thus, the proposed framework is designed to use the Transformer's capabilities while ensuring collaboration among vehicles to improve the accuracy of their respective local trajectory predictions without sharing their raw training data. The model was pruned to enhance computational and communication efficiency. Consequently, the proposed framework reconciles the distribution, data privacy, computational overhead, and latency constraints required for vehicular environments.

The remainder of this chapter is structured as follows: Section 3.2 presents the proposed framework, detailing the proposed trajectory prediction model and the proposed FL architecture. Section 3.3 presents the experimental setup and results, describing the datasets used, the loss function, the evaluation metrics, the simulation environment, and the results obtained. Finally, Section 3.4 concludes the chapter.

3.2 . Proposed FedVANET-TP framework

The proposed mobility framework, illustrated in Figure 3.1, aims to achieve high accuracy and low latency while minimizing computational and communication resource requirements. To meet these objectives, a lightweight neural network is employed, designed with a reduced number of layers and neurons per layer to maintain high accuracy while significantly reducing both computational and communication costs. The model is based on an FL ar-

chitecture, enabling vehicles to collaborate in improving their local prediction models without sharing their raw data, which are often large in volume and sensitive. This approach thus enhances the privacy of the updates exchanged between vehicles, while reducing the communication overhead. Specifically, the proposed system operates in a scenario comprising M vehicles moving within a VANET and collaborating through an FL framework. Each vehicle k embeds an artificial neural network model $f_k(w_k)$, where w_k represents the model parameters. This model is trained locally using the vehicle dataset D_k to perform trajectory prediction tasks. This dataset is created by combining the target vehicle's date (its spatial coordinates (x, y), velocity v_{vel}, and acceleration v_{acc}) with the identical information from up to eight neighboring vehicles. This combination forms the input features, denoted as X^k. The dataset also contains the labels Y^k, which correspond to the predicted future coordinates $(\tilde{x}, \tilde{y})$ of the target vehicle. Local training, therefore, consists of minimizing a loss (or objective) function that quantifies the error between the predicted and actual values, and can be expressed as follows:

$$\min_{w_k} \mathcal{L}_k(w_k) = \frac{1}{|\mathcal{D}_k|} \sum_{(X_i^k, Y_i^k) \in \mathcal{D}_k} \mathcal{L}(f_k(X_i^k), Y_i^k) \tag{3.1}$$

Where: $\mathcal{L}$ is the objective function, $|\mathcal{D}_k|$ represents the number of samples in the local dataset, $f_k(X_i^k)$ represents the predicted coordinates $(\hat{x}_i, \hat{y}_i)$ of the local vehicle, and i is the current sample index. The training process performed locally by each vehicle is summarized in Algorithm 3.2.1.

The FL framework then coordinates vehicle communication, enabling each vehicle to transmit its local parameters to an aggregation server, which ag-

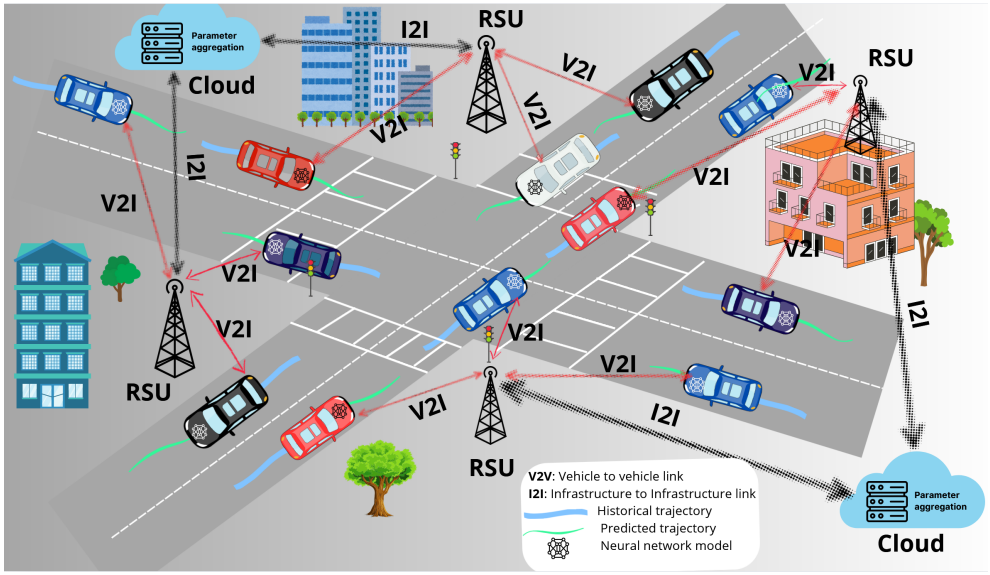

Figure 3.1: The proposed overall trajectory prediction framework architecture.

gregates them to constitute the global model and redistributes the updated global parameters to all participating vehicles. The proposed approach is therefore structured around two main components: the trajectory prediction model and the FL framework. These two components are discussed in detail in the following subsections.

3.2.1 . Proposed trajectory prediction model

To achieve our performance objectives, we opted for the Transformer model due to its inherent capacity to efficiently capture and process long-term spatial and temporal dependencies. This methodological choice is crucial, as it enables us to accurately model the complex interactions between the target vehicle and its environment, thereby ensuring accurate and realistic future trajectory predictions over an extended prediction horizon. To mitigate the high computational costs intrinsic to the standard Transformer architecture, a customization strategy is implemented. This adaptation involved pruning structural blocks deemed of minor interest in trajectory prediction tasks and reducing the number and size of layers within the blocks that are used. This optimization successfully and significantly reduced the model's computational footprint without compromising the required high precision.

In the subsequent paragraphs of this subsection, the standard Transformer architecture is first presented, followed by a detailed description of the proposed customized version specifically designed for our trajectory prediction model.

Algorithm 3.2.1 Local model training and update

Input: Local dataset D_k, local epochs number E and the learning rate η
Output: Updated local parameter w_k^{t+1}
Initialize local model parameters $w_k^t \leftarrow w_0$
for each epoch $e = 1$ to E **do**
    for each batch $(X, y) \in D_k$ **do**
        Compute predicted coordinates $\hat{y} = \text{forward}(X, w_k^t)$
        Compute loss $\mathcal{L} = \text{mse}(y, \hat{y})$
        Compute gradients $\nabla w_k^t = \text{backward}(\mathcal{L})$
        Update parameters : $w_k^{t+1} \leftarrow w_k^t - \eta \nabla w_k^t$
    end for
end for
Return w_k^{t+1}

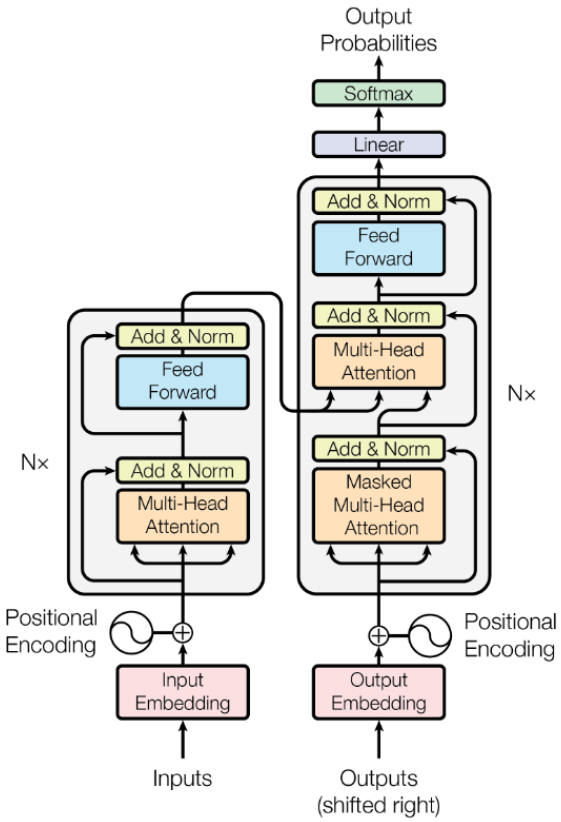

Figure 3.2: The standard Transformer model architecture. source: Vaswani et al. (2017) [1]

The standard Transformer architecture

Initially introduced by Vaswani et al. [1] for natural language processing (NLP) tasks, the Transformer model employs an encoder–decoder architecture, a traditional approach used in machine translation models. It extends this framework to sequence prediction by replacing recurrent mechanisms (such as RNNs or LSTMs) and convolutional operations (CNNs) with attention-based layers exclusively. As illustrated in Figure 3.2, the standard Transformer consists of the following main components:

- *Tokenization:* Tokenization is the first step in text preprocessing for Transformer-based models. It involves segmenting raw text into basic linguistic units, called tokens (such as words, subwords, or characters), to make the input interpretable by the model. Each token is then mapped to a numerical identifier from a predefined vocabulary, producing a sequence of integer indices that serves as the model's input. Modern Transformer-based models such as BERT [235] and GPT [236] employ subword tokenization, which efficiently handles rare words while maintaining a

manageable vocabulary size. Tokenization occurs prior to embedding and positional encoding, forming the entry point of the Transformer. It plays a crucial role in determining the quality of learned representations and the model's overall performance.

- *Embedding blocks:* Prior to processing, input words or tokens are converted into continuous vector representations using a dedicated embedding layer. This step projects each token into a dense vector space where semantic similarities are preserved. The encoder and decoder utilize separate embedding layers. Furthermore, embeddings are scaled by $\sqrt{d_{\text{model}}}$ (where d_{model} is the dimension of word vectors) prior to the addition of positional encoding. This scaling is vital to balance the magnitudes of the values and stabilize the training process.

- *Positional Encoding (PE):* Unlike recurrent networks, the Transformer processes all input tokens in parallel, thus lacking an inherent notion of sequence order. To inject positional information into the sequence, a positional encoding is added to each embedding vector. These encodings are derived from sinusoidal functions of different frequencies, defined by

$$PE(pos, 2i) = sin(\frac{pos}{10000^{\frac{2i}{d_{\text{model}}}}}) \tag{3.2}$$

and

$$PE(pos, 2i+1) = cos(\frac{pos}{10000^{\frac{2i}{d_{\text{model}}}}}) \tag{3.3}$$

where pos is the token position and i is the dimension index. This formulation enables the model to learn relative positional relationships while preserving natural periodicity, thereby improving generalization to sequences longer than those encountered during training.

- *Encoder :* As defined in [7], the encoder block consists of $N = 6$ identical layers of artificial neurons, each comprising two sub-layers: the first sub-layer is a multi-head self-attention mechanism, which dynamically weights the positions in the input sequence to determine which are most relevant for each word. The second is a fully connected feed-forward network, which processes each word independently while maintaining the sequence order. A residual connection [237] is applied around both sub-layers, and its output is then passed through a layer normalization step [238]. These mechanisms contribute to stabilizing the learning process and facilitating the propagation of gradients. To ensure consistency of residual connections, all sub-layers, as well as the embedding layers, produce output vectors of dimension $d_{\text{model}} = 512$ (as defined in [7]).

- *Decoder :* Also comprising six identical layers, the decoder extends the encoder's architecture by integrating a multi-head cross-attention sublayer. This component allows the decoder to contextualize its predictions by integrating the sequence representations produced by the encoder. Moreover, the self-attention mechanism within the decoder is specifically adapted to prevent each word from attending to subsequent words. This masking process is combined with a one-position shift of the output representations to ensure that each word is processed by attending only to preceding tokens. This design enables the model to generate each word sequentially, using information from the previously predicted outputs to guide the current prediction.

- *Attention mechanisms:* The attention mechanism is a function designed to extract relevant contextual information from a sequence. It determines, for a given element, which other parts of the sequence are most relevant for its processing. Three distinct vectors represent each element: a query (Q), which expresses what the element seeks to understand in context; a key (K), which characterizes the informational content of other elements in the sequence; and a value (V), which contains the actual information to be retrieved. Attention scores (similarities between queries and keys) are used to weight the values, and the final output is computed as a weighted sum of these values. The operations of the attention mechanism are based on two main components: Scaled Dot-Product Attention and Multi-Head Attention, which are described below.

  - *Scaled Dot-Product Attention:* This attention mechanism enables the model to evaluate the contextual relevance among different elements within a sequence, such as between words in a sentence. It dynamically weights the sequence information according to its contextual importance, determining which parts of the input sequence should receive the most focus when processing a given element. In practice, the mechanism computes the similarity between a query Q and each key K_i using a dot product:

$$score_i = Q \cdot K_i^T \tag{3.4}$$

    The scores are then scaled by dividing them by the square root of the key dimension

$$score_i = \frac{Q \cdot K_i^T}{\sqrt{d_k}} \tag{3.5}$$

    The scaled scores are subsequently passed through a softmax function [239] to obtain probability weights.

$$weights_i = softmax(\frac{Q \cdot K_i^T}{\sqrt{d_k}}) \tag{3.6}$$

Finally, these weights are used to combine the corresponding values V_i, resulting in a weighted aggregation of the most contextually relevant information.

$$Attention(Q, K, V) = softmax(\frac{Q \cdot K^T}{\sqrt{d_k}}) \cdot V \tag{3.7}$$

- *Multi-Head Attention:* The Multi-Head Attention mechanism enhances the core principle of scaled dot-product attention by allowing the model to capture diverse contextual relationships in parallel. Instead of a single operation across the entire representation space, the mechanism linearly projects the query (Q), key (K), and value (V) vectors into multiple lower-dimensional subspaces. Each subspace forms an independent attention head that executes its own scaled dot-product attention. This parallel processing capability allows several heads to focus on distinct contextual aspects, for example, capturing syntactic dependencies, semantic meanings, or long-range relationships, simultaneously. The outputs from all heads are then concatenated and subjected to a final linear transformation to generate the block's output representation.

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h) \cdot W^0, \tag{3.8}$$

where

$$head_i = Attention(Q \cdot W_i^Q, K \cdot W_i^K, V \cdot W_i^V), \tag{3.9}$$

with W^Q, W^K, W^V and W^O the query, key, value and output learning matrices respectively. The original Transformer architecture introduced in [7] employs $h = 8$ parallel attention heads, each working on vectors of reduced dimensionality ($d_k = d_v = \frac{d_{\text{model}}}{h} = 64$).

This setup enables the model to jointly attend to multiple representation subspaces while maintaining a computational cost comparable to a single, full-dimensional attention layer.

- *Feed-Forward Networks (FFN):* Each layer in both the encoder and the decoder blocks incorporates a fully connected feed-forward network, which follows the attention mechanism. This network consists of two

linear transformations separated by a non-linear activation function, typically ReLU [240] or GELU [241]:

$$FFN(x) = max(0, xW_1 + b_1) \cdot W_2 + b_2 \tag{3.10}$$

Where max is the activation function (ReLu or GELU), x is the input vector for a given position, with dimension d_{model}, b_1 is the bias vector of the first layer, with dimension d_{ff}, W_1 is the weight matrix of the first linear layer, with dimensions d_{model} x d_{ff}, W_2 is the weight matrix of the second linear layer, with dimensions d_{ff} x d_{model}, and b_2 is the bias vector of the second layer, with dimension d_{model}. This component enables the model to learn abstract and non-linear representations from the information captured by the attention mechanism, thereby enhancing its expressive capacity and allowing it to model complex dependencies between token features.

- *Output layer:* The softmax function is applied at the final stage of the decoder during output generation. After the last decoder layer, the model outputs a vector of logits (unnormalized probability scores) with a dimension equal to the vocabulary size. The softmax function converts these logits into normalized probabilities:

$$P(y_i) = \frac{e^{z_i}}{\sum_j e^{z_j}} \tag{3.11}$$

Where z_i is the score for token i. This step allows the model to assign a probability to each word to be the next output token in the generated sequence, with the most probable token is selected as the output.

The customized Transformer architecture

The proposed model is based on a customized and optimized Transformer architecture for trajectory prediction within vehicular networks. Unlike the standard Transformer, this customized version is specifically designed to enhance predictive performance on numerical sequential data while reducing computational and communication costs to meet the latency and resource constraints inherent to vehicular networks. To achieve this objective, several architectural adaptations were implemented: (i) The tokenization process was removed, as historical trajectory data are inherently numerical and do not require transformation into discrete word tokens. (ii) The number of stacked layers in both the encoder and decoder was reduced to a single layer instead of six, as used in the original Transformer, thereby reducing model complexity. (iii) The dimensionality of the internal sub-layers was significantly reduced

to minimize computational overhead while maintaining the model's ability to capture relevant temporal dependencies.

The overall architecture of the proposed Transformer-based trajectory prediction approach, illustrated in Figure 3.3, consists of four main components: an input embedding block, an encoder, a decoder, and an output layer, which are described in detail below.

- *Input Embedding Block:* The input embedding block transforms the raw numerical trajectory features into a dense latent representation suitable for attention-based processing. Each input frame, composed of features such as position, velocity, and acceleration, is passed through two fully connected layers with nonlinear activations (ReLU and GELU), and L2 regularization. Batch normalization is applied after each layer to stabilize training and accelerate convergence. This block maps the input features to an embedding space of dimension $d_{\text{model}} = 32$, which serves as the foundation for subsequent attention computations. To preserve the temporal order of the sequence, positional encodings are generated using sinusoidal functions and added to the embeddings.

- *Encoder Block:* The encoder processes the embedded input sequence to extract high-level contextual representations. It consists of a multi-head self-attention mechanism, followed by a feed-forward network with non-linear activation. Each sub-layer is surrounded by a residual connection and layer normalization, ensuring stable gradient propagation and efficient learning. In this customized version, the encoder employs only one layer with two attention heads, each having a key dimension (d_k) of 8, and a feed-forward network dimension of 64. This design drastically reduces computational complexity while retaining the

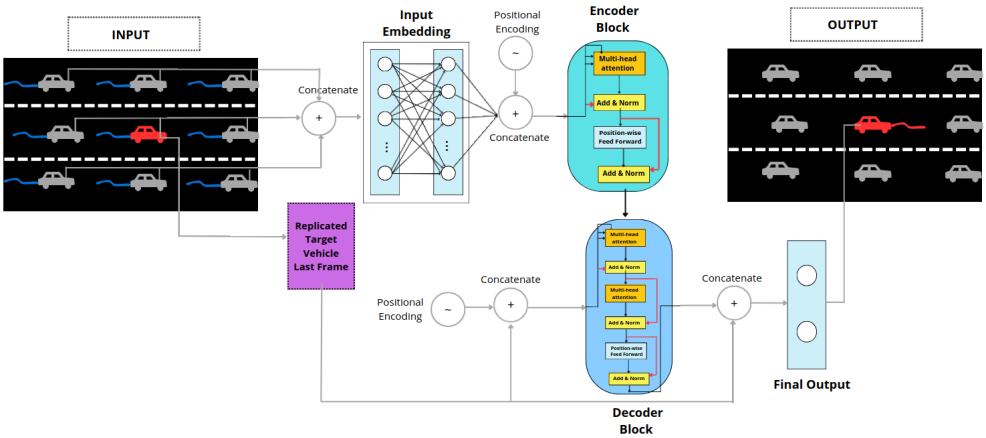

Figure 3.3: The proposed Transformer-based trajectory prediction model architecture.

model's ability to capture dependencies across the input frames.

- *Decoder Block:* The decoder takes as input both the encoded representation and the last observed position, speed, and acceleration of the target vehicle in order to predict its future positions. This block integrates two attention mechanisms: a self-attention mechanism, allowing the decoder to process previously predicted frames while masking future positions to maintain autoregressive properties; and a cross-attention mechanism, which aligns the decoder's internal representation with the encoder's output, enabling the model to leverage the contextual information of the input sequence. Each attention sub-layer is followed by dropout, residual connections, and layer normalization. The decoder employs a single layer with two attention heads and an embedding dimension of 32, which is consistent with the encoder.
- *Output Layer:* The final output layer aggregates the decoder's representations to generate continuous predictions of the target vehicle's future positions. After flattening the decoder's last hidden state, a fully connected layer with linear activation produces the output vector of dimension 2, corresponding to the predicted $\hat{x}$ and $\hat{y}$ coordinates.

This prediction model is trained locally on each vehicle by using Algorithm 3.2.1. In this chapter, we adopted an FL architecture to enable vehicles to collaborate during the training of their local models. This approach enhances the accuracy of predictions while ensuring the privacy of the data, which is often sensitive in vehicular environments. To implement this FL architecture, the Flower AI framework [14] was selected, which provides a flexible and efficient infrastructure for federated learning architectures implementation. The Flower framework and the adopted FL architecture are presented in the following subsection.

3.2.2 . Proposed Federated Learning architecture

The proposed FL architecture is built on the principle of decentralized collaboration among multiple entities, referred to as clients. These clients jointly train a global model without exchanging their local data. As illustrated in Figure 3.1, in the proposed framework, each vehicle acts as a client, maintaining its own trajectory dataset and locally training the customized Transformer-based prediction model. A central aggregation server, located in the cloud and accessible through Road-Side Units (RSUs), coordinates the overall training process.

Federated Learning Process

The FL process follows an iterative training cycle composed of the following steps:

1. *Global model initialization:* The central server initializes the global model with random parameters.

2. *Global model distribution:* At the beginning of each communication round, the server randomly selects a fraction of the connected vehicles (50% in our case). It sends them a fit message containing the current global model parameters and the local training configuration to initialize the training process. This partial participation strategy is adopted to limit computational costs and communication overhead, as involving all clients in every round would lead to significant synchronization latency without a proportional gain in global model performance.

3. *Local model training:* Upon receiving the fit message, each selected vehicle initializes its local model with the global parameters and performs local training using its private dataset.

4. *Local model update transmission :* After completing local training, each vehicle sends its updated model parameters to the central server.

5. *Model Aggregation:* The server aggregates updates received from the selected vehicles using the FedAvg algorithm proposed by McMahan et al.[7], the most widely used aggregation method in federated learning. The global model parameters are updated as:

$$w_{t+1} = \sum_{k \in S_v} \frac{n_k}{n} \cdot w_t^k \tag{3.12}$$

$$n = \sum_{j \in S_v} n_j \tag{3.13}$$

   where:

   - w_{t+1} denotes the global updated parameters,
   - w_t^k represents the local updated parameters of vehicle k,
   - S_v is the number of selected vehicles in the current round,
   - n_k denotes the number of local data samples belonging to vehicle k, and
   - n represents the total number of samples across all selected vehicles.

6. *Model evaluation:* The server assesses the performance of the aggregated global model by issuing an evaluate message containing the current global model parameters to a subset of vehicles, which compute evaluation metrics on their local validation data and return the results.

7. *Repetition until convergence:* Steps (2)–(6) are repeated until a predefined convergence criterion is met.

   The analysis of the results from several ablation studies concerning the convergence of the proposed trajectory prediction model allowed us to identify the minimum number of local training iterations required to achieve satisfactory convergence, characterized by a low variation in the evaluation metrics. On this basis, a convergence criterion of 20 communication rounds is adopted, which represents the best compromise between computational efficiency and prediction accuracy.

Flower federated AI framework

To implement the proposed FL architecture, the Flower framework introduced by Beutel et al. [14] is used. Flower is an open-source platform specifically designed for research and deployment in federated learning systems. It provides high flexibility in client configuration, customization of aggregation strategies, and communication management between the central server and participant clients. Thanks to its modular design, Flower integrates seamlessly with deep learning libraries such as TensorFlow, PyTorch, and scikit-learn, while supporting heterogeneous environments where clients may have differing data distributions and computational capacities, without having to handle the complex infrastructure and communication logic. In this chapter, Flower was used to coordinate distributed training among vehicles, synchronize model weights, and execute the FedAvg aggregation strategy.

The experimental setup, along with the results of the conducted experiments, is presented in the following section.

3.3 . Experimental setup and results

This section presents the dataset used to train the prediction model, the loss function, the evaluation metrics, the implementation tools, and the experimental outcomes. The state-of-the-art approaches used for comparison are also described.

3.3.1 . Dataset

We used publicly available real-world datasets, namely the NGSIM US-101 and NGSIM I-80 datasets [242], to train the proposed trajectory prediction model. Each dataset consists of three 15-minute recordings, totaling 45 minutes, and representing different highway traffic conditions: light, moderate, and heavy. Sampled at 10 Hz, both datasets contain over $1,048,570$ entries distributed across 18 columns. From each dataset, the unique vehicle identifiers, provided in the datasets, is utilized to extract specific subsets corre-

sponding to the vehicles selected for our experiments. The choice of vehicles was influenced by the amount of individual data available per vehicle. Accordingly, for each selected vehicle, a subset focusing on the target vehicle and its immediate neighboring vehicles is extracted, considering the following four features: the local lateral and longitudinal coordinates ($Local\_X$ and $Local\_Y$), the velocity (v_Vel), and the acceleration (v_Acc). To better support long-term prediction, each subset was downsampled to 5 Hz. The resulting dataset was then split into training, validation, and test sets with a ratio of 70%, 15% and 15% respectively.

3.3.2 . Loss function

For optimizing the proposed Transformer-based trajectory prediction model, the Mean Squared Error (MSE) is employed as the loss function. MSE is widely used in regression tasks because it effectively measures the discrepancy between predicted and actual values. It is defined as:

$$MSE = \frac{1}{|D_k|} \cdot \sum_{i=1}^{|D_k|} (y_i - \hat{y}_i)^2 \tag{3.14}$$

Where $|D_k|$ denotes the number of samples of the dataset of the vehicle k, y_i the ground truth, and $\hat{y}_i$ the predicted values of the sample i. This function penalizes large deviations more heavily, making it particularly suitable for trajectory prediction tasks where minimizing significant spatial errors is crucial. Thus, minimizing MSE leads the model to produce more accurate and stable predictions of future vehicle positions.

3.3.3 . Evaluation metrics

To assess the quality of the trajectories predicted by our Transformer-based model, several metrics, commonly used in trajectory prediction methods, are employed.

- *Average Displacement Error (ADE):* The ADE measures the mean error between all predicted positions and the ground-truth positions across the entire sequence:

  $$ADE = \frac{1}{T} \cdot \sum_{t=1}^{T} ||\hat{y}_t - y_t|| \tag{3.15}$$

  where T denotes the temporal sequence length, $\hat{y}_t$ is the predicted position at time step t, and y_t is the corresponding ground-truth position. This metric reflects the overall accuracy of the predicted trajectory over the full prediction window.

- *Final Displacement Error (FDE):* The FDE focuses specifically on the final position of the predicted sequence, measuring the distance between the last predicted point and the last ground-truth point:

$$FDE = ||\hat{y}_T - y_T|| \tag{3.16}$$

This metric is particularly important for evaluating the prediction accuracy at the prediction horizon, where errors can accumulate.

- *Root Mean Squared Error (RMSE):* The RMSE extends the Mean Squared Error to all positions within the sequence:

$$RMSE = \sqrt{\frac{1}{T} \cdot \sum_{t=1}^{T} (\hat{y}_t - y_t)^2} \tag{3.17}$$

RMSE is sensitive to large deviations, making it useful for identifying predictions with significant errors at specific points in the trajectory.

- *Accuracy based on FDE:* Finally, the fraction of correctly predicted trajectories is also evaluated, considering a prediction to be correct if its FDE is below a given threshold δ:

$$Accuracy = \frac{1}{N} \cdot \sum_{i=1}^{N} 1(||\hat{y}_T^i - y_T^i|| < \delta) \tag{3.18}$$

where N is the number of trajectories considered, and $\mathbf{1}(\cdot)$ is the indicator function. This metric provides an intuitive measure of the proportion of reliable trajectory predictions within a specified spatial tolerance.

Together, these metrics enable us to evaluate both the overall accuracy of predicted trajectories and the precision at the prediction horizon, as well as sensitivity to large deviations and the practical reliability of predictions for applications such as autonomous driving and trajectory planning.

3.3.4 . Implementation details

In this chapter, the Federated Learning (FL) architecture is implemented by deploying a customized Flower API and using the FedAvg aggregation algorithm. The entire system was developed in Python 3.11, relying on the TensorFlow [243] and Keras [244] deep learning libraries. Local model training was performed on each vehicle using the AdamW [245] optimizer, with an initial learning rate of 0.001 that was gradually decreased according to a cosine decay schedule. The scheduler used $1,000$ decay steps, and the minimum learning rate at the end of the schedule ($alpha$) was set to 0.0001. The number of local training epochs was fixed at 20, as was the total number of communication rounds between the server and vehicles. The batch size was set to 16. All these values are the optimal values of the proposed framework,

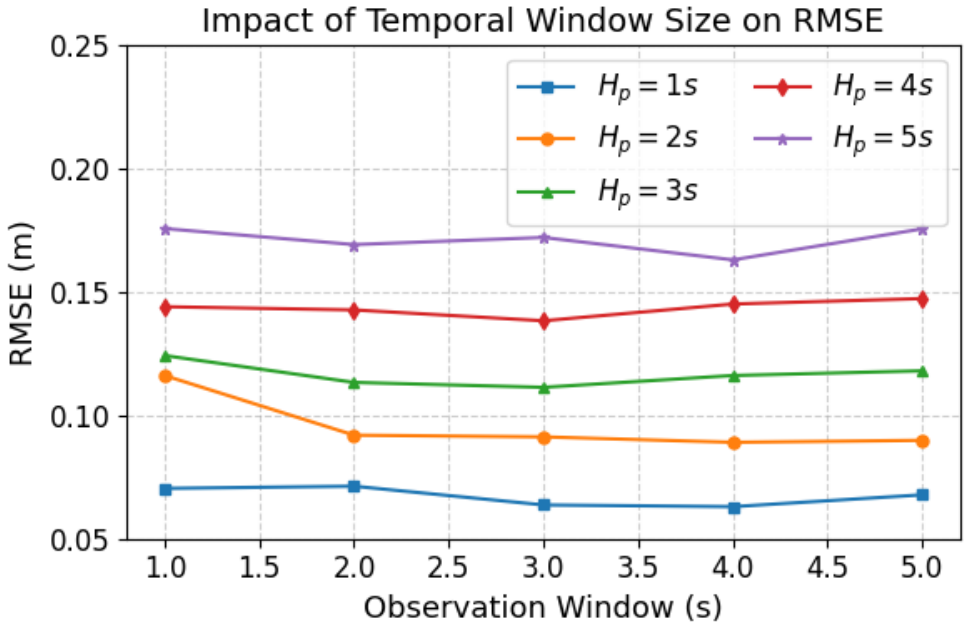

Figure 3.4: Impact of the temporal window size on prediction accuracy

Table 3.1: Trajectory prediction performance analysis on the NGSIM US-101 and I-80 datasets for different prediction horizons.

	US-101			I-80		
Prediction horizon	RMSE	ADE	FDE	RMSE	ADE	FDE
1 s	0.0632	0.0709	0.0904	0.0594	0.0635	0.0747
2 s	0.0908	0.0981	0.1323	0.0686	0.0712	0.0944
3 s	0.1140	0.1246	0.1682	0.0994	0.1137	0.1452
4 s	0.1286	0.1417	0.1843	0.1225	0.1435	0.1829
5 s	0.1458	0.1676	0.2085	0.1303	0.1502	0.2065

obtained after several ablation studies, involving tools such as GridSearchCV, from scikit-learn [246] library, and Keras Tuner [247] from the Keras library.

The entire system was implemented, and all experiments were conducted on a laptop equipped with a 13th-generation Intel Core i9 processor (20 CPU cores), 32 GB of RAM, running Ubuntu 24.04 LTS (64-bit).

3.3.5 . Simulation results

Proposed model performance was evaluated using the MSE loss function and the RMSE metric. An ablation study, illustrated in Figure 3.4, enabled us to set the observation window of historical trajectories to 3s, as this configuration yields the highest prediction accuracy (lowest RMSE) across all prediction horizons (H_p). Figure 3.5 illustrates the training and validation loss

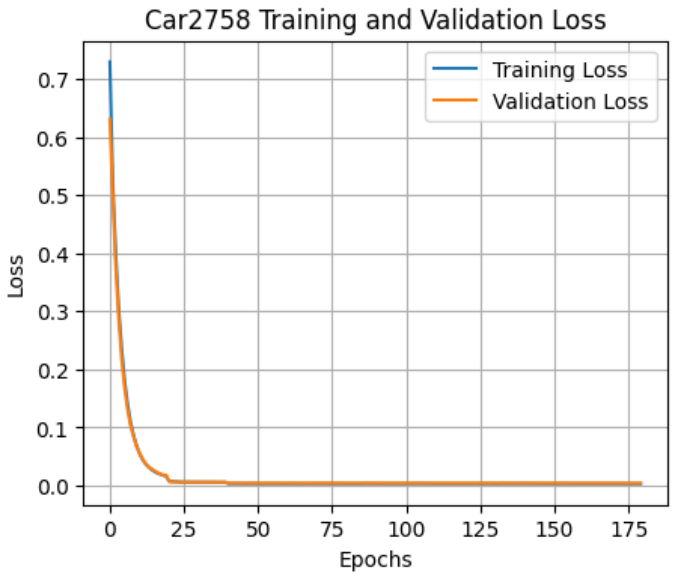

Figure 3.5: The evolution of the loss function for a representative vehicle.

Table 3.2: RMSE(m) comparison between our model and state-of-the-art methods

Dataset	Model	1s	2s	3s	4s	5s
NGSIM	iNATRan [248]	0.39	0.96	1.61	2.42	3.43
	AS-LSTM [101]	0.34	0.99	1.77	2.75	3.99
	FAHEFL [134]	0.34	0.46	0.60	0.75	-
	STA-LSTM [98]	0.10	0.20	0.31	0.43	0.56
	FedVANET-TP (Ours)	**0.06**	**0.09**	**0.11**	**0.13**	**0.14**

for a representative vehicle. The horizontal axis represents the cumulative epochs across the communication rounds in which the vehicle participated. The model begins to converge as early as the 25th epoch, reaching stability around the 40th epoch with very low MSE values. The nearly identical and very low errors shown by the training and validation curves demonstrate high predictive accuracy, rapid convergence, and minimal overfitting.

The results, illustrated in Figure 3.6, present the mean aggregated RMSE for prediction horizons ranging from 1 to 5 seconds across varying vehicle densities. Although the error increases alongside both the prediction horizon and the number of vehicles—driven by higher uncertainty and interaction complexity—it remains remarkably low. This demonstrates that the proposed approach delivers consistent, stable performance across all configurations, demonstrating its robustness across diverse vehicular scenarios. Notably, the scenario involving 5 vehicles yielded the lowest RMSE values across all pre-

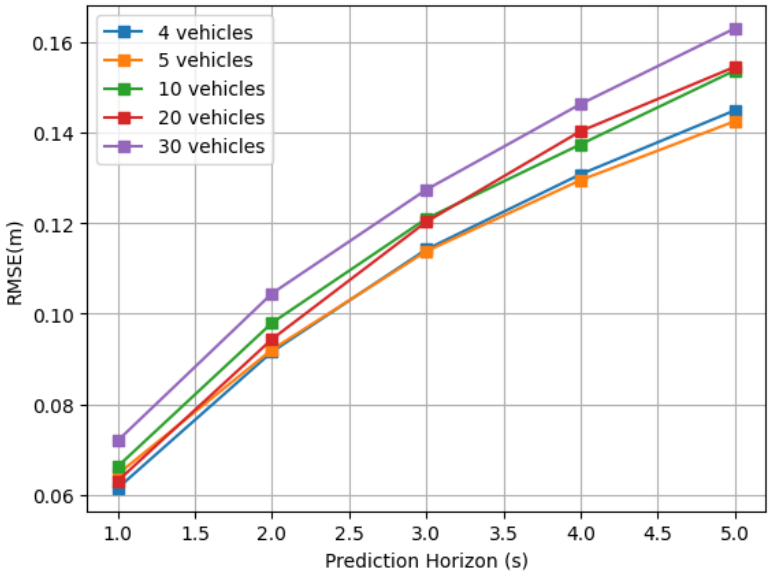

Figure 3.6: Impact of vehicle density on the mean aggregated RMSE on the FedAvg server

diction horizons (both short- and long-term). Consequently, we adopted this configuration for the remainder of the proposed model evaluations. To ensure scalability, the architecture supports the creation of 5-vehicle clusters, each orchestrated by a dedicated virtual cloud server via a local RSU. Table 3.1 reports the values of the performance evaluation metrics RMSE, ADE, and FDE of the proposed model on the NGSIM US-101 and I-80 datasets for prediction horizons ranging from 1 to 5 seconds. For short-term prediction (1 s), the model achieves high accuracy on both datasets, with RMSE values of 0.0632 m on US-101 and 0.0594 m on I-80. As the prediction horizon increases, all error metrics exhibit a gradual and expected rise, reaching maximum RMSE values of 0.1458 m and 0.1303 m at 5 s on US-101 and I-80, respectively. Despite this increase, the RMSE, ADE, and FDE values remain consistently low and closely aligned across the two datasets, indicating stable performance and strong generalization capabilities of the proposed framework across different highway scenarios. Furthermore, Table 3.2 shows that the proposed model outperforms existing methods, by achieving approximately 40% accuracy improvement at 1s prediction horizon.

Table 3.3 summarizes the communication overhead for the FedVANET-TP system. While each vehicle incurs a minimal cost of 0.21 MB per round (two messages), the server handles a higher volume due to the coordination of the aggregation process. Despite this, the model maintains a low RMSE, achieving an efficient trade-off between communication and accuracy. Furthermore, vehicle per round participation varies due to the FedAvg sampling

Table 3.3: Server-side and vehicle-side average communication overhead in FedVANET-TP

Node	Type	Total Rounds	Sent / Round	Received / Round	Total Sent	Total Received
FedAvg Server	Message Count	20	7	7	140	140
	Message Size (MB)	–	0.72	0.72	14.46	14.46
Vehicle 2608	Message Count	6	2	2	12	12
	Message Size (MB)	–	0.21	0.21	1.26	1.26
Vehicle 2618	Message Count	11	2	2	22	22
	Message Size (MB)	–	0.21	0.21	2.32	2.32

strategy and intermittent vehicular connectivity. This non-uniform contribution reflects realistic deployment conditions.

Regarding communication costs, the FAHEFL [134] baseline employs an artificial neural network with 431,080 trainable parameters. This incurs in a server-side communication overhead of 6.56 MB per round for two vehicles (comprising 1.64 MB for both upload and download per vehicle, excluding metadata). In contrast, the proposed model utilizes only 25,818 trainable parameters, corresponding to 0.39 MB per round for two vehicles. This represents a reduction in communication overhead approximately by a factor of 16 compared to this relatively large model. Since the number of trainable parameters is directly proportional to the computational burden, these results demonstrate that the proposed approach outperforms state-of-the-art both centralized and decentralized models not only in terms of accuracy, but also in communication and computational efficiency.

3.3.6 . Comparison models

the proposed approach is benchmark against both centralized and distributed baselines using the NGSIM dataset. Regarding the centralized existing methods, Table 3.2, compares the proposed approach against the following methods:

- iNATRan in [248] introduces a Transformer-based model designed to capture complex social and temporal interactions through graph and sparse self-attention mechanisms.

- STA-LSTM in [98] employs dual-level attention mechanisms to identify critical temporal dependencies and rank spatial interactions, specifically demonstrating how learned attention weights can explain complex maneuvers such as lane-changing behaviors.

- AS-LSTM [101] utilizes an integration of social convolutional pooling and attention layers to enhance vehicle trajectory prediction accuracy.

We also compared our model with the following distributed model:

- FAHEFL [134]: a trajectory forecasting approach using FL. By integrating Fast partially Adaptive Homomorphic Encryption (FAHE1) [249], the framework ensures data confidentiality during nodal communication and server-side aggregation.

3.4 . Conclusion

This chapter presented a trajectory prediction framework based on a customized Transformer architecture within a Federated Learning (FL) paradigm. Experimental results demonstrate that the proposed approach outperforms both state-of-the-art centralized and distributed models in terms of prediction accuracy, achieving lower RMSE values. Furthermore, the model exhibits superior communication and computational efficiency compared to existing federated learning-based approaches, while maintaining strong generalizability across different vehicular scenarios. These findings confirm the feasibility of deploying the model in distributed, collaborative, and highly dynamic vehicular environments with intermittent connectivity.

However, despite these advantages, the proposed system faces several limitations, including:

- **Single point of failure:** when the aggregation server fails or is compromised by a cyberattack, the entire system is blocked or compromised.
- **Communication overhead:** the system requires frequent communication between the server and all participating vehicles at each round, which can lead to increased latency in highly mobile vehicular environments and potential network saturation.
- **Security vulnerabilities:** although raw data does not leave the vehicles, several attacks remain possible, including model poisoning, data poisoning, model inversion, Sybil attacks, and Denial-of-Service (DoS) attacks.

  Convergence issue: In VANETs, vehicles frequently join and leave the federated system, which can slow convergence of the prediction model.

To address these limitations and move toward a more resilient architecture, the following chapter explores a fully decentralized trajectory prediction framework that operates without a central server. By eliminating the central server in favor of peer-to-peer coordination, this next approach aims to enhance system robustness, scalability, efficiency, and autonomy in fully distributed vehicular environments.

4 - FDG-VTP: A Fully Decentralized Gossip Vehicular Trajectory Prediction Model

4.1 . Introduction

The vehicle trajectory prediction approach presented in the previous chapter achieved significant improvements in prediction accuracy and notable gains in communication and computational efficiency. These results were driven by effective data preprocessing and a meticulous configuration of the neural network and the Federated Learning architecture.

However, this system relies on a central orchestration server, which introduces several structural limitations. In particular, bottlenecks may arise due to vehicles with limited computational or communication capabilities, as the aggregation server must wait for model updates from all participating vehicles at each training round before performing aggregation and proceeding to the next round. Moreover, the frequent communications between vehicles and the server generate substantial overhead, while the server itself represents a single point of failure [250]. Consequently, any vehicle losing connectivity with the Roadside Unit (RSU) is automatically excluded from the system.

To address these limitations, this chapter introduces a fully decentralized trajectory prediction model based on the Decentralized Federated Learning (DFL) paradigm. Named FDG-VTP, this framework eliminates the need for central coordination by using direct vehicle-to-vehicle (V2V) communication. To effectively manage intermittent connectivity in highly mobile environments, FDG-VTP is combined with a gossip-type peer-to-peer communication protocol for robust model update dissemination.

Despite their potential, current DFL-based mobility prediction frameworks necessitate a compromise between predictive precision and communication overhead [139]. While existing literature often prioritizes model convergence [251], it frequently neglects the critical roles of network reliability and the inherent heterogeneity of vehicular datasets. In this context, this chapter proposes a novel collaborative trajectory prediction framework that integrates the highly accurate Transformer-based prediction approach presented in the previous chapter (Section 3.2.1) with a robust gossip-driven communication algorithm (presented in Section 2.2.8 of Chapter 2), specifically designed for fully decentralized vehicular ad hoc networks (VANETs).

The properties of robustness, flexibility, efficiency, and implementation simplicity motivated the adoption of gossip-based communication protocol in the fully decentralized collaborative trajectory prediction approach proposed in this chapter.

Furthermore, the communication protocol is enhanced with a robust aggregation mechanism that selects only those model parameters expected to improve the local model of the receiving vehicle. A control parameter is also introduced to delay the transmission of updates until they have been sufficiently trained locally, thereby avoiding the dissemination of uninformative or ineffective updates.

The following sections of this chapter present the proposed approach in detail, including the communication protocol, the aggregation strategy, and the experimental results obtained.

4.2 . FDG-VTP proposed model

The proposed model enables vehicles to communicate directly via vehicle-to-vehicle (V2V) interactions, without requiring any third-party coordination equipment, in order to exchange the parameters of their respective local models. This enables each vehicle to benefit from its neighbors' data to enhance its prediction accuracy, without sharing raw data, which is often both voluminous and sensitive.

Thus, in contrast to the centralized scenario discussed in Section 3.2 in the previous chapter, a network of N mobile vehicles was considered in a VANET that collaborate autonomously without the aid of any infrastructure-based coordination equipment. It is assumed that vehicles interact via On-Board Units (OBUs) that rely on the Dedicated Short Range Communication (DSRC) protocol, standardized under IEEE 802.11p, as an alternative to costly infrastructures such as 5G. Each target vehicle k, surrounded by other connected vehicles, performs local training of the customized neural network for trajectory prediction tasks using the Algorithm 3.2.1. The vehicles use the same datasets presented in the previous chapter in Section 3.3.1. To collaborate efficiently while managing bandwidth constraints, each vehicle communicates only with a limited subset of neighboring vehicles within its radio coverage, as enabled by the gossip-based protocol.

Therefore, the proposed methodology comprises two main components: the customized Transformer-based trajectory prediction model and a gossip-based peer-to-peer communication protocol. The following sections provide a detailed presentation of these components.

4.3 . Customized Transformer-based trajectory prediction model

This component maintains the same configuration, data preprocessing techniques, and time-windowing algorithms as the model in Section 3.2.1.

Algorithm 4.3.1 Proposed gossip-based Communication mechanism

Input: Local dataset D_k
Randomly generate local initial parameters w_0
Initialize local parameters $w_k^t \leftarrow w_0$
while stop_event **is not set do**
    $Broadcast\ discovery\ messages$ **in parallel** every $30\ seconds$
    $w_k^{t+1} \leftarrow$ trainAndUpdate(D_k, w_k^t) **in parallel**
    if there are neighbors in neighbor list **then**
        Send $w_k^{t+1}$to selected neighbors **in parallel**
    end if
    Receive messages from neighbors **in parallel**
    Handle received messages **in parallel**
    Monitor neighbors **in parallel**
end while

4.4 . Proposed Gossip-based communication algorithm

Due to its inherent advantages, including simplicity of implementation, resilience to node failures and message loss, and resilience to dynamic topologies, and high scalability, as discussed in Section 2.2.8, a gossip-based protocol has selected for the fully decentralized system proposed in this chapter. To mitigate the limitations of such protocols, such as message redundancy leading to high signaling overhead and slow convergence rates (also addressed in Section 2.2.8), several optimization strategies have been developed. These strategies aim to minimize communication costs and accelerate convergence time. They are integrated into the corresponding modules of Algorithm 4.3.1, which presents the proposed communication protocol and consists of the following components:

- *Neighbor discovery messages and topology management*: Each vehicle periodically broadcasts, via the User Datagram Protocol (UDP) protocol, a discovery message containing the following fields: *message type*, *sender ID*, *Internet Protocol (IP) address*, and *listening port*. An ablation study, illustrated in Figure 4.1, led us to establish a 30-second interval between successive discovery messages (D_t) for highway scenarios; in such environments, the low relative velocities between vehicles traveling in the same direction result in a relatively stable neighborhood topology. In contrast, for urban scenarios characterized by frequent turns and higher topological volatility, we adopted shorter intervals of 5–10 seconds.

  Upon receiving a discovery or response message, the receiving vehicle determines whether the sender is already present in its neighbor list. If the sender is not listed, the receiver appends the sender's ID, IP address, and port number to its neighbor table and initializes an obsolescence counter. Furthermore, if the incoming message is of the "dis-

covery" type, the receiver transmits a response message containing its own identification and connection details. In contrast, if the sender is already present in the neighbor list, the receiver simply resets the associated obsolescence counter. Additionally, each vehicle monitors its registered neighbors' activity every 10 minutes. Any neighbor remaining inactive for more than 5 minutes is considered to have left the system and is consequently removed from the list. This process ensures bidirectional communication and maintains an up-to-date record of active neighbors.

- *Sending Parameters*: The message transmission process relies on an adaptive aggregation mechanism that controls communication frequency to limit network overhead while maintaining the effectiveness of collaborative learning. At each training round t, a dynamic aggregation frequency is computed by the following equation:

$$F_t^k = F_{init}^k + (F_{fin}^k - F_{init}^k) \cdot (1 - e^{-\lambda t}), \tag{4.1}$$

where F_{init}^k and F_{fin}^k represent the initial and final aggregation frequencies, and λ regulates the progression speed. The impact of the initial parameter sharing frequency was analyzed in Table 4.1, justifying the selection of a value of 7 (7 local training cycles before sending) as the optimal initial dissemination frequency (f_{init}) for each vehicle. f_{fin} is set to 1.

The transmission of local model parameters occurs only when the current iteration (t) meets the condition imposed by this aggregation frequency. This strategy prevents the premature or excessive dissemina-

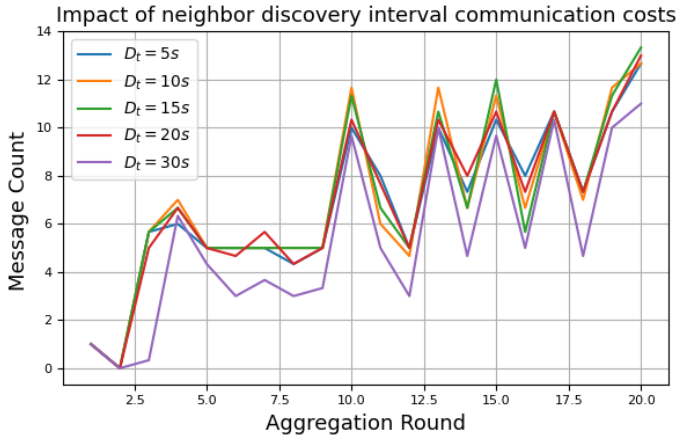

Figure 4.1: Impact of neighbor discovery interval (D_t) on per-round communication costs

tion of weakly informative updates, particularly during the early stages of training.

When transmission is enabled, the local model weights are first quantized to reduce their size and limit communication costs. The quantized parameters are then encapsulated into a structured message containing the identifier of the sending vehicle, its connection information, the model weights, and the size of the local training dataset. The message is subsequently compressed to further reduce the volume of transmitted data, and then fragmented into multiple segments, which are sent sequentially to a subset of neighboring vehicles selected by the gossip communication protocol. Data exchange is performed over direct TCP connections to ensure reliable delivery of the fragments.

- *Receiving parameters*: Upon accepting an incoming connection from a neighboring vehicle, the receiving node buffers the transmitted data, incrementally assembling the message fragments until the full payload is received. Once all fragments have been collected, the receiver reconstructs the original message by performing defragmentation, followed by decompression and deserialization. This process yields a structured message containing the sender's identifier, the transmitted model parameters, and metadata such as the size of the sender's local training dataset.

  For valid weight messages received from neighboring vehicles, the transmitted parameters are used in the aggregation process described below.

- *Local parameter aggregation process*:

  Upon receiving model updates from neighboring vehicles, each vehicle aggregates these updates with its local model parameters using the aggregation strategy proposed in Algorithm 4.4.1. This strategy jointly accounts for the sizes of the training datasets held by the sender vehicles and a reliability factor that reflects the usefulness of the received

Table 4.1: Average Communication Cost and RMSE per Communication Round for Different f_{init} Values

f_{init}	Messages Sent	Messages Received	Total Messages	RMSE
1	5	7	12	0.0617
4	4	6	10	0.0650
6	3	6	9	0.0640
7	2	5	7	0.0626
10	2	4	6	0.0716
15	2	4	6	0.0613

updates. The reliability factor r_n^k, associated with the received update w_n^k, is computed by the receiving vehicle k with respect to the Euclidean divergence D_n^k of w_n^k from the local model w_k, and is defined as follows:

$$r_n^k = e^{-\frac{D_n^k}{\sigma^2}}, \tag{4.2}$$

with

$$D_n^k = \sum_i (w_k - w_n^k)^2, \tag{4.3}$$

in this formulation:

- σ determines the rate at which the reliability r_n^k decays as the distance D_n^k increases; a smaller σ results in a more rapid decrease in trust for even minor divergences, whereas a larger σ allows for a higher tolerance of variations between the received and local models.

The final aggregated model parameters are then computed using the following weighted averaging scheme:

$$\bar{w} = \frac{s_k\, w_k + \sum_{n=1}^{M} \left(s_n \cdot r_n^k \cdot w_n\right)}{s_k + \sum_{n=1}^{M} \left(s_n \cdot r_n^k\right)}, \tag{4.4}$$

where:

- w_k denotes the local parameters,
- w_n denotes the parameters received from neighboring vehicle n,
- s_k and s_n correspond to the sizes of the local dataset and the dataset of vehicle n, respectively,
- M is the number of model updates received by vehicle k from its neighbors (up to a maximum of three messages) and accumulated up to the current communication round.

Once the aggregation process is complete, the resulting weights are used to update the local model parameters; the updated model is then used for the subsequent local training phase.

This ensures that the global model converges towards a high-performance state even in the presence of heterogeneous data or potential outliers in the VANET environment.

Algorithm 4.4.1 Local aggregation process

Input: Local parameters w_k, received parameters w_n^k, local parameters size s_k, received parameters size s_n^k, sensitivity parameter σ
Output: Aggregated weights $\bar{w}$
Initialize total reliability $r_{tot} \leftarrow 0$
Initialize aggregated weights $\bar{w} \leftarrow 0$ ▷ Initialize with zero matrices

for each i in w_k **do**
    $\bar{w}^i \leftarrow w_k^i \times s_k$ ▷ Add local model contribution
end for
$r_{tot} \leftarrow r_{tot} + s_k$ ▷ Add contributions from neighbors
for each j in w_n^k **do**
    $D_n^k \leftarrow \sum_i (w_k^j - w_n^{k,j})^2$ ▷ Compute the divergence
    $r_n^k \leftarrow e^{-\frac{D_n^k}{\sigma^2}}$ ▷ Compute the neighbor's reliability
    for each i in w_n^k **do**
        $\bar{w}^i \leftarrow \bar{w}^i + w_n^{k,i} \times s_n^k \times r_n^k$ ▷ Compute the aggregation
    end for
    $r_{tot} \leftarrow r_{tot} + s_n^k \times r_n^k$ ▷ Compute the total reliability
end for
for each i in $\bar{w}$ **do**
    $\bar{w}^i \leftarrow \bar{w}^i / r_{tot}$ ▷ Normalize aggregated weights by total reliability
end for
return $\bar{w}$ ▷ Return normalized aggregated weights

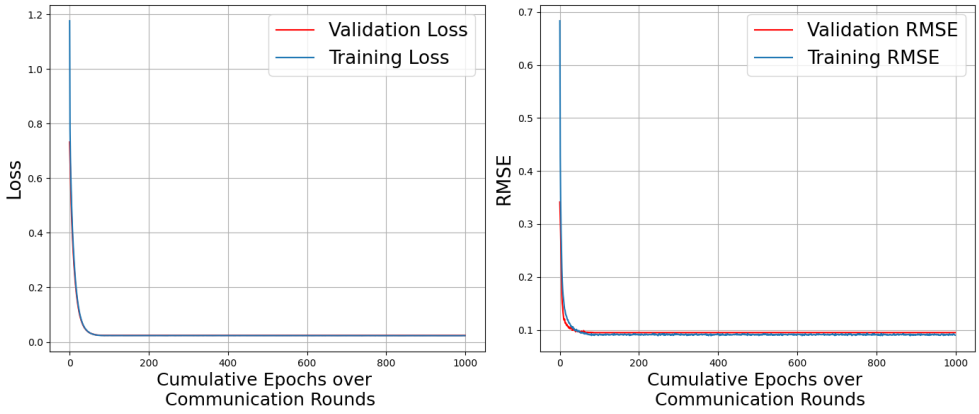

Figure 4.2: Loss function and RMSE convergence trends for a representative vehicle (car 2621) using the proposed model

The experimental setup and the results obtained are discussed in the following section.

4.5 . Experimental setup and results

The experimental setup in this chapter relies on the datasets previously introduced in Section 3.3.1. Similarly, the performance is evaluated using the MSE loss function and the RMSE metric, as specified in Section 3.3.2 and Section 3.3.3. the discussion hereafter centers on the specific implementation of the decentralized environment, followed by a detailed presentation of the results and a comparative study against state-of-the-art methods.

4.5.1 . Implementation details

A local instance of the prediction model is deployed on each participating vehicle and adopts the same architecture and configuration as the model presented in Section 3.2.1 of the previous chapter. According to the results obtained from the ablation studies, the hyperparameters of the communication algorithm are defined as follows: $f_{init} = 7$, $f_{fin} = 1$, and $\sigma = 5.0$. The User Datagram Protocol UDP [252] and Transmission Control Protocol (TCP) [253] transport protocols are used, respectively, to transmit discovery messages with minimal communication overhead and to exchange local model parameters with high reliability. The JavaScript Object Notation (JSON) [254] format is used for all messages exchanged between vehicles within the proposed gossip-based protocol. The hardware platform used in this study is the same as that described in Section 3.3.4 of the previous chapter.

4.5.2 . Simulation results

This section first presents the results obtained with the proposed framework and then compare them with the performance of the comparison models. Figure 4.2 presents the performance indicators regarding the trajectory prediction accuracy of the proposed model. This figure shows the evolution of the MSE loss and the RMSE evaluation metrics across training epochs accumulated over all communication rounds. The results highlight very fast convergence, along with low MSE and RMSE values that are nearly identical across both the training and validation datasets, thereby demonstrating the strong generalization capability of the proposed model. Table 4.2 reports the numerical RMSE results obtained on the NGSIM US-101 and NGSIM I-80 datasets

Table 4.2: Model RMSE (m) on the two datasets NGSIM US-101 and US I-80

Prediction horizon	US-101	US-I-80
1 s	0.0622	0.0522
2 s	0.0903	0.0620
3 s	0.1054	0.0661
4 s	0.1214	0.0904
5 s	0.1253	0.1163
6 s	0.1342	0.1347
7 s	0.1432	0.1360

Table 4.3: RMSE (m) comparison with different baselines methods (NGSIM US-101 dataset)

Model	1 s	2 s	3 s	4 s	5 s
TrajectoFormer [119]	0.47	0.84	1.33	1.97	2.56
MODA [255]	0.40	1.02	1.76	2.69	3.86
SA-STGCN [122]	0.34	0.65	0.97	1.14	1.55
AS-LSTM [101]	0.10	0.20	0.31	0.43	0.56
FDG-VTP	**0.06**	**0.09**	**0.11**	**0.12**	**0.13**

for different prediction horizons (H_p). The table highlights low RMSE values, reaching 0.0622 m and 0.0522 m at a 1 s prediction horizon for the NGSIM US-101 and NGSIM I-80 datasets, respectively. At a 5s prediction horizon, the errors remain limited, with RMSE values of 0.1253 m and 0.1163m on the two datasets, respectively. Even at a 7s prediction horizon, the RMSE remains low, reaching 0.1432m for NGSIM US-101 and 0.1360 m for NGSIM I-80. The results demonstrate consistently low errors across both datasets. Similar to the centralized FL–based model in Section 3.3.5, this approach achieves an approximately 36% improvement in prediction accuracy at 1s prediction horizon.

Table 4.3 and Figure 4.3 compare the performance of the proposed model, evaluated using the RMSE metric, with that of several state-of-the-art approaches from the literature. The reported curves and numerical results clearly demonstrate the superiority of the proposed method, which consistently achieves lower RMSE values than all competing approaches.

The analysis of the impact of the number of neighboring vehicles with which each target vehicle communicates is presented in Figure 4.4. The results indicate that prediction accuracy improves as the number of neighbors increases for short-term prediction horizons (up to 3 s). Beyond this threshold, accuracy tends to stabilize regardless of fleet size. This behavior can be

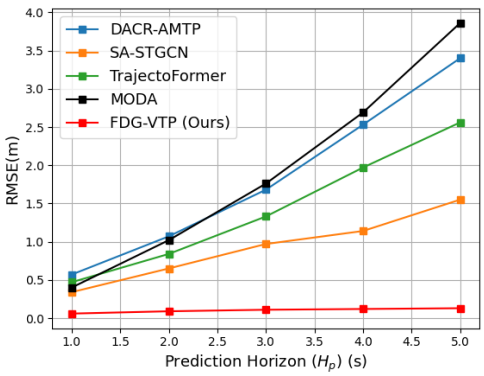

Figure 4.3: RMSE comparison with State-of-the-art baselines methods

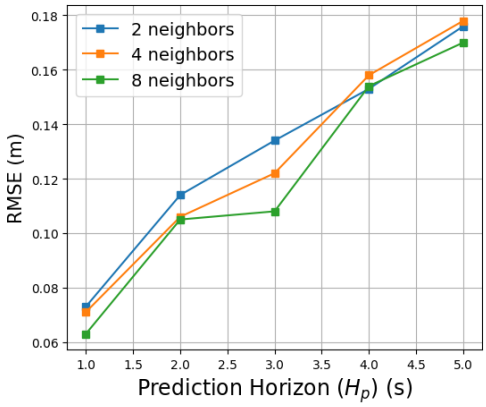

Figure 4.4: Impact of the number of neighboring vehicles on RMSE

explained by the fact that a larger set of neighboring vehicles enables each node to leverage a richer pool of information to refine its local model. However, for longer prediction horizons, this benefit is progressively offset by the accumulation of errors introduced by each vehicle's local prediction model.

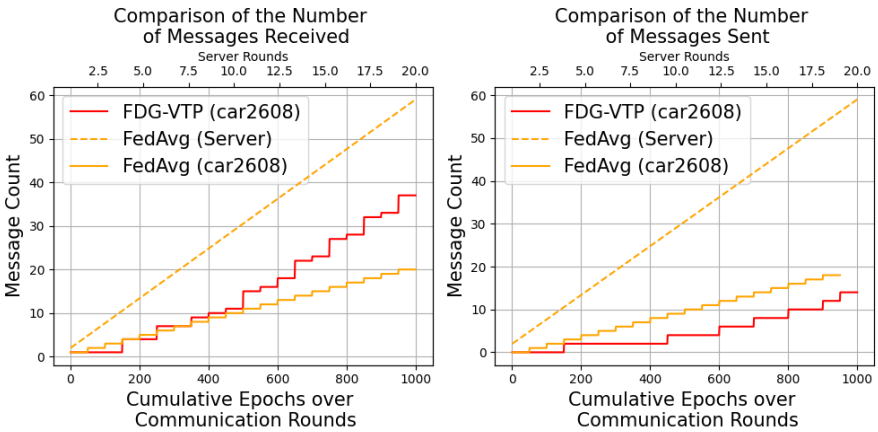

Figure 4.5: Comparison of received and sent message counts between the proposed model and the FL-FedAvg-based method

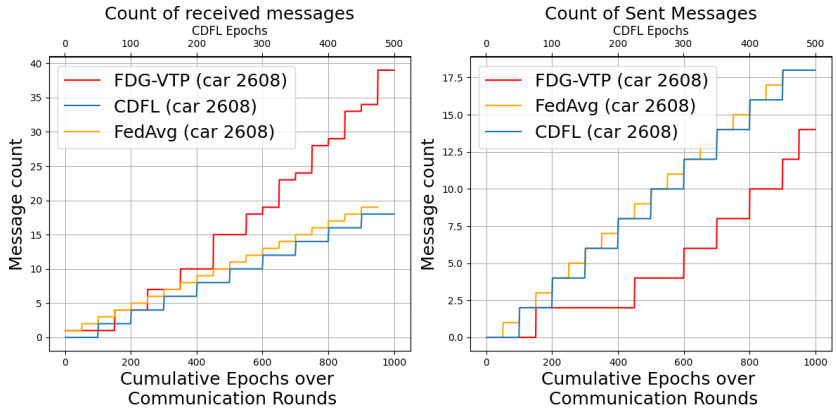

Figure 4.6: Comparison of received and sent message counts between the proposed model and the baselines

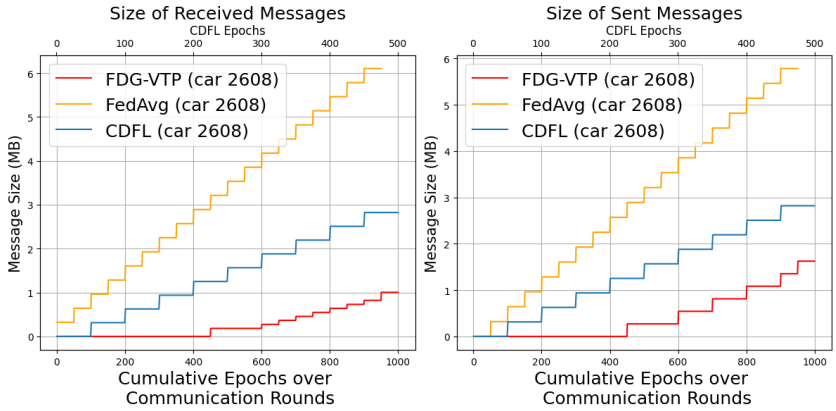

Figure 4.7: Comparison of received and sent message sizes between the proposed model and the baselines

Regarding the communication overhead of the proposed algorithm, Figures 4.5 and 4.6 show that, FDG-VTP receives approximately 2 times as many messages as CDFL. This is due to the regular broadcast of neighbor-discovery messages by surrounding vehicles, as well as the transmission of model weights after each training round, once the initial delayed iterations have been completed by neighbor vehicles. However, as shown in Figure 4.5, FDG-VTP outperforms FedAvg in bidirectional communication, reducing the number of received messages by about 4% and the number of transmitted messages by 64%. Compared to CDFL, it also achieves a 22% reduction in the total number of messages sent. This reduction is explained by the fact that discovery messages originating from already known neighbors are not processed by the target vehicle. Furthermore, thanks to quantization and compression,Figure 4.7 shows that the proposed approach significantly reduces reduces the size of received messages by approximately 64% compared to CDFL and over 80%

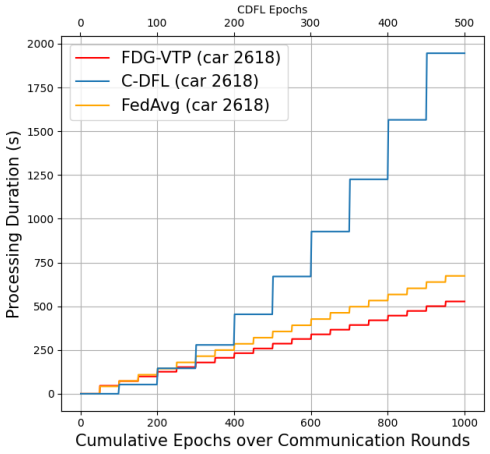

Figure 4.8: Comparison of processing times between the proposed framework and baselines methods

compared to FedAvg. Similarly, the size of sent messages is reduced by about 42% and 72% relative to CDFL and FedAvg, respectively.

Moreover, as illustrated in Figure 4.8, the proposed approach reduces the computation time by about 72% compared to CDFL and 22% compared to FedAvg.

4.5.3 . Comparison models

The state-of-the-art (SOTA) mobility prediction models employed as baselines in this study are as follows

- AS-LSTM [101] A description of this approach is provided in Section 3.3.6.
- **TrajectoFormer [119]:** This model uses a customized Transformer architecture, along with neighbor selection and time-windowing techniques, to effectively capture spatiotemporal dependencies among vehicles.
- **MODA [255]:** This model, termed multimodal and dynamics-aware (MODA) interaction Neural Network, utilizes an LSTM encoder-decoder architecture enhanced by a Graph Attention Network (GAT) [256]. A Conditional Variational Autoencoder (CVAE) [257] is also incorporated, allowing the model to output multimodal trajectories.
- **SA-STGCN [122]:** This model is based on a dynamic Spatio-Temporal Surrounding-Aware Graph Convolutional Network (SA-STGCN) for simultaneously predicting the motion and trajectories of all vehicles in a given scene. A Graph Convolutional Network (GCN) [258] is used to capture the spatial dependencies between vehicles, while a Temporal Convolution Network (TCN) [259] is employed for temporal correlation.

The proposed communication algorithm is compared with the following SOTA baselines:

- **FedAvg [7]:** The Federated Averaging (FedAvg) algorithm is one of the foundational algorithms of Federated Learning (FL). It enables multiple nodes to collaborate on training an AI model by exchanging only the model parameters, thereby enhancing the privacy of training data and reducing communication costs. FedAvg relies on a central coordination server to optimize resource utilization and security strategies.
- **C-DFL [146]:** C-DFL is a Decentralized Federated Learning (DFL) framework that incorporates compressed communications. Within this framework, each node alternately performs multiple local updates and several inter-node communication rounds during a single training cycle. It was proposed to balance communication efficiency with model consensus, while ensuring convergence guarantees.

4.6 . Conclusion

This chapter introduces FDG-VTP, a novel vehicle trajectory prediction framework designed for fully decentralized vehicular ad-hoc networks (VANETs). It integrates an optimized Transformer model for highly accurate trajectory forecasting with an efficient, asynchronous gossip-based communication algorithm. Experimental results demonstrate that FDG-VTP outperforms existing centralized and decentralized trajectory prediction models in terms of higher precision, lower computational resource usage, and reduced communication overhead. This high-performing, efficient, and robust framework also ensures the privacy of raw vehicle data by preventing its transmission within the system. However, this approach requires strict security measures to guarantee the confidentiality, integrity, and availability of the local model parameters exchanged between vehicles. This issue is addressed in the following chapter, which presents a method to secure vehicle-to-vehicle communications, as well as to detect malicious behavior and exclude the responsible vehicle in this fully decentralized system.

5 - PPIR: A Privacy-preserving and Intrusion-Resilient framework for Gossip-based Communication Algorithm in VANETs

5.1 . Introduction

The previous chapter demonstrated that an artificial neural network architecture based on the Transformer model, when properly configured, can achieve highly accurate trajectory prediction performance, comparable to, or even exceeding, that of centralized federated learning (FL)-based approaches. It was also shown that a gossip-based communication algorithm, combined with an appropriate aggregation strategy and reinforced by a mechanism that prevents premature or excessive dissemination of weakly informative updates, particularly during the early stages of training, achieves significantly superior communication and computational efficiency compared to centralized FL-based approaches.

This chapter focuses on integrating a security and privacy layers into the fully decentralized and collaborative trajectory prediction system for Vehicular Ad-hoc Networks (VANETs) introduced in the Chapter 4. Security breaches in VANETs pose serious threats, compromising sensitive data and endangering the physical safety of road users and their property. These systems continuously exchange critical information, including vehicle states, trajectories, and increasingly, learned model parameters. Consequently, confidentiality, integrity, authenticity, and availability become essential requirements, especially in fully decentralized environments where trust assumptions are minimal.

FL and its decentralized variant (DFL) have been adopted to enable collaborative learning while keeping raw data local. However, decentralization significantly expands the attack surface. In VANETs, adversaries may exploit eavesdropping, impersonation, Sybil attacks, model poisoning, backdoor insertion, or denial-of-service attacks to disrupt the learning process, manipulate shared parameters, or infer sensitive information. Moreover, compromised vehicles may behave in a Byzantine manner by selectively transmitting corrupted updates or attempting to isolate honest nodes, thereby directly threatening system integrity and availability.

Existing approaches are limited by their dependence on infrastructure, high computational overhead, and inadequate protection against insider attacks. To overcome these limitations, this chapter introduces a fully decentralized security framework, termed the Privacy-Preserving and Intrusion-Resilient decentralized gossip-based trajectory prediction framework for VANETs (PPIR).

The proposed framework relies on a data-centric approach that performs plausibility and consistency checks using innovation errors derived from an extended Kalman filter [260] to detect and exclude misbehaving vehicles. Furthermore, a unified elliptic curve Diffie–Hellman (ECDH)-based [16] encryption and signature scheme, combined with quantization, compression, and fragmentation techniques, is employed to reduce communication overhead while ensuring security, privacy, and robustness under intermittent connectivity and adversarial conditions. Several cyber-attack scenarios were evaluated on the proposed system to assess its performance. The results demonstrate the high effectiveness of the proposed approach and its superiority over baseline methods in terms of intrusion detection and mitigation performance, as well as computational efficiency.

The following sections successively present the encryption and digital signature mechanisms used to preserve privacy within the system, and the malicious behavior detection scheme introduced to counter internal threats.

5.2 . System model

The framework introduced in this chapter constitutes a data-driven misbehavior-detection and mitigation system (MDS), integrated with a unified elliptic-curve-based encryption and digital signature process. It is designed for Vehicular Ad-hoc Networks (VANETS), where vehicles engage in direct V2V communication to collaboratively train an artificial neural network, thereby enhancing the accuracy of their respective local trajectory predictions. This environment uses the gossip-based communication protocol described in Section 4.4 of the previous chapter, while the underlying neural network model follows the architecture presented in Section 4.3 of that chapter.

The following sections present these security techniques and their implementation within the proposed framework.

5.2.1 . Encryption and digital signature mechanism

To maintain a high level of security while minimizing computational overhead, a unified signcryption scheme based on the Elliptic Curve Diffie-Hellman (ECDH) [16] key agreement and the Elliptic Curve Digital Signature Algorithm (ECDSA) [17] is employed.

Before transmission, the sending vehicle k and the recipient n one establish a shared secret. The vehicle k hold an elliptic curve key pair (d_k, Q_k), where d_k denotes the private key and

$$Q_k = d_k G \tag{5.1}$$

the corresponding public key, with G being the generator point of the selected elliptic curve.

When the two vehicles k and n communicate, the shared secret is derived using the ECDH protocol as follows:

$$S_{k,n} = d_k Q_k = d_k d_n G. \tag{5.2}$$

This shared secret $S_{k,n}$ is processed through a key derivation function (KDF) to obtain a symmetric encryption key $s_{k,n}$:

$$s_{k,n} = KDF(S_{k,n}) \tag{5.3}$$

Model updates m are then encrypted using a symmetric encryption algorithm:

$$c = Enc_{s_{k,n}}(m) \tag{5.4}$$

In parallel, a digital signature is generated over the encrypted message to ensure authenticity and integrity:

$$\sigma = Sign_{d_k}(c) \tag{5.5}$$

The transmitted message consists of the tuple (c, σ, Q_k). Upon reception, vehicle n verifies the signature using the sender's public key Q_k and decrypts the message only if the verification succeeds. Any failure in these steps results in immediate message rejection.

The following section describes in detail the system for detecting and mitigating malicious behaviors proposed in this chapter.

5.2.2 . Proposed misbehavior detection system

This subsection presents the components of the proposed misbehavior detection system (MDS) illustrated in Figure 5.1. The system adopts a data-centric approach, filtering malicious or anomalous model updates (rather than raw data such as position or speed) prior to aggregation process. Each node independently evaluates received updates using complementary plausibility and dynamic consistency analyses. The consistency analyses rely on innovation errors computed by an Extended Kalman Filter (EKF). The final decision is derived from a weighted aggregation of the individual detection indicators. These analyses are performed by algorithms that are described as follows:

- **EKFTracker**: to identify malicious or anomalous behavior within the vehicular network, a tracking mechanism based on the Extended Kalman Filter (EKF), illustrated in 5.2.1, is implemented. This algorithm dynamically models the evolution of local model weights among neighboring vehicles and detects significant statistical deviations.

  Each target vehicle k maintains an independent EKF state associated with each neighboring vehicle n in the network. This state, denoted as

$x^n \in \mathbb{R}^d$, where d denotes the total number of model parameters, represents the current estimate of the expected weights of the vehicle n. It serves as a reference for calculating both prediction and innovation.

The EKF algorithm relies on two fundamental components: the transition model and the observation model.

- *Transition model (f)*: The state transition is based on a local gradient estimate, computed as the difference between the previous and current weights, scaled by the learning rate η. The predicted state is defined as:

$$x^n_{t|t-1} = x^n_{t-1} - \eta \nabla x^n_{t-1}, \tag{5.6}$$

with

$$\nabla x^n_{t-1} = \frac{w^n_{t-1} - w^n_t}{\eta} \tag{5.7}$$

where w^n_t denotes the received weights vector from the vehicle n at the iteration t.

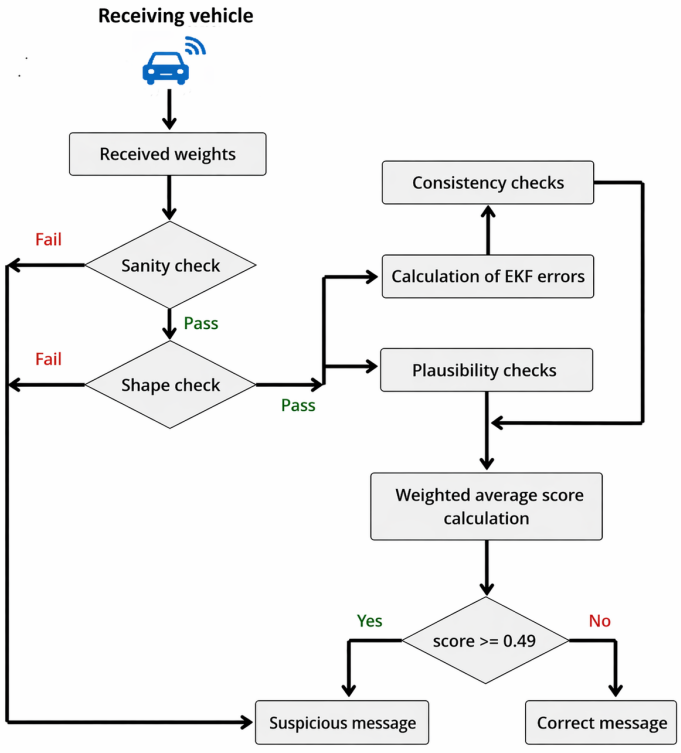

Figure 5.1: Proposed misbehavior detection system

The covariance prediction follows:

$$P_{t|t-1} = F P_{t-1} F^\top + Q \tag{5.8}$$

where F is the diagonal Jacobian of the transition model and Q the process noise covariance.

- *Observation model (h)*: The observation function maps the predicted state back into the measurement space, accounting for precision constraints (e.g., float16 quantization effects). It corresponds to the received weights, possibly quantized:

$$z_t = h(x_t) + v_t \tag{5.9}$$

where $h(\cdot)$ reconstructs the weights from the state and $v_t \sim N(0, R)$ denotes the measurement noise.

The innovation vector is defined as:

$$\nu_t = z_t - h(x_{t|t-1}) \tag{5.10}$$

The Kalman gain is computed as:

Algorithm 5.2.1 EKFTracker

Input: Received weights w_t^n, weight shapes $\mathcal{S}$, learning rate η
Output: Innovation vector ν_t^n
$d \leftarrow \sum_{s \in S} \prod s$
$Q \leftarrow 0.01\, I_d$ ▷ Process noise
$R \leftarrow 0.05\, I_d$ ▷ Measurement noise

function Prediction(n)
    $\nabla x_{t-1}^n \leftarrow \frac{w_{t-1}^n - w_t^n}{\eta}$ ▷ Estimate gradient
    $x_{t|t-1}^n \leftarrow x_{t-1}^n - \eta \nabla x_{t-1}^n$ ▷ State prediction
    $F \leftarrow I_d - \eta \cdot 10^{-2}$ ▷ Jacobian
    $P_{t|t-1}^n \leftarrow F P_{t-1}^n F^\top + Q$ ▷ Covariance prediction
end function

function Update(n, w_t^n)
    $z_t^n \leftarrow \text{vec}(w_t^n)$ ▷ Observation
    $\hat{z}_t^n \leftarrow h(x_{t|t-1}^n)$ ▷ Predicted observation
    $\nu_t^n \leftarrow z_t^n - \hat{z}_t^n$ ▷ Innovation
    $K_t^n \leftarrow P_{t|t-1}^n \left(P_{t|t-1}^n + R + 10^{-4} I_d \right)^{-1}$ ▷ Kalman gain
    State update: $x_t^n \leftarrow x_{t|t-1}^n + K_t^n \nu_t^n$
    $P_t^n \leftarrow (I_d - K_t^n) P_{t|t-1}^n$ ▷ Covariance update
    return ν_t^n
end function

$$K_t = P_{t|t-1} \left(P_{t|t-1} + R \right)^{-1} \tag{5.11}$$

State and covariance are updated as follows:

$$x_t = x_{t|t-1} + K_t \nu_t \tag{5.12}$$

$$P_t = \left(I_d - K_t \right) P_{t|t-1} \tag{5.13}$$

Given the high dimensionality of neural network parameters, the EKF-Tracker algorithm uses sparse matrices for covariance calculations and the determination of the Kalman gain (K). This approach significantly reduces the memory footprint and computational complexity, ensuring that real-time tracking remains feasible within resource-constrained vehicular environments.

- **ConsistencyChecker**: Illustrated in Algorithm 5.2.2, This layer uses an Extended Kalman Filter (EKF) to assess the dynamic consistency of the update process. It relies on two metrics:

Algorithm 5.2.2 ConsistencyChecker

Input: Neighbor vehicle ID n, EKFTracker
Output: Consistency result $\mathcal{R} = \{MC, BC\}$
Initialize sliding window $W_n \leftarrow \emptyset$
function CheckConsistency(n)
    Initialize $\mathcal{R} \leftarrow \{0, 0\}$
    $\nu_n \leftarrow$ last EKF innovation of vehicle n
                                ▷ Step 1: Magnitude check
    if $\max(|\nu_n|) > \tau_{\max}$ **then**
        $\mathcal{R}.MC \leftarrow 1$
        return $\mathcal{R}$
    end if
    Append ν_n to W_n
                                ▷ Step 2: Robust statistical bounds
    $\text{median} \leftarrow \text{median}(W_n)$
    $\text{MAD} \leftarrow 1.4826 \cdot \text{median}(|W_n - \text{median}|)$
    $\text{MAD} \leftarrow \max(\text{MAD}, \epsilon)$
    $L \leftarrow \text{median} - 1.5 \cdot \text{MAD}$
    $U \leftarrow \text{median} + 1.5 \cdot \text{MAD}$
    if $\nu_n < L$ **or** $\nu_n > U$ **then**
        $\mathcal{R}.BC \leftarrow 1$
    end if
    return $\mathcal{R}$
end function

- *Magnitude Check (MC)* : For each peer n, this check analyzes the filter's innovation (ν_n), which is the discrepancy between the EKF's predicted value and the received update. If the maximum absolute value of this innovation vector exceeds a threshold (τ_{max}), the update is deemed inconsistent.

$$MC = \begin{cases} 1, & \text{if } max|\nu_n| > \tau_{\text{max}}, \\ 0, & \text{otherwise.} \end{cases} \tag{5.14}$$

- *Bound Check (BC)*: This robust statistical analysis applies a Hampel filter to a sliding window of past innovation errors. It computes a median and Median Absolute Deviation (MAD) to define a dynamic confidence interval (between a lower bound and a upper bound). Any innovation error falling outside this interval is considered an outlier.

$$BC = \begin{cases} 1, & \text{if } \nu_n \notin [lower, upper], \\ 0, & \text{otherwise.} \end{cases} \tag{5.15}$$

with

$$lower = median(\nu) - 1.5 \cdot MAD \tag{5.16}$$

$$upper = median(\nu) + 1.5 \cdot MAD \tag{5.17}$$

- **PlausibilityChecker**: Illustrated in 5.2.3, this layer examines the likelihood of a received update compared to the previously received updates from the same vehicle. It relies on two main metrics:

  - *Norm Check (NC)*: This analysis computs the L_2 norm of the received weights from vehicle n. If this norm exceeds a predefined threshold (N_{max}), the update is flagged as a potential malicious update.

$$NC = \begin{cases} 1, & \text{if } \|w_n\|_2 > N_{\text{max}}, \\ 0, & \text{otherwise.} \end{cases} \tag{5.18}$$

  - *Difference Check (DC)*: This temporal analysis compares the current update with the last update received from the same sender. The absolute difference is computed; if the 95^{th} percentile of this difference exceeds a threshold (Δ_{max}), the update is considered anomalous.

$$DC = \begin{cases} 1, & \text{if } Percentile_{95}(\Delta_n) > \Delta_{max}, \\ 0, & \text{otherwise.} \end{cases} \tag{5.19}$$

with

$$\Delta_n = |w_n^t - w_n^{t+1}| \tag{5.20}$$

Prior to applying the Plausibility and Consistency checks, the system performs a suite of lightweight verification filters:

- **Numerical integrity validation** to ensure the absence of NaN and Inf values in the received update
- **Tensor shape conformity verification** to ensure that the shape of the received update matches the expected shape of the model

Serving as a first line of defense, these filters immediately isolate clearly corrupted updates.

Algorithm 5.2.3 PlausibilityChecker

Input: Neighbor ID n, received weights w_n, previous weights $\bar{w}_n$, local weights w_k
Parameters: $N_{\max}$ (max norm), $\Delta_{\max}$ (max update variation)
Output: Plausibility result $\mathcal{R} = \{NC, DC\}$
Initialize $\mathcal{R} \leftarrow \{0, 0\}$
$\tilde{w}_n \leftarrow \text{vec}(w_n)$ ▷ Flatten received weights
if $\text{std}(\tilde{w}_n) < 10^{-3}$ **then**
    $\mathcal{R}.NC \leftarrow 1$
else
    $n_n \leftarrow \|\tilde{w}_n\|_2$
    if $n_n > N_{\max}$ **then**
        $\mathcal{R}.NC \leftarrow 1$
    else
        if $n_n \notin [0.5\, n_n^{\min},\ 2.0\, n_n^{\max}]$ **then**
            $\mathcal{R}.NC \leftarrow 1$ ▷ Norm Consistency Check
        end if
    end if
end if
if $\bar{w}_n \neq \varnothing$ **then**
    $\hat{w}_n \leftarrow \text{vec}(\bar{w}_n)$
    $\Delta_n \leftarrow |\tilde{w}_n - \hat{w}_n|$
    if $\text{percentile}_{95}(\Delta_n) > \Delta_{\max}$ **then**
        $\mathcal{R}.DC \leftarrow 1$ ▷ Difference Consistency Check
    end if
end if
Update historical norm statistics and store w_n
return $\mathcal{R}$

Individual binary results are aggregated into a final decision through a weighted risk score. Each check is multiplied by a predefined confidence weight, summed, and then normalized. To ensure strict anomaly detection, a penalty is applied if at least one check fails. An update is definitely rejected if its final score falls below 0.49; otherwise, it is validated for the global model update.

The following section presents the threat model adopted in this study.

5.3 . The threat model

This section presents the threat model used in this thesis to evaluate the reliability and resilience of the proposed system. Within this framework, we assume that the privacy-preserving mechanisms implemented, including signcryption, are secure and cannot be compromised. Nevertheless, malicious vehicles (MVs) can impersonate legitimate vehicle credentials, thereby gaining access to the system and interacting with honest participants. Once admitted, these adversarial vehicles can launch a wide range of attacks. This study focuses on three major threat classes.

- **Model poisoning attacks:** This category include the following types of attacks:
  - **Random-weights attacks:** In this attack, local model parameters are replaced with values drawn from a uniform distribution and disseminated to neighboring vehicles;
  - **Backdoor attacks:** In this attach a fixed constant value is injected into a predefined subset (20%) of the model parameters prior to sharing;
  - **Model scaling attacks:** This attack amplify local parameters by a scaling factor before transmission; and
  - **Sign-flipping attacks**, in which the sign of each model parameter is inverted.
- **Denial-of-service (DoS) attacks:**, The objective of this category of attack is to disrupt the collaborative learning process by overwhelming the network or specific vehicles with excessive messages or malformed updates, thereby degrading availability and increasing communication latency.
- **Sybil attacks:** In this category of attack, a single malicious vehicle creates multiple identities to disproportionally influence the learning process, bias aggregation strategy, and undermine the trust assumptions of the decentralized system.

The severity of each attack is controlled by an intensity factor, where larger absolute values correspond to more disruptive behavior but also increase detectability. These attacks adversely affect global model convergence and degrade the accuracy of local models, potentially leading to severe trajectory prediction errors, large-scale traffic disruptions, and, in extreme cases, catastrophic accidents. In addition to adversarial behavior, non-malicious failures are also considered, such as transmission failures, packet loss, or synchronization errors, which may similarly degrade system reliability.

The following section presents the configuration of the experimental environment and discusses the results obtained by the proposed malicious behavior detection system (MDS).

5.4 . Experimental setup and results

The Misbehavior Detection System (MDS) and the privacy-preserving signcryption mechanisms proposed in this chapter are implemented as security and confidentiality layers within the fully decentralized trajectory prediction framework. This framework, based on Decentralized Federated Learning (DFL), was previously detailed in Section 4.5 of the preceding chapter.

The entire system is deployed locally within each target vehicle k. Consequently, each target vehicle encrypts its messages before transmission using the confidentiality layer and uses the security layer's MDS to analyze, in real time and at each communication iteration, the model weights received from its peers. When these weights satisfy all validity, plausibility, and consistency checks, they are integrated into the local aggregation process. In contrast, any update deemed suspicious is rejected, and the sending vehicle is permanently excluded from further exchanges. The simulation environment used in this chapter,including the Transformer-based trajectory prediction model configuration, the gossip-type communication protocol settings, and the software and hardware platforms—is identical to that presented in the previous chapter. Likewise, the datasets employed for training and evaluating the trajectory prediction model and the loss function used remain unchanged.

5.4.1 . Evaluation metrics

The evaluation metric for the trajectory prediction model is identical to that used in Section 4.5 of the previous chapter. To assess the performance of the proposed misbehavior detection system (MDS), a confusion matrix is used, one of the most commonly used tools for evaluating classification models [209]. Based on this matrix, the Accuracy, Recall, Precision, and F1-score metrics are computed to quantify the effectiveness and reliability of the proposed MDS. These metrics are defined as follows:

- *Accuracy*: Measures the proportion of correctly classified updates, in-

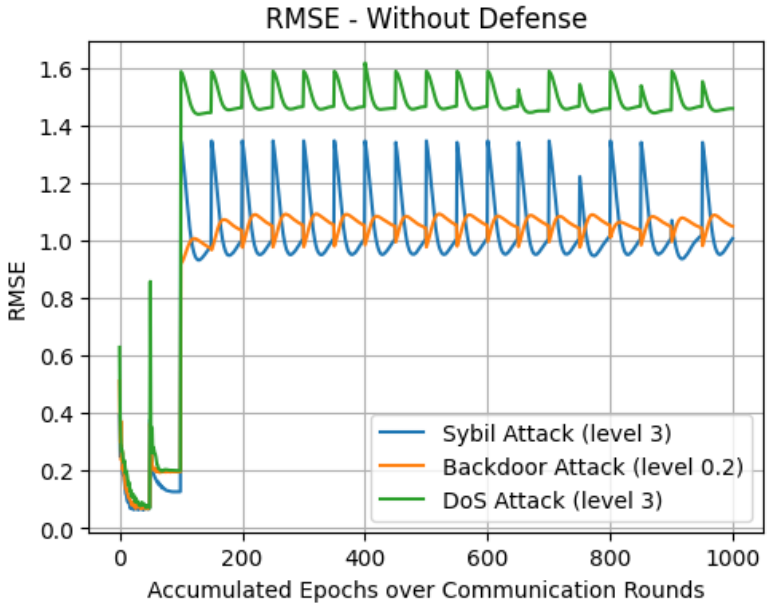

Figure 5.2: Impact of the unmitigated attacks on RMSE.

cluding both positive and negative instances, among all evaluated updates.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.21}$$

- *Precision*: Represents the proportion of correctly identified positive instances among all instances predicted as positive.

$$Precision = \frac{TP}{TP + FP} \tag{5.22}$$

- *Recall*: Also known as Sensitivity or the True Positive Rate (TPR), this metric measures the proportion of actual positive instances that are correctly identified by the model.

$$Recall = \frac{TP}{TP + FN} \tag{5.23}$$

- *F1-score*: This metric provides a balanced evaluation by calculating the harmonic mean of precision and recall, making it particularly useful when both false positives and false negatives carry significant importance.

$$F1 - score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{5.24}$$

The simulation and evaluation results are discussed in the following subsection.

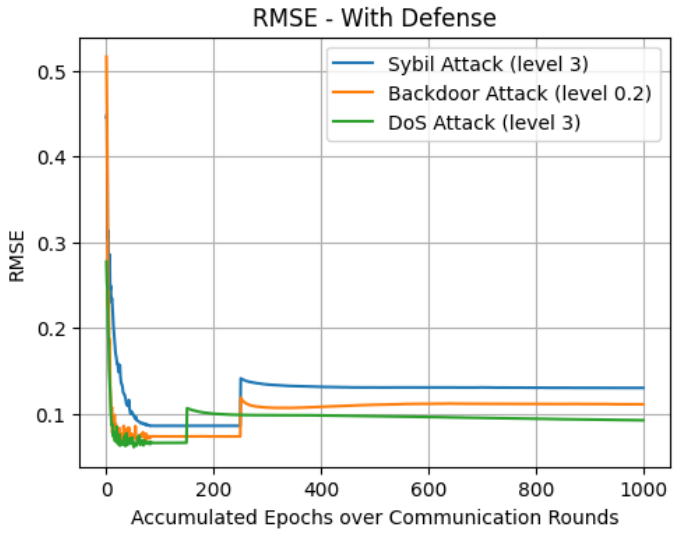

Figure 5.3: Impact of the attacks mitigation on RMSE.

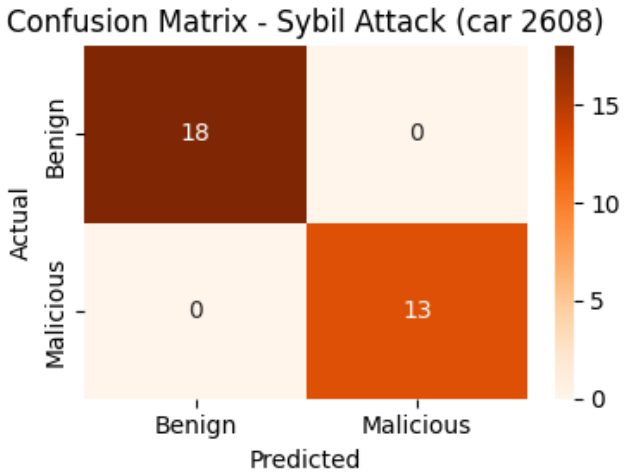

Figure 5.4: Confusion matrix of a representative vehicle under Sybil attack.

5.4.2 . Simulation results

The proposed DFL-based trajectory prediction framework was evaluated across three scenarios: baseline (no attack), unmitigated attacks, and attacks with defenses activated. Under normal conditions (no attack scenario), the proposed model converges by the 20th communication round, achieving an RMSE of about 0.06 m. As illustrated in Figure 5.2, the framework is highly sensitive to unmitigated attacks. Even low-intensity attacks severely degrade performance and prevent global convergence. DoS attacks proved to be the most critical and the most difficult to detect. The Figure 5.3 shows that inte-

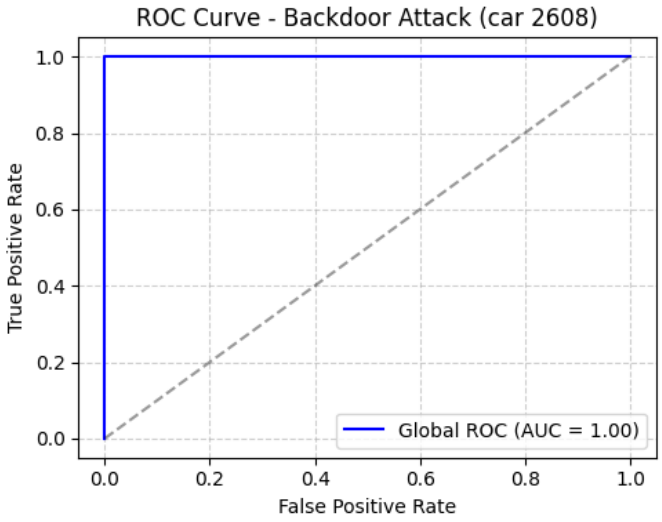

Figure 5.5: ROC curve of a representative vehicle under backdoor attack.

grating the proposed MDS mechanism restores performance to near-baseline levels. Notably, the confusion matrix for mitigating a Sybil attack (Figure 5.4) and the ROC curve for mitigating a backdoor attack (Figure 5.5) by a representative vehicle both demonstrate perfect classification, with zero false positives and zero false negatives, reflecting an ideal separation between benign and malicious update classes. Consequently, the MDS achieves 100% accuracy, precision, recall, and F1-score under specific conditions: namely, backdoor attacks with an intensity of 0.2 (corresponding to a manipulation of 20% of the attacked model's parameters), and Sybil attacks involving three or more fake vehicle IDs, with a malicious node ratio of 20%. For DoS attacks, the best observed performance yields an accuracy of 85.71%, a precision of 50%, a recall of 100%, and an F1-score of 66.67%, as shown in Figure 5.6 and Table 5.1. This table also compares the performance of the proposed MDS against LA-DETECTS [222] state-of-the-art baseline. The results indicate that the proposed approach significantly outperforms the baseline in detecting Sybil attacks and achieves strong performance in detecting DoS attacks, although slightly below that of the baseline.

The proposed MDS runs locally within each target vehicle, thus requiring no additional communication overhead. Moreover, as shown in Figure 5.7, the proposed MDS achieves an average processing time per round of approximately 144 ms, compared to 207 ms for the state-of-the-art baseline, representing a reduction of approximately 30%. It is worth noting that these values may vary depending on the receiving vehicle's computational capabilities and the concurrent processing workload.

In terms of complexity, the EKFTracker—uses sparse diagonal matrices—and

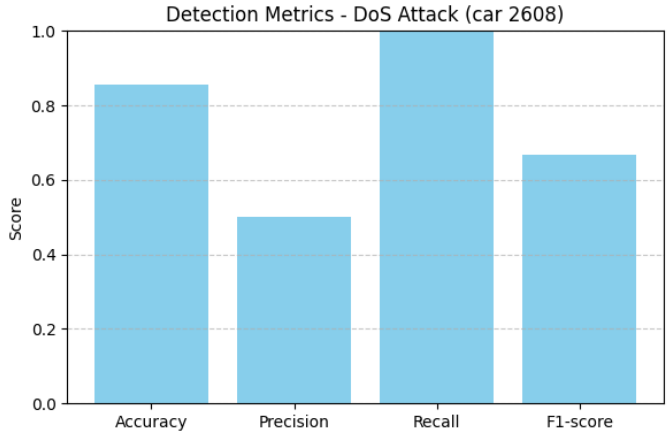

Figure 5.6: Detection metrics of a representative vehicle under DoS attack.

Table 5.1: Performance comparison between the proposed MDS and State-of-the-art baselines

Attack	DoS		Sybil	
	LA-DETECTS[222]	PPIR	LA-DETECTS[222]	PPIR
Accuracy	99.99	85.71	50.73	100.00
Precision	99.99	50.00	56.18	100.00
Recall	99.99	100.00	63.85	100.00
F1-Score	99.99	66.67	60.24	100.00

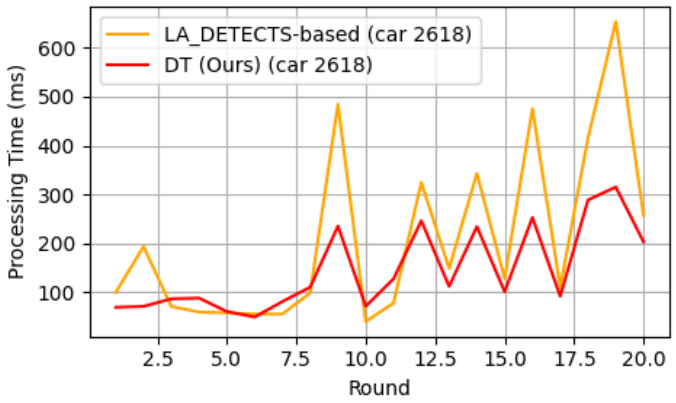

Figure 5.7: Processing time comparison between the proposed MDS and the LA-DETECTS-based basline.

the ConsistencyChecker operate in linear time $O(d)$. Meanwhile, PlausibilityChecker achieves quasi-linear complexity $O(d \cdot log(d))$ due to its statistical

checks. Consequently, the overall proposed MDS overhead is dominated by the PlausibilityChecker, while other modules remain lightweight. Memory usage scales as $O(V \cdot d)$, where V denotes the number of weight vectors received per round. This ensures the scalability of the proposed MDS while maintaining efficiency in peer-to-peer vehicular network settings.

5.5 . Conclusion

This chapter proposes an approach to enhance the security and confidentiality of local model update exchanges within the fully decentralized federated learning-based mobility prediction system introduced in the previous chapter.

The proposed method, termed PPIR (Privacy-Preserving and Intrusion Resilient decentralized gossip-based trajectory prediction framework for VANETs), integrates a unified encryption and digital signature-based privacy-preserving mechanism relying on elliptic-curve Diffie–Hellman operations. This secure framework strengthens a data-centric malicious behavior detection system (MDS).

The MDS relies on consistency analyses based on innovation errors computed through an Extended Kalman Filter, by comparing the predicted updates for each peer vehicle with the updates actually received from that vehicle. These consistency checks are combined with plausibility analyses that assess the likelihood of newly received model weights with respect to the historical weight trajectory of the same vehicle.

Experimental results demonstrate that PPIR outperforms state-of-the-art methods in detecting various types of malicious behaviors, while introducing no additional communication overhead and maintaining a low computational cost.

6 - General Conclusion and Perspectives

6.1 . General conclusion

The emergence of smart cities integrating Intelligent Transportation Systems (ITS) imposes stringent technological requirements on autonomous vehicles, which need permanent and ultra-reliable connectivity to operate effectively. These vehicles continuously exchange information with roadside infrastructure, cloud or edge servers, and neighboring vehicles through Vehicle-to-Infrastructure (V2I), and Vehicle-to-Vehicle (V2V) communications. This communication capability enables autonomous vehicles to adopt increasingly intelligent behaviors, such as avoiding collisions, adjusting their trajectories in real time, and coordinating actions between vehicles, thereby allowing better traffic optimization.

However, this increased connectivity, combined with VANETs' decentralized coordination and their open wireless channels, expands the attack surface within these vehicular networks. The exposure of communication links, distributed learning processes, and onboard decision-making systems introduces vulnerabilities to adversarial manipulation, data poisoning, identity spoofing, denial-of-service attacks, and model corruption. In these safety-critical systems, security breaches do not merely compromise data confidentiality or privacy; they may directly impact vehicle control decisions, potentially leading to cascading failures, traffic disruption, or physical harm. Security, robustness, and trust management, therefore, become fundamental pillars of reliable intelligent transportation systems.

Understanding and developing effective mobility models is essential for implementing efficient security solutions. Accurate trajectory prediction enhances situational awareness, supports anomaly detection, facilitates consistency verification of updates shared among collaborating vehicles, and enables proactive rather than reactive security strategies. In other words, effective mobility modeling is an enabling component for adaptive and efficient security solutions in vehicular networks. This led us to devote a significant part of this thesis to the study of mobility models.

A critical analysis of existing mobility forecasting approaches revealed several structural and operational limitations. For instance, traditional centralized mobility methods rely on transmitting raw vehicular data to remote servers for processing and model training. While effective in controlled settings, such solutions incur substantial communication overhead, high bandwidth consumption, and significant computational burden at aggregation servers, thereby limiting scalability and introducing single points of failure. Furthermore, centralized data collection raises serious privacy concerns, as sensitive mobility

traces and behavioral patterns are exposed.

Although FL centralized and decentralized collaborative paradigms address certain privacy risks by keeping data local, these systems require robust and efficient mechanisms to strengthen security and privacy. Several approaches have been proposed in this context. However, they are generally costly in terms of computational costs and communication overhead. Complex cryptographic algorithms, heavy consensus schemes, or redundant validation mechanisms often introduce significant latency and resource consumption, making them poorly suited to highly dynamic VANET environments where real-time responsiveness is critical.

Moreover, collaborative AI-based mobility models frequently prioritize model accuracy and training efficiency while overlooking the integrity and trustworthiness of exchanged model updates. This lack of security robustness is particularly critical in adversarial settings, where compromised participants may inject malicious gradients or manipulated weights, degrading convergence or embedding stealthy backdoors into shared updates. At the same time, most of existing security defense mechanisms are not fully adapted to VANET-specific constraints, including high mobility, intermittent connectivity, rapidly evolving topologies, heterogeneous onboard computational capabilities, and strict latency requirements.

Based on this observation, this thesis aims to contribute to the improvement of existing methods in the literature by proposing a new framework capable of jointly addressing the three following coupled challenges:

- **A high-precision trajectory prediction model:** Suitable for cooperative autonomous driving and enabling each vehicle to predict its future path with minimal errors.
- **Collaborative and decentralized communication-efficient algorithm:** Capable of enabling vehicles to directly share their local model parameters to enhance mutual prediction accuracy without the need for coordinating infrastructure, while incorporating an effective aggregation strategy. At the same time, remaining compatible with bandwidth and latency constraints.
- **Robust security and privacy layers:** Ensuring the integrity, availability, and confidentiality of the updates exchanged between vehicles.

To achieve these objectives, this dissertation first addresses the challenge of mobility prediction by proposing a highly accurate trajectory prediction model based on a customized and optimized Transformer architecture. To meet the second objective, a fully decentralized peer-to-peer gossip-based communication algorithm is proposed. This algorithm is enhanced by an efficient aggregation strategy and a mechanism that prevents the transmission

of premature or insufficiently trained updates. Finally, security and privacy layers have been integrated into this fully decentralized, Transformer-based collaborative trajectory prediction system.

By integrating highly accurate predictive model, optimized communication strategies, and security and privacy mechanisms, this thesis contributes toward the development of scalable, resource-aware, and trustworthy intelligent vehicular systems.

The main contributions of this thesis can be concluded as follows:

- **FedVANET-TP**: An FL-based framework for vehicular networks enabling collaborative trajectory prediction using a customized Transformer-based model. By using a pruned artificial neural network architecture tailored to trajectory prediction tasks and sharing only local model updates among vehicles through an aggregation server, FedVANET-TP reduces computational and communication overhead and achieves a lower RMSE compared to state-of-the-art baseline models, while preserving the privacy of sensitive vehicle data.

- **FDG-VTP**: A fully decentralized trajectory prediction framework incorporating an optimized communication algorithm. This algorithm integrates an aggregation strategy that accounts for local prediction model dataset sizes and a reliability factor associated with received weights. It further integrates update scheduling mechanisms as well as quantization and compression techniques to reduce communication overhead while maintaining high predictive accuracy. Experimental results demonstrate superior performance compared to both centralized and decentralized baselines while maintaining a reduced computational overhead.

- **PPIR**: A security and confidentiality layers integrated into the fully decentralized trajectory prediction framework. PPIR combines elliptic-curve Diffie–Hellman-based signcryption with a malicious behavior detection system relying on Extended Kalman Filter-based consistency checks and plausibility verification of received weights. Experimental evaluations demonstrate that the proposed approach outperforms state-of-the-art methods in detecting backdoor and Sybil attacks, while achieving performance comparable to state-of-the-art methods under DoS attacks.

Collectively, these contributions demonstrate that fully decentralized collaborative learning in VANET environments can simultaneously achieve high prediction accuracy, communication efficiency, and strong security guarantees. They represent a significant step toward reliable and resilient intelligent mobility systems.

The promising results obtained in this research can be further extended to tackle additional challenges. To this end, future work and potential improvements are outlined in the next section.

6.2 . Perspectives

Several research directions emerge from this work and open promising perspectives for further investigation.

A first extension can involve introducing a hierarchical clustering architecture in which vehicles are organized into dynamic clusters, enabling efficient intra-cluster collaboration, while inter-cluster coordination ensures global model consistency. Such a multi-tier collaborative structure would improve scalability, reduce communication complexity, and enhance robustness in large-scale vehicular deployments.

A second direction can concern the development of adaptive and context-aware aggregation mechanisms. Future work could explore reinforcement learning–based or trust-aware aggregation strategies that dynamically adjust to varying network density, mobility patterns, and adversarial behaviors. This would further enhance resilience in highly dynamic VANET environments.

Another important perspective lies in integrating more lightweight and energy-efficient learning mechanisms. Knowledge distillation, or sparse training techniques could be investigated to further reduce computational overhead while preserving predictive accuracy.

Ragarding security and privacy, deeper analysis of advanced, coordinated attack strategies such as collusion attacks, adaptive poisoning, model inversion, and stealthy Byzantine behavior, will be investigated. Enhancing the proposed MDS to anticipate evolving adversarial strategies would significantly strengthen the trustworthiness of the system.

Additionally, future research could investigate the incorporation of lightweight and communication-efficient blockchain mechanisms or distributed ledger technologies to enhance trust management, traceability of model updates, and accountability among collaborating vehicles in fully decentralized settings.

Finally, under specific conditions, real-world vehicular network experiments could be envisioned to evaluate the system performance under realistic traffic and communication conditions, as well as vehicle data heterogeneity.

Collectively, these research directions aim to advance toward scalable, autonomous, and self-securing collaborative vehicular intelligent systems capable of operating reliably in next-generation smart mobility systems.

Résumé détaillé de la thèse

Les systèmes de transport intelligents (ITS), éléments clés des villes intelligentes, s'appuient sur l'intégration de technologies avancées afin d'améliorer l'efficacité, la sécurité, la durabilité et le confort des transports. L'émergence des véhicules autonomes a profondément transformé le paysage des transports intelligents en introduisant des capacités avancées de perception, de prise de décision autonome et de communication en temps réel. Au sein des réseaux véhiculaires de type VANET (Vehicular Ad-hoc Networks), les véhicules peuvent communiquer entre eux, interagir avec l'infrastructure routière ainsi qu'avec les différents objets connectés composant les systèmes ITS. Ils forment ainsi un environnement distribué, dynamique et fortement interopérable.

Cependant, cette connectivité accrue expose les véhicules à un large éventail de vulnérabilités. La sécurité des protocoles de communication devient alors un enjeu majeur, notamment face à des menaces telles que l'usurpation d'identité, les attaques Sybil, les attaques par déni de service ou encore les attaques par empoisonnement. Dans ce contexte, cette thèse a pour objectif d'analyser ces menaces spécifiques et de proposer des mécanismes de sécurisation adaptés aux contraintes propres aux véhicules connectés, notamment la forte mobilité, l'exigence de faible latence, les ressources limitées et la nature collaborative des échanges.

Par ailleurs, comprendre, savoir modéliser et anticiper les mouvements des véhicules est essentiel pour mieux appréhender les défis liés à la sécurité et à la protection de la vie privée dans les environnements véhiculaires. C'est pourquoi la première partie de cette thèse est consacrée à l'étude des modèles de mobilité. Cette étude nous a permis d'identifier deux grandes familles de modèles de mobilité : les modèles classiques et les modèles basés sur l'intelligence artificielle (IA). Les modèles classiques s'appuient principalement sur des formulations mathématiques, des lois physiques ou des approches statistiques afin de représenter le comportement des véhicules. Leurs hypothèses et paramètres sont généralement faciles à interpréter, ils sont bien adaptés aux applications de planification urbaine et à la modélisation des flux réguliers de trafic routier. Toutefois, leurs performances demeurent limitées dans les scénarios complexes ou lorsqu'il s'agit de représenter le comportement de véhicules individuels. À l'inverse, les modèles de mobilité basés sur l'IA, notamment l'apprentissage automatique et l'apprentissage profond, apprennent directement à partir des données et offrent de meilleures performances dans les situations complexes ainsi que pour la modélisation des comportements individuels, notamment dans les tâches de prédiction de trajectoires. Néanmoins, ces approches nécessitent généralement de grandes quantités de données ainsi qu'une importante capacité de calcul.

Dans la mesure où cette thèse vise à intégrer des mécanismes de sécurité directement au niveau des véhicules, nos travaux s'intéressent principalement aux comportements individuels. Nous avons donc choisi de nous focaliser sur les modèles de mobilité basés sur l'IA.

Au sein de cette catégorie, nous avons identifié trois grands groupes de modèles. Le premier regroupe les approches centralisées, dans lesquelles toutes les données sont collectées au sein d'une infrastructure centrale. Cette centralisation permet d'entraîner les modèles d'IA sur un volume important de données et d'obtenir ainsi de très bonnes performances. Toutefois, elle présente des risques élevés de fuite de données ainsi que des coûts importants en matière de calcul et de communication.

Le deuxième groupe repose sur le paradigme de l'apprentissage fédéré (Federated Learning – FL). Dans ce cadre, les données ainsi que les copies locales des modèles d'IA demeurent au niveau des véhicules, tandis qu'un serveur d'agrégation coordonne la collaboration entre les différents participants. Cette approche permet de réduire les coûts de communication, de répartir les calculs d'entraînement entre les véhicules et d'améliorer la confidentialité des données. Cependant, les modèles basés sur le FL restent exposés à plusieurs limitations, notamment les risques de fuite des paramètres des modèles (poids et biais) lors des échanges avec le serveur d'agrégation, ainsi que le problème du point unique de défaillance lié à la dépendance envers ce serveur.

Le troisième groupe concerne les approches basées sur l'apprentissage fédéré décentralisé (Decentralized Federated Learning – DFL). Ce paradigme supprime le serveur central de coordination et privilégie les communications directes entre véhicules. Il améliore ainsi l'évolutivité du système et renforce la confidentialité des paramètres échangés. En revanche, l'absence de serveur central ralentit généralement la convergence des modèles locaux. Afin d'améliorer cette convergence, ces approches nécessitent souvent de mécanismes de communication complexes impliquant un choix et un paramétrage minutieux des protocoles de communication.

À l'issue de cette étude des modèles de mobilité, nous avons analysé ces mécanismes de communication utilisés dans les approches basées sur le DFL afin d'explorer les modèles de mobilité collaboratifs fondés sur ce paradigme. Cette analyse nous a permis d'identifier deux schémas de communication particulièrement adaptés : le schéma Peer-to-Peer (P2P) classique et une de ses variantes appelée Gossip. Dans les réseaux P2P classiques, chaque véhicule communique directement avec l'ensemble des autres nœuds. Dans le cadre des modèles de mobilité collaboratifs, cette approche permet d'améliorer les performances de prédiction, la convergence et la robustesse des modèles. Cependant, elle réduit l'évolutivité du système et augmente la latence ainsi que les coûts de calcul et de communication. À l'inverse, dans le schéma Gos-

sip, chaque véhicule communique uniquement avec un sous-ensemble limité de véhicules. Cette approche réduit les coûts de communication et améliore l'évolutivité du système, mais souvent au détriment des performances des modèles entrainés.

Par ailleurs, ces deux schémas de communication P2P sans fil favorisent la propagation rapide de logiciels malveillants et exposent les systèmes à de nombreuses menaces de sécurité ciblant les modèles d'IA, en particulier dans les environnements d'apprentissage collaboratif fédéré. Ces menaces permettent notamment la réalisation d'attaques par empoisonnement visant à dégrader les performances des modèles ou à ralentir, voire empêcher, leur convergence ; d'attaques Sybil cherchant à amplifier les effets de l'empoisonnement en multipliant artificiellement le nombre d'attaquants ; ainsi que d'attaques par déni de service (DoS) destinées à saturer les ressources de calcul ou la bande passante des nœuds ciblés. Afin de contrer ces menaces, plusieurs mécanismes de sécurité ont été proposés dans la littérature. Parmi eux, le chiffrement homomorphique permet de transmettre et d'agréger les paramètres des modèles sans avoir à les déchiffrer. La confidentialité différentielle (Differential Privacy – DP), quant à elle, consiste à ajouter du bruit aux paramètres afin d'empêcher la reconstitution des données d'entraînement. D'autres techniques sont également utilisées, telles que la blockchain, l'agrégation robuste, le calcul multipartite sécurisé, les pseudonymes dynamiques, les mécanismes de réputation ainsi que les systèmes de détection et de prévention des comportements malveillants.

À l'issue de cette revue de la littérature, nous avons développé un modèle de prédiction de trajectoire basé sur des réseaux de neurones artificiels de type Transformer, spécifiquement adapté et optimisé pour les tâches de prédiction de trajectoires. Nous avons ensuite implémenté ce modèle au sein d'une architecture FL afin de permettre à chaque véhicule de collaborer avec les autres véhicules du système. Cette collaboration permet à chaque véhicule cible de tirer parti de la richesse des données de ses voisins pour améliorer les performances de ses prédictions locales, sans pour autant partager les données sensibles utilisées pour l'entrainementd des modèles. Les véhicules échangent uniquement les paramètres de leurs modèles locaux, préservant ainsi la confidentialité des données brutes tout en favorisant l'apprentissage collaboratif. Les résultats ont montré des performance de prédiction supérieures aux méthodes de compraison avec des coûts de calcul et de communication également réduits.

Comme deuxième contribution, nous avons étendu cette approche basée sur le FL vers une architecture totalement décentralisée fondée sur le DFL. En plus du modèle de prédiction basé sur le Transformer, cette approche intègre un algorithme de communication P2P de type Gossip. Afin de réduire les coûts de communication et d'améliorer la convergence tout en garantissant

de hautes performances et un niveau de sécurité renforcé, nous avons intégré à cet algorithme deux mécanismes complémentaires : un mécanisme retardant l'envoi des poids jusqu'à ce qu'ils soient suffisamment entraînés localement par chaque véhicule ; un mécanisme d'agrégation pondérée prenant en compte à la fois la fiabilité des poids reçus ainsi que la taille des jeux de données de l'émetteur et du récepteur. Les résultats ont démontré une meilleure efficacité des calculs et des communications.

La dernière contribution de cette thèse consiste à intégrer, au sein de ce système totalement décentralisé, une solution de sécurité et de préservation de la vie privée adaptée aux contraintes des environnements véhiculaires. Cette solution repose sur un système de détection et de réponse aux anomalies fondé sur des vérifications de cohérence et de plausibilité utilisant les erreurs d'innovation d'un filtre de Kalman étendu. Ce mécanisme permet de détecter en temps réel les comportements malveillants et d'exclure les véhicules suspects du processus collaboratif. Il est complété par un mécanisme léger de chiffrement et de signature numérique unifié garantissant l'authenticité, la confidentialité et l'intégrité des messages échangés. L'approche de sécurité proposée a été validée expérimentalement à travers la simulation de plusieurs types d'attaques, notamment des attaques par empoisonnement, des attaques Sybil et des attaques par déni de service (DoS). Les résultats obtenus montrent des performances comparables à celles de l'état de l'art en matière de détection des comportements malveillants et de résilience, tout en réduisant les coûts de calcul d'environ 30%, ce qui rend cette solution plus adaptée aux contraintes des systèmes VANET embarqués.

Bibliography

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[2] H. Hartenstein and L. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *IEEE Communications Magazine*, vol. 46, no. 6, pp. 164–171, 2008.

[3] S. Corson and J. Macker, "Mobile ad hoc networking (manet): Routing protocol performance issues and evaluation considerations," Tech. Rep., 1999.

[4] L. C. Hua, M. H. Anisi, L. Yee, and M. Alam, "Social networking-based cooperation mechanisms in vehicular ad-hoc network—a survey," *Vehicular Communications*, vol. 10, pp. 57–73, 2017.

[5] M. Kakkasageri and S. S. Manvi, "Information management in vehicular ad hoc networks: A review," *Journal of network and computer applications*, vol. 39, pp. 334–350, 2014.

[6] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," *Journal of computer security*, vol. 15, no. 1, pp. 39–68, 2007.

[7] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[8] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Federated learning for ultra-reliable low-latency v2v communications," in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1–7.

[9] J. Posner, L. Tseng, M. Aloqaily, and Y. Jararweh, "Federated learning in vehicular networks: Opportunities and solutions," *IEEE Network*, vol. 35, no. 2, pp. 152–159, 2021.

[10] P. Kairouz and H. B. McMahan, "Advances and open problems in federated learning," *Foundations and trends in machine learning*, vol. 14, no. 1-2, pp. 1–210, 2021.

[11] R. W. van der Heijden, S. Dietzel, T. Leinmüller, and F. Kargl, "Survey on misbehavior detection in cooperative intelligent transportation systems," *IEEE Communications Surveys Tutorials*, vol. 21, no. 1, pp. 779–811, 2019.

[12] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Generation Computer Systems*, vol. 115, pp. 619–640, 2021.

[13] J. Zhao, F. Huang, L. Liao, and Q. Zhang, "Blockchain-based trust management model for vehicular ad hoc networks," *IEEE Internet of Things Journal*, vol. 11, no. 5, pp. 8118–8132, 2024.

[14] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, J. Fernandez-Marques, Y. Gao, L. Sani, K. H. Li, T. Parcollet, P. P. B. De Gusmão *et al.*, "Flower: A friendly federated learning research framework," *arXiv preprint arXiv:2007.14390*, 2020.

[15] E. T. Martínez Beltrán, M. Q. Pérez, P. M. S. Sánchez, S. L. Bernal, G. Bovet, M. G. Pérez, G. M. Pérez, and A. H. Celdrán, "Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges," *IEEE Communications Surveys Tutorials*, vol. 25, no. 4, pp. 2983–3013, 2023.

[16] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[17] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ecdsa)," *International journal of information security*, vol. 1, no. 1, pp. 36–63, 2001.

[18] M. G. Beiró, A. Panisson, M. Tizzoni, and C. Cattuto, "Predicting human mobility through the assimilation of social media traces into mobility models," *EPJ Data Science*, vol. 5, no. 1, p. 30, 2016.

[19] J. Nzouonta, N. Rajgure, G. Wang, and C. Borcea, "Vanet routing on city roads using real-time vehicular traffic information," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 7, pp. 3609–3626, 2009.

[20] V. D. Khairnar and S. N. Pradhan, "Mobility models for vehicular ad-hoc network simulation," in *2011 IEEE Symposium on Computers Informatics*, 2011, pp. 460–465.

[21] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical review E*, vol. 62, no. 2, p. 1805, 2000.

[22] M. J. Lighthill and G. B. Whitham, "On kinematic waves ii. a theory of traffic flow on long crowded roads," *Proceedings of the royal society of london. series a. mathematical and physical sciences*, vol. 229, no. 1178, pp. 317–345, 1955.

[23] J. A. Laval and L. Leclercq, "A mechanism to describe the formation and propagation of stop-and-go waves in congested freeway traffic," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 368, no. 1928, pp. 4519–4541, 2010.

[24] D. Ngoduy, "Instability of cooperative adaptive cruise control traffic flow: A macroscopic approach," *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 10, pp. 2838–2851, 2013.

[25] J. Harri, F. Filali, and C. Bonnet, "Mobility models for vehicular ad hoc networks: a survey and taxonomy," *IEEE Communications Surveys Tutorials*, vol. 11, no. 4, pp. 19–41, 2009.

[26] M. Kezia and K. Anusuya, "Mobility models for internet of vehicles: a survey," *Wireless Personal Communications*, vol. 125, no. 2, pp. 1857–1881, 2022.

[27] J. Yoon, M. Liu, and B. Noble, "Random waypoint considered harmful," in *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428)*, vol. 2, 2003, pp. 1312–1321 vol.2.

[28] C. Bettstetter, "Smooth is better than sharp: A random mobility model for simulation of wireless networks," in *Proceedings of the 4th ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, 2001, pp. 19–27.

[29] D. S. Gaikwad and M. Zaveri, "A novel mobility model for realistic behavior in vehicular ad hoc network," in *2011 IEEE 11th International Conference on Computer and Information Technology*, 2011, pp. 597–602.

[30] P. I. Richards, "Shock waves on the highway," *Operations research*, vol. 4, no. 1, pp. 42–51, 1956.

[31] W.-L. Jin, "A multi-commodity lighthill–whitham–richards model of lane-changing traffic flow," *Transportation Research Part B: Methodological*, vol. 57, pp. 361–377, 2013.

[32] J. Sun, C. Li, J. Ding, J. Yang, and Z. Liu, "A markov chain based traffic flow control model for reducing vehicles' co2 emissions," in *2015 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, 2015, pp. 250–255.

[33] S. Krauß, "Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics," 1998.

[34] H. Ou and T.-Q. Tang, "An extended two-lane car-following model accounting for inter-vehicle communication," *Physica A: Statistical Mechanics and Its Applications*, vol. 495, pp. 260–268, 2018.

[35] S. Ding, X. Chen, Z. Fu, and F. Peng, "An extended car-following model in connected and autonomous vehicle environment: Perspective from the cooperation between drivers," *Journal of Advanced Transportation*, vol. 2021, no. 1, p. 2739129, 2021.

[36] F. Legendre, V. Borrel, M. D. de Amorim, and S. Fdida, "Modeling mobility with behavioral rules: The case of incident and emergency situations," in *Asian Internet Engineering Conference*. Springer, 2006, pp. 186–205.

[37] P. G. Gipps, "A behavioural car-following model for computer simulation," *Transportation research part B: methodological*, vol. 15, no. 2, pp. 105–111, 1981.

[38] S. Kharrazi, M. Almén, E. Frisk, and L. Nielsen, "Extending behavioral models to generate mission-based driving cycles for data-driven vehicle development," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1222–1230, 2019.

[39] H. X. Liu, X. Wu, W. Ma, and H. Hu, "Real-time queue length estimation for congested signalized intersections," *Transportation research part C: emerging technologies*, vol. 17, no. 4, pp. 412–427, 2009.

[40] P. B. Mirchandani and N. Zou, "Queuing models for analysis of traffic adaptive signal control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 1, pp. 50–59, 2007.

[41] J. Kim and S. Bohacek, "A survey-based mobility model of people for simulation of urban mesh networks," *Proc. MeshNets*, pp. 1–11, 2005.

[42] Q. Zheng, X. Hong, and J. Liu, "An agenda based mobility model," in *39th Annual Simulation Symposium (ANSS'06)*. IEEE, 2006, pp. 8–pp.

[43] M. Schwamborn, N. Aschenbruck, and P. Martini, "A realistic trace-based mobility model for first responder scenarios," in *Proceedings of the 13th ACM international conference on modeling, analysis, and simulation of wireless and mobile systems*, 2010, pp. 266–274.

[44] A. Förster, A. Bin Muslim, and A. Udugama, "Trails-a trace-based probabilistic mobility model," in *Proceedings of the 21st ACM international conference on modeling, analysis and simulation of wireless and mobile systems*, 2018, pp. 295–302.

[45] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "Sumo–simulation of urban mobility: an overview," in *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*. ThinkMind, 2011.

[46] L. Owen, Y. Zhang, L. Rao, and G. McHale, "Traffic flow simulation using corsim," in *2000 Winter Simulation Conference Proceedings (Cat. No.00CH37165)*, vol. 2, 2000, pp. 1143–1147 vol.2.

[47] J. Härri, F. Filali, C. Bonnet, and M. Fiore, "Vanetmobisim: generating realistic mobility patterns for vanets," in *Proceedings of the 3rd international workshop on Vehicular ad hoc networks*, 2006, pp. 96–97.

[48] G. D. Cameron and G. I. Duncan, "Paramics—parallel microscopic simulation of road traffic," *The Journal of Supercomputing*, vol. 10, no. 1, pp. 25–53, 1996.

[49] M. Fellendorf and P. Vortisch, "Microscopic traffic flow simulator vissim," in *Fundamentals of traffic simulation*. Springer, 2010, pp. 63–93.

[50] L. Smith, R. Beckman, and K. Baggerly, "Transims: Transportation analysis and simulation system," Los Alamos National Lab.(LANL), Los Alamos, NM (United States), Tech. Rep., 1995.

[51] J. Barceló and J. Casas, "Dynamic network simulation with aimsun," in *Simulation approaches in transportation analysis: Recent advances and challenges*. Springer, 2005, pp. 57–98.

[52] PTV Group, "Ptv visum – transportation planning software," https://www.ptvgroup.com, 2024, accessed: July 2025.

[53] K. W Axhausen, A. Horni, and K. Nagel, *The multi-agent transport simulation MATSim*. Ubiquity Press, 2016.

[54] Caliper Corporation, "Transmodeler – traffic simulation software," https://www.caliper.com/transmodeler, 2024, accessed: July 2025.

[55] A. R. Hamadou Adamou, J. Ben-Othman, L. Mokdad, and A. Benzerbadj, "Enhancing roundabout performance measures through a new generic simulink-based simulator," in *Proceedings of the 2023 13th International Conference on Information Communication and Management*, 2023, pp. 1–8.

[56] A. Varga and R. Hornig, "An overview of the omnet++ simulation environment," in *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, 2008, pp. 1–10.

[57] T. R. Henderson, M. Lacage, G. F. Riley, C. Dowell, and J. Kopena, "Network simulations with the ns-3 simulator," *SIGCOMM demonstration*, vol. 14, no. 14, p. 527, 2008.

[58] R. Riebl, H.-J. Günther, C. Facchi, and L. Wolf, "Artery: Extending veins for vanet applications," in *2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*. IEEE, 2015, pp. 450–456.

[59] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, 1959.

[60] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[61] L. Zhang, Q. Liu, W. Yang, N. Wei, and D. Dong, "An improved k-nearest neighbor model for short-term traffic flow prediction," *Procedia-Social and Behavioral Sciences*, vol. 96, pp. 653–662, 2013.

[62] P. Cong, Y. Xiao, X. Wan, M. Deng, J. Li, and X. Zhang, "Dacr-amtp: Adaptive multi-modal vehicle trajectory prediction for dynamic drivable areas based on collision risk," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 9, pp. 5339–5360, 2024.

[63] V. Katariya, M. Baharani, N. Morris, O. Shoghli, and H. Tabkhi, "Deeptrack: Lightweight deep learning for vehicle trajectory prediction in highways," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 18 927–18 936, 2022.

[64] V. Jakkula, "Tutorial on support vector machine (svm)," *School of EECS, Washington State University*, vol. 37, no. 2.5, p. 3, 2006.

[65] P. Shiguihara, A. D. A. Lopes, and D. Mauricio, "Dynamic bayesian network modeling, learning, and inference: A survey," *IEEE Access*, vol. 9, pp. 117 639–117 648, 2021.

[66] L. E. Peterson, "K-nearest neighbor," *Scholarpedia*, vol. 4, no. 2, p. 1883, 2009.

[67] L. Rokach and O. Maimon, "Decision trees," in *Data mining and knowledge discovery handbook*. Springer, 2005, pp. 165–192.

[68] R. Izquierdo, I. Parra, J. Muñoz-Bulnes, D. Fernández-Llorca, and M. A. Sotelo, "Vehicle trajectory and lane change prediction using ann and svm classifiers," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 1–6.

[69] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, A. M. Umar, O. U. Linus, H. Arshad, A. A. Kazaure, U. Gana, and M. U. Kiru, "Comprehensive review of artificial neural network applications to pattern recognition," *IEEE Access*, vol. 7, pp. 158 820–158 846, 2019.

[70] P. Kumar, M. Perrollaz, S. Lefèvre, and C. Laugier, "Learning-based approach for online lane change intention prediction," in *2013 IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 797–802.

[71] R. Bani-Hani, S. H. Aljbour, and M. Shurman, "Autonomous vehicles trajectory prediction approach using machine learning test," in *2023 14th International Conference on Information and Communication Systems (ICICS)*, 2023, pp. 1–6.

[72] I. Rish *et al.*, "An empirical study of the naive bayes classifier," in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, no. 22. Seattle, USA, 2001, pp. 41–46.

[73] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[74] Y. Huang, J. Du, Z. Yang, Z. Zhou, L. Zhang, and H. Chen, "A survey on trajectory-prediction methods for autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 3, pp. 652–674, 2022.

[75] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: Lstm cells and network architectures," *Neural computation*, vol. 31, no. 7, pp. 1235–1270, 2019.

[76] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[77] G. Van Houdt, C. Mosquera, and G. Nápoles, "A review on the long short-term memory model," *Artificial intelligence review*, vol. 53, no. 8, pp. 5929–5955, 2020.

[78] R. Jiang, H. Xu, G. Gong, Y. Kuang, and Z. Liu, "Spatial-temporal attentive lstm for vehicle-trajectory prediction," *ISPRS International Journal of Geo-Information*, vol. 11, no. 7, p. 354, 2022.

[79] F. Altché and A. de La Fortelle, "An lstm network for highway trajectory prediction," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 353–359.

[80] B. Kim, C. M. Kang, J. Kim, S. H. Lee, C. C. Chung, and J. W. Choi, "Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network," in *2017 IEEE 20Th international conference on intelligent transportation systems (ITSC)*. IEEE, 2017, pp. 399–404.

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

[82] M. Freitag and Y. Al-Onaizan, "Beam search strategies for neural machine translation," *arXiv preprint arXiv:1702.01806*, 2017.

[83] N. Deo and M. M. Trivedi, "Multi-modal trajectory prediction of surrounding vehicles with maneuver based lstms," in *2018 IEEE intelligent vehicles symposium (IV)*. IEEE, 2018, pp. 1179–1184.

[84] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[85] K. O'shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.

[86] J. Wu, "Introduction to convolutional neural networks," *National Key Lab for Novel Software Technology. Nanjing University. China*, vol. 5, no. 23, p. 495, 2017.

[87] V. Bharilya and N. Kumar, "Machine learning for autonomous vehicle's trajectory prediction: A comprehensive survey, challenges, and future research directions," *Vehicular Communications*, vol. 46, p. 100733, 2024.

[88] G. Xie, A. Shangguan, R. Fei, W. Ji, W. Ma, and X. Hei, "Motion trajectory prediction based on a cnn-lstm sequential model," *Science China Information Sciences*, vol. 63, no. 11, p. 212207, 2020.

[89] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 1549–15 498.

[90] R. Chandra, U. Bhattacharya, A. Bera, and D. Manocha, "Traphic: Trajectory prediction in dense and heterogeneous traffic using weighted interactions," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 8475–8484.

[91] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *International conference on machine learning*. PMLR, 2015, pp. 2342–2350.

[92] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. Schmid, "Vectornet: Encoding hd maps and agent dynamics from vectorized representation," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 522–11 530.

[93] B. Kim, S. H. Park, S. Lee, E. Khoshimjonov, D. Kum, J. Kim, J. S. Kim, and J. W. Choi, "Lapred: Lane-aware prediction of multi-modal future trajectories of dynamic agents," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 636–14 645.

[94] M. Fu, T. Zhang, W. Song, Y. Yang, and M. Wang, "Trajectory prediction-based local spatio-temporal navigation map for autonomous driving in dynamic highway environments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6418–6429, 2022.

[95] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of deep learning," *Neurocomputing*, vol. 452, pp. 48–62, 2021.

[96] D. Soydaner, "Attention mechanism in neural networks: where it comes and where it goes," *Neural Computing and Applications*, vol. 34, no. 16, pp. 13 371–13 385, 2022.

[97] T. Zhang and Z. Wang, "Improve the lstm trajectory prediction accuracy through an attention mechanism," in *2022 IEEE Transportation Electrification Conference Expo (ITEC)*, 2022, pp. 190–195.

[98] L. Lin, W. Li, H. Bi, and L. Qin, "Vehicle trajectory prediction using lstms with spatial–temporal attention mechanisms," *IEEE Intelligent Transportation Systems Magazine*, vol. 14, no. 2, pp. 197–208, 2022.

[99] D. Yu, H. Lee, T. Kim, and S.-H. Hwang, "Vehicle trajectory prediction with lane stream attention-based lstms and road geometry linearization," *Sensors*, vol. 21, no. 23, p. 8152, 2021.

[100] Z. Yang, Z. Gao, F. Gao, C. Shi, L. He, and S. Gu, "Intelligent vehicle moving trajectory prediction based on residual attention network," *World electric vehicle journal*, vol. 13, no. 3, p. 47, 2022.

[101] Z. Yang, D. Liu, and L. Ma, "Vehicle trajectory prediction based on lstm network," in *2022 International Conference on Artificial Intelligence and Computer Information Technology (AICIT)*, 2022, pp. 1–4.

[102] D. Cheng, X. Gu, C. Qian, C. Du, and J. Wang, "Vehicle trajectory prediction with interaction regions and spatial–temporal attention," *IEEE Access*, vol. 11, pp. 130 850–130 859, 2023.

[103] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "The performance of lstm and bilstm in forecasting time series," in *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 3285–3292.

[104] D. Qu, S. Wang, H. Liu, and Y. Meng, "A car-following model based on trajectory data for connected and automated vehicles to predict trajectory of human-driven vehicles," *Sustainability*, vol. 14, no. 12, p. 7045, 2022.

[105] S. Gao, Z. Zhao, X. Liu, Y. Jiao, C. Song, and J. Zhao, "Vehicle lane change multistep trajectory prediction based on data and cnn_bilstm model," *Journal of Advanced Transportation*, vol. 2024, no. 1, p. 7129562, 2024.

[106] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.

[107] Z. Sheng, Y. Xu, S. Xue, and D. Li, "Graph-based spatial-temporal convolutional network for vehicle trajectory prediction in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 17 654–17 665, 2022.

[108] X. Li, X. Ying, and M. C. Chuah, "Grip: Graph-based interaction-aware trajectory prediction," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 3960–3966.

[109] Y. Chang and X. Wang, "Vehicle trajectory prediction with multimodal and dynamics-aware interaction neural networks," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 12, pp. 18 059–18 072, 2024.

[110] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *arXiv preprint arXiv:1707.01926*, 2017.

[111] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," *arXiv preprint arXiv:1709.04875*, 2017.

[112] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4883–4894, 2020.

[113] Y. Huang, H. Bi, Z. Li, T. Mao, and Z. Wang, "Stgat: Modeling spatial-temporal interactions for human trajectory prediction," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6272–6281.

[114] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, 2021.

[115] K. Zhang, X. Feng, L. Wu, and Z. He, "Trajectory prediction for autonomous driving using spatial-temporal graph attention transformer," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 22 343–22 353, 2022.

[116] F. Giuliari, I. Hasan, M. Cristani, and F. Galasso, "Transformer networks for trajectory forecasting," in *2020 25th international conference on pattern recognition (ICPR)*. IEEE, 2021, pp. 10 335–10 342.

[117] H. Liao, Y. Li, Z. Li, C. Wang, Z. Cui, S. E. Li, and C. Xu, "A cognitive-based trajectory prediction approach for autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 4, pp. 4632–4643, 2024.

[118] M. Kim, B. I. Kwak, J.-U. Hou, and T. Kim, "Robust long-term vehicle trajectory prediction using link projection and a situation-aware transformer," *Sensors*, vol. 24, no. 8, p. 2398, 2024.

[119] F. Amin, K. Gharami, and B. Sen, "Trajectoformer: Transformer-based trajectory prediction of autonomous vehicles with spatio-temporal neighborhood considerations," *International Journal of Computational Intelligence Systems*, vol. 17, no. 1, p. 87, 2024.

[120] Z. Wang, J. Guo, Z. Hu, H. Zhang, J. Zhang, and J. Pu, "Lane transformer: A high-efficiency trajectory prediction model," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 4, pp. 2–13, 2023.

[121] L. Cheng, Y. Qin, K. Yang, Z. Chen, and X. Tang, "Toward safe motion planning for autonomous driving in highway," *IEEE Transactions on Vehicular Technology*, vol. 74, no. 2, pp. 2491–2502, 2025.

[122] H. Sadid and C. Antoniou, "Dynamic spatio-temporal graph neural network for surrounding-aware trajectory prediction of autonomous vehicles," *IEEE Transactions on Intelligent Vehicles*, pp. 1–14, 2024.

[123] A. Tucker and K. Marsh, "Speeding through the pandemic: Perceptual and psychological factors associated with speeding during the covid-19 stay-at-home period," *Accident Analysis & Prevention*, vol. 159, p. 106225, 2021.

[124] F. M. Shiri, T. Perumal, N. Mustapha, and R. Mohamed, "A comprehensive overview and comparative analysis on deep learning models: Cnn, rnn, lstm, gru," *arXiv preprint arXiv:2305.17473*, 2023.

[125] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.

[126] X. Cheng, R. Zhang, J. Zhou, and W. Xu, "Deeptransport: Learning spatial-temporal dependency for traffic condition forecasting," in *2018 International Joint Conference on Neural Networks (IJCNN)*, 2018, pp. 1–8.

[127] H. Cui, T. Nguyen, F.-C. Chou, T.-H. Lin, J. Schneider, D. Bradley, and N. Djuric, "Deep kinematic models for kinematically feasible vehicle trajectory predictions," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 10 563–10 569.

[128] X. Mo, Y. Xing, and C. Lv, "Recog: A deep learning framework with heterogeneous graph for interaction-aware trajectory prediction," *arXiv preprint arXiv:2012.05032*, 2020.

[129] N. Bansal, R. S. Bali, K. Jakhar, M. S. Obaidat, N. Kumar, S. Tanwark, and J. J. P. C. Rodrigues, "Htfm: Hybrid traffic-flow forecasting model for intelligent vehicular ad hoc networks," in *ICC 2021 - IEEE International Conference on Communications*, 2021, pp. 1–6.

[130] T. M. Sakho and J. B. Othman, "Fedvanet-tp: Federated trajectory prediction model for vanets," in *2023 10th International Conference on Wireless Networks and Mobile Communications (WINCOM)*, 2023, pp. 1–6.

[131] Y. Belal, S. Ben Mokhtar, H. Haddadi, J. Wang, and A. Mashhadi, "Survey of federated learning models for spatial-temporal mobility applications," *ACM Transactions on Spatial Algorithms and Systems*, vol. 10, no. 3, pp. 1–39, 2024.

[132] X. Zhou, R. Ke, Z. Cui, Q. Liu, and W. Qian, "Stfl:spatio-temporal federated learning for vehicle trajectory prediction," in *2022 IEEE 2nd International Conference on Digital Twins and Parallel Intelligence (DTPI)*, 2022, pp. 1–6.

[133] B. Li, Y. Jiang, W. Sun, W. Niu, and P. Wang, "Fedvanet: Efficient federated learning with non-iid data for vehicular ad hoc networks," in *2021 IEEE Global Communications Conference (GLOBECOM)*, 2021, pp. 1–6.

[134] M. Han, K. Xu, S. Ma, A. Li, and H. Jiang, "Federated learning-based trajectory prediction model with privacy preserving for intelligent vehicle," *International journal of intelligent systems*, vol. 37, no. 12, pp. 10 861–10 879, 2022.

[135] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the forty-first annual ACM symposium on Theory of computing*, 2009, pp. 169–178.

[136] Z. Wang and T. Yan, "Federated learning-based vehicle trajectory prediction against cyberattacks," in *2023 IEEE 29th International Symposium on Local and Metropolitan Area Networks (LANMAN)*, 2023, pp. 1–6.

[137] A. Lalitha, S. Shekhar, T. Javidi, and F. Koushanfar, "Fully decentralized federated learning," in *Third workshop on bayesian deep learning (NeurIPS)*, vol. 12, 2018.

[138] K. Jayaram, V. Muthusamy, G. Thomas, A. Verma, and M. Purcell, "Lambda fl: Serverless aggregation for federated learning," in *International Workshop on Trustable, Verifiable and Auditable Federated Learning*, vol. 9, 2022.

[139] M. A. Dinani, A. Holzer, H. Nguyen, M. A. Marsan, and G. Rizzo, "A gossip learning approach to urban trajectory nowcasting for anticipatory ran management," *IEEE Transactions on Mobile Computing*, vol. 23, no. 5, pp. 6288–6303, 2024.

[140] M. Movahedian, M. Dolati, and M. Ghaderi, "Adaptive model aggregation for decentralized federated learning in vehicular networks," in *2023 19th International Conference on Network and Service Management (CNSM)*, 2023, pp. 1–9.

[141] H. Ye, L. Liang, and G. Y. Li, "Decentralized federated learning with unreliable communications," *IEEE journal of selected topics in signal processing*, vol. 16, no. 3, pp. 487–500, 2022.

[142] C. Che, X. Li, C. Chen, X. He, and Z. Zheng, "A decentralized federated learning framework via committee mechanism with convergence guarantee," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 12, pp. 4783–4800, 2022.

[143] D. Chen, T. Deng, J. Jia, S. Feng, and D. Yuan, "Mobility-aware decentralized federated learning with joint optimization of local iteration and leader selection for vehicular networks," *Computer Networks*, vol. 263, p. 111232, 2025.

[144] B. Le Bars, A. Bellet, M. Tommasi, E. Lavoie, and A.-M. Kermarrec, "Refined convergence and topology learning for decentralized sgd with heterogeneous data," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2023, pp. 1672–1702.

[145] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent," *Advances in neural information processing systems*, vol. 30, 2017.

[146] W. Liu, L. Chen, and W. Zhang, "Decentralized federated learning: Balancing communication and computing costs," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 8, pp. 131–143, 2022.

[147] T. Sun, D. Li, and B. Wang, "Decentralized federated averaging," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 4, pp. 4289–4301, 2022.

[148] H. Tang, X. Lian, M. Yan, C. Zhang, and J. Liu, "d^2: Decentralized training over decentralized data," in *International Conference on Machine Learning*. PMLR, 2018, pp. 4848–4856.

[149] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[150] N. Lu, N. Cheng, N. Zhang, X. Shen, and J. W. Mark, "Connected vehicles: Solutions and challenges," *IEEE Internet of Things Journal*, vol. 1, no. 4, pp. 289–299, 2014.

[151] T. Z. Tahi, "Resource allocation in c-v2x: A review," *arXiv preprint arXiv:2401.15756*, 2024.

[152] S. U. Bhover, A. Tugashetti, and P. Rashinkar, "V2x communication protocol in vanet for co-operative intelligent transportation system," in *2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, 2017, pp. 602–607.

[153] B. Shabir, M. A. Khan, A. U. Rahman, A. W. Malik, and A. Wahid, "Congestion avoidance in vehicular networks: A contemporary survey," *IEEE Access*, vol. 7, pp. 173 196–173 215, 2019.

[154] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," *IEEE Communications Surveys Tutorials*, vol. 7, no. 2, pp. 72–93, 2005.

[155] M. Yang and Y. Yang, "An efficient hybrid peer-to-peer system for distributed data sharing," *IEEE Transactions on Computers*, vol. 59, no. 9, pp. 1158–1171, 2010.

[156] J. P. Muñoz-Gea, J. Malgosa-Sanahuja, P. Manzanares-Lopez, J. C. Sánchez-Aarnoutse, and A. M. Guirado-Puerta, "A hybrid topology architecture for p2p file sharing systems," in *International Conference on Software and Data Technologies*. Springer, 2006, pp. 220–229.

[157] R. Venkadeshan and M. Jegatha, "Super peer deployment in unstructured peer-to-peer networks," in *Advances in Computing and Information Technology: Proceedings of the Second International Conference on Advances in Computing and Information Technology (ACITY) July 13-15, 2012, Chennai, India-Volume 1*. Springer, 2012, pp. 661–669.

[158] M. Assran, N. Loizou, N. Ballas, and M. Rabbat, "Stochastic gradient push for distributed deep learning," in *International Conference on Machine Learning*. PMLR, 2019, pp. 344–353.

[159] A. Nabli, E. Belilovsky, and E. Oyallon, "A^2cid^2: Accelerating asynchronous communication in decentralized deep learning," *Advances in Neural Information Processing Systems*, vol. 36, pp. 47 451–47 474, 2023.

[160] L. Bottou *et al.*, "Online learning and stochastic approximations," *On-line learning in neural networks*, vol. 17, no. 9, p. 142, 1998.

[161] D. Davis, D. Drusvyatskiy, S. Kakade, and J. D. Lee, "Stochastic subgradient method converges on tame functions," *Foundations of computational mathematics*, vol. 20, no. 1, pp. 119–154, 2020.

[162] J. Jiang and L. Hu, "Decentralised federated learning with adaptive partial gradient aggregation," *CAAI Transactions on Intelligence Technology*, vol. 5, no. 3, pp. 230–236, 2020.

[163] D. Naik, P. Grace, N. Naik, P. Jenkins, D. Mishra, and S. Prajapat, "An introduction to gossip protocol based learning in peer-to-peer federated learning," in *2023 IEEE International Conference on ICT in Business Industry Government (ICTBIG)*, 2023, pp. 1–8.

[164] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2508–2530, 2006.

[165] V. Khatana and M. V. Salapaka, "D-distadmm: A o(1/k) distributed admm for distributed optimization in directed graph topologies," in *2020 59th IEEE Conference on Decision and Control (CDC)*, 2020, pp. 2992–2997.

[166] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson, "Optimal parameter selection for the alternating direction method of multipliers (admm): Quadratic problems," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 644–658, 2015.

[167] V. W.-H. Luk, A. K.-S. Wong, C.-T. Lea, and R. W. Ouyang, "Rrg: redundancy reduced gossip protocol for real-time n-to-n dynamic group communication," *Journal of Internet Services and Applications*, vol. 4, no. 1, p. 14, 2013.

[168] J. Tu, J. Zhou, and D. Ren, "An asynchronous distributed training algorithm based on gossip communication and stochastic gradient descent," *Computer Communications*, vol. 195, pp. 416–423, 2022.

[169] E. Hallaji, R. Razavi-Far, M. Saif, B. Wang, and Q. Yang, "Decentralized federated learning: A survey on security and privacy," *IEEE Transactions on Big Data*, vol. 10, no. 2, pp. 194–213, 2024.

[170] Y. Wan, Y. Qu, W. Ni, Y. Xiang, L. Gao, and E. Hossain, "Data and model poisoning backdoor attacks on wireless federated learning, and the defense mechanisms: A comprehensive survey," *IEEE Communications Surveys Tutorials*, vol. 26, no. 3, pp. 1861–1897, 2024.

[171] R. Gosselin, L. Vieu, F. Loukil, and A. Benoit, "Privacy and security in federated learning: A survey," *Applied Sciences*, vol. 12, no. 19, p. 9901, 2022.

[172] U. Demir, Y. E. Sagduyu, T. Erpek, H. Jafari, S. Kompella, and M. Xue, "Distributed federated learning for vehicular network security: Anomaly detection benefits and multi-domain attack threats," *arXiv preprint arXiv:2505.23706*, 2025.

[173] V. Tolpegin, S. Truex, M. E. Gursoy, and L. Liu, "Data poisoning attacks against federated learning systems," in *European symposium on research in computer security*. Springer, 2020, pp. 480–501.

[174] J. Steinhardt, P. W. W. Koh, and P. S. Liang, "Certified defenses for data poisoning attacks," *Advances in neural information processing systems*, vol. 30, 2017.

[175] M. Fang, X. Cao, J. Jia, and N. Gong, "Local model poisoning attacks to {Byzantine-Robust} federated learning," in *29th USENIX security symposium (USENIX Security 20)*, 2020, pp. 1605–1622.

[176] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," *Advances in neural information processing systems*, vol. 30, 2017.

[177] A. DasGupta, "The trimmed mean," in *Asymptotic theory of statistics and probability*. Springer, 2008, pp. 271–278.

[178] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *International conference on machine learning*. Pmlr, 2018, pp. 5650–5659.

[179] C. Feng, A. H. Celdrán, J. Von der Assen, E. T. M. Beltrán, G. Bovet, and B. Stiller, "Dart: A solution for decentralized federated learning model robustness analysis," *Array*, vol. 23, p. 100360, 2024.

[180] J. R. Douceur, "The sybil attack," in *International workshop on peer-to-peer systems*. Springer, 2002, pp. 251–260.

[181] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 2013, pp. 387–402.

[182] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *International conference on artificial intelligence and statistics*. PMLR, 2020, pp. 2938–2948.

[183] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015, pp. 1322–1333.

[184] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in pharmacogenetics: An {End-to-End} case study of personalized warfarin dosing," in *23rd USENIX security symposium (USENIX Security 14)*, 2014, pp. 17–32.

[185] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on Security and Privacy (SP)*, 2017, pp. 3–18.

[186] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *2019 IEEE Symposium on Security and Privacy (SP)*, 2019, pp. 739–753.

[187] Y. Chen, H. Chen, Y. Zhang, M. Han, M. Siddula, and Z. Cai, "A survey on blockchain systems: Attacks, defenses, and privacy preservation," *High-Confidence Computing*, vol. 2, no. 2, p. 100048, 2022.

[188] M. Conti, E. Sandeep Kumar, C. Lal, and S. Ruj, "A survey on security and privacy issues of bitcoin," *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 3416–3452, 2018.

[189] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients-how easy is it to break privacy in federated learning?" *Advances in neural information processing systems*, vol. 33, pp. 16 937–16 947, 2020.

[190] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *2019 IEEE Symposium on Security and Privacy (SP)*, 2019, pp. 691–706.

[191] C. Fung, C. J. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," *arXiv preprint arXiv:1808.04866*, 2018.

[192] S. Iqbal, P. Ball, M. H. Kamarudin, and A. Bradley, "Simulating malicious attacks on vanets for connected and autonomous vehicle cybersecurity: A machine learning dataset," in *2022 13th International Symposium on*

Communication Systems, Networks and Digital Signal Processing (CSNDSP), 2022, pp. 332–337.

[193] L. Tamilselvan and V. Sankaranarayanan, "Prevention of blackhole attack in manet," in *The 2nd international conference on wireless broadband and ultra wideband communications (AusWireless 2007)*. IEEE, 2007, pp. 21–21.

[194] J. Mirkovic and P. Reiher, "A taxonomy of ddos attack and ddos defense mechanisms," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, pp. 39–53, 2004.

[195] V. J. Manès, H. Han, C. Han, S. K. Cha, M. Egele, E. J. Schwartz, and M. Woo, "The art, science, and engineering of fuzzing: A survey," *IEEE Transactions on Software Engineering*, vol. 47, no. 11, pp. 2312–2331, 2019.

[196] Z. Xu, F. Jiang, L. Niu, J. Jia, and R. Poovendran, "Poster: Brave: Byzantine-resilient and privacy-preserving peer-to-peer federated learning," in *Proceedings of the 19th ACM Asia Conference on Computer and Communications Security*, 2024, pp. 1934–1936.

[197] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," in *International conference on the theory and application of cryptology and information security*. Springer, 2001, pp. 514–532.

[198] X. Yuan, J. Liu, B. Wang, W. Wang, B. Wang, T. Li, X. Ma, and W. Pedrycz, "Fedcomm: A privacy-enhanced and efficient authentication protocol for federated learning in vehicular ad-hoc networks," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 777–792, 2024.

[199] F. Pub, "Secure hash standard (shs)," *Fips pub*, vol. 180, no. 4, p. 2012, 2012.

[200] N. F. PUB, "202 federal information processing standards publication: Sha-3 standard: Permutation-based hash and extendable-output functions," 2015.

[201] P. FIPS, "198 (federal information processing standards publication) the keyed hash message authentication code (hmac)," *Information Technology Labratory, National Institute of Standards and Technology, Gaithersburg, MD*, pp. 20 899–8900, 2008.

[202] Y. Yahata, K. Sugiura, and H. Matsutani, "A scalable secure fault tolerant aggregation for p2p federated learning," in *2024 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, 2024, pp. 222–231.

[203] P. Papadimitratos, L. Buttyan, T. Holczer, E. Schoch, J. Freudiger, M. Raya, Z. Ma, F. Kargl, A. Kung, and J.-P. Hubaux, "Secure vehicular communication systems: design and architecture," *IEEE Communications Magazine*, vol. 46, no. 11, pp. 100–109, 2008.

[204] D. H. Yum and P. J. Lee, "Identity-based cryptography in public key management," in *European Public Key Infrastructure Workshop*. Springer, 2004, pp. 71–84.

[205] R. Matam and S. Tripathy, "Wrsr: wormhole-resistant secure routing for wireless mesh networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2013, no. 1, p. 180, 2013.

[206] R. Nath and P. K. Sehgal, "Sd-aodv: A protocol for secure and dynamic data dissemination in mobile ad hoc network," *arXiv preprint arXiv:1107.3363*, 2011.

[207] M. Erritali, B. El Ouahidi, and D. Bourget, "Secured geographic routing protocol for vehicular ad hoc networks (vanets)," in *International Conference on Networked Systems*. Springer, 2013, pp. 311–315.

[208] S. Li, E. C.-H. Ngai, and T. Voigt, "An experimental study of byzantine-robust aggregation schemes in federated learning," *IEEE Transactions on Big Data*, vol. 10, no. 6, pp. 975–988, 2024.

[209] P. Lv, L. Xie, J. Xu, X. Wu, and T. Li, "Misbehavior detection in vehicular ad hoc networks based on privacy-preserving federated learning and blockchain," *IEEE Transactions on Network and Service Management*, vol. 19, no. 4, pp. 3936–3948, 2022.

[210] Y. Gao, L. Zhang, L. Wang, K.-K. R. Choo, and R. Zhang, "Privacy-preserving and reliable decentralized federated learning," *IEEE Transactions on Services Computing*, vol. 16, no. 4, pp. 2879–2891, 2023.

[211] T. Bao, L. Xu, L. Zhu, L. Wang, R. Li, and T. Li, "Privacy-preserving collaborative filtering algorithm based on local differential privacy," *China Communications*, vol. 18, no. 11, pp. 42–60, 2021.

[212] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318.

[213] M. Asad, S. Shaukat, E. Javanmardi, J. Nakazato, N. Bao, and M. Tsukada, "Secure and efficient blockchain-based federated learning approach for vanets," *IEEE Internet of Things Journal*, vol. 11, no. 5, pp. 9047–9055, 2024.

[214] N. Prema, "Efficient secure aggregation in vanets using fully homomorphic encryption (fhe)," *Mobile networks and applications*, vol. 24, no. 2, pp. 434–442, 2019.

[215] S. Gyawali, Y. Qian, and R. Q. Hu, "A privacy-preserving misbehavior detection system in vehicular communication networks," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 6147–6158, 2021.

[216] A. C. Yao, "Protocols for secure computations," in *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*, 1982, pp. 160–164.

[217] S. Narkedimilli, R. A. Kumar, N. Kumar, R. P. Reddy *et al.*, "Fl-deco-bc: A privacy-preserving, provably secure, and provenance-preserving federated learning framework with decentralized oracles on blockchain for vanets," *arXiv preprint arXiv:2407.21141*, 2024.

[218] I. Damgård, V. Pastro, N. Smart, and S. Zakarias, "Multiparty computation from somewhat homomorphic encryption," in *Annual cryptology conference*. Springer, 2012, pp. 643–662.

[219] O. Friha, M. A. Ferrag, M. Benbouzid, T. Berghout, B. Kantarci, and K.-K. R. Choo, "2df-ids: Decentralized and differentially private federated learning-based intrusion detection system for industrial iot," *Computers & Security*, vol. 127, p. 103097, 2023.

[220] F. Mansouri, M. Tarhouni, B. Alaya, and S. Zidi, "A distributed intrusion detection framework for vehicular ad hoc networks via federated learning and blockchain," *Ad Hoc Networks*, vol. 167, p. 103677, 2025.

[221] Z. Zhang, X. Cao, J. Jia, and N. Z. Gong, "Fldetector: Defending federated learning against model poisoning attacks via detecting malicious clients," in *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, 2022, pp. 2545–2555.

[222] R. Sultana, J. Grover, M. Tripathi, and P. Sharma, "La-detects: Local and adaptive data-centric misbehavior detection framework for vehicular technology security," *IEEE Open Journal of Vehicular Technology*, vol. 6, pp. 145–169, 2025.

[223] W. Issa, N. Moustafa, B. Turnbull, N. Sohrabi, and Z. Tari, "Blockchain-based federated learning for securing internet of things: A comprehensive survey," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–43, 2023.

[224] Y. Jiang, B. Ma, X. Wang, G. Yu, P. Yu, Z. Wang, W. Ni, and R. P. Liu, "Blockchained federated learning for internet of things: A comprehensive survey," *ACM Computing Surveys*, vol. 56, no. 10, pp. 1–37, 2024.

[225] N. El Ioini and C. Pahl, "A review of distributed ledger technologies," in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer, 2018, pp. 277–288.

[226] A. R. Javed, M. A. Hassan, F. Shahzad, W. Ahmed, S. Singh, T. Baker, and T. R. Gadekallu, "Integration of blockchain technology and federated learning in vehicular (iot) networks: A comprehensive survey," *Sensors*, vol. 22, no. 12, p. 4394, 2022.

[227] J.-H. Chen, M.-R. Chen, G.-Q. Zeng, and J.-S. Weng, "Bdfl: A byzantine-fault-tolerance decentralized federated learning method for autonomous vehicle," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 9, pp. 8639–8652, 2021.

[228] I. Ali, Y. Chen, N. Ullah, M. Afzal, and W. HE, "Bilinear pairing-based hybrid signcryption for secure heterogeneous vehicular communications," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 5974–5989, 2021.

[229] L. Wang, X. Zhao, Z. Lu, L. Wang, and S. Zhang, "Enhancing privacy preservation and trustworthiness for decentralized federated learning," *Information Sciences*, vol. 628, pp. 449–468, 2023.

[230] M. Tang, Z. Huang, and G. Deng, "Fedl: Confidential deep learning for autonomous driving in vanets based on functional encryption," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 12, pp. 21 074–21 085, 2024.

[231] P. Sharma and H. Liu, "A machine-learning-based data-centric misbehavior detection model for internet of vehicles," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4991–4999, 2021.

[232] F. A. Ghaleb, M. Aizaini Maarof, A. Zainal, B. A. S. Al-Rimy, F. Saeed, and T. Al-Hadhrami, "Hybrid and multifaceted context-aware misbehavior detection model for vehicular ad hoc network," *IEEE Access*, vol. 7, pp. 159 119–159 140, 2019.

[233] C. Zhang, X. Lin, R. Lu, P.-H. Ho, and X. Shen, "An efficient message authentication scheme for vehicular communications," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 6, pp. 3357–3368, 2008.

[234] P. Liu, Q. He, Y. Chen, S. Jiang, B. Zhao, and X. Wang, "A lightweight authentication and privacy-preserving aggregation for blockchain-enabled federated learning in vanets," *IEEE Transactions on Consumer Electronics*, vol. 71, no. 1, pp. 1274–1287, 2025.

[235] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, 2019, pp. 4171–4186.

[236] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving language understanding by generative pre-training," 2018.

[237] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[238] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[239] J. Bridle, "Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters," *Advances in neural information processing systems*, vol. 2, 1989.

[240] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 315–323.

[241] D. Hendrycks, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

[242] U.S. Department of Transportation, Federal Highway Administration, "Next generation simulation (ngsim): U.s. highway 101 dataset," https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm, 2006, accessed: 2025-11-03.

[243] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[244] F. Chollet *et al.*, "Keras," https://github.com/fchollet/keras, 2015.

[245] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[246] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler *et al.*, "Api design for machine learning software: experiences from the scikit-learn project," *arXiv preprint arXiv:1309.0238*, 2013.

[247] T. O'Malley, E. Bursztein, J. Long, F. Chollet, H. Jin, L. Invernizzi *et al.*, "Keras Tuner," https://github.com/keras-team/keras-tuner, 2019.

[248] X. Chen, H. Zhang, F. Zhao, Y. Cai, H. Wang, and Q. Ye, "Vehicle trajectory prediction based on intention-aware non-autoregressive transformer with multi-attention learning for internet of vehicles," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–12, 2022.

[249] E. L. Cominetti and M. A. Simplicio, "Fast additive partially homomorphic encryption from the approximate common divisor problem," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2988–2998, 2020.

[250] S. Augenstein, A. Hard, K. Partridge, and R. Mathews, "Jointly learning from decentralized (federated) and centralized data to mitigate distribution shift," *arXiv preprint arXiv:2111.12150*, 2021.

[251] S. R. Pokhrel and J. Choi, "Federated learning with blockchain for autonomous vehicles: Analysis and design challenges," *IEEE Transactions on Communications*, vol. 68, no. 8, pp. 4734–4746, 2020.

[252] J. Postel, "User datagram protocol. rfc 768," 1980.

[253] ——, "Transmission control protocol," Tech. Rep., 1981.

[254] T. Bray, "The javascript object notation (json) data interchange format," Tech. Rep., 2014.

[255] Y. Chang and X. Wang, "Vehicle trajectory prediction with multimodal and dynamics-aware interaction neural networks," *IEEE Transactions on Vehicular Technology*, 2024.

[256] A. G. Vrahatis, K. Lazaros, and S. Kotsiantis, "Graph attention networks: a comprehensive review of methods and applications," *Future Internet*, vol. 16, no. 9, p. 318, 2024.

[257] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Conditional variational autoencoder for prediction and feature recovery applied to intrusion detection in iot," *Sensors*, vol. 17, no. 9, p. 1967, 2017.

[258] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[259] S. Bai, J. Z. Kolter, and V. Koltun, “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling,” *arXiv preprint arXiv:1803.01271*, 2018.

[260] G. Welch, G. Bishop *et al.*, “An introduction to the kalman filter,” 1995.