Technical Report Documentation Page

1. Report No.	**2. Government Accession No.** N/A	**3. Recipient's Catalog No.** N/A
4. Title and Subtitle Privacy-preserving Machine Learning Models for Traffic Forecasting		**5. Report Date** November 17, 2025
		6. Performing Organization Code N/A
7. Author(s) Mahmoud N Mahmoud, Ph.D. https://orcid.org/0000-0003-3059-7912 Adom Isaac		**8. Performing Organization Report No.** N/A
9. Performing Organization Name and Address North Carolina Agricultural and Technical State University, 1601 E Market St, Greensboro, NC 27411		**10. Work Unit No. (TRAIS)** N/A
		11. Contract or Grant No. 69A3552348327
12. Sponsoring Agency Name and Address CARMEN+ University Transportation Center The Ohio State University, 930 Kinnear Rd., Columbus, OH 43212		**13. Type of Report and Period Covered** Final (June 2023 to Oct 2025)
		14. Sponsoring Agency Code N/A
15. Supplementary Notes Conducted in cooperation with the U.S. Department of Transportation, Federal Highway Administration.		

16. Abstract

This report presents a comprehensive investigation into privacy-preserving traffic management, explainable artificial intelligence for autonomous systems, and cybersecurity in AV control. The research addresses critical challenges facing Intelligent Transportation Systems (ITS) and autonomous vehicles through four interconnected contributions. We introduce a secure and privacy-preserving traffic forecasting framework that combines Inner Product Functional Encryption (IPFE) with k-anonymity mechanisms to protect driver location data while enabling accurate traffic flow prediction through a hybrid deep learning architecture. We apply Concept Relevance Propagation (CRP), a bias-resistant explainable AI technique, to provide transparent concept-level explanations for traffic detection models in autonomous vehicles, enhancing trust and interpretability. We leverage CRP-generated explanations to automate dataset annotation for perception models, significantly reducing manual labeling effort while producing datasets that yield superior model performance. Finally, we develop an explainability-guided detection framework for trojan backdoor attacks in regression-based AV steering networks, achieving high detection rates for visible triggers and strong resilience against stealthy invisible variants. Together, these contributions establish new benchmarks for trustworthy AI in transportation, addressing fundamental challenges in data privacy, model transparency, and system security while demonstrating practical applicability for real-world deployment.

17. Key Words		18. Distribution Statement	
Privacy-preserving Machine Learning, Traffic Forecasting, Functional Encryption		No restrictions. This document is available to the public through NTIS: National Technical Information Service Springfield, Virginia 22161	
19. Security Classif.(of this report)	**20. Security Classif.(of this page)**	**21. No. of Pages**	**22. Price**
Unclassified	Unclassified	99	N/A

Form DOT F 1700.7 (8-72) **Reproduction of completed page authorized**

CARMEN+

Center for Automated Vehicles Research with Multimodal Assured Navigation

USDOT University Transportation Centers Program

Final Report: Privacy-preserving Machine Learning Models for Traffic Forecasting

P.I.	Project Info:
Mahmoud N Mahmoud	Grant No. 69A3552348327
North Carolina Agriculture and Technical State University	DUNS: 832127323
Electrical and Computer Engineering	EIN #: 31-6025986
	Project Effective: August 1, 2024 Project End: August 30, 2025 Submission: September 30, 2025

Consortium Members:

The Ohio State University College of Engineering · TEXAS The University of Texas at Austin · UCI University of California, Irvine · North Carolina Agricultural and Technical State University

DISCLAIMER

Abstract

This report examines privacy-preserving traffic management, explainable artificial intelligence for autonomous systems, and cybersecurity in AV control. The work addresses challenges in Intelligent Transportation Systems (ITS) and autonomous vehicles through four related contributions. We present a secure and privacy-preserving traffic forecasting framework that combines Inner Product Functional Encryption (IPFE) with k-anonymity mechanisms to protect driver location data while enabling accurate traffic flow prediction using a hybrid deep learning architecture. We use Concept Relevance Propagation (CRP), a bias-resistant explainable AI technique, to provide transparent concept-level explanations for traffic detection models in autonomous vehicles, improving trust and interpretability. We use CRP-generated explanations to automate dataset annotation for perception models, reducing manual labeling effort while producing datasets that improve model performance. We also present an explainability-guided detection framework for trojan backdoor attacks in regression-based AV steering networks, achieving high detection rates for visible triggers and strong resilience against stealthy invisible variants. These contributions address challenges in data privacy, model transparency, and system security while showing practical applicability for real-world deployment.

Executive Summary

This report synthesizes four research contributions addressing critical challenges in Intelligent Transportation Systems (ITS) and autonomous vehicle (AV) technologies under the U.S. Department of Transportation's University Transportation Centers Program. The overarching theme is the development of robust, transparent, and secure AI-driven solutions that balance operational effectiveness with privacy protection, explainability, and cybersecurity. Chapter 2 presents a novel framework that integrates Inner Product Functional Encryption (IPFE) with k-anonymity to enable secure traffic forecasting while protecting sensitive driver location data, achieving high forecasting accuracy while maintaining strong privacy guarantees against collusion attacks. Chapter 3 introduces Relevance-Based Explainable AI (RB-XAI) using Concept Relevance Propagation (CRP) to provide transparent, concept-level explanations for traffic detection models in autonomous systems, enhancing trust and enabling regulatory compliance. Chapter 4 extends XAI applications to automate dataset annotation for perception models, significantly reducing manual labeling effort while producing annotations that yield superior model performance. Chapter 5 presents an explainability-guided framework for detecting trojan backdoor attacks in regression-based AV steering networks, achieving high detection rates for visible triggers and strong resilience against stealthy invisible variants. These contributions address DOT strategic priorities by enabling proactive congestion management while protecting citizen privacy, enhancing transparency and trust in autonomous systems, reducing barriers to perception model development, and strengthening cybersecurity for safety-critical applications.

Table of Contents

Chapter 1: Introduction

1.1 Motivation and DOT Relevance

Traffic congestion remains a critical challenge for modern transportation systems, with profound impacts on productivity, quality of life, economic activity, and the environment. Prolonged travel times, wasted fuel, increased operational costs, and heightened carbon emissions highlight the inadequacy of traditional congestion mitigation strategies in keeping pace with urbanization and the rapid growth of vehicle ownership. Recent data analytics underscore the severity of the problem, with major U.S. cities experiencing annual productivity losses exceeding $1,800 per driver and over 100 hours wasted in traffic. These trends reinforce the urgency of deploying innovative congestion management strategies that move beyond infrastructure expansion or traditional punitive measures.

Intelligent Transportation Systems (ITS) have emerged as the cornerstone of this effort, enabling real-time data collection, analysis, and decision-making through advances in sensing, communications, and artificial intelligence. Within ITS, Vehicular Ad-Hoc Networks (VANETs) are particularly promising, as this technology leverages vehicle onboard computing and vehicle-to-infrastructure (V2I) communication to enhance traffic management, improve road safety, and optimize overall transportation efficiency. However, the very data required for effective traffic forecasting—the spatiotemporal routes of drivers—are highly sensitive. The temporal mobility information of each driver qualifies as behavioral biometric signatures, uniquely identifying individuals and exposing personal routines. Consequently, ensuring privacy protection in VANET-based traffic management is not only a technical necessity but also a prerequisite for user trust and system adoption.

Simultaneously, the deployment of artificial intelligence in mission-critical sectors like transportation has enabled Autonomous Vehicles (AVs) to leverage deep learning-based models for real-time perception and control. AVs operate across four key phases: perception, localization, planning, and control, relying on sensors like LiDAR and RADAR. The perception phase is fundamental, involving complex deep learning tasks like road surface extraction and object recognition, which require extensive, detailed dataset annotation. However, despite significant advancements in AV technology, complete public acceptance remains a challenge due to the "black box" nature of their decision-making processes. This opacity undermines trust and raises concerns about transparency, regulatory compliance, accountability, safety, and security, issues that have become even more pressing in light of recent AV incidents.

1.2 Traffic Forecasting and Privacy-Preserving Systems

Short-term traffic flow forecasting primarily focuses on predicting traffic flow conditions in a few or hundreds of minutes. As a prominent research area in ITS, traditional short-term traffic flow forecast methods often face limitations in accuracy and reliability. Statistical models or early deep learning approaches like Stacked Autoencoder (SAE), Convolutional Neural Network (CNN), or Long Short-Term Memory (LSTM) have struggled to fully capture the nonlinear, stochastic relationship between traffic flow and time influenced by environmental and behavioral variability. While several works propose hybrid deep learning algorithms to jointly model spatial, temporal, and periodic features of traffic flow prediction, they often treat these aspects independently and, critically, neglect the privacy of the underlying driver data.

To address these concerns, Chapter 2 presents a secure and efficient privacy-preserving traffic forecasting framework that integrates advanced cryptographic mechanisms with deep learning. The proposed scheme divides the traffic management area into cells (geographic regions), each assigned a unique identification number, where drivers report encrypted location data to the Traffic Management Center (TMC). Utilizing functional encryption, the TMC aggregates the encrypted location data while revealing only minimal information, which serves as input to a multilayer deep learning model for traffic flow prediction. This model identifies and extracts hidden characteristics within the input traffic flow data, constructing a traffic density map that highlights probable regions of congestion. Importantly, the encrypted reports and decryption utilize functional encryption keys, k-anonymous reporting, and safeguards against collusion attacks, ensuring that no subset of entities can compromise driver privacy.

1.3 Explainable AI for Autonomous Vehicle Perception

The emerging field of eXplainable Artificial Intelligence (XAI) presents an opportunity to make AI decisions in AVs understandable to humans. However, XAI techniques are not widely adopted in the AV sector, leading to missed opportunities to improve transparency and safety within AV systems. While some studies have surveyed the advantages, challenges, and methods of integrating different XAI techniques into the AV domain, the focus of contemporary research on XAI for autonomous systems has primarily centered on explaining the behavior of models in tasks like semantic segmentation and object detection, utilizing traditional attribution-based XAI techniques like LIME, SHAP, Saliency Maps, and GRAD-CAM.

Chapter 3 addresses this gap by employing Concept Relevance Propagation (CRP), a bias-resistant relevance-based XAI algorithm, to provide transparent concept-level explanations for the behavior of traffic detection models used in AVs for traffic perception. CRP, an advanced approach extending Layerwise Relevance Propagation (LRP), goes beyond traditional attribution maps by generating explanations that automatically identify and visualize relevant concepts within the input space. This insight sheds light on the crucial latent concepts and areas responsible for the behavior of traffic detection models used in AVs, aiming to boost transparency, understanding, and trust in autonomous systems.

1.4 Automated Dataset Annotation via Explainable AI

As AI advances, building efficient models requires extensive, diverse datasets, increasing the need for annotated data. Manual annotation is time-consuming, costly, and often prone to inconsistencies, especially when facing real-world complexities. While companies offer data annotation and management services that streamline computer vision workflows, these services are costly and still depend on manually annotated datasets for pre-training, especially when applying auto-labeling features to new, custom datasets, where performance remains minimal.

Chapter 4 addresses the dual challenge of transparency and automated annotation in AV perception model development by introducing a novel framework leveraging the bias-resistant Concept Relevance Propagation (CRP) XAI technique. This framework enhances model interpretability and automates dataset annotation for perception tasks. By integrating Relevance Maximization, CRP provides transparent explanations by pinpointing highly critical concepts and input regions used for network encodings that influence object detection. Additionally, the approach combines CRP with semi-supervised learning to generate high-quality automated annotations, significantly streamlining the annotation process and reducing manual effort. Results show that models trained on auto-annotated data achieve higher mAP scores with lower latency than models trained on pre-annotated datasets, offering a faster, more cost-effective solution for perception model development.

1.5 Functional Encryption and Cryptographic Components

The privacy-preserving traffic forecasting framework presented in Chapter 2 relies on advanced cryptographic mechanisms, specifically Inner Product Functional Encryption (IPFE). Functional encryption allows for the encryption of messages while enabling a designated decryptor to compute the output of a function on the encrypted message using a decryption key without being able to learn the message itself. IPFE, a specific type of functional encryption, allows for the computation of the inner product of two encrypted vectors, enabling secure aggregation of driver location data while preserving individual privacy.

The cryptographic design incorporates k-anonymity mechanisms, where each driver encrypts their true location cell along with k-1 dummy cells, embedding the actual location within a broader anonymity set. This approach, combined with fresh random nonces in every encryption, enforces semantic security and prevents ciphertext correlation or trajectory inference. The scheme guarantees confidentiality of individual location reports while still enabling the TMC to perform aggregate traffic forecasting, establishing a foundation for trustworthy, privacy-preserving traffic management systems.

1.6 Trojan Detection in Autonomous Vehicle Control Systems

The deployment of artificial intelligence in critical infrastructure systems has enabled AVs to use sophisticated deep neural networks that fuse inputs from LiDAR, RADAR, vision, and inertial sensors for real-time steering control. While this data-driven autonomy improves adaptability in dynamic traffic scenes, it also broadens the system's attack surface. Among these threats, trojan backdoor attacks—stealthy malicious manipulations embedded during training—can covertly hijack model behavior, forcing dangerous trajec-

tory deviations. Exacerbating this risk is the opaque nature of DNN systems, where non-intuitive latent representations obscure effective analysis and regulatory auditing.

Chapter 5 presents an explainability-guided detection framework designed for regression-based AV steering control systems, addressing security gaps in existing defenses. The approach repurposes Grad-CAM and Concept Relevance Propagation (CRP) as active security tools, generating multi-level spatial and conceptual attribution maps that expose the rationale behind steering decisions. By analyzing explanations from benign and trojaned samples across varying poisoning rates, the framework reveals telltale indicators of backdoor compromise like saliency drift, spatial deformation, and conceptual divergence. These explanation-derived features empower lightweight binary classifiers that detect trojaned behavior with high fidelity, without requiring prior knowledge of trigger patterns or access to clean reference datasets.

1.7 Research Contributions Across Chapters

This report presents a comprehensive investigation into privacy-preserving traffic management, explainable artificial intelligence for autonomous systems, and cybersecurity in AV control. The contributions span four interconnected research areas:

- **Privacy-Preserving Traffic Forecasting:** A novel framework combining functional encryption with deep learning for secure, accurate traffic flow prediction while protecting driver location privacy.
- **Explainable AI for Traffic Detection:** Application of Concept Relevance Propagation (CRP) to provide transparent, concept-level explanations for traffic detection models in autonomous vehicles.
- **Automated Dataset Annotation:** Leveraging XAI techniques to automate dataset annotation for perception models, significantly reducing manual labeling effort while maintaining high quality.
- **Trojan Detection via Explainability:** An explainability-guided framework for detecting trojan backdoor attacks in regression-based AV control systems, bridging the gap between classification-oriented detectors and continuous-output control models.

1.8 Report Organization

The remainder of this report is organized as follows. Chapter 2 presents the privacy-preserving traffic forecasting framework using functional encryption and deep learning. Chapter 3 describes the relevance-based explainable AI approach for traffic detection in autonomous systems. Chapter 4 details the automated dataset annotation framework leveraging explainable AI. Chapter 5 presents the explainability-guided trojan detection framework for AV control systems. Finally, Chapter 6 synthesizes the cross-chapter insights, emphasizing shared themes, privacy-utility tradeoffs, machine learning contributions, cryptographic innovations, and real-world DOT impact.

Chapter 2: Privacy-Preserving Traffic Forecasting Using Functional Encryption and Deep Learning

2.1 Introduction

Traffic congestion remains a critical challenge for modern transportation systems, with profound impacts on productivity, quality of life, economic activity, and the environment. Prolonged travel times, wasted fuel, increased operational costs, and heightened carbon emissions[39] highlight the inadequacy of traditional congestion mitigation strategies in keeping pace with urbanization and the rapid growth of vehicle ownership. Recent Inrix location-based data analytic report[47], underscore the severity of the problem, with major U.S. cities like Chicago and New York being the top-ranked cities in 2024, experiencing annual productivity losses exceeding $1,800 per driver and over 100 hours wasted in traffic. These trends reinforce the urgency of deploying innovative congestion management strategies that move beyond infrastructure expansion or traditional punitive measures.

With this challenge, Intelligent Transportation Systems (ITS) have emerged as the cornerstone of this effort[87], enabling real-time data collection, analysis, and decision-making through advances in sensing, communications, and artificial intelligence. Within ITS, Vehicular Ad-Hoc Networks (VANETs) are particularly promising, as this technology leverages vehicle onboard computing and vehicle-to-infrastructure (V2I) communication, as shown in Fig. 2.1 to enhance traffic management, improve road safety, and optimize overall transportation efficiency[21]. Thus, contemporary research focuses on designing preventive techniques that take advantage of VANET to reduce congestion[52], [81], [112], [114]. Unlike conventional navigation applications that are reactive to congestion and rely on potentially biased user reports, VANET based traffic management systems allow for proactive predictive modeling based on continuous real-time location updates from participating drivers to alleviate traffic congestion issues. This capability enables traffic management centers (TMCs) to analyze and aggregate timely traffic patterns from driver location reports to construct dynamic density maps, identify emerging congestion hotspots, and proactively guide drivers through alternative route recommendations before bottlenecks materialize, thus improving traffic flow. However, the very data required for this traffic forecast, the spatiotemporal routes of drivers, are highly sensitive. The temporal mobility information of each driver qualifies as behavioral biometric signatures, just as fingerprints[112]; they often uniquely identify individuals and expose personal routines. Consequently, knowing this temporal route information raises privacy concerns, with implications ranging from profiling by third-party entities such as insurers or travel brokers to exploitation by malicious actors. Therefore, en-

suring privacy protection in VANET based traffic management is not only a technical necessity but also a prerequisite for user trust and system adoption. Moreover, it should be noted that various trials have been explored to minimize urban congestion, such as enhancing transportation infrastructure, charging traffic fines, offering route information, enforcing traffic regulations, and boosting public transportation[68], [93]. Despite these efforts, several factors contribute to the persistence of congestion in urban areas. These include rapid population growth, increased vehicle ownership, and the lure of urban centers for economic opportunities. Yet, addressing urban congestion while preserving the drivers' privacy remains a complex and ongoing challenge.

Also, short-term traffic flow forecast primarily focuses on predicting traffic flow condition in a few or hundreds of minutes. As a prominent research area in ITS, traditional short-term traffic flow forecast methods often face limitations in accuracy and reliability. For instance, statistical models or early deep learning (DL) approaches like Stacked Autoencoder (SAE), Convolutional Neural Network (CNN) or Long Short Term Memory (LSTM)[28], [46], [64], [69], [117], have struggled to fully capture the nonlinear, stochastic relationship between traffic flow and time influenced by environmental and behavioral variability. While several works propose hybrid DL algorithms to jointly model spatial, temporal, and periodic features of traffic flow prediction, they often treat these aspects independently and, critically, neglect the privacy of the underlying driver data.

To address these concerns, we propose a secure and efficient privacy-preserving traffic forecasting framework that integrates advanced cryptographic mechanisms with deep learning. Our proposed scheme divides the traffic management area into cells (geographic regions) each assigned a unique identification number (ID), where drivers report encrypted location data to the TMC. Utilizing functional encryption (FE), the TMC then aggregates the encrypted location data while revealing only the minimal information, this serves as input to our multilayer DL model for traffic flow prediction. This model identifies and extracts hidden characteristics within the input traffic flow data, constructing a traffic density map (i.e. heat map) that highlights probable regions of congestion. Drivers can then reroute their journey to avoid these regions. Importantly, the encrypted reports and decryption performed within the scheme utilizes a variety of functional encryption keys, k-anonymous reporting, and a single decryption key, incorporating safeguards against collusion attacks, ensuring that no subset of entities can compromise driver privacy by combining partial keys or ciphertexts. The proposed holistic collusion-resistant research bridges a significant gap in designing an efficient VANET traffic management system, particularly in major and mid-sized cities experiencing rapid urbanization and traffic congestion. The primary contributions of this work are enumerated as follows:

1. We propose a novel privacy-preserving location reporting scheme for traffic management systems, based on Inner Product Functional Encryption (IPFE)[14]. This scheme incorporates advanced functional encryption, k-anonymity and decryption techniques to safeguard the privacy of driver route information, while allowing access to specific encrypted data. Consequently, this approach maintains the confidentiality of sensitive data, while facilitating the prediction of future traffic congestion and enabling the creation of accurate traffic forecast density maps.

2. We developed a DL-based model for predicting traffic flow, integrating a hybrid architecture that combines Convolutional Long Short-Term Memory (Conv-LSTM) to capture spatial and short-term temporal dependencies, Bidirectional LSTM (Bi-LSTM) to extract long-term periodic trends, and a

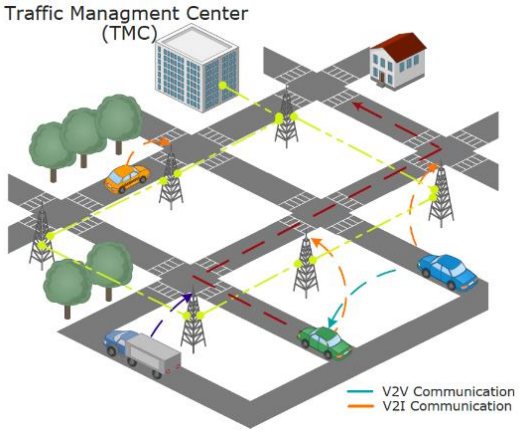


Figure 2.1: Conceptual framework of Vehicular Ad-hoc Networks.

Squeeze-and-Excitation (SE) module to enhance feature representation. Together, these components enable accurate modeling of complex traffic dynamics.

3. The proposed scheme was rigorously evaluated using both synthetic and real-world traffic data. This two stage evaluation entails: measuring the efficiency, overhead, and privacy guarantees of the encryption-based reporting scheme, including resistance to collusion, and assessing the forecasting performance of the hybrid model against both historical and contemporary research baselines.

The subsequent sections of this chapter are structured as follows: Section II reviews literature, while Section III outlines the system models and the design objectives. Section IV details the preliminaries, while section V presents a thorough overview of our proposed privacy-preserving traffic forecast system. The privacy and security analysis and performance evaluation are provided in sections VI and VII, respectively. Finally, Section VIII summarizes the conclusions drawn from our study.

2.2 Related Work

2.2.1 Privacy-Preserving Route Reporting

Recent advancements in literature have introduced a variety of privacy-preserving route reporting mechanisms for ITS. Most of these studies utilize pseudonyms, homomorphic encryption (HE), differential privacy (DP) and k-anonymous algorithms, further enhanced by blockchain technologies to bolster data integrity and privacy. These innovations, detailed in studies[41], [50], [60], [80], [113], aim to enhance data integrity and privacy. Furthermore, other traffic management techniques leverage transferable Federated Learning (FL) and Graph Convolutional Network (GCN) approaches[101], [107] for crowdsensed data, have emerged as cutting-edge solutions for addressing the challenges of data scarcity and improving traffic management efficiency while safeguarding the privacy of crowdsourced data in ITS. While these strategies significantly contribute to preserving user identity and sensitive information protection, they encounter notable limitations including scalability issues, system complexity, blockchain overhead, heightened security vulnerabilities, and dependency on internet connectivity. Moreover, these strategies may incur substantial costs and

present considerable barriers during their adoption and integration within the existing traffic management infrastructure. As such, while promising, these schemes may fall short in addressing the nuanced demands of dynamic, real-time traffic management scenarios, underscoring the need for continued innovation and adaptation in this rapidly evolving field.

2.2.2 Deep Learning for Traffic Forecasting

Traffic flow forecasting initially included three primary model types: parametric, non-parametric, and hybrid. Parametric models like ARIMA excel in analyzing time series data for traffic forecasting on expressways and urban roads[38], [54], with innovations such as Kohonen-ARIMA (KARIMA)[100] subset ARIMA[53], and seasonal ARIMA[104] enhancing their precision for nonlinear data. Non-parametric models, including K-Nearest Neighbor (KNN) and Support Vector Regression (SVR)[30], adapt well to complex data relationships but can face optimization hurdles and susceptibility to local minima.. Hybrid models combine the strengths of both, using techniques from ARIMA, Empirical Mode Decomposition (EMD), Singular Value Decomposition (SVD), and Neural Networks (NNs) to achieve superior accuracy and robustness in predicting traffic flow[72], [96]. DL further advances traffic flow prediction with Lv et al.[64] showcasing the effectiveness of Stacked Autoencoders (SAEs) in surpassing traditional methods like Support Vector Machines (SVMs) and Feedforward Neural Networks (FNNs) in estimating traffic flows. DL models in traffic flow forecasting, can be segmented into short-term, long-term, and hybrid models. For short-term predictions, DL models incorporating CNNs, GCNs, and their variants have been effective in capturing spatial-temporal traffic patterns[66], [67], [69], [102], [115], yet they struggle with temporal sequence data, where past information crucially predicts future outcomes. The introduction of LSTM networks by Tian et al.[98] highlighted their superiority in capturing temporal dynamics, paving the way for subsequent variants[65], [116] that further illustrate LSTMs' proficiency in long-term forecasting. However, these models often overlook the impact of road network layouts. Hybrid models[20], [117] merging CNNs for spatial insight and LSTMs for temporal analysis have markedly improved traffic prediction, merging the strengths of both to enhance traffic management. Nonetheless, the success of these sophisticated models hinges on the quality and availability of traffic data. The process of data collection and analysis, especially from motorists and connected vehicles, raises significant user privacy and data security concerns, necessitating stringent data protection protocols that adds complexity to these forecasting systems.

Despite the scarcity or limited endeavors in both research and development to fully address the dual challenges of ensuring user privacy and data security in traffic data collection and creating dependable traffic forecasting systems, existing studies offer promising directions. For instance, Xia et al. [107] present a system that combines GCN with FL for modeling traffic patterns. This approach utilizes GCN for identifying spatial dependencies in traffic data and employs FL for privacy-preserving collaborative learning without sharing raw data. Though ingenuous, this innovative system grapples with hurdles, including scalability issues, communication bottlenecks, susceptibility to adversarial threats, integration complexities with existing systems, and limited adaptability across different environments, highlighting the need for further research. To overcome the limitations identified in existing research and offer a holistic solution, we introduce a novel, lightweight privacy-preserving traffic forecasting system. Our system uniquely leverages functional encryption based on cryptography for scalable, internet-independent, and efficient privacy-preserving solution. It enables intricate encryption and computation on encrypted traffic data, safeguarding data security

and privacy without compromise. Further enriching our solution, we incorporate a hybrid Conv-LSTM and Bi-LSTM model with an SE module, enhancing the extraction and analysis of crucial temporal-spatial dynamics, alongside short-term and long-term traffic patterns. This approach significantly boosts the forecast accuracy and precision, setting a new benchmark for traffic forecasting systems in terms of privacy preservation and operational efficiency with exceptional forecast reliability.

2.3 System Models and Design Objectives

This section provides an overview of the system model, which includes the network model, threat model, and the proposed scheme's design goals.

2.3.1 Network Model

As shown in Fig. 2.2, our considered network model includes three main entities: the vehicle-side (drivers), traffic management center (TMC), and a key distribution center (KDC). The role of each entity is described below.

- *Drivers (D)*: As primary components of the traffic management system, each vehicle D sends its encrypted location information periodically to the TMC. Communication between drivers and the TMC is either direct or indirect through a gateway (Roadside unit). A set of D, $\mathbb{D} = \{D_i, 1 \leq i \leq |\mathbb{D}|\}$, form the network.

- *TMC*: As the central control and monitoring hub, the TMC uses encrypted location information from drivers for traffic flow analysis, congestion detection, and route planning in real time.

- *KDC*: The KDC is a crucial offline entity responsible for preserving secure communication and data privacy by providing drivers D and the TMC, respectively, with unique encryption and functional decryption keys.

2.3.2 Threat Model

We adopt an honest but curious adversarial model for our privacy-preserving traffic management system, involving three entities: the KDC, the TMC, and the Drivers. The KDC functions as an offline, setup-only authority responsible for initializing cryptographic keys. The TMC is assumed to compute traffic aggregates correctly but may act curiously by attempting to infer sensitive information such as driver locations, trajectories, or mobility patterns. Drivers are generally honest in submitting anonymized traffic reports (the specific anonymization mechanism lies outside the scope of this chapter), yet adversarial behavior may arise if a subset of drivers colludes with one another or with the TMC to extract private information about non-colluding participants. We also consider external adversaries $\mathscr{A}$ that may attempt to eavesdrop on, manipulate, or inject traffic data through the communication channels between drivers and the TMC. Finally, we assume the KDC does not collude post-setup; collusion involving a malicious KDC would compromise confidentiality in a single-authority functional encryption setting like ours. This threat model reflects realistic risks in vehicular networks, focusing on three principal adversaries; a curious TMC, colluding drivers, and external adversaries, with TMC–driver collusion representing the strongest practical adversary.

2.3.3 Design Goals

In our proposed scheme, we anticipate achieving the following objectives.

- *Privacy Preservation*: Our design seeks to develop robust mechanisms that protect the location and identity of drivers (via unauthorized access and monitoring prevention) while enabling effective traffic management and congestion mitigation.
- *Real-time Traffic Forecasting*: By leveraging advanced predictive models' accuracy and real-time traffic forecasting capabilities, our system aims to provide reliable and informed traffic data, enabling proactive congestion management and efficient route planning.
- *Scalability and Efficiency*: Our system is designed to be scalable for real-time deployment and operation under dynamic traffic conditions. This scalability extends to accommodating increasing drivers, expanding map sizes, and handling various network loads, ensuring efficient performance with minimal latency and computational overhead.
- *Secure Communication*: With secure communication channels, our scheme utilizes secure protocols and cryptography techniques to guarantee the integrity, security, and confidentiality of data exchanged between system entities.

2.4 Preliminaries

2.4.1 Functional Encryption

Functional encryption (FE) refers to a type of cryptography that allows for the encryption of a message x using a key k to get $Enc_k(x)$, as well as the ability of a designated decryptor to compute the output of a function f on the encrypted message using a decryption key dk without being able to learn the message itself (i.e., $Dec_{dk}(Enc_k(x)) = f(x)$) [13]. Recently, the focus on FE has been increasing, especially on how to design efficient schemes for limited classes of functions or polynomials, such as linear [1], [7] or quadratic [11]. in this chapter, we focus on a specific type of functional encryption known as inner product functional encryption (IPFE)[14], which allows for the computation of the inner product of two encrypted vectors. In an IPFE framework, when provided with the encryption of a vector x and a functional decryption key linked to a vector y, one can exclusively derive the dot product result $(x \cdot y)$ by decrypting the encrypted form of x, all without gaining access to the actual values of x. IPFE involves three distinct parties, outlined as follows.

- *KDC*: The KDC produces an encryption key for the encryptor and a single functional decryption key for the decryptor.
- *Encryptor*: The encryptor encrypts the plaintext vector x into the ciphertext and sends it to the decryptor.
- *Decryptor*: The decryptor uses the functional decryption key dk_y obtained from the KDC to evaluate and access $(x \cdot y)$, where x and y are the plaintext vector and the encrypted vector, respectively. The decryptor is obliged to maintain non-collusion with the KDC.

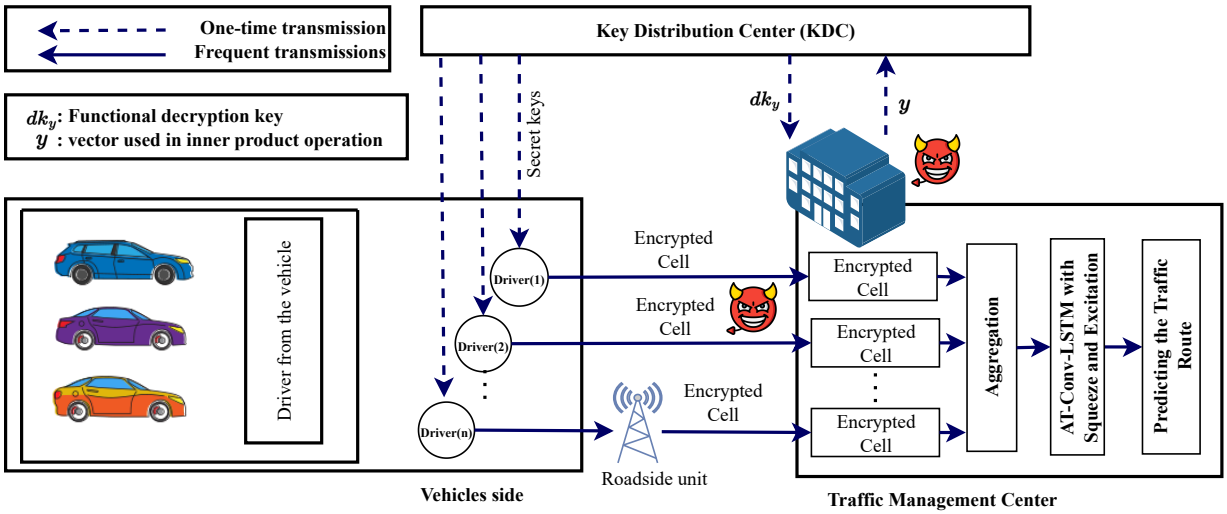


Figure 2.2: Illustration of the Privacy-Preserving Traffic Management System

2.4.2 Convolution/ LSTM and Bi-LSTM

CNN and LSTM are powerful deep-learning architectures widely used in computer vision and natural language processing. CNNs use a combination of convolutional layers, pooling layers, and fully-connected layers to extract features from an image and then classify the image into one of the predefined classes. CNNs are particularly suitable for object recognition, facial recognition, and image segmentation tasks. On the other hand, LSTM networks are mainly used for natural language processing tasks such as language translation, sentiment analysis, and text generation. LSTM networks are composed of multiple layers of memory units, which are responsible for storing information from the past and using it to make predictions. They are particularly powerful when understanding data sequences, such as sentences, and predicting what comes next. A combination of CNN and LSTM, known as Conv-LSTM, is usually used to improve the performance of a neural network. The wide adoption of Conv-LSTM is due to their high accuracy. The purpose of using attention-based Conv-LSTM is to make the near-future predictions accurate and timely.

2.4.3 Attention Mechanism

An attention mechanism allows deep learning models to selectively focus on certain parts of the input when making predictions. It is particularly useful in natural language processing and image recognition tasks. In these tasks, the model must be able to identify and understand specific parts of the input to make accurate predictions. The attention mechanism is implemented by adding an attention layer to the neural network, which learns to assign weights to different input parts. These weights are then used to create a weighted sum of the input, which is then passed to the next network layer. Attention mechanisms have been shown to improve the performance of neural networks on a wide range of tasks and are now widely used in many state-of-the-art models. An attention-based Conv-LSTM combines attention mechanisms and Conv-LSTMs to provide accurate forecasting.

2.4.4 Squeeze-and-excitation

Squeeze-and-excitation (SE)[44] is a type of attention mechanism that aims to improve the feature representation of a neural network. It works by first compressing the feature maps' spatial dimensions, reducing the number of channels. The resulting feature maps are then passed through an excitation module, which learns to assign weights to different channels based on their importance. These weights are then used to recalibrate the feature maps, improving the network's overall feature representation. SE has been shown to improve the performance of neural networks on various tasks such as image classification, object detection, and semantic segmentation, particularly in architectures like CNNs. It can be added to existing architectures like CNNs or convolutional LSTMs as a module.

2.5 Proposed Scheme

As depicted in Fig. 2.2, our proposed framework comprises two primary components: 1) Privacy-Preserving Location Reporting and Aggregation for Drivers, and 2) Traffic Forecasting through Deep Learning. The first component encompasses system initialization, driver location reporting, and server-side aggregation of information for traffic monitoring. The second component involves a deep learning-based traffic forecasting algorithm. Our model utilizes Conv-LSTM on aggregated driver data to predict short- and long-term traffic patterns while ensuring driver privacy. Additionally, our model incorporates an attention mechanism and a squeeze-and-excitation block, significantly improving performance. The following subsections explain the details of each building block. For clarity, the mathematical symbols used in the scheme are summarized in Table 2.1. Fig. 2.4 further illustrates the key phases of the privacy-preserving reporting and forecasting pipeline through a sequence diagram.

2.5.1 Drivers Location Reporting and Aggregation

We assume that the traffic management area is divided into a set of geographic areas called cells, as illustrated in Figure 2.3. Each cell is assigned a unique identifier, similar to zip codes.

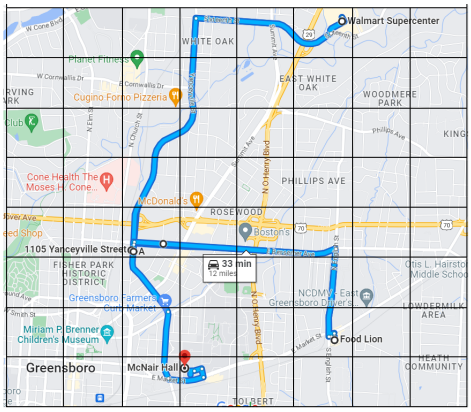


Figure 2.3: A traffic management area partitioned into distinct geographic zones (i.e., cells).

- To report their location, each $D_i \in \mathbb{D}$, where $\{1 \leq i \leq |\mathbb{D}|\}$, employ IPFE scheme[14] to conceal their association with a specific cell, denoted as $l_i^j[t] = 1$, where $1 \leq j \leq |\mathscr{L}|$, and $\mathscr{L}$ represents the total

number of grid cells within a given reporting area. Additionally, drivers encrypt the remaining $(k-1)$ dummy cells with a value of zero to maintain k-anonymity [95]. The outcome is a set of k ciphertexts, labeled as $C_i^1[t]$ through $C_i^k[t]$, which are subsequently transmitted to the decryptor (i.e. TMC). This encryption mechanism safeguards the confidentiality of the driver's precise location.

- At each reporting interval t, the TMC receives encrypted cell information $C_i^j[t]$ from all drivers and applies the functional decryption key dk to compute the aggregate driver density for each cell j (i.e. $Dec_{dk}([C_1^j[t],\ldots,C_{|\mathbb{D}|}^j[t]]) = \sum_{i=1}^{|\mathbb{D}|} l_i^j[t]$, where $l_i^j[t]$ denotes the plaintext occupancy status of grid cell j reported by D_i at time slot t). If fewer than $|\mathbb{D}|$ ciphertexts are received for a given cell, the TMC compensates with dummy ciphertexts encrypting zero, preserving consistency in the aggregate computation. Aggregation is restricted to fixed-length, non-overlapping time windows (e.g. every Δt minutes), ensuring each ciphertext contributes exactly once and eliminating overlap-based differencing attacks. The process further enforces a minimum cohort threshold γ, releasing aggregates only when at least γ distinct drivers contribute within a reporting window. γ is defined by the assumed collusion bound α (fraction of drivers that may collude with the TMC) and the required minimum of honest contributors h, and is computed as $\gamma \geq \left\lceil \frac{h}{1-\alpha} \right\rceil$. This guarantees that even if the TMC colludes with a subset of drivers, the reports of non-colluding participants remain indistinguishable within a sufficiently large anonymity set. Aggregates that do not satisfy the threshold are merged, thereby mitigating both small-cohort leakage and differencing risks, and ensuring that published outputs expose no information beyond the authorized cell-level counts.

The main phases of the route report are described as follows.

System Initialization

During system initialization, the KDC computes and distributes the following: (a) Public parameters; (b) Driver's encryption keys; and (c) TMC's functional decryption key.

a) Public Parameters Generation: To generate the public parameters, the KDC should:

Setup $\left(1^\lambda, \mathscr{F}_\mathbb{D}\right)$: The algorithm first generates secure parameters as $\mathscr{G} := (\mathbb{G}, p, g) \leftarrow \text{GroupGen}\left(1^\lambda\right)$, and then generates several samples as $a_i \leftarrow_R \mathbb{Z}_p^1, \mathbf{a}_i := (1, a_i)^\top, \forall i \in \{1,\ldots,|\mathbb{D}|\}$, in addition to $\mathbf{W}_i \leftarrow_R \mathbb{Z}_p^{1\times 2}, u_i \leftarrow_R \mathbb{Z}_p^1$. Then, it generates the master public key and master private key as

$\mathbf{mpk} := (\mathscr{G}, [\mathbf{a}_i]^1, \mathbf{W}_i\mathbf{a}_i]), \mathbf{msk} := (\mathbf{W}_i, u_i)_{i\in\{1,\ldots,|\mathbb{D}|\}}$

b) Drivers' Encryption Keys Generation: KDC constructs and distribute $|\mathbb{D}|$ encryption keys to the drivers in the network as follows: $\mathbf{pk}_i := (\mathscr{G}, [\mathbf{a}_i], [\mathbf{W}_i\mathbf{a}_i], u_i)$.

c) TMC's Functional Decryption Key Generation: To enable secure aggregation, the KDC constructs a vector of ones, denoted as $\mathbf{y}_{1\times|\mathbb{D}|}$, whose length equals the number of drivers in the network. This vector enforces that, when evaluated in an inner product with encrypted driver reports, the result corresponds to the total number of drivers present in a given grid cell j. Using this vector, the KDC computes the functional decryption key dk as:

$$dk := \mathbf{d}_i^\top \leftarrow (y_i\mathbf{W}_i)_{i\in|\mathbb{D}|}, z \leftarrow \sum_{i\in|\mathbb{D}|} y_i u_i$$

[1]Note that $[x] = g^x$. In our representation, we adopt the bracket notation implicitly from [29], which is widely recognized and used as a standard in the cryptographic community.

Table 2.1: Main notations.

Notation	Description
$\mathbb{D}$	Number of drivers
$\mathbf{pk}_i$	Encryption keys of Drivers
$\mathscr{L}$	Total number of grid cells
γ	Minimum cohort threshold
$l_i^j[t]$	Status of cell j reported by driver D_i at time t
$C_i^j[t]$	Encrypted status of cell j reported by D_i at time t
$\mathbb{G}, p, g$	Public parameters for the functional encryption
dk	Functional decryption keys
$\mathbf{X}[t^s]$	Current traffic density over $t^s - n, ..., t^s$
$\mathbf{X}[t^d]$	Daily historical traffic density over $t^d - n, ..., t^d$
$\mathbf{X}[t^w]$	Weekly historical traffic density over $t^w - n, ..., t^w$
$G[t^s]$	Output from the CNN
$H_2[t^s]$	The LSTM hidden state indicating the spatial-temporal feature for time step t^s
C, H	Channel and spatial dimensions of the Squeeze operation
$G[t^s], G^{'}[t^s]$	Output of CNN and Squeeze and excitation
$H_a[t^s]$	The output of Conv-SE-LSTM at each time step t^s
τ	Time interval
β_k	The attention value

This operation is equivalent to aggregating the secret shares across all drivers to generate a single decryption key for the TMC. Crucially, KDC issues only one functional key and strictly focuses on the aggregate function defined by $\mathbf{y}$. No alternative functional keys (e.g. sparse vectors or selector functions) are distributed. This restriction guarantees that the TMC can recover only aggregate driver densities at the cell level and is cryptographically prevented from isolating or reconstructing any individual driver's report, even under repeated queries.

Reporting Drivers Locations

For each reporting period t, driver D_i encrypts the cell j information, $\forall 1 < j < |\mathscr{L}|$, and generate the ciphertext $C_i^j[t]$. This encryption ensures that the cell information is kept private and only authorized parties can access it. Each cell information is encrypted separately, allowing the TMC to compute the aggregated reports for cell j without learning the individual reports themselves. The encrypted cell information is generated as follows.

Encrypt $\left(\mathbf{pk}_i, l_i^j\right)$: The algorithm first generates a random nonce $r_i^j \leftarrow_R \mathbb{Z}_{p\, j\in\{1,...,K\}}$ and then computes

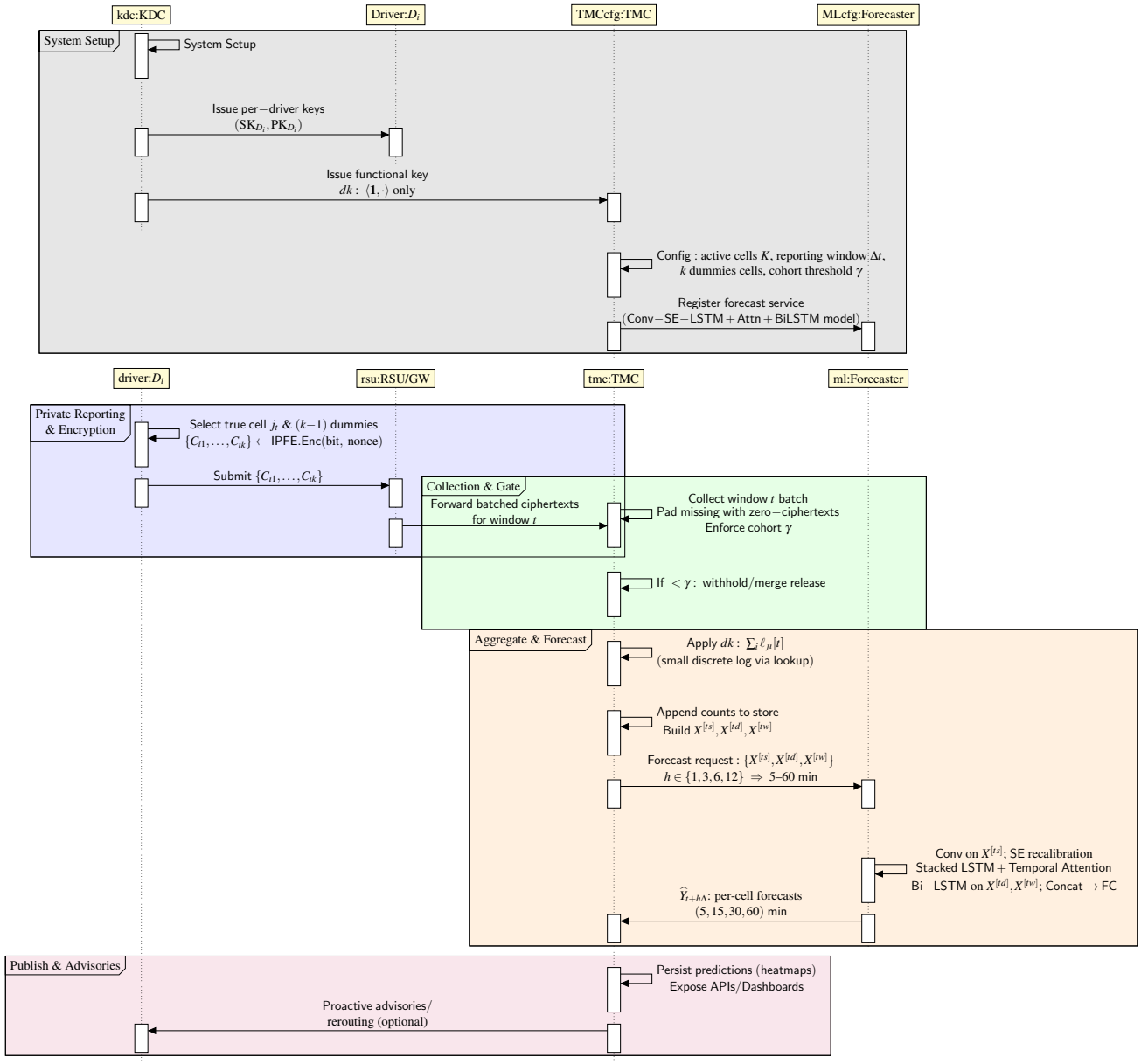


Figure 2.4: End-to-end workflow of the proposed framework, illustrating the sequential interactions from system setup, private driver reporting and encryption, secure aggregation at the TMC, to deep learning based forecasting.

the ciphertext as

$$\mathbf{C}_i^j[t] := \left(\left[t_i^j \right] \leftarrow [\mathbf{a}_i r_i^j], \left[\mathbf{c}_i^j \right] \leftarrow [l_i^j[t] + u_i + \mathbf{W}_i \mathbf{a}_i r_i^j] \right).$$

It should be noted that the drivers do not need to report the encryption status for all cells within the reporting area. Instead, they can employ K-anonymity [95] to selectively report only a subset of cells, thereby ensuring privacy and reducing computational overhead.

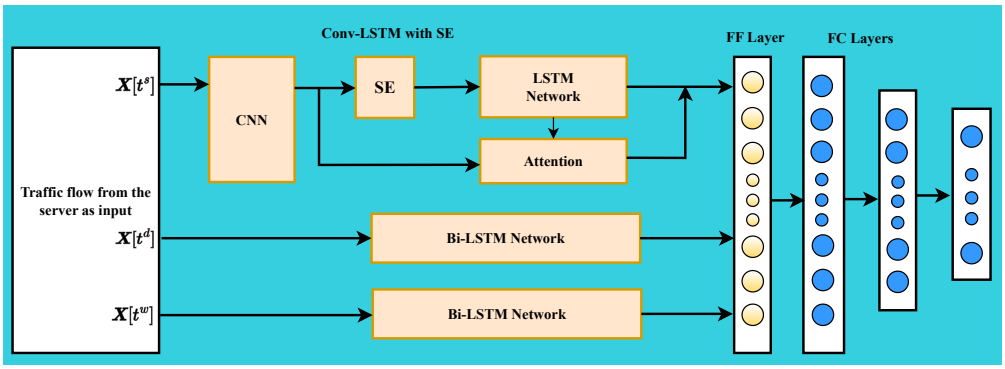


Figure 2.5: Architecture of Attention-Based Conv-LSTM Network.

Aggregating the Drivers Reports

After collecting all the D's encrypted locations (c_t) at time t, represented as $c_t = [C_1^j[t], C_2^j[t], \ldots, C_{|\mathbb{D}|}^j[t]]$, the TMC first verifies that the minimum cohort threshold γ is satisfied and then applies the functional decryption key dk to obtain the total aggregated traffic density by computing:

$$= \frac{\prod_{i\in[\mathbb{D}]} \left(\left[\mathbf{y}^\top \mathbf{c}_i\right] / \left[\mathbf{d}_i^\top t_i\right] \right)}{[z]}$$

$$= \frac{\prod_{i\in[\mathbb{D}]} \left(\left[\mathbf{y}^\top \mathbf{c}_i\right] / \left[\mathbf{y}^\top \mathbf{W}_i \mathbf{a}_i r_i^j\right] \right)}{[z]}$$

$$= \frac{\prod_{i\in[\mathbb{D}]} \left(\left[\mathbf{y}^\top (l_i^j[t] + \mathbf{u}_i + \mathbf{W}_i \mathbf{a}_i r_i^j)\right] / \left[\mathbf{y}^\top \mathbf{W}_i \mathbf{a}_i r_i^j\right] \right)}{[z]}$$

$$= \frac{\prod_{i\in[\mathbb{D}]} \left[\mathbf{y}^\top l_i^j[t] + \mathbf{y}^\top \mathbf{u}_i + \mathbf{y}^\top \mathbf{W}_i \mathbf{a}_i r_i^j - \mathbf{y}^\top \mathbf{W}_i \mathbf{a}_i r_i^j\right]}{[z]}$$

$$= \prod_{i\in[\mathbb{D}]} \left[\mathbf{y}^\top l_i^j[t] + \mathbf{y}^\top \mathbf{u}_i + \mathbf{y}^\top \mathbf{W}_i \mathbf{a}_i r_i^j - \mathbf{y}^\top \mathbf{W}_i \mathbf{a}_i r_i^j - \mathbf{y}^\top \mathbf{u}_i\right]$$

$$= \prod_{i\in[\mathbb{D}]} \mathbf{y}^\top l_i^j[t]$$

$$= \sum_{i=1}^{|\mathbb{D}|} l_i^j[t]$$

Solving the discrete logarithm is not a challenging task due to the relatively small value of $(\sum_{i=1}^{|\mathbb{D}|} l_i^j[t])$. While many methods have been introduced to compute the discrete logarithm, such as Shank's baby-step giant-step algorithm [91], we resorted to using a lookup table to compute it efficiently in a light-weight manner. By performing the above steps, the result $(\sum_{i=1}^{|\mathbb{D}|} l_i^j[t])$ is the summation of the drivers passing through grid cell j at each reporting period t. After the aggregation, the TMC can use the encrypted information to forecast traffic conditions, such as traffic density and congestion, as explained in the next section.

2.5.2 Deep learning-based Traffic Forecasting

Traffic Flow Process Formulation: The process of traffic flow prediction can be formulated mathematically as the drivers' density and congestion patterns within each cell under the traffic monitoring area. This formulation involves the analysis of historical density, real-time density, and future density. As shown in Fig. 2.6, at the current time t, the objective is to predict the traffic flow of a specific grid cell at the time interval $(t+h\Delta)$ for a given prediction horizon, utilizing the past traffic status. Let $X^j[\tau]$ denote the traffic flow of the j^{th} observation route during the τ^{th} time interval. The traffic flow values $X^j[\tau]$ correspond to $\tau = t-n\Delta,\ldots,t-\Delta,t$. Here, $\Delta = 5$ minutes, $n = 15$, and $h = 1,3,6,12,$. This means that 75-minute historical data will be used to predict the traffic flow of the next 5, 15, 30, and 60 minutes.

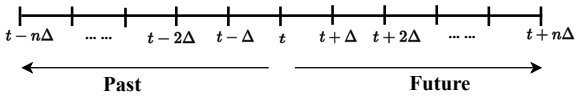


Figure 2.6: The Traffic forecasting time horizon.

We create three spatiotemporal traffic flow matrices to capture the temporal and spatial aspects of traffic flow. This involves combining historical traffic flow data from neighboring locations at different time scales, including the current moment t^s, daily patterns t^d, and weekly trends t^w. The matrix $X[t^s]$ specifically represents the current historical traffic density. It considers a time window spanning from $t^s - n$ to t^s where each column of this matrix can be represented as the status of the reporting area at time t^s denoted as $X[t^s]$ = $\left[\sum_{i=1}^{|D|} l_i^1[t^s], \sum_{i=1}^{|D|} l_i^2[t^s], ..., \sum_{i=1}^{|D|} l_i^{\mathscr{L}}[t^s]\right]^T$. The following matrix defines $X[t^s]$ with dimensions $\mathscr{L} \times n$, where $\mathscr{L}$ is the number of reporting cells, and n is the size of the time window used for analysis.

$$\begin{bmatrix} X[t^s-n] \\ . \\ . \\ . \\ . \\ X[t^s] \end{bmatrix}^T = \begin{bmatrix} \sum_{i=1}^{i=|D|} l_i^1[t^s-n] & .. & \sum_{i=1}^{i=|D|} l_j^1[t^s] \\ \sum_{i=1}^{i=|D|} l_i^1[t^s-n] & .. & \sum_{i=1}^{i=|D|} l_j^2[t^s] \\ . & . & .. \\ . & . & .. \\ . & . & .. \\ \sum_{i=1}^{i=|D|} l_i^{\mathscr{L}}[t^s-n] & .. & \sum_{i=1}^{i=|D|} l_j^{\mathscr{L}}[t^s] \end{bmatrix}$$

The next matrix defines the historical traffic densities with daily periodicity (i.e., in the previous day d) over the same time period $t^d - n, ..., t^d, ..., t^d + n$. The traffic data with daily periodicity can be obtained by considering the previous and following n time intervals of the same moment as time t^s from the preceding day. This can be represented as the matrix $X[t^d]$.

$$\begin{bmatrix} X[t^d-n] \\ . \\ . \\ . \\ . \\ X[t^d+n] \end{bmatrix}^T = \begin{bmatrix} \sum_{i=1}^{i=|D|} l_i^1[t^d-n] & .. & \sum_{i=1}^{i=|D|} l_j^1[t^d+n] \\ \sum_{i=1}^{i=|D|} l_i^1[t^d-n] & .. & \sum_{i=1}^{i=|D|} l_j^2[t^d+n] \\ . & . & .. \\ . & . & .. \\ . & . & .. \\ \sum_{i=1}^{i=|D|} l_i^{\mathscr{L}}[t^d-n] & .. & \sum_{i=1}^{i=|D|} l_j^{\mathscr{L}}[t^d+n] \end{bmatrix}$$

Similarly, the next matrix defines the historical traffic densities with weekly periodicity (i.e., in the previous week $'w'$) over the same time period $t^w-n,...,t^w,...,t^w+n$. Historical traffic flow data is constructed with weekly periodicity by considering previous and subsequent n time intervals of the same moment as time t^s in the last week as follows $X[t^w]$.

$$\begin{bmatrix} X[t^w-n] \\ . \\ . \\ . \\ . \\ X[t^w+n] \end{bmatrix}^T = \begin{bmatrix} \sum_{i=1}^{i=|D|} l_i^1[t^w-n] & ... & \sum_{j=L}^{i=|D|} l_j^1[t^w+n] \\ \sum_{i=1}^{i=|D|} l_i^1[t^w-n] & ... & \sum_{j=L}^{i=|D|} l_j^2[t^w+n] \\ . & . & ... \\ . & . & ... \\ . & . & ... \\ \sum_{i=1}^{i=|D|} l_i^L[t^w-n] & ... & \sum_{j=L}^{i=|D|} l_j^L[t^w+n] \end{bmatrix},$$

Deep Learning-based Forecasting: The traffic forecasting model utilized by TMC is based on an attention-based Convolutional Sqeeze and Excitation and Long Short-Term Memory (Conv-SE-LSTM) deep learning architecture. The model's structure is depicted in Fig. 2.5. The Conv-SE-LSTM module serves as the primary component of the proposed model, focusing on capturing the spatial-temporal features of traffic flow. The Conv-SE-LSTM module combines a CNN, a SE, and an LSTM network, as illustrated in Fig. 2.5. The CNN component comprises two convolutional layers, while the LSTM component comprises two LSTM layers. The input to the Conv-LSTM module is a spatial-temporal traffic flow matrix denoted as $X[t^s]$, which represents the current historical traffic flow of the reporting area to be predicted. The main components of the proposed model are described as follows.

1) Convolutional Block: To extract spatial features, a two-dimensional convolution operation is applied to the traffic flow data $X[t^s]$ at time t^s. The convolution operation involves a two-dimensional convolution kernel filter, which slides over the flow data to acquire the local perceptual domain. The convolution operation can be expressed as

$$Y[t^s] = \sigma(W_s * X[t^s] + b_s), \tag{2.1}$$

where W_s represents the filter weights, b_s is the bias term, $X^s[t]$ denotes the input traffic flow at time t^s, $*$ denotes the convolution operation, σ represents the activation function, and $Y[t^s]$ is the output of the first convolutional layer. This process helps in extracting spatial features from the neighboring observation

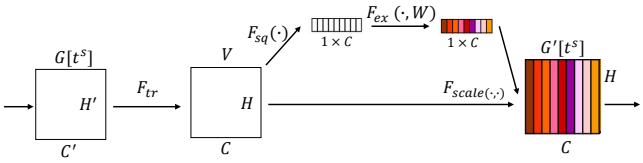


Figure 2.7: The weighting mechanism within the Squeeze-and-Excitation block.

locations. $G[t^s]$ represents the output of the second convolutional layer.

$$G[t^s] = \sigma(W_{s_2} * Y[t^s] + b_{s_2}) \tag{2.2}$$

After processing the current spatiotemporal information through the two convolutional layers, the output is then connected to the squeeze and excitation module.

2) Squeeze-and-Excitation (SE): In the SE, convolution transformation is represented by F_{tr}, which maps the input $G[t^s]$ to feature mappings V where $V \in \mathbb{R}^{H \times C}$ (see Fig. 2.7). The feature mappings V undergo a squeeze operation, which aggregates the feature maps across their spatial dimensions (H) to generate a channel descriptor. This descriptor captures the global distribution of channel-wise feature responses, allowing all network layers to access information from the entire receptive field. Subsequently, the excitation operation, implemented through a self-gating mechanism, takes the channel descriptor as input and produces modulation weights specific to each channel. These weights are then applied to the feature mappings V, generating the output of the SE block. This output can be directly fed into subsequent layers of the network. In our model, one dimensional SE is applied to the input $G[t^s]$ to generate the output is $G'[t^s]$, which is input to the LSTM module. The complete architecture for the SE module is given in Fig. 2.8.

3) LSTM: Long-term dependencies within sequential data can be efficiently captured using the LSTM architecture, making it particularly suitable for handling extended sequential patterns. In our model, we employ multiple LSTM layers to capture higher-level traffic flow features. The first LSTM processes the sequence output from the SE module $G^{'}[t^s] = [G'[t^s - n], \ldots, G'[t^s - 1], G[t^s]]$ and calculates the hidden state for each time step $H_1[t^s] = [H_1[t^s - n], ..., H_1[t^s - 1], H_1[t^s]]$. Then the hidden state sequence $H_1[t^s]$ is input into the second LSTM layer to calculate the hidden state $H_2[t^s]$ as the output, which indicates the spatial-temporal feature for time step t^s. LSTM layers are stacked so that each subsequent layer receives the hidden state of the previous layer. As a result, the model can capture increasingly complex patterns and dependencies within the sequential data. The diagram in Fig. 2.9 visually represents the used LSTM layers and their sequential connections.

4) Attention Mechanism: The standard LSTM cannot determine the importance of different parts within a traffic flow sequence. To address this limitation, an attention mechanism is introduced. This attention mechanism enables the model to automatically identify varying levels of importance for different segments of the traffic flow sequence at different time steps. The incorporation of the attention mechanism with the Conv-LSTM module is depicted in Fig. 2.9, providing a visual representation of its functionality. The output of Conv-SE-LSTM at each time step t^s is computed as a weighted summation of the output of the LSTM network $H_2[t^s]$ follows:

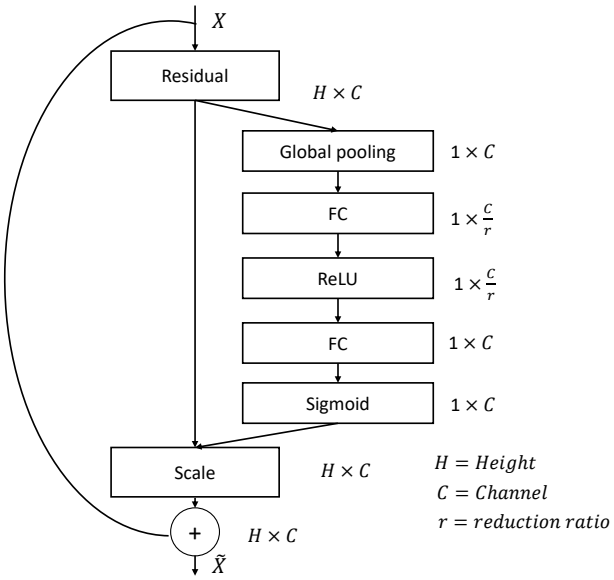


Figure 2.8: Squeeze-and-Excitation module architecture.

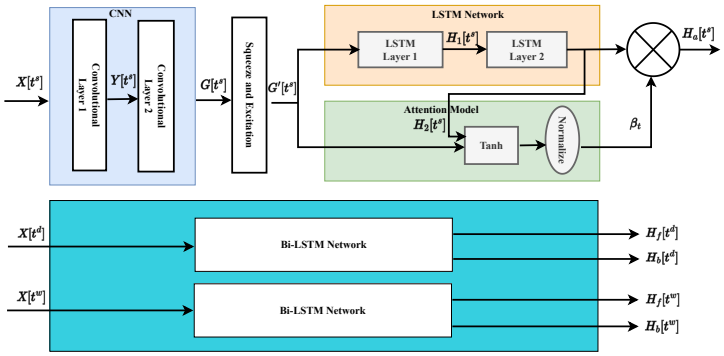


Figure 2.9: The Conv-SE-LSTM module with an attention mechanism.

$$H_a[t^s] = \sum_{k=1}^{n+1} \beta_k H_2[t^s - (k-1)] \tag{2.3}$$

where $n+1$ is the length of flow sequence and β_k is the temporal attention value at time step $t-(k-1)$. The attention value β_k can be computed as

$$\beta_k = \frac{exp(s_k)}{\sum_{k=1}^{n+1} exp(s_k)} \tag{2.4}$$

The scores $s = (s_1, s_2, ..., s_{n+1})^T$ indicate the importance of each part in the traffic flow sequence, which can be obtained as

$$s_t = V_s^T tanh(W_{hs} G[t^s] + W_{ls} H_2[t^s]) \tag{2.5}$$

where V_s^T, W_{hs} and W_{ls} are the learnable parameters and $H_2[t^s]$ is the hidden output from the Conv-LSTM network.

5) Bidirectional LSTM (Bi-LSTM): A module based on bi-directional LSTM networks is employed to

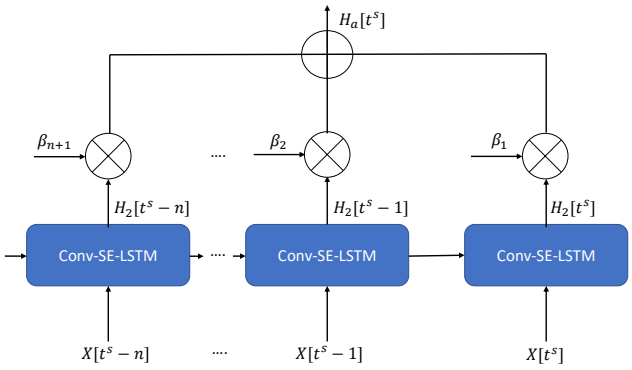


Figure 2.10: The attention mechanism with Conv-LSTM networks.

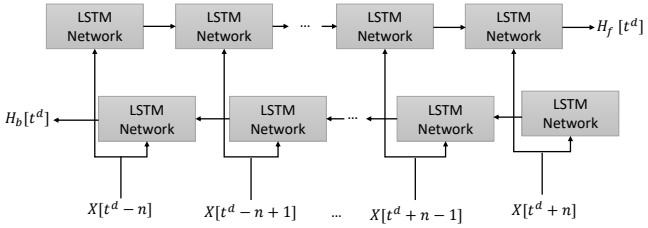


Figure 2.11: The structure of Bi-LSTM networks.

extract periodic features and capture such a temporal dependency from the daily $\mathbf{X}[t^d]$ and $\mathbf{X}[t^w]$ weekly densities. The hidden states of forward and backward passes are combined as the output. This way, more features from both directions can be captured, improving the prediction performance. Fig. 2.11 illustrates the overall structure of the bi-directional LSTM module used in the model.

As shown in Fig. 2.9, H_t^a can be obtained after the processing by the attention Conv-LSTM and Bi-LSTM modules, the spatial-temporal features, the daily periodicity features $H_t^{d,f}$, $H_t^{d,b}$ the weekly periodicity features $H_t^{w,f}$ and $H_t^{w,b}$. Then, all these features are concatenated into a feature vector and then input by two regression layers to perform forecasting. Also, Fig. 2.9 shows the spatial-temporal features $H_a[t]$, the daily periodicity features $H_f[t^d]$, $H_b[t^d]$ and the weekly periodicity features $H_f[t^w]$ and $H_b[t^w]$ can be obtained after the processing by the attention Conv-SE-LSTM and Bi-LSTM modules. Then, these features are concatenated into a feature vector fed into two regression layers to carry out forecasting.

Architecture Remarks. In our model, we utilize SE layers to enhance the performance of CNNs by adaptively recalibrating the channel-wise feature responses. The SE layer employs global pooling to reduce the spatial dimensions of the input data, generating a channel descriptor for each channel. This descriptor is then processed through a fully connected layer to generate channel weights. These weights are utilized to scale the original feature maps, enabling the network to selectively emphasize different regions of the input data based on the specific task at hand. The attention mechanism is also employed to selectively focus on specific segments of the input data rather than processing the entire input indiscriminately. Attention is commonly used in sequence-to-sequence models like Recurrent Neural Networks (RNNs) and Transformer-based models, particularly when dealing with variable-length input sequences. The model can assign weights to different parts of the sequence by employing attention mechanisms based on their relative

importance for the given task. This allows the model to effectively allocate its attention and resources to the most relevant portions of the input sequence.

2.6 Privacy and Security Analysis

Proposition 1. *Confidentiality of Location Reports*

Proof: During system initialization, each driver D_i is provisioned with a unique encryption key pk_i, derived from independently sampled randomness (a_i, W_i, u_i) under the IPFE scheme. These keys provide encryption capability only; no driver receives functional decryption material. Consequently, ciphertexts generated by one driver are computationally inaccessible to all others. The TMC, by contrast, is issued a single functional decryption key dk, scoped exclusively to the inner product with the all-ones vector $\mathbf{y}$[14]. This key allows recovery of aggregate driver counts per cell, expressed as $\langle \mathbf{y}, x \rangle$, but reveals no individual component x_i. The issuance of only one function scoped key prevents selector queries or arbitrary decryptions that could isolate individuals. Hence, the scheme guarantees confidentiality of individual location reports while still enabling the TMC to perform aggregate traffic forecasting.

Proposition 2. *Unlinkability of Encrypted Cells*

Proof: Ciphertexts generated by one driver, whether for identical or different cells, are computationally unlinkable under the known-ciphertext model employed. For each driver D_i and cell $j \in \{1, \ldots, K\}$, the IPFE encryption algorithm incorporates a fresh random nonce $r_i^j \leftarrow_R \mathbb{Z}_p$. This ensures that repeated encryptions of the same plaintext yield distinct ciphertexts, eliminating deterministic patterns and guaranteeing semantic or Indistinguishability under Chosen-Plaintext Attack (IND-CPA) security. To further conceal the true report, each driver enforces k-anonymity[95] by encrypting $K-1$ dummy cells alongside the actual cell, thereby embedding the true location within a broader anonymity set. The combined effect of nonce-induced randomness and dummy-cell padding prevents adversaries, whether external or colluding with the TMC, from correlating ciphertexts across time or cells, thus rendering them unlinkable to any specific driver trajectory.

Proposition 3. *Anonymity of Location Reports*

Proof: Insider and outsider adversaries cannot compromise the anonymity of drivers' location reports. With the IPFE[14] cryptosystem employed, even if the TMC colludes with a subset of drivers, the coalition learns no additional information about non-colluding drivers' reports beyond the authorized aggregates, provided a minimum cohort threshold γ is enforced, the aggregation windows are fixed and non-overlapping, and the decryption key is scoped exclusively to the aggregate function. In this scenario, the colluding coalition observes all ciphertexts, the plaintext of colluding drivers, and only aggregate sums via the function-limited key dk. Differencing attacks are neutralized by releasing aggregates only when at least γ distinct contributors are present, ensuring that honest drivers' inputs are masked within a sufficiently large anonymity set. Fixed, non-overlapping time windows further prevent adaptive cohort-splitting and overlap-based inference, while function scoping ensures that only one decryption key tied to the all-ones vector is available, precluding selector-style queries that could isolate individuals. Meanwhile, fresh random nonces applied at each encryption step guarantee IND-CPA security, and k-anonymity obliges drivers to report $K-1$ dummy cells alongside their true location, preventing ciphertext correlation or trajectory inference. Under these

constraints, the coalition's posterior knowledge about any non-colluding driver's bit is negligibly stronger than its prior, and thus no per-driver information beyond the aggregate counts is revealed.

2.7 Performance Analysis

The proposed schemes were implemented in Python on a Lambda GPU workstation equipped with the following specifications: 2xQuadro RTX 8000 GPUs, 2-Way NVLink, Intel i9-9820X CPU (10 Cores), 128 GB of RAM, and a 2 TB NVMe SSD. This workstation came pre-installed with the latest versions of essential libraries such as CUDA, Jupyter, Pytorch, Tensorflow, and Keras. For our implementation, we utilized two datasets:

- **SUMO Dataset:** To assess the encryption component of our project, we generated a set of random trips based on real maps. We started by obtaining a genuine map of Greensboro, North Carolina, USA, from the OpenStreetMap project[78]. The traffic management area covered an $8\,\text{km} \times 8\,\text{km}$ region, divided into 40 cells, each measuring $1\,\text{km} \times 1\,\text{km}$. To create real and random routes, we employed the "Simulation of Urban MObility" (SUMO) software[48]. All results presented are the averages from 30 different runs (See Fig. 2.12).

- **PeMS Dataset:** This dataset was sourced from the Performance Measurement System (PeMS), supported by California Department of Transportation (Caltrans)[16]. We used the PeMS14 dataset, covering traffic data from 2001 to 2023 across California's major metropolitan areas. The data, collected from nearly 40,000 sensors, is mostly recorded at 5-minute intervals, with some available at 30-second intervals for more detailed historical and real-time traffic analysis. For our study we focused on two specific scenarios: freeway and urban traffic, training and evaluating our proposed model with data from 183 sensors in District 10, specifically on Freeway SR99-S, as well as 12 sensors from District 4 on Street I980 in Oakland. This enabled robust analysis across both freeway and urban traffic conditions.

We then assess the proposed privacy-preserving traffic management forecasting system from three perspectives: Computation Overhead, Communication Overhead, and Traffic Flow Forecasting.

2.7.1 Computation Overhead

The computation overhead is quantified through two key metrics: the cryptographic key size provisioned by the KDC and the encrypted message size transmitted to the TMC ($D_k + E_m$). For D_k, each driver receives a unique key $pk_i := (G, [a_i], [W_i a_i], u_i)$, generated over the asymmetric BN256 pairing curve with 256-bit security. With each group element of 32 bytes, a driver requires two group points (64 bytes) and one small field element (2 bytes), yielding a lightweight 66-byte route encryption key. For E_m, each encrypted cell consists of two group elements (32 bytes $\times$ 2 group elements = 64 bytes). Under k-anonymity[95], the transmitted payload grows linearly with k (total encrypted cells), resulting in a message size of $64k$ bytes. For an 80-cell grid, this corresponds to just 5.12 KB, easily handled by on-board units and existing V2I standards. Compared with state-of-the-art schemes in Table 2.2, our IPFE-based design incurs only $\mathcal{O}(k)$ modular

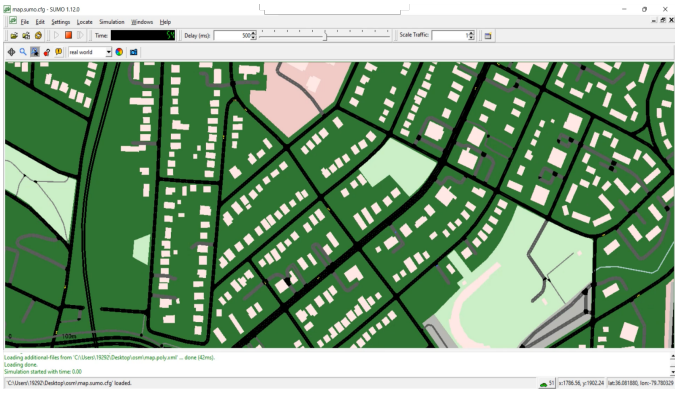
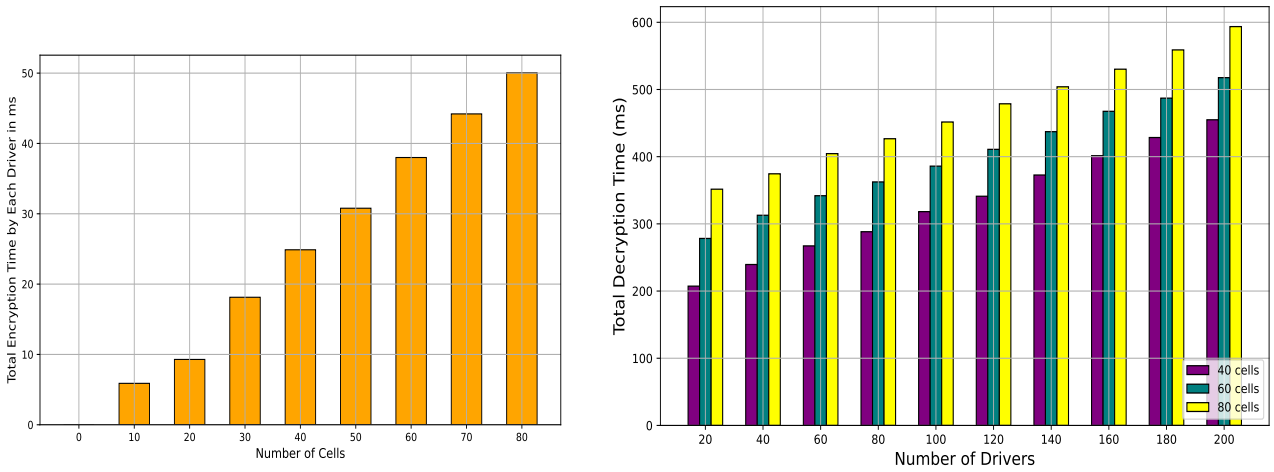


Figure 2.12: Synthetic dataset generation using SUMO



(a) Encryption times vs Number of Cells.

(b) Decryption Time vs Number of Drivers.

Figure 2.13: Computation Overhead Analysis.

exponentiations with constant-size ciphertexts for the driver-side encryptions and $O(|\mathbb{D}| \cdot K)$ modular exponentiations for the TMC aggregated decryptions. Competing methods are significantly costlier: blockchain schemes add $\mathcal{O}(\log n)$ consensus overhead, additive HE grows as $\mathcal{O}(K \cdot L \cdot \lambda^3)$ with minute-level delays, and FL+DP requires $\mathcal{O}(|D| \cdot d \cdot R)$ repeated server gradient updates with $O(dR)$ client gradient uploads, thereby slowing convergence. In contrast, our design guarantees predictable linear growth with drivers encrypting in $\approx$12–35 *ms* for $k \in [24, 48]$ per window, and the TMC executes a single $\mathcal{O}(|D| \cdot K)$ decryption across all cells with full parallelization. Even at metropolitan scale ($|D| = 10^5$, $K = 100$), aggregate decryption completes in under 1*s* on modern multi-core/GPU servers, ensuring real-time performance. These results confirm the efficiency of our cryptographic design: lightweight for resource constrained vehicles, scalable for dense urban deployments, and decisively more practical than blockchain, HE, or FL based alternatives.

2.7.2 Communication Overhead

In our simulations, we enforce a minimum cohort size of $\gamma = 20$, derived from the collusion-resilience condition $\gamma \geq \left\lceil \frac{h}{1-\alpha} \right\rceil$, with $\alpha = 0.5$ (up to 50% colluders) and $h = 10$ honest contributors per reporting window.

Table 2.2: Comparative Analysis of Privacy and Non-Privacy Schemes

Scheme	Computational complexity	Communication complexity	Scalability	Forecasting	Remarks
Proposed	**Driver:** $\mathcal{O}(K)$ bit-ops; **TMC:** $\mathcal{O}(\lvert\mathbb{D}\rvert \cdot K)$ bit-ops per Δt	**bytes$_{\text{up,driver}}$** $= K \cdot \lvert G\rvert$ (independent of $\lvert\mathbb{D}\rvert$)	**Linear/Feasible if** $\mathcal{O}(\lvert\mathbb{D}\rvert \cdot K)_{tmc} < \Delta t$	✓	**Lowest driver cost; Scalable; Accurate forecasting**
(Blockchain & Pseudonyms) [113]	Signature $\mathcal{O}(1)$ ops; Consensus $\mathcal{O}(\log n)$	Constant per tx = $\mathcal{O}(s_{tx})$; bytes$_{\text{out}} \Rightarrow \mathcal{O}(N_{\text{peers}} \cdot s_{tx})$	Limited throughput; Latency $\propto 1/\text{TPS}$	✗	Consensus bottleneck; unsuitable for real-time
(Blockchain & CPPA) [60]	Signature $\mathcal{O}(1)$ ops; Consensus $\mathcal{O}(\log n)$	Per auth. = $\mathcal{O}(s_{tx})$bytes; Broadcast = $\mathcal{O}(N_{\text{peers}} \cdot s_{tx})$	Consensus latency $\mathcal{O}(T)$	✗	High latency; Unsuitable for high-freq. traffic data
Additive HE [41]	Driver: $\mathcal{O}(K \cdot \lambda^3)$ bit-ops; TMC: $\mathcal{O}(K \cdot L \cdot \lambda^3)$ bit-ops per Δt	bytes$_{\text{up,driver}}$ = $\mathcal{O}(K \cdot s_{ct}^{HE})$	Super-linear; Delay in sec–min range	✗	Very high computation; Impractical for short real-time Δt
Differential Privacy (DP) [50]	Driver: $\mathcal{O}(K)$ Noise injection; TMC: $\mathcal{O}(\lvert\mathbb{D}\rvert \cdot K)$ Noise vector aggre.	bytes$_{\text{up,driver}}$ = $\mathcal{O}(K) \approx K \cdot s_{\text{val}}$; $(s_{ct}^{HE} \gg s_{\text{val}})$	Linear in $\lvert\mathbb{D}\rvert + K$	✓	Forecast accuracy ↓ degrades at strong ε ↓ levels
(FL & DP) [80]	Clients: $\mathcal{O}(dR)$; Server: $\mathcal{O}(\lvert\mathbb{D}\rvert \cdot d \cdot R)$	bytes$_{\uparrow,\text{client/R}}$ = bytes$_{\downarrow,\text{server/R}} \approx \mathcal{O}(1) \cdot d \cdot s_{\text{elem}}$	Scales to many drivers $\lvert\mathbb{D}\rvert \times R$	✓	High comms. overhead; Slow convergence with DP

* ✓ = supported; ✗ = not supported; R rounds; d gradient; G, s_{ct}^{HE} ciphertext sizes; s_{elem} byte/element; s_{val} per value-size; λ security parameter; $\mathcal{O}(1)$ small/constant cost; Δt forecast window; ε DP privacy; s_{tx} tx size; N_{peers} neighbors; n set size; TPS transactions/s; T consensus period; L HE layers

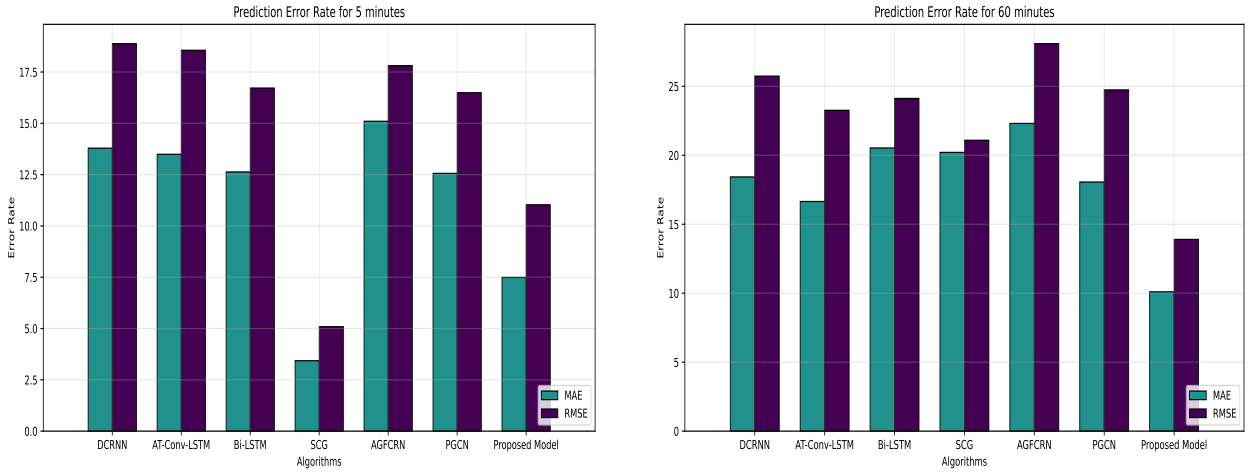
On the driver side, uplink communication is limited to transmitting constant-size ciphertexts $|G|$ for K active cells. This uplink is independent of the total number of drivers $|D|$. Using a BN256-based IPFE implementation, each encrypted cell is 64 bytes, so a driver reporting $K = 80$ active cells transmits about 5.12 KB per window. Even with $|D| = 500$ drivers, the aggregate uplink remains only ~ 0.005 MB per reporting interval Δt, a negligible bandwidth demand for vehicular networks. By contrast, additive HE expands payloads to $\mathcal{O}(K \cdot s_{\text{ct}}^{\text{HE}})$, where $s_{\text{ct}}^{\text{HE}} \gg |G|$, and induces super-linear decryption costs at the server/TMC. Blockchain systems impose $\mathcal{O}(s_{\text{tx}})$ per transaction and additional $\mathcal{O}(N_{\text{peers}} \cdot s_{\text{tx}})$ broadcast overhead. FL+DP requires bidirectional gradient transfers on the order of $\mathcal{O}(d \cdot s_{\text{elem}})$ each over R rounds, compounding both communication and convergence delays. Central DP preserves the $\mathcal{O}(K)$ byte growth per driver ($\approx K \cdot s_{\text{val}}$) but reduces forecast accuracy under strong privacy guarantees (small ε) while still obligating the TMC to aggregate $|D|$ vectors as summarized Table 2.2. We validated these properties with a prototype built using the GoFE library in Python for our IPFE setup[32] over a synthetically generated Greensboro-area traffic trace with $|D| = 200$ drivers and 80 geographic cells. Simulation results confirm linear scaling, with Fig. 2.13a depicting a direct linear relationship between the encryption time and K, where encrypting 80 cells requires only ~ 50 *ms* per driver. Similarly, TMC decryption time also scales linearly with K, where decrypting 200 driver presences across 80 cells completes in under 600 *ms* as illustrated in Fig. 2.13b. Thus, the overall complexity remains bounded by $\mathcal{O}(K)$ per driver and $\mathcal{O}(|D| \cdot K)$ at the TMC, both of which comfortably satisfy the feasibility condition $\mathcal{O}(|D| \cdot K)\, t_{\text{mc}} < \Delta t$, where the total work $\mathcal{O}(|D| \cdot K)\, t_{\text{mc}}$ duration utilized by the TMC is $< \Delta t$. Hence, our IPFE-based design avoids the heavy communication overheads of blockchain consensus, HE expansion, and FL gradient exchanges, while maintaining forecasting accuracy and strong privacy guarantees.

Table 2.3: Hyper-parameter tuning

Hyper-parameter	Value	Selected Best Value
Units	32, . . . , 512	488
Activation	`relu`, `tanh`, `sigmoid`	`relu`
Dropout	True, False	True
Learning rate	1×10^{-4} to 1×10^{-2}	0.0003

Table 2.4: Different Optimizer comparison

Optimizer	MAE	MAPE	RMSE
`SGD`	39.45	62.73	45.66
`ADADELTA`	18.75	19.13	24.09
`RMSProp`	12.06	14.25	15.67
`ADAGRAD`	9.80	10.18	13.93
`ADAM`	7.94	8.50	11.03



(a) Mean Absolute Error and Root Mean Square Error

(b) Mean Absolute Error and Root Mean Square Error

Figure 2.14: Error Rate Assessment for Short-Term Traffic Flow Forecasting.

2.7.3 Traffic Flow Forecast

This subsection evaluates the traffic forecasting model (Conv-LSTM) both with and without the squeezing and excitation algorithms, attention mechanism, and Bi-LSTM. In order to measure our suggested scheme against comparable traffic forecasting methods found in the literature, we selected three commonly used

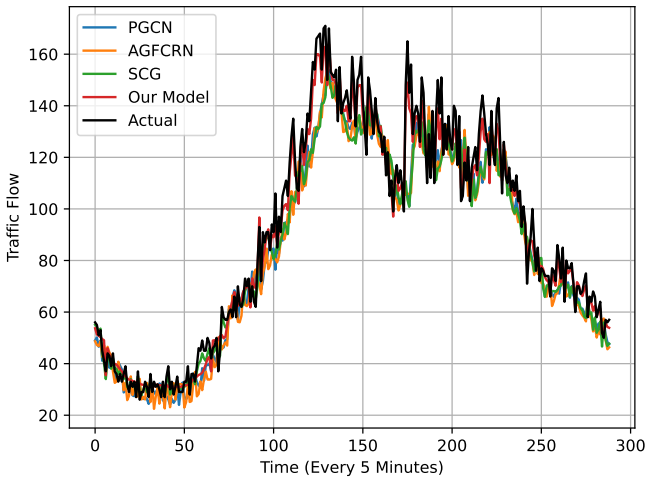


Figure 2.15: Comparison of our privacy-preserving model Predictions vs other non-privacy-preserving models and actual traffic flow over a 300-Minute interval.

performance indices. These measures, `Mean Absolute Error (MAE)`, `Mean Absolute Percentage Error (MAPE)`, and `Root Mean Square Error (RMSE)` assess the accuracy of predictive models in regression analysis.

- *Mean Absolute Error:* The MAE is calculated using the formula

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|F_p - F_t| \tag{2.6}$$

- *Mean Absolute Percentage Error:* The MAPE is calculated as follows:

$$MAPE(\%) = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{F_p - F_t}{F_t}\right| \times 100 \tag{2.7}$$

- *Root Mean Square Error:* The RMSE is determined by the formula:

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(F_p - F_t)^2} \tag{2.8}$$

where F_p represents the predicted traffic flow and F_t represents the true traffic flow.

1. **Experimental Data and Evaluation:** Using the PeMS dataset, the hyperparameters of the forecasting Conv-LSTM model are fine-tuned utilizing the Tensorflow Keras Tuner. The tuning process

Table 2.5: Prediction performance with various proposed modules for prediction of urban area traffic flow.

Algorithm	Measure	**5min**	**15min**	**30 min**	**60 min**
Conv LSTM (Stage 1)	MAE	9.05	10.80	10.28	10.50
	MAPE (%)	9.94	11.73	10.25	11.33
	RMSE	12.32	14.73	14.28	14.21
Bi-Conv LSTM (Stage 2)	MAE	12.57	18.07	12.37	14.31
	MAPE (%)	13.18	19.08	12.9	13.73
	RMSE	18.8	24.16	18.55	21.16
AT-Bi-Conv LSTM (Stage 3)	MAE	8.19	9.45	9.21	10
	MAPE (%)	8.86	9.56	9.60	10.71
	RMSE	11.33	13.14	13.01	13.94
AT-Bi-Conv-SE LSTM (Stage 4)	MAE	**7.94**	**8.66**	**9.88**	**10.10**
	MAPE (%)	**8.5**	**9.22**	**10.66**	**10.75**
	RMSE	**11.03**	**12.10**	**13.8**	**13.9**

involved exploring a range of hyperparameter values, including unit limits for feed-forward layers ranging from 32 to a maximum of 512, likewise exploring various activation functions ranging `ReLU`, `Sigmoid`, and `Tanh`. Lastly, we investigated different learning rates within 1×10^{-4} to 1×10^{-2}. Table 2.3 contains the tuning process outcomes and the optimal hyperparameter values. Additionally, we performed a comparative analysis on adopting five different optimizers for our model. The optimizers used were Stochastic Gradient Descent (SGD), ADADELTA, Root Mean Square Propagation (RMSProp), Adaptive Gradient (ADAGRAD), and Adaptive Moment Estimation (ADAM). The analysis results are presented in Table 2.4, where the ADAM consistently outperforms other optimizers regarding error reduction. Consequently, we selected ADAM as the optimizer for our final model.

2. **Forecast Performance Evaluation:** Here, we demonstrate the efficacy of our proposed hybrid model for traffic flow prediction at a particular Point of Interest (POI) on Street I980 in Oakland, District 4, by utilizing a number of crucial elements, including an attention mechanism (AT), a squeeze-and-excitation (SE) module, and a Bi-LSTM module. Our hybrid model was built in four stages using the TensorFlow framework[31], Beginning with the Conv-LSTM (Conv LSTM) model (Stage 1). Then, integrating the Conv LSTM model with a Bi-LSTM module to form a Bi-Conv LSTM model (Stage 2). The Bi-Conv LSTM model was enhanced further by adding an attention mechanism to produce an AT-Bi-Conv LSTM model (Stage 3). Lastly, we fuse a squeeze-and-excitation module

Table 2.6: Performance comparison of different Models for Urban traffic flow prediction.

Horizon	Measure	DCRNN[58]	AT-Conv-LSTM[117]	Bi-LSTM[65]	SCG[67]	AGFCRN[57]	PGCN[90]	Our Model
5 min	MAE	13.79	13.49	12.63	**3.43**	15.10	12.56	**7.49**
	MAPE (%)	10.7	10.1	10.49	8.60	9.67	8.74	**8.5**
	RMSE	18.88	18.56	16.72	**5.09**	17.81	16.49	**11.03**
15 min	MAE	14.79	14.34	15.09	**6.89**	16.71	13.43	**8.66**
	MAPE (%)	11.5	10.8	12.28	11.61	10.14	9.88	**9.22**
	RMSE	20.43	20.08	18.34	**8.43**	22.68	17.91	**12.10**
30 min	MAE	16.05	15.48	17.41	11.15	19.53	15.62	**9.88**
	MAPE (%)	12.4	11.4	14.5	16.89	12.82	11.61	**10.66**
	RMSE	21.18	21.26	19.72	14.71	25.94	19.33	**13.8**
60 min	MAE	18.43	16.65	20.53	20.21	22.31	18.06	**10.10**
	MAPE (%)	14.2	12.3	16.9	17.36	15.53	13.89	**10.75**
	RMSE	25.74	23.26	24.12	21.09	28.09	24.74	**13.9**

to the previous AT-Bi-Conv LSTM model, resulting in an AT-Bi-Conv-SE LSTM model (Stage 4). The outcomes, as detailed in Table 2.5, highlight our final hybrid model (AT-Bi-Conv-SE LSTM model) as the best performing model, achieving the lowest MAE and RMSE error rates of 7.94% and 11.03% respectively for a 5 minutes prediction time, while posing a 10.1% MAE value and 13.9% RMSE value for a prediction horizon of 60 minutes. Also, Table 2.5 shows stage two (Bi-Conv LSTM model) as the worst performing stage, with the highest MAE and RMSE error rates of 12.57% and 18.8% respectively for a 5 minutes prediction horizon, as well as, 14.31% MAE value and 21.16% RMSE value for a prediction time of 60 minutes, indicating a decrease in performance, after the addition of the Bi-LSTM module. For instance, a significant increase in MAE and RMSE error rates from 10.5% and 14.21% respectively in stage 1 to MAE and RMSE error rates of 14.31% and 21.16% respectively in stage 2. Conversely, a substantial performance improvement was witnessed from stage 2 to stage 3, likewise from stage 3 to stage 4 (best performing model) across all prediction horizons. Concretely, this comprehensive approach underscores the effectiveness of our model in accurately forecasting traffic flow and positions it as a leading solution for traffic management and analysis. Fig. 2.15 shows the prediction performance of the proposed model, emphasizing its superior forecasting accuracy due to its coherence with the referenced actual traffic flow compared to the flow predictions of other baselines forecasting models.

Furthermore, we comprehensively compared our proposed hybrid (AT-Bi-Conv-SE-LSTM) model and other established contemporary approaches for short-term traffic flow predictions spanning various prediction time horizons (5, 15, 30, and 60 minutes). The comparative approaches encompass Diffusion Convolutional Recurrent Neural Network (DCRNN)[58], Attention-Based Conv-LSTM Network (AT-Con-LSTM)[117], Bidirectional LSTM network[65], STFSA Convolutional Neural Network Gated Recurrent Unit (SCG)[67], Adaptive Spatial-Temporal Fusion Graph Convolutional Network (AGFCRN)[57] and Progressive Graph Convolutional Network (PGCN)[90]. Table 2.6 showcases the comparison of prediction accuracy (error rates) across different models using the MAE, MAPE, and RMSE indices. Notably, our proposed hybrid (AT-Bi-Conv-SE LSTM) model emerged

as the overall best, consistently delivering exceptionally low MAE and RMSE rates of 7.49% and 11.03%, respectively, for a 5-minute forecast. For the same prediction time, AGFCRN shows the highest MAE and RMSE rates of 15.10% and 17.81% respectively, making it the least effective for the same forecast duration. Other models, including DCRNN, AT-Con-LSTM, Bi-LSTM, SCG and PGCN, showed improved performances (descension of MAE and RMSE rates) over AGFCRN (least performing), with SCG being the superior model for shorter prediction times. Fig. 2.14a affirms these findings, as we can visualize a reduction in the MAE and RMSE rates (improved model performance) moving from the least performing AGFCRN to the best performing SCG forecasting model for a 5-minute forecast. Similarly, from Table 2.6, for a 60-minute forecast, AGFCRN remains the least efficient with the highest MAE and RMSE rates of 22.31% and 28.09% respectively, while our proposed model was the best-performing forecasting model with the least MAE and RMSE rates of 10.1% and 13.9% respectively (significantly reducing errors compared to AGFCRN and PGCN). Common to the behavior observed in Fig. 2.14a for the 5-minute forecast, Fig. 2.14b provides a visual illustration of the ascension in model performance for the forecasting algorithms moving from AGFCRN (the least performing algorithm), DCRNN, AT-Con-LSTM, Bi-LSTM, SCG, PGCN to our proposed hybrid model (best performing), in decreasing order of MAE and RMSE rates. It is essential to note, the trend of increasing MAE and RMSE rates with longer prediction horizons is consistent across all models as witnessed in both Tables 2.5 and 2.6. However, the SCG model, while excellent for short predictions (5 and 15 minutes), from Table 2.6, falls short for longer horizons (30 and 60 minutes) compared to our proposed model. This indicating how reactionary the SCG model is, as well as underlining the superior capability of our proposed hybrid model in providing precise short-term traffic forecasts, essential for dynamic traffic management, incident response, and enhancing mobility, safety, and the overall efficiency of the transportation network.

2.8 Conclusion

this chapter presented a novel, secure, and efficient framework for privacy-preserving traffic forecasting that addresses both the precision demands of modern ITS and the sensitivity of driver location data. By combining IPFE with k-anonymity, the proposed scheme supports encrypted route reporting and aggregation while preventing disclosure of individual trajectories, even under collusion. A hybrid Conv-LSTM and Bi-LSTM model, enhanced with a SE module, operates on the aggregated encrypted data to capture complex spatial-temporal traffic dynamics and deliver reliable forecasts. Extensive evaluations on both synthetic and real-world datasets confirmed the framework's scalability, low overheads, and resilience, achieving high forecast accuracy particularly at critical congestion points. Unlike prior methods, the proposed system integrates strong cryptographic guarantees with deep learning, establishing a new benchmark for trustworthy, privacy-preserving traffic forecasting and demonstrating clear potential for deployment in real-world ITS environments.

Chapter 3: RB-XAI: Relevance-Based Explainable AI for Traffic Detection in Autonomous Systems

3.1 Introduction

The recent transformative evolution of artificial intelligence (AI) has significantly impacted various industries, including transportation, healthcare, finance, and cybersecurity. This evolution is propelled by sophisticated AI algorithms, which empower autonomous systems, such as AVs and Unmanned Aerial Vehicles (UAVs or drones), with versatile capabilities in navigation, learning, decision-making, and collaboration, spanning various industries. UAVs have become indispensable across applications such as agriculture, infrastructure inspection, package delivery, and disaster response [76], [79], due to their agility in accessing remote or hazardous locations. Simultaneously, AVs promise to revolutionize the transportation industry with their autonomous navigation capabilities, leveraging advanced sensors for their perception, localization, planning, and control operations [9].

Despite the remarkable strides in AVs, a significant impediment to its widespread societal acceptance, like many other intelligent systems persists in the form of the "black box" stereotype. This stereotype symbolizes the opacity in AV decision-making [3], creating a challenge of understanding and trust. Recent accidents involving AVs [12], [92] and growing concerns related to ethics and security vulnerabilities underscore the urgency of transparent solutions to address this challenge comprehensively. Overcoming the "black box" hurdle requires making the decision-making processes of AVs transparent to build stakeholder trust and acceptance. Achieving this transparency is not only essential for regulatory compliance and ethical considerations but also for ensuring the safety, security, and accountability of autonomous systems as they navigate through our dynamic and complex world.

The emerging field of XAI presents an opportunity to make AI decisions in AVs understandable to humans. However, besides the potential, XAI techniques are not widely adopted in the AV sector. Many AV developers and researchers have not fully embraced these techniques, leading to missed opportunities to improve transparency and safety within AV systems. This underutilization has also hindered innovation and collaboration within the field of XAI for AVs. However, this underutilization contrasts sharply with the extensive XAI research in areas like medicine [56], [73], [88], [119], IoT [118], cybersecurity [74], [111] and several others [22], [24], [37], [97], where XAI methods are actively explored and used. While some studies have surveyed the advantages, challenges, and methods of integrating different XAI techniques into the AV domain [9], the focus of contemporary research on XAI for autonomous systems has primarily

centered on explaining the behavior of models in tasks like semantic segmentation and object detection, utilizing traditional attribution-based XAI techniques like LIME, SHAP, Saliency Maps, GRAD-CAM, etc.

In our work, we employ the CRP XAI algorithm, a bias-resistant relevance-based XAI algorithm to proffer transparent concept–level explanations for the behavior of traffic detection models used in AVs, for traffic perception. CRP, an advanced approach extending the Layerwise Relevance Propagation (LRP) technique [84], goes beyond traditional attribution maps, by generating explanations that automatically identify and visualizes relevant examples within the input space. This insight sheds light on the crucial latent concepts and areas within the input space responsible for the behavior of traffic detection models [2] used in AVs. This research aims to boost transparency, understanding, trust, and ultimately lay the groundwork for XAI integration in AV development, fostering safer and more widely accepted autonomous systems. Concretely,

- Employing XAI techniques, specifically the CRP algorithm, our paper analyzes traffic detection in AVs. The CRP algorithm, a hybrid approach incorporating Relevance Maximization (Rel-Max), can identify examples that are highly relevant to network latent encodings. This provides nuanced insights into the features influencing the decision-making process of the traffic detection model.
- Deploying the YOLO (You Only Look Once) object detection model on our customized traffic dataset, merging Open Image Dataset Version 7 and the Microsoft COCO Dataset, our study extensively evaluates the proposed CRP explainer. The assessment hinges on Faithfulness, measuring the accuracy of CRP explanations in reflecting the detection model decisions, and Complexity metrics, gauging the comprehensibility of the generated CRP explanations. This dual evaluation provides a thorough assessment of the CRP explainer's performance.
- To enhance the YOLO model, we further integrate the Convolutional Block Attention Module (CBAM) into its feature extraction segment, to assess the impact of CBAM on the performance of both the YOLO model and the CRP XAI algorithm.

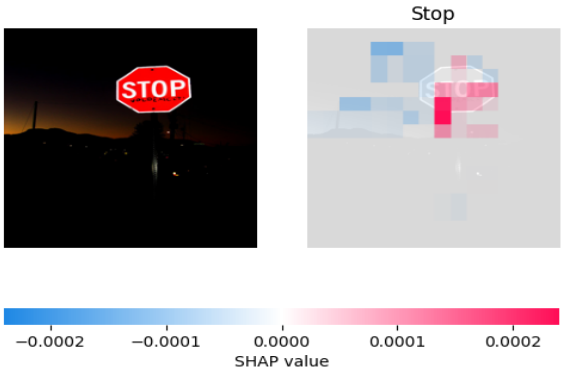
The subsequent sections of this chapter are structured as follows: Section II reviews literature, while Section III outlines our proposed method. Results and findings are presented in Section IV, followed by a summary of conclusions and potential future avenues of research in Section V.

3.2 Related Work

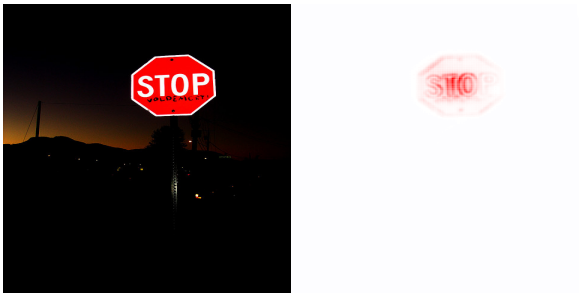
3.2.1 Explainable Artificial Intelligence (XAI)

XAI has emerged as a remedy for the inherent opacity of intricate AI models, especially Deep Neural Networks (DNNs), spanning critical domains such as healthcare, transportation, energy, security, finance, and criminal justice. Initially focused on healthcare, XAI ensures transparency in AI decisions, facilitating their integration into clinical workflows. For example, [56], [88], [119] suggest ensemble XAI approaches, melding SHapley Additive Explanations (SHAP) and Gradient-weighted Class Activation Mapping (Grad-CAM++) [88], [119], along with Class Activation Mapping (CAM) and Saliency map [56] algorithms, for retrospective visual explanations in the classification of COVID-19 and pneumonia from medical scans.

Similarly, in [8], [73], SHAP is utilized to elucidate early diagnosis of brain tumors [8] and chronic kidney disease [73] from medical scans using DNNs. In the energy sector, Machlev et al. [70] utilize Local Interpretable Model-agnostic Explanations (LIME), Occlusion-Sensitivity, and GRAD-CAM to expound on outcomes produced by Convolutional Neural Network Power Quality Disturbance (CNN-PQD) classifiers. They introduce an assessment metric, employing Binary scores and Intersection over Union (IoU) scores, to evaluate the explainability of both XAI techniques and classifiers, fostering trust by providing comprehensible rationales for AI decisions, benefiting professionals and users.



(a) **SHAP Explanation**



(b) **CRP Explanation**

Figure 3.1: Comparing XAI Algorithms

3.2.2 XAI for Autonomous Systems

In the field of transportation, specifically autonomous systems, the exploration and application of XAI techniques to enhance transparency in the behavior of models used by AVs have been relatively limited, as mentioned earlier. However, Mankodiya et al. [71] present a proposal for an XAI integrated AV system. This system utilizes GradCAM and Saliency maps XAI techniques to provide comprehensive explanations, visualizations, and insights into the intricate workings of semantic road segmentation model layers, elucidating the perception actions of AVs. Additionally, Hogan et al. [24] [42] tackle the challenges of interpretability in AI systems employed in UAVs. They achieve this by adapting KernelSHAP, an optimized variant of the traditional SHAP, for object detection tasks in aerial imagery. The KernelSHAP explainer offers quantitative insights into robust performance, detecting biases, and accurately attributing positive contributions in real-world images. The outcomes of this study highlight the significant potential of KernelSHAP as an XAI

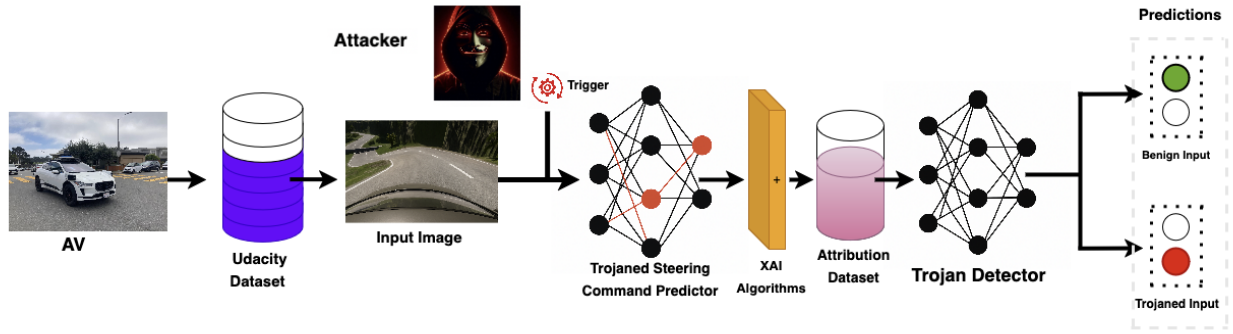


Figure 3.2: Framework of the Relevance-Based eXplainable Autonomous Vehicle Traffic Multi-Detection System

algorithm, especially in safety-critical applications where precise and interpretable insights are of utmost importance.

Despite the remarkable contributions of attribution-based XAI techniques (LIME, SHAP, Grad-CAM, Saliency Maps, etc.) in enhancing model transparency, these techniques largely used in literature have inherent limitations. SHAP like other attribution map techniques, as shown in Fig. 3.1a, focuses on providing insights to specific predictions (Stop prediction), by attributing prediction using positive (red) and negative (blue) masks on specific input features. Yet, lacking a holistic understanding of the model decision-making process, leaving critical aspects of model behavior unexplored. A notable drawback of these approaches is their susceptibility to biases due to their symmetrical treatment of positive and negative feature contributions, as depicted in Fig. 3.1a. Thus, the potential to obscure imbalances in feature importance, leading to biased interpretations and impacting the reliability (faithfulness) of these XAI algorithms. Moreover, the computational demands of these methods and the absence of standardized metrics pose challenges for comprehensive quantitative assessments and hinder the consistent comparison of their effectiveness across applications.

With its emphasis on relevance maximization and low operational latency, CRP addresses these attribution maps challenges particularly for AVs, by discerning and highlighting crucial latent concepts within the input space responsible for the behavior of traffic detection models. In Fig. 3.1b, the concept of "STOP text in an octagon" is highlighted, representing knowledge learned by the detection model for perceiving stop signs during AV traffic perception. Provision of transparent concept-level explanations as shown in Fig. 3.1b ensures a more nuanced and accurate understanding of model decisions, making CRP a valuable tool to enhancing reliability (faithfulness), transparency and ease of comprehension (less complexity). Additionally, the capabilities of CRP help mitigate biases, offering a more robust, efficient, and interpretable solution for real-time traffic perception operation in AVs.

3.3 Methodology

In this section, we present the methodological contributions of our work through a structured framework illustrated in Fig. 3.2, consisting of three main stages. Firstly, the `Input Space` incorporates a custom annotated traffic dataset with 21,000 samples across 7 classes, depicting real-time on-road scenarios. Secondly, the `Traffic Detection Model` involves a medium-sized YOLOv8 model, which we further in-

Table 3.1: Main Notations

Notation	Description
$\mathscr{C}$	Number of object classes
$\mathscr{B}$	Number of predicted bounding boxes per cell
S_p	Scaling parameter
$\mathscr{N}_{\text{cell}}$	Total number of cells in the grid
δ, σ	ReLU, sigmoid activation functions
$\mathscr{L}$	Loss function
W	Learnable model weights
X	Feature map
c, s	Channel and spatial parameters for the attention operation
$c_{ij}, \hat{c}_{ij}$	Predicted and ground truth confidence score in cell (i, j)
λ	Weight for loss calculation
1_{ij}	Indicator function for traffic detection in cell (i, j)
R	Relevance Quantity
$p^c_{ij}, \hat{p}^c_{ij}$	Predicted and ground truth class probabilities in cell (i, j)

fused with a CBAM, to enhance traffic perception. This enables the impact observation of the CBAM on the performance of the CRP explainer. Finally the `XAI Algoritm`, where we utilized a CRP explainer to provide post-hoc concept-level explanations for the behavior of AV traffic detection models, specifically the YOLOv8 model, in traffic perception. Subsequently, further details on these contributions are elaborated with reference to notations from Table 3.1, outlining the process of obtaining concept-based explanations using CRP for traffic detection.

3.3.1 Traffic Detection Model

Traffic detection models serve as a cornerstone for AVs, pivotal in ensuring safety, guiding navigation, and aiding decision-making by promptly and accurately identifying and tracking objects. These models play a crucial role in optimizing traffic flow, enforcing rule compliance, facilitating efficient route planning, and enhancing overall situational awareness for a comprehensive and secure autonomous driving experience. Our chosen traffic detection model, You Only Look Once (YOLO), introduced by Redmon et al. [82], stands out for its real-time capabilities, particularly beneficial in applications where low latency is imperative, as seen in AVs. As illustrated in Fig. 3.2, the architecture of YOLO encompasses a Convolutional Neural Network (CNN)-based Backbone with a Feature Pyramid Network (FPN) for multi-scale feature extraction, complemented by a Detection Head comprising convolutional layers, and adopting a unified approach to object detection, predicting bounding boxes ($\mathscr{B}$), confidence scores, and class probabilities ($\mathscr{C}$) in a single forward pass. Its fundamental process involves dividing an input image into grids ($\mathscr{N}_{\text{cell}}$), where each grid cell predicts bounding box coordinates (t_x, t_y, t_w, t_h) and class probabilities. YOLO utilizes softmax activation for class probability predictions, producing simultaneous results for all bounding boxes and classes

in a single pass. This amalgamation of components makes YOLO an exceptionally efficient and real-time object detection framework. The YOLO loss function ($\mathscr{L}_{\text{YOLO}}$) encompasses three main components: the localization loss ($\mathscr{L}_{\text{loc}}$), the confidence score loss ($\mathscr{L}_{\text{conf}}$), and the classification loss ($\mathscr{L}_{\text{cls}}$). The $\mathscr{L}_{\text{loc}}$ evaluates the accuracy of predicted bounding box coordinates. From Equation 3.1, it is computed as the sum of squared differences between predicted (t_x, t_y, t_w, t_h) and ground truth $(\hat{t}_x, \hat{t}_y, \hat{t}_w, \hat{t}_h)$ bounding box coordinates, multiplied by the localization loss weight (λ_{coord}). Here, 1_{ij}^{obj} serves as an indicator function, signifying object presence in cell (i, j) with 1 and 0 otherwise. The confidence score loss ($\mathscr{L}_{\text{conf}}$), detailed in Equation 3.2, evaluates predictions for both object and no-object scenarios. The summation considers cases where an object is present ($1ij^{\text{obj}}$) and where there is no object (1_{ij}^{noobj}). This formulation ensures the model is trained to predict confidence scores accurately, distinguishing between cells with and without objects. The weights (λ_{conf}) enable fine-tuning the importance of each term in the $\mathscr{L}_{\text{conf}}$ function during training. Moreover, the $\mathscr{L}_{\text{cls}}$ of the YOLO model assesses the accuracy of predicted class probabilities (p_{ij}^{c}) compared to ground truth class indicators (1_{ij}^{noobj}). From Equation 3.3, a sum is taken over all cells, bounding boxes, and classes, penalizing deviations between predicted class probabilities and true class indicators when an object is present. The weight λ_{cls} allows for adjusting the impact of the classification loss during training.

Therefore, the $\mathscr{L}_{\text{YOLO}}$ as denoted in Equation 3.4, is a comprehensive expression that integrates these three components to facilitate accurate bounding box localization, confidence score prediction, and class classification during model training.

$$\begin{aligned}\mathscr{L}_{\text{loc}} = \lambda_{\text{coord}} \sum_{i=0}^{\mathscr{N}_{\text{cell}}} \sum_{j=0}^{\mathscr{B}} 1_{ij}^{obj} \big[(t_x - \hat{t}_x)^2 + (t_y - \hat{t}_y)^2 \\ + (t_w - \hat{t}_w)^2 + (t_h - \hat{t}_h)^2 \big]\end{aligned} \tag{3.1}$$

$$\begin{aligned}\mathscr{L}_{\text{conf}} = \sum_{i=0}^{\mathscr{N}_{\text{cell}}} \sum_{j=0}^{\mathscr{B}} \Big[\lambda_{\text{conf}} 1_{ij}^{obj} (c_{ij} - \hat{c}_{ij})^2 \\ + \lambda_{\text{conf}} 1_{ij}^{\text{noobj}} (c_{ij} - \hat{c}_{ij})^2 \Big]\end{aligned} \tag{3.2}$$

$$\begin{aligned}\mathscr{L}_{\text{cls}} = \lambda_{\text{cls}} \sum_{i=0}^{\mathscr{N}_{\text{cell}}} \sum_{j=0}^{\mathscr{B}} \sum_{c=0}^{\mathscr{C}} 1_{ij}^{obj} \big[\hat{p}_{ij}^{c} \log \left(p_{ij}^{c} \right) \\ + \left(1 - \hat{p}_{ij}^{c} \right) \log \left(1 - p_{ij}^{c} \right) \big]\end{aligned} \tag{3.3}$$

$$\mathscr{L}_{\text{YOLO}} = \mathscr{L}_{\text{loc}} + \mathscr{L}_{\text{conf}} + \mathscr{L}_{\text{cls}} \tag{3.4}$$

3.3.2 Attention Mechanism

Attention mechanisms in deep learning play a vital role in enhancing model performance by addressing various challenges such as adapting to variable-length sequences, improving interpretability, optimizing memory usage, handling long-term dependencies, and enabling faster training through parallelization. They

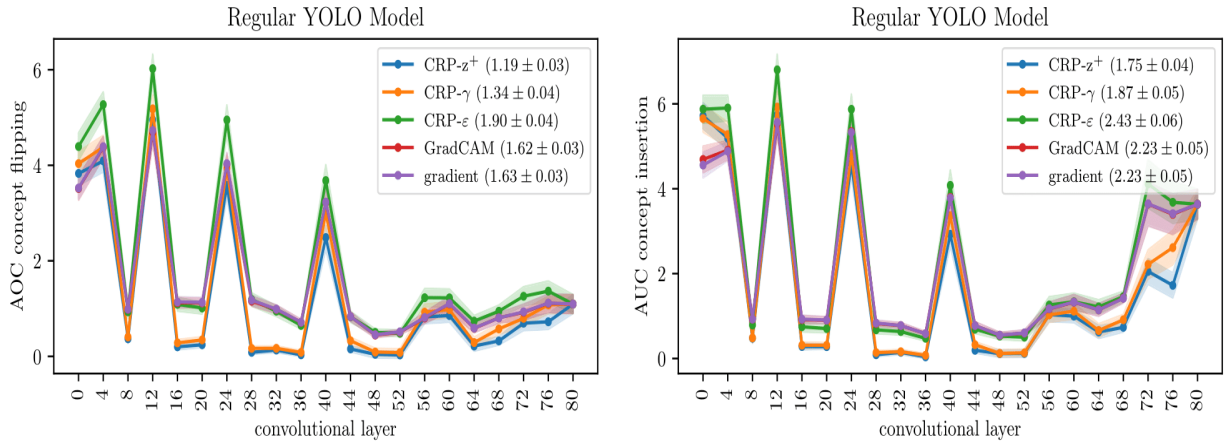


Figure 3.3: Comparing the faithfulness of various XAI approaches in attributing concepts on the YOLO model.

are particularly crucial for tasks involving diverse inputs and contribute significantly to various architectures. To boost the localization and classification performance of our YOLO model as well as improve spatial awareness and feature extraction robustness, we leverage the CBAM mechanism [106]. CBAM is strategically applied to the intermediate layers of the YOLO backbone as depicted in Fig. 3.2, integrating both channel and spatial attention mechanisms to selectively emphasize essential channel information and capture contextual details by focusing on relevant spatial locations. In the channel and spatial attention mechanisms, the scaling parameters S_{pc} and S_{ps}, are computed by applying a sigmoid activation function to the result of a double transformation using learnable weights (W_1 and W_2) on average and max pooled input feature maps (X) respectively as detailed in Equations 3.5 and 3.7. The resulting weighted channel X_c and spatial X_s feature maps are obtained through element-wise multiplication of the original feature maps with their calculated corresponding scaling parameters in Equations 3.6 and 3.8. This process emphasizes specific channels based on relevance and highlights spatial regions pertinent to the task. The Final Attended Feature Map (X_{att}) from Equations 3.9 is computed through the element-wise multiplication of X_c and X_s.

$$S_{pc} = \sigma\left(W_2\delta\left(W_1\,\text{avgpool}(X)\right)\right) \tag{3.5}$$

$$X_c = S_{pc}\cdot X \tag{3.6}$$

$$S_{ps} = \sigma\left(W_2\delta\left(W_1\,\text{maxpool}(X)\right)\right) \tag{3.7}$$

$$X_s = S_{ps}\cdot X \tag{3.8}$$

$$X_{att} = X_c \odot X_s \tag{3.9}$$

This fusion process integrates both the channel and spatial attention properties to capture rich contextual information, aiding YOLO to better understand the context of objects in images, leading to enhanced object localization and recognition, particularly in complex scenes.

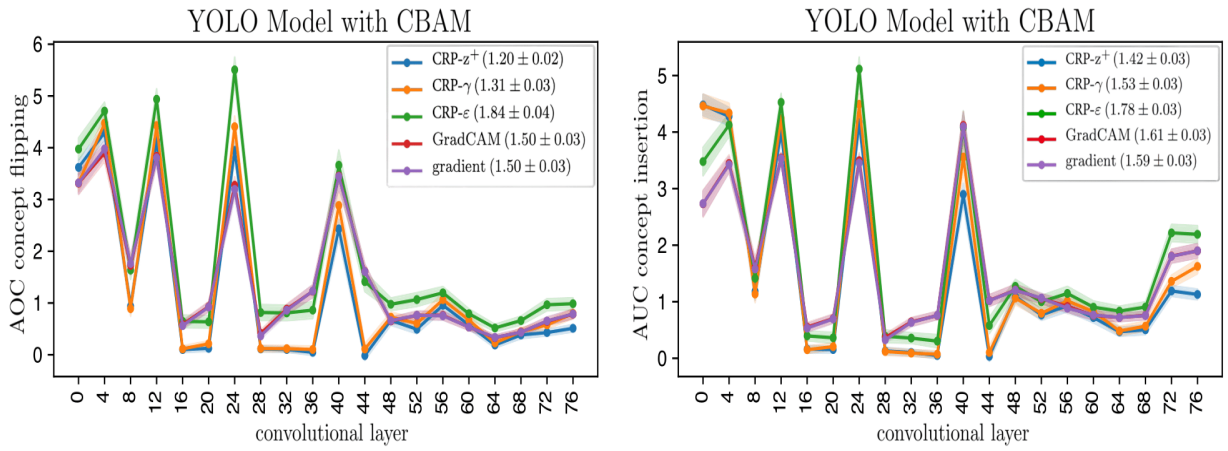


Figure 3.4: Comparing the faithfulness of various XAI approaches in attributing concepts on the Attention model.

3.3.3 eXplainable Artificial Intelligence (XAI)

As AI models advance in complexity, the interpretability of their decision-making processes diminishes, creating challenges in critical domains such as healthcare, finance, and autonomous systems where understanding AI decisions is paramount. XAI has emerged in response, driven by researchers and practitioners recognizing the need for transparency and interpretability in consequential applications. This transparency is essential for ethical AI use, fostering trust, accountability, error analysis, and the identification of model mistakes and biases. XAI facilitates human-in-the-loop collaboration regarding regulations and oversight in industries where a clearer understanding of AI models is required. It plays a vital role in making deep learning models accessible, understandable, and trustworthy, promoting responsible AI deployment across diverse applications.

In this study, the CRP algorithm [2] is utilized as a bias-resistant framework, extending the LRP [84] to offer a nuanced methodology for interpreting AI models. CRP introduces relevance maximization to propagate relevance, disentangling relevance flows associated with learned concepts. This facilitates the computation of concept-conditional relevance maps, offering insights into "what" models identify and "where" they focus their attention—providing both localized and global concept-based explanations. The relevance decomposition ($R_{i\leftarrow j}^{(l-1,l)}(\mathbf{X} \mid \theta \cup \theta_l)$) formula in Equation 3.10 embedded with filtering functionality, computes the relevance of a feature $\mathbf{X}_i$ at layer $(l-1)$ concerning a neuron j at layer l, considering conditions $(\mathbf{X} \mid \theta \cup \theta_l)$. It quantifies the importance of $\mathbf{X}_i$ for the activation of neuron j, incorporating conditions associated with both the entire model (θ) and the layer l (θ_l). The normalization term z_{ij} ensures appropriate relevance flow from neuron j to neuron i, with z_j representing the total relevance flow into neuron j. The indicator function δ_{jc_l} acts as a selector based on conditions c_l, indicating whether relevance should propagate further. Finally, $R_j^l(\mathbf{X} \mid \theta)$ represents the relevance of neuron j at layer l for the input feature $\mathbf{X}$, considering conditions θ. This nuanced approach enables a detailed understanding of attributions related to latent representations in traffic detection models.

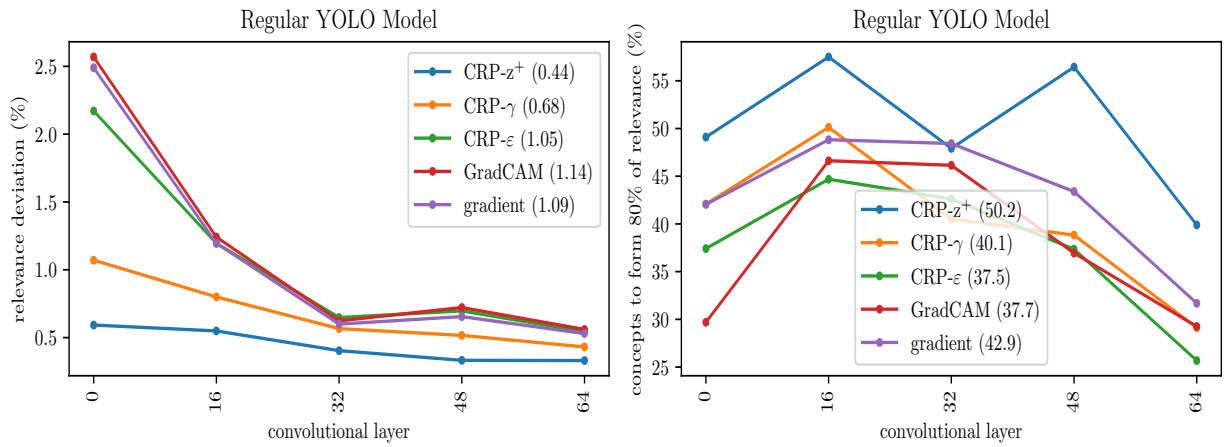


Figure 3.5: Comparing the complexity of various XAI approaches in attributing concepts on the YOLO model.

$$R_{i\leftarrow j}^{(l-1,l)}(\mathbf{X} \mid \theta \cup \theta_l) = \frac{z_{ij}}{z_j} \cdot \sum_{c_l \in \theta_l} \delta_{jc_l} \cdot R_j^l(\mathbf{X} \mid \theta) \tag{3.10}$$

By modifying the backward pass of LRP, CRP generates a concept-conditional heatmap, incorporating conditions corresponding to specific concepts of interest. The resulting pixel-level output explanations from CRP address concerns about activation-based example selection for latent concept representation, offering a holistic understanding of the traffic detection model decision-making processes. For a traffic detection model denoted as, $f : \mathbb{R}^n \rightarrow \mathbb{R}^{N\times(n_c+4)}$ with an output, Equation 3.11 initializes the relevance propagation. Here, $R_{(b,c)}^L$ represents the relevance of a feature $\mathbf{X}_i$ for bounding box k with coordinates (t_x, t_y, t_w, t_h) of class y. The initialization is determined by the Kronecker deltas δ_{bk} and δ_{cy}, ensuring relevance is attributed only to the specified bounding box and class. The term $f_c(\mathbf{X})$ denotes the output of the model for class c, serving as the starting point for relevance propagation.

$$R_{(b,c)}^L(\mathbf{X} \mid \theta) = \delta_{bk}\delta_{cy} f_c(\mathbf{X}) \tag{3.11}$$

In essence, the seamless integration of CRP into traffic detection models provides a dual advantage. Firstly, it generates transparent concept-level explanations, enhancing interpretability and reliability, particularly in AVs for traffic perception. Secondly, CRP explanations effectively mitigate biases, presenting a more robust, efficient, and interpretable solution for real-time traffic perception in AVs.

3.4 Experimental Results and Discussion

The success of an AI model hinges significantly on the quality and quantity of the utilized data. In our study, we developed a robust traffic detection model using two custom annotated datasets tailored for object detection and other computer vision tasks: The Open Images dataset by Krasin et al. [35] from Google Research and the Common Objects in Context (MS COCO) traffic dataset from Microsoft Research by Lin

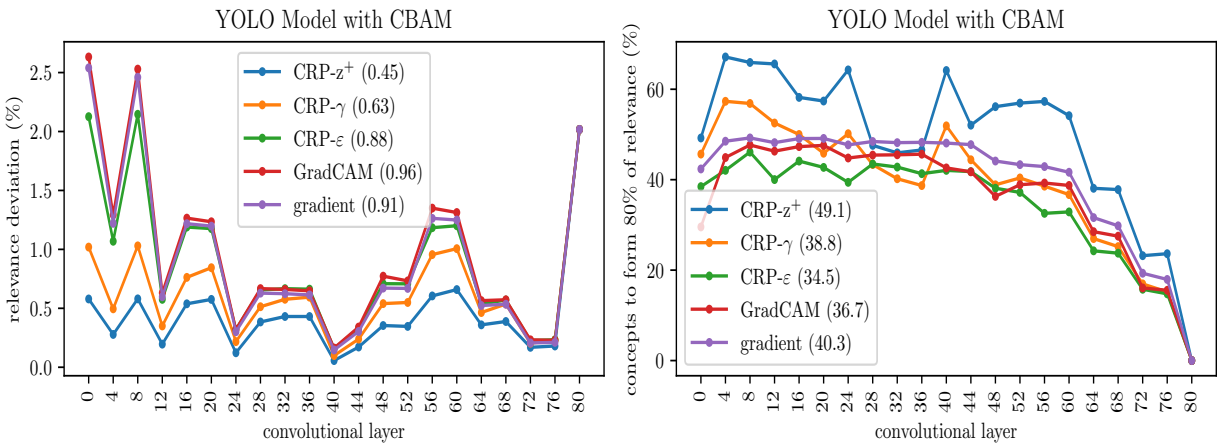


Figure 3.6: Comparing the complexity of various XAI approaches in attributing concepts on the Attention model.

Table 3.2: Traffic Detection Model Evaluation

Model	**mAP**	**Latency** (*GFLOPs*)	**Duration** (*hrs*)
YOLO Model	0.651	46.8	14.729
Attention Model	**0.696**	48.2	**14.704**
Edge-YOLO [59]	0.473	10.3	> 18
Faster-RCNN [109]	0.489	21.1	> 16

et al. [61]. A combined dataset of over 21,000 images across seven classes follows an 80%:20% train-test split and further dividing 20% of the training data for validation to fine-tune hyperparameters. We trained and fine-tuned a medium-sized YOLOv8 model on this dataset for up to 90 epochs on a Lambda GPU workstation, applying a `ReduceLROnPlateau` strategy to adjust the learning rate by a factor of 0.2 upon stalling validation loss improvements after five epochs.

Subsequently, we comprehensively evaluate the performance of our YOLOv8 traffic detection model and the CRP explainer, both before and after integrating the CBAM mechanism, allowing us to discern the impact of CBAM on both the detection model and the interpretability provided by CRP. Ultimately, we assessed the proposed relevance-based traffic detection system from three perspectives: Traffic Detection Evaluation, XAI Algorithm Evaluation, and Computational Overhead.

3.4.1 Traffic Detection Evaluation

As previously highlighted, our evaluation follows a dual approach, examining the performance of the traffic detection model both before and after the integration of the CBAM mechanism. Furthermore, we benchmark these models against contemporary MS COCO dataset related detection models commonly found in the literature. Notably, conventional models like YOLOv8 and DeepLab Model (DDDLM) [37] are typically appraised using the mean Average Precision (mAP) metric—a pivotal measure of their ability to accurately

Table 3.3: Evaluated scores for various XAI approaches in terms of faithfulness and complexity

XAI Algorithms	**Faithfulness** (↑)				**Complexity** (↓)			
	Concept Flipping (↓)		Concept Insertion(↑)		Explanation Deviation		Comprehension of 80% of attr. (%)	
	Reg_{model}	Att_{model}	Reg_{model}	Att_{model}	Reg_{model}	Att_{model}	Reg_{model}	Att_{model}
CRP - z^+	1.19	1.20	1.75	1.42	**0.44**	**0.45**	50.2	49.1
CRP - γ	1.34	1.31	1.87	1.53	0.68	0.63	40.1	38.8
CRP - ε	**1.90**	**1.84**	**2.43**	**1.78**	1.05	0.88	**37.5**	**34.5**
GRADCAM	1.62	1.50	2.23	1.61	1.14	0.96	37.7	36.7
Gradient	1.63	1.50	2.23	1.59	1.09	0.91	42.9	40.3

Table 3.4: Table Type Styles

XAI Algorithms	**Correlation** (ρ)		**RMSE**		**Duration (sec)**
	Reg_{model}	Att_{model}	Reg_{model}	Att_{model}	
CRP	**0.855**	**0.833**	**0.161**	**0.159**	**23.4**
LRP	0.806	0.772	0.221	0.228	54.1
Latent Activation Maps	0.718	0.769	0.391	0.356	> 520
GRADCAM	0.420	0.454	0.266	0.262	> 390

detect and classify instances across diverse classes.

- *mean Average Precision:* mAP as the average of the Average Precision (AP_i) values across all classes ($\mathscr{C}$).

$$mAP = \frac{1}{\mathscr{C}} \sum_{i=1}^{\mathscr{C}} AP_i \tag{3.12}$$

where AP_i represents the Average Precision computed as the area under the interpolated precision-recall curve for class i.

From Table 3.2, the YOLOv8 model without CBAM achieved a mAP of 65.1% on our custom traffic dataset. This performance outshone the Edge-YOLO model (47.3%) from Liang et al. [59] by a substantial margin of 17.3%. Similarly, our YOLOv8 model surpassed the Faster-RCNN model (48.9%) from Li et al. [109] by a significant difference of 16.2%.

Notably, Table 3.2 demonstrates that our Attention Model (YOLOv8 model infused with a CBAM mechanism) emerged as the best performing model, achieving a remarkable mAP value of 69.6%. This superiority is evident as it outperforms every other model in the table by an average margin exceeding 4.5% in mAP value, on an MS COCO related dataset. Moreover, the higher mAP value emphasized the enhanced accuracy

and reliability of our Attention Model, making it a compelling choice for AVs in ensuring robust real-time traffic perception in diverse and challenging real-world scenarios.

3.4.2 XAI Algorithm Evaluation

In tandem with the traffic detection evaluation, we introduce two key model-agnostic quantitative metrics for the evaluation the robustness and interpretability of our proposed CRP explainer, benchmarked with other explainers for 100 randomly chosen predictions:

Faithfulness: This metric measures the degree to which explanations truly represents features utilized during the internal workings of a model during inference. It quantifies the extent to which explanations reliably reflect the decision-making process of a model. The measure primarily involves two techniques: `Concept Flipping` and `Concept Insertion`. Inspired by the pixel flipping experiment, these techniques use latent concepts instead of input features, with spatial sum-aggregation computing relevance scores for each concept in a layer, treating convolutional channels as distinct concepts. For Concept Flipping, relevant channels are successively deactivated (set to zero activation), and output changes are measured to reflect the impact on the model's decision. Conversely, Concept Insertion involves initializing filters with zero activation and successively restoring relevant concepts, observing the resulting model output changes. The most faithful explanation must have higher values for both the concept flipping and insertion technique, as well as demonstrates a significant decline in performance during flipping and a substantial improvement during insertion. This decline and rise in performance are glaring from the faithfulness values of all the XAI algorithms of Table 3.3 for both the regular YOLOv8 model (Red) and the Attention model (Blue). The CRP - ε explainer, leveraging different rules of concept relevance propagation (z^+, γ, & ε) , emerges as the best-performing XAI technique. It achieves the highest faithfulness scores for with 1.90 flipping and 2.43 insertion scores, outperforming GRADCAM (1.62 & 2.23) and Gradient (1.63 & 2.23) explainers for the regular YOLOv8 model. Similarly, for the attention model, CRP - ε stands out as the most faithful explainer, though there was no substantial rise from 1.84 flipping to 1.78 insertion scores, it was the best performing explainer compared to the scores of GRADCAM (1.50 & 1.61) and Gradient (1.50 & 1.59) explainers.

Additionally, for the regular YOLOv8 model, comparing the performance plots for concept flipping (left) and insertion (right) of Fig. 3.3, there is an evident decline (stall) and rise in performance respectively after the 68th convolutional layer. Likewise, performance rise and decline after the 68th convolutional layer is observed, for concept insertion (right) and flipping (left) in the faithfulness plot of the attention model as show in Fig. 3.4.

Complexity: It gauges how easily a human can understand and comprehend the explanation provided by an XAI method. While faithfulness is crucial, presenting explanations in an understandable manner is essential for non-experts, enhancing the overall usability of the explainer, especially in AV applications. Complexity encompasses two key facets: `Explanation Deviation` and `Comprehension of 80 % of All Attributions` . Explanation Deviation assesses the diversity and range of explanations provided by an XAI method for different predictions per class, indicative of its robustness and versatility. A low deviation value suggests precise and similar explanations within the same class, reducing complexity. This level of diversity is computed using the standard deviation of latent concept attributions per class. The final deviation value for concept relevance scores $R_j(\mathbf{X}_i)$ for class $t$is with mean attribution $\bar{R}_j$ over m_s class

samples and m_c concepts is calculated using Equations 3.13 and 3.14.

$$\sigma_f = \frac{1}{m_t}\sum_{t}^{m_t}\left(\frac{1}{m_c}\sum_{j}^{m_c}\sqrt{\frac{1}{m_s-1}\sum_{i}^{m_s}\left(R_j\left(\mathbf{X}_i\right)-\bar{R}_j\right)^2}\right) \tag{3.13}$$

$$\bar{R}_j = \frac{1}{m_s}\sum_{i}^{m_s} R_j\left(\mathbf{X}_i\right) \tag{3.14}$$

The second facet measures number of concepts required to comprehend 80 % of all attributions. This is essential for user understanding and trustworthiness of the XAI method. A smaller number of concepts needed for comprehension is favorable. From Table 3.3, CRP variant explainers, utilizing the same propagation rules, exhibit the lowest complexity values (0.44 deviation and 37.5 % comprehension scores) for the regular YOLOv8 model, surpassing GRADCAM (1.14 & 37.7 %) and Gradient (1.09 & 42.9 %) explainers. This result is visualized in Fig. 3.5, for the deviation (left) and comprehension (right) plots. Similarly, for the attention model, CRP (0.45 & 34.5 %) stands out as the least complex explainer, outshining GRADCAM (0.96 & 36.7 %) and Gradient (0.91& 40.3%) techniques, as presented in Fig. 3.6 and Table 3.3. Notably, the CBAM attention mechanism contributes to reduced complexity for all XAI explainers, with observed reduction in complexity values for the attention model compared to the regular YOLOv8 model for the deviation and comprehension facets.

To comprehensively assess the robustness and interpretability of XAI techniques in detection models, we introduce the metrics, `Correlation (`ρ`)` and `Root Mean Square Error (RMSE)`. These metrics are gaged using Context (C_s) and Sensitivity (S_s), and they provide insights into the consistency of concept sensitivity under varying background conditions. (C_s) defines the ratio of positive attributions outside the predicted traffic bounding box to the overall sum, thus, revealing concept utilization variations across traffic classes, and (S_s) reflects the concept responsiveness to input perturbations. The computation of ρ and RMSE is depicted in Equations Equation 3.15 and 3.16.

$$\rho = \sum_{i}\frac{\left(C_{si}-\bar{C}_s\right)\left(S_{si}-\bar{S}_s\right)}{\sqrt{\sum_j\left(C_{sj}-\bar{C}_s\right)^2}\sqrt{\sum_k\left(S_{sk}-\bar{S}_s\right)^2}} \tag{3.15}$$

$$\text{RMSE} = \sqrt{\frac{1}{m}\sum_{i}\left(C_{si}-S_{si}\right)^2} \tag{3.16}$$

with $\bar{C}_s$ as the average Context score and the mean Sensitivity score $\bar{S}_s$ computed over m evaluated concepts.

From Table 3.4, the ρ and RMSE performances of four different XAI methods are compared for both the regular and attention models. Again, CRP was the best performing method for the regular YOLOv8 model, exhibiting the highest positive correlation (0.855) and the lowest RMSE score (0.161). This suggests consistent concept influence on decisions across diverse inputs. CRP surpassed LRP (0.806 & 0.221), Latent Activation Maps (0.718 & 0.391) and GRADCAM Maps (0.420 & 0.266) for the regular model. Similarly, for the attention model, CRP (0.833 & 0.159) outshines LRP (0.772 & 0.228), Latent Activation Maps (0.769 & 0.356) and GRADCAM Maps (0.454 & 0.262) as presented in Table 3.4, highlighting its superior

performance in maintaining concept sensitivity and overall robustness.

3.4.3 Computational Overhead

In our analysis, we examine the computational burden of the traffic detection model, considering both baseline and CBAM-integrated versions, alongside the CRP explainer in our proposed relevance-based traffic detection system. This evaluation is juxtaposed with the computational costs of other detection models in literature. For the traffic detection model, we measure model execution time in hours and computing performance in Giga Floating Point Operations Per Second (GFLOPS). Table II shows that the attention model had the lowest model execution time at 14.704 hours, yet its computing performance was the highest at 48.2 GFLOPS, aligning with expectations. Attention mechanisms contribute to selective computation, reduced input dimensionality, enhanced learning efficiency, and parallelization opportunities. Despite additional computations in both forward and backward passes, this trade-off is justified by the improved mAP performance it delivers. The regular YOLOv8 model closely follows with a model execution time of 14.729 hours and a latency performance of 46.8 GFLOPS. In contrast, the Edge-YOLO model [59] and the Faster-RCNN model [109] had longer execution durations (over 18 hours and 16 hours, respectively) and lower latency performance (10.3 GFLOPS and 21.1 GFLOPS, respectively).

Transitioning to the evaluation of the latency of our CRP explainer, benchmarked against other XAI approaches, we randomly sampled 100 predictions. Column 4 of Table IV highlights that CRP exhibited the best execution latency at 23.4 seconds compared to 54.1 seconds for LRP. Moreover, CRP showcased significantly lower latency compared to activation and its advanced variant GRADCAM, which both had latencies above 520 seconds and 390 seconds, respectively. This low latency, coupled with its enhanced performance, positions CRP as a pragmatic choice for real-time applications with resource constraints, such as AV.

3.5 Conclusion

In this study, we introduce a bias-resistant CRP XAI algorithm to proffer transparent concept–level explanations for the behavior of detection models used in autonomous systems, for traffic perception. The work further incorporates a CBAM into the YOLOv8 detection model (Attention Model) to evaluate its impact on model performance and the CRP explainer.

Our comprehensive evaluation demonstrates the effectiveness of the attention mechanism, resulting in improved detection model performance and reduced run time post-CBAM integration. The study identifies a trade-off between explanation faithfulness and complexity, positioning our CRP explainer as the optimal compromise for both the regular YOLOv8 model and the attention model. Compared to other XAI techniques, CRP stands out for its faithfulness and low complexity, offering valuable insights into critical concepts influencing model decisions. This makes CRP suitable for human-in-the-loop collaboration in industries requiring a clear understanding of AI models, aiding in regulations and oversight. The results also highlights the efficacy of CRP in maintaining consistent concept influence, contributing to the robustness, reliability, and versatility of explanations across different input scenarios. These enhanced performances coupled with its low latency, positions CRP as a pragmatic choice for real-time applications with resource constraints, such as AVs.

Chapter 4: Automating Dataset Annotation for Perception Models via eXplainable AI: A Concept Relevance Propagation Approach

4.1 Introduction

Artificial Intelligence (AI) has recently undergone a remarkable metamorphosis reshaping sectors, including transportation, healthcare, finance, and cybersecurity, endowing autonomous systems with advanced navigation, decision-making, and collaboration capabilities. In transportation, Unmanned Aerial Vehicles (UAVs) and Autonomous Vehicles (AVs) stand out. UAVs, lauded for their reach and dexterity, are essential in areas like package delivery, agriculture, emergency response, and infrastructure inspection. Similarly, AVs hold the potential to reshape transportation by offering precise, safety-driven autonomy without human oversight.

AVs operate across four key phases: perception, localization, planning, and control,[9] relying on sensors like LiDAR and RADAR. The perception phase is fundamental, involving complex deep learning (DL) tasks like road surface extraction and object recognition, which require extensive, detailed dataset annotation for object detection, lane detection, and segmentation. However, annotating such data is time-consuming, costly, and often prone to inconsistencies, especially when facing real-world complexities. Despite significant advancements in AV technology, like all other intelligent systems, AV's complete public acceptance remains a challenge due to the "black box" nature of their decision-making[3]. This opacity undermines trust and raises concerns about transparency, regulatory compliance, accountability, safety, and security, issues that have become even more pressing in light of recent AV incidents[12], [92]. eXplainable Artificial Intelligence (XAI) seeks to bridge this gap by proffering transparent, interpretable model insights that enhance trust in autonomous systems through textual, visual, and feature-importance explanations[9]. While XAI has been extensively applied in fields like healthcare[88], [119] and cybersecurity[74], its adoption in autonomous system perception tasks, such as object detection and segmentation, remains limited. Although some studies have explored the benefits, challenges, and strategies for incorporating XAI into the AV domain[9], most contemporary research focuses narrowly on explaining model behavior in these tasks[71]. This overlooks a broader opportunity to leverage XAI beyond its traditional explainability role, to enhance both safety and performance in AV systems.

As AI advances, building efficient models requires extensive, diverse datasets, increasing the need for annotated data in some scenarios. To address the time-intensive nature of manual annotation[36], [86], Corso[25] advocates leveraging advanced AI techniques to automate the process. Companies like Roboflow[105], Meta[45], and SuperAnnotate[94] offer data annotation and management services that streamline computer

vision (CV) workflows, to enhance labeling quality and consistency, while reducing annotation time to facilitate the development of robust perception models. However, these services are costly and still depend on manually annotated datasets for pre-training, especially when applying the auto-labeling features to new, custom datasets[94], [105], where performance remains minimal.

Our work addresses the dual challenge of transparency and automated annotation in AV perception model development by introducing a novel framework leveraging the bias-resistant Concept Relevance Propagation (CRP) XAI technique[2]. This framework enhances model interpretability and automates dataset annotation for perception tasks. By integrating Relevance Maximization (Rel-Max)[2], CRP provides transparent explanations by pinpointing highly critical concepts and input regions used for network encodings, that influence object detection, improving both model transparency and reliability. Additionally, we combine CRP with semi-supervised learning to generate high-quality automated annotations, significantly streamlining the annotation process and reducing manual effort to under 189 seconds for a 15,000-sample dataset. Our results show that models trained on our auto-annotated data achieved at least 1.4% higher mAP scores with lower latency than models trained on pre-annotated datasets. This offers a faster, more cost-effective solution for perception model development while promoting safer, more transparent autonomous systems. In specific, our contributions are as follows:

- We propose a pipeline to enhance model interpretability, efficiency, and automate dataset annotation for perception models. Using a custom dataset comprising Open Images Dataset V7[35] and Microsoft COCO Dataset[61], partitioned into 75% raw and 25% annotated data, we train an object detection model on the annotated subset and apply CRP to interpret model behavior on the raw data. To further optimize performance, we incorporate a Convolutional Block Attention Module (CBAM)[106] into the feature extraction layers, improving feature selection and boosting both detection accuracy and model transparency, ensuring reliable autonomous systems.

- Our approach leverages CRP-generated concept-level explanations[2] to automate data annotation. By transforming heatmap-based explanations of each test point into bounding box contours, we effectively localize relevant concepts across the raw dataset, eliminating the need for manual labeling while validating the feasibility of automated annotations. This method drastically reduces annotation time and cost while ensuring high-quality, consistent labels, offering a scalable and efficient solution for dataset preparation in complex object detection tasks.

- We comprehensively evaluate our system by comparing models trained on our auto-annotated dataset against those trained on datasets labeled via pre-annotation[35], [61], active learning[23], [77], [110], and Roboflow's auto-labeling[105] methods, all of equal size. This comparative analysis demonstrates the effectiveness of our approach. Additionally, we assess the CRP explainer's performance using metrics such as faithfulness and complexity, providing deeper insights into XAI techniques in object detection tasks, offering a novel perspective on the interplay between annotation strategies and model explainability.

The reminder of this chapter is organized as follows in the sections that follow: While Section 4.3 describes our proposed approach, Section4.2 reviews contemporary literature. Section4.4 presents the results and findings, and Section4.5 offers a summary of the conclusions and possible directions for further research.

4.2 Related Work

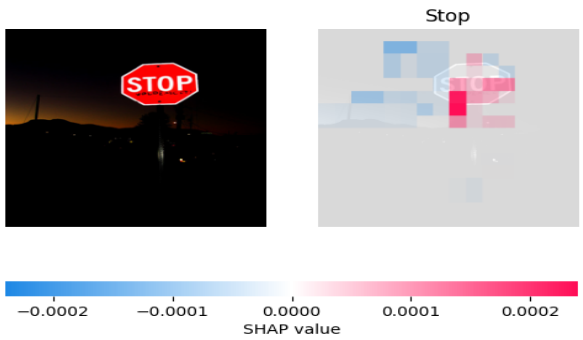
4.2.1 eXplainable AI (XAI) Techniques in Object Detection

Attribution-based XAI techniques (LIME, SHAP, Saliency Maps, etc.) widely employed in literature have intrinsic limitations to their significant role of improving model transparency. SHAP[63], for instance, uses positive (red) and negative (blue) masks to explain predictions (Stop class), as shown in Fig. 4.1a[4] (Preliminary study). While effective for pixel-level attributions, these methods fail to capture higher-level, concept-driven explanations essential for precise annotation tasks. In Fig. 4.1a[4], the most significant positive contribution to the "stop sign" prediction is misattributed to extraneous regions, including the bottom-left area of the octagonal shape and surrounding pixels outside the stop sign itself. Such inaccuracies render attribution-based techniques unsuitable for the proposed high-quality automated annotation framework. Another notable limitation of SHAP, is their vulnerability to biases due to its symmetric treatment of positive and negative feature contributions, which can obscure feature importance imbalances and lead to skewed interpretations, affecting dependability (faithfulness). Additionally, these methods have high computational demands, thereby limiting their thorough utilization across diverse applications, particularly in large-scale or resource-constrained scenarios.

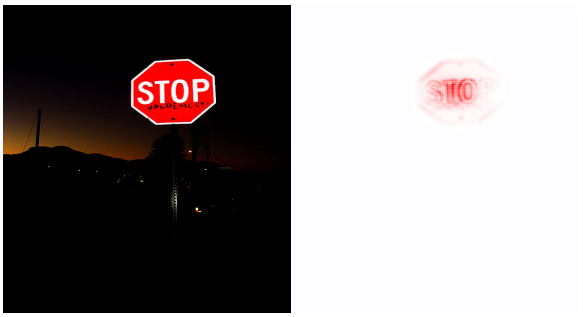
In contrast, CRP tackles the difficulties of traditional XAI attribution maps by by prioritizing concept-level relevance over pixel-level attributions within the input space. For example, Fig. 4.1b[4] emphasizes the concept of a "STOP text in an Octagon" showing how the model learns to identify stop signs during traffic perception. By focusing on higher-level, concept-aware features, CRP enables transparent, interpretable insights into model decision-making processes, while reducing attribution biases, and ensuring precise localizations. This makes CRP better suited for advance automated annotation pipelines requiring high accuracy and scalability, for robust object detection solutions.

4.2.2 Synergy Between Attention Mechanisms and XAI

Attention mechanisms in DL dynamically prioritize key input features, enhancing model performance by focusing on the most relevant information. By computing attention scores and generating weighted input feature representations, these mechanisms improve decision-making and model robustness. Recent studies indicate that attention mechanisms can also enhance the interpretability of post-hoc XAI techniques. For instance, Lee[51] employed Luong attention to highlight critical EMG signals for predicting finger joint angles, while Shi[89] utilized a deformable attention module (DAM) to emphasize infection regions, improving both model accuracy and interpretability using GRADCAM and Layer-wise Relevance Propagation (LRP). However, existing literature lack direct analysis on how attention mechanisms affect the performance of XAI methods in object detection, particularly their impact on the Concept Relevance Propagation (CRP) algorithm. Furthermore, the potential of CRP for automating data annotation in object detection remains unexplored. Our work bridges these gaps by investigating the influence of CBAM on CRP, demonstrating improved XAI performance, and introducing CRP as an effective tool for automated dataset annotation, addressing a critical bottleneck in object detection pipelines.



(a) **SHAP Explanation.**



(b) **CRP Explanation**

Figure 4.1: Comparison of Interpretability Between Attribution-Based 4.1a and Concept Relevance-Based 4.1b XAI Algorithms

4.2.3 Data Annotation Techniques for Object Detection

Annotation is critical for developing object detection models, providing the ground truth data necessary for training and evaluation. Manual annotation, previously the standard, involved experts generating precise bounding boxes and class labels[83], requiring approximately 35 seconds per object class annotation[36], [86], to ensure high accuracy. But posing significant challenges in terms of time and cost, particularly with large-scale datasets. As datasets grew in size and complexity, this process became a bottleneck for efficient model development, especially in applications like AV perception. To address this, the AI community has developed advanced techniques such as active learning[23], semi-supervised learning[18], self-supervised learning[19], and synthetic data generation[34] to reduce dependence on manual labeling. While these approaches improve annotation efficiency, they still rely on manually labeled data for pre-trained models and require extensive iterative training, particularly for unseen, custom datasets. Industry solutions like `Roboflow` and `SuperAnnotate` provide automated annotation tools but struggle with novel datasets, requiring partial manual annotations and remaining prohibitively expensive[94], [105].

Although studies have examined the influence of annotation quality, efficiency, and methods on model performance[6], [103], the application of XAI for automated annotation remains largely unexplored. Our work bridges this gap by employing the CRP XAI algorithm to both interpret object detection model behavior and automate dataset annotation processes. This innovative approach enhances training data preparation efficiency while extending the application of XAI beyond traditional interpretability, offering a scalable solution for object detection model development.

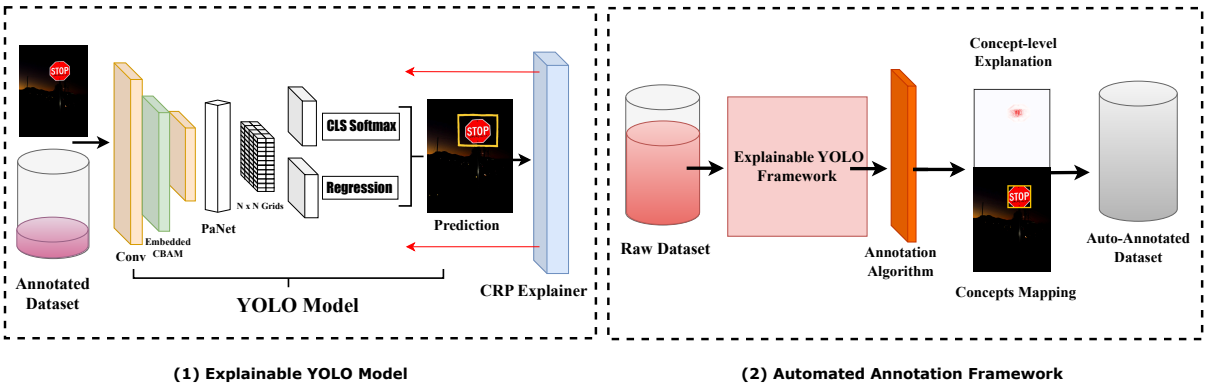


Figure 4.2: **Framework of the Relevance-Based Explainable Automated Annotation System.** The framework showcases two components: (1) An Explainable CBAM-enhanced YOLO Model, and (2) An Automated Annotation Framework, which uses CRP-generated concept-level explanations from the YOLO model to streamline and automate data annotation.

4.3 Methodology

Overview: Our approach involves two core processes: developing an explainable object detection model and leveraging it for automated dataset annotation. As illustrated in Fig. 4.2, we design a CBAM-enhanced You Only Look Once (YOLO) detection model to boost detection accuracy and augment the interpretability of the CRP explainer. This explainable model is then applied to automate the annotation of a 15,000-sample dataset spanning seven categories, significantly reducing reliance on manual labeling. Partitioning the dataset into 75% raw, unannotated test data and 25% annotated training data, with 20% of the training set reserved for validation, the detection model is trained and subsequently evaluated. The CRP explainer generates concept-level explanations for each test sample, visualized as heatmaps that localize and delineate relevant concepts into bounding box contours. This automated annotation process, informed by model-driven insights, offers a scalable, efficient alternative to traditional manual labeling methods. Detailed descriptions of each stage are provided in the subsequent sections with notations referenced in Table 4.1.

YOLO-based Object Detection Framework. Autonomous systems heavily rely on object detection models for accurate and timely decision-making. We utilized the YOLO model, first introduced by Redmon et al. [82], in our study because of its real-time capabilities, making it essential for applications requiring minimal latency. The YOLO architecture as illustrated in Fig. 4.2, features a CNN-based Backbone for feature extraction, a Path Aggregation Network (PANet) for enhanced multi-scale feature fusion, and a convolutional layer-based Detection Head that predicts bounding boxes ($\mathscr{B}$), confidence scores, and class probabilities ($\mathscr{C}$) in a single pass. YOLO divides an input image into grids ($\mathscr{N}_{\text{cell}}$), with each grid cell predicting bounding box coordinates (t_x, t_y, t_w, t_h) and class probabilities using softmax activation, producing simultaneous results for all bounding boxes and classes. We have updated the YOLO model to include an attention mechanism to enhance its performance and interpretability (see Section 4.3.1) so it can be used for automated annotation tasks (see Section 4.3.2).

The performance of YOLO depends on its loss function ($\mathscr{L}_{\text{YOLO}}$), which consists of three key compo-

Table 4.1: Main Notations.

Notation	Description
$\mathscr{N}_{\text{cell}}$	Total number of cells in the grid
$\mathscr{L}$	Loss function
C, B	Object classes and predicted bounding boxes
T	Set of IoU thresholds
f	Object Detection Model
a, w	Activations and learnable model weights
δ, σ	ReLU, sigmoid activation functions
X	Feature map
$c_{ij}, \hat{c}_{ij}$	Predicted and ground truth confidence score in cell (i, j)
λ	Weight for loss calculation
I_{ij}	Indicator function for traffic detection in cell (i, j)
R	Relevance Quantity
$p^c_{ij}, \hat{p}^c_{ij}$	Predicted and ground truth class probabilities in cell (i, j)

nents: Localization loss ($\mathscr{L}_{\text{loc}}$), Classification loss ($\mathscr{L}_{\text{cls}}$), and Confidence score loss ($\mathscr{L}_{\text{conf}}$). $\mathscr{L}_{\text{loc}}$ (Equation 4.1) measures the accuracy of predicted bounding box coordinates (t_x, t_y, t_w, t_h) against the ground truth $(\hat{t}_x, \hat{t}_y, \hat{t}_w, \hat{t}_h)$, weighted by localization loss weight (λ_{coord}) and indicated by the function (I^{obj}_{ij}) for object presence. $\mathscr{L}_{\text{cls}}$ (Equation 4.2) compares predicted class probabilities (p^c_{ij}) to ground truth class indicators (I^{noobj}_{ij}), punishing deviations when an object is present and modulating its impact with weight (λ_{cls}). $\mathscr{L}_{\text{conf}}$ (Equation 4.3) evaluates confidence ratings for both object and no-object cases, using weights (λ_{conf}) to balance their importance and guarantee precise confidence score predictions. Equation 4.4 incorporates these elements into the overall $\mathscr{L}_{\text{YOLO}}$ loss function, enabling precise bounding box localization, confidence score prediction, and class classification during training.

$$\begin{aligned}\mathscr{L}_{\text{loc}} = \lambda_{\text{coord}} \sum_{i=0}^{\mathscr{N}_{\text{cell}}} \sum_{j=0}^{\mathscr{B}} I_{ij}^{obj} \big[(t_x - \hat{t}_x)^2 + (t_y - \hat{t}_y)^2 \\ + (t_w - \hat{t}_w)^2 + (t_h - \hat{t}_h)^2 \big]\end{aligned} \tag{4.1}$$

$$\begin{aligned}\mathscr{L}_{\text{cls}} = \lambda_{\text{cls}} \sum_{i=0}^{\mathscr{N}_{\text{cell}}} \sum_{j=0}^{\mathscr{B}} \sum_{c=0}^{\mathscr{C}} I_{ij}^{obj} \big[\hat{p}^c_{ij} \log\left(p^c_{ij}\right) \\ + \left(1 - \hat{p}^c_{ij}\right) \log\left(1 - p^c_{ij}\right) \big]\end{aligned} \tag{4.2}$$

Table 4.2: Distribution of Object Classes in Training and Test Sets for Autonomous Vehicle Perception.

Classes	Training Set	Test Set
Traffic Light	752	2,256
Bicycle	895	2,670
Stop Sign	611	1,832
Car	1,350	4,061
Motorcycle	824	2,476
Bus	513	1,539
Truck	471	1,414
Total	**5,416**	**16,248**

$$\mathscr{L}_{\text{conf}} = \sum_{i=0}^{\mathscr{N}_{\text{cell}}} \sum_{j=0}^{\mathscr{B}} \Big[\lambda_{\text{conf}} I_{ij}^{obj} \left(c_{ij} - \hat{c}_{ij}\right)^2 + \lambda_{\text{conf}} I_{ij}^{\text{noobj}} \left(c_{ij} - \hat{c}_{ij}\right)^2 \Big] \tag{4.3}$$

$$\mathscr{L}_{\text{YOLO}} = \mathscr{L}_{\text{loc}} + \mathscr{L}_{\text{cls}} + \mathscr{L}_{\text{conf}} \tag{4.4}$$

4.3.1 Enhancing Object Detection Performance with Attention Mechanisms

To understand the durability, robustness, and dependability of the CRP XAI technique, we explored how model generalization improvements via attention mechanisms and dataset augmentation could impact our CRP explainer and the overall auto annotation process. In general, attention mechanisms enhance neural networks by selectively focusing on the most relevant parts of the input data, quantifying feature relevance using attention scores, and creating weighted combinations to highlight significant features. This allows neural networks to handle long-term dependencies, optimize memory usage, and adapt to variable-length sequences. In our work, we utilized the CBAM [106] to enhance the resilience of feature extraction, spatial awareness, and localization performance of our YOLO model. CBAM is applied to the intermediate layers of the YOLO backbone, incorporating both spatial and channel attention processes to collect contextual details. This is achieved by focusing on important spatial regions and selectively highlighting important channel data. The attention process involves two sequential sub-processes: the channel attention mechanism followed by the spatial attention mechanism. These mechanisms work together to refine feature representations and improve the model's ability to detect objects. Equation 4.5 explain how the scaling parameters $\mathbf{S}_{channel}$ and $\mathbf{S}_{spatial}$ in the channel and spatial attention processes, respectively, are calculated. This involves

applying a sigmoid activation function $\sigma(.)$ to the output of a double transformation utilizing learnable weights ($\mathbf{W}_1$ and $\mathbf{W}_2$) on average and max pooled input feature maps ($\mathbf{X}$) after applying a ReLU activation function $\delta(.)$ to the output. The resulting weighted channel $\mathbf{X}_{channel}$ and spatial $\mathbf{X}_{spatial}$ feature maps are obtained through element-wise multiplication of the original feature maps with their corresponding scaling parameters, as seen in Equation 4.5. These weighted feature maps permit the selective emphasis of relevant channels and spatial regions crucial to the task. The element-wise multiplication of $\mathbf{X}_{channel}$ and $\mathbf{X}_{spatial}$ yields the final attended feature map ($\mathbf{X}_{\text{att}}$). This fusion process integrates both channel and spatial attention properties to capture rich contextual information, aiding the YOLO model in better understanding the context of features in images, leading to enhanced object localization and recognition, particularly in complex scenes as will be shown in Section 4.4.1.

$$\begin{aligned}
\mathbf{S}_{channel} &= \sigma\left(\mathbf{W}_2\delta\left(\mathbf{W}_1\,\text{avgpool}(\mathbf{X})\right)\right) \\
\mathbf{S}_{spatial} &= \sigma\left(\mathbf{W}_2\delta\left(\mathbf{W}_1\,\text{maxpool}(\mathbf{X})\right)\right) \\
\mathbf{X}_{channel} &= \mathbf{S}_{channel}\cdot\mathbf{X} \\
\mathbf{X}_{spatial} &= \mathbf{S}_{spatial}\cdot\mathbf{X} \\
\mathbf{X}_{att} &= \mathbf{X}_{channel}\odot\mathbf{X}_{spatial}
\end{aligned} \tag{4.5}$$

The enhanced performance of the YOLO model also extends to improved interpretability of post-hoc XAI algorithms, particularly the CRP XAI algorithm. The attention process strategically filters out noise and hones in on critical input information, ultimately reducing the available pool of features from which our CRP explainer selects relevant ones contributing to crucial concepts responsible for model decisions. Thus, in addition to improved detection performance, deploying the attention mechanism allows us to observe its positive impact on the utility of the CRP explainer (see Section 4.4.2).

4.3.2 XAI for Automated Annotation

As AI algorithms become increasingly sophisticated and widely utilized in critical domains such as autonomous systems, the need for understanding AI decisions has become paramount. XAI addresses this need by enhancing transparency and interpretability of deep learning models. In our work, we leverage XAI techniques, specifically the CRP algorithm, to improve the automated annotation process for object detection tasks. Providing in-depth insights into the model's decision-making process allows CRP to generate high-quality annotations automatically, significantly reducing the time and cost associated with manual data labeling. This approach not only enhances the efficiency of dataset creation but also improves the overall performance and reliability of our YOLO-based object detection model. Furthermore, by integrating XAI into our workflow, we create a more accountable and interpretable system, which is important for the development and deployment of safe and trustworthy AI technologies in various computer vision applications, particularly in the context of autonomous systems and traffic sign detection.

Concept Relevance Propagation (CRP)

In our work, the CRP XAI algorithm [2] is used as an advanced, bias-resistant technique that builds upon Layer-wise Relevance Propagation (LRP) [10], [84] to provide detailed comprehensible explanations of DL

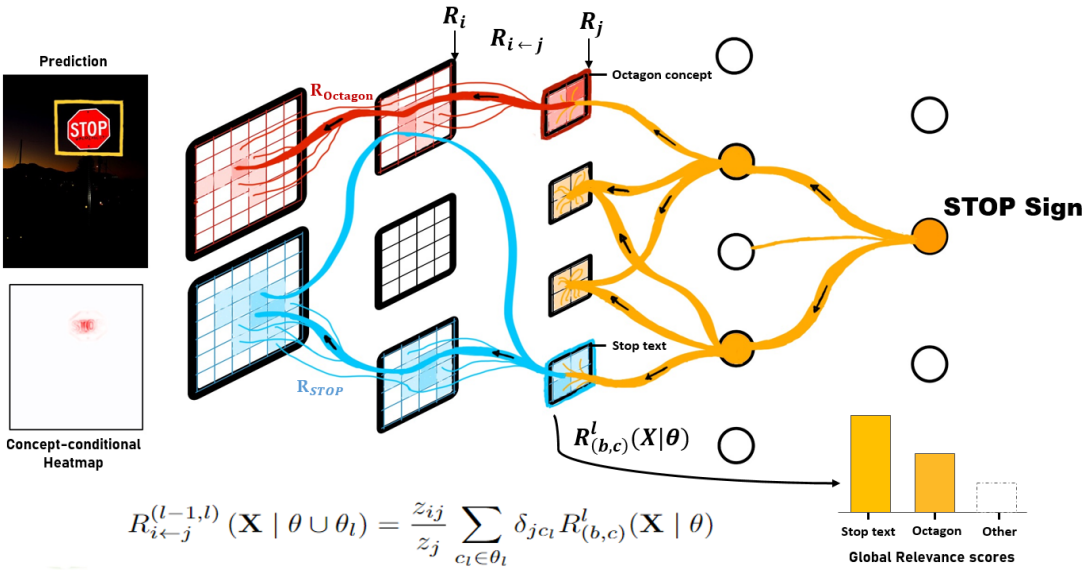


Figure 4.3: CRP process from output prediction to preceding layers in an object detection model. The figure illustrates how relevance is propagated from the detected "STOP Sign" class through the network, highlighting key concepts such as the octagon shape and the stop text.

model reasoning. Traditionally, LRP explains model predictions by attributing the relevance of the output score to neurons in the network, from the output layer back to the input features (pixels), highlighting the importance of these neurons in the inference process. CRP enhances this by decomposing relevance flows using Relevance Maximization (RelMax) and introducing conditions that target specific learnt concepts for more in-depth relevance backpropagation. CRP employs binary masking of relevance tensors to compute concept-conditional relevance maps $\mathbf{R}(x \mid \theta)$, where x represents the input data point and θ denotes conditions such as output categories or specific concepts (e.g., the "Octagon" shape in a stop sign). This approach yields both local and global explanations, providing insights into `what` concepts the model recognizes, `where` these concepts are located within the input image, and `how` they contribute to the model's prediction. To illustrate how LRP works, assuming a DL model $f(x) = f_1 \circ \cdots \circ f_l(x)$, where $f(x)$ represents the forward pass prediction and $f_l, \ldots, f_1$ are the network layers. For a particular layer, pre-activations $\mathbf{z}_{ij}$ maps input $\mathbf{x}_i$ to output j as shown in Equation 4.6, where $\mathbf{x}_i$ is the input and $\mathbf{w}_{ij}$ is the weight. Aggregated pre-activations and activations for the next layer are shown in Equations 4.7 and 4.8, respectively. Equation 4.9 illustrates how LRP distributes relevance R_j from output j to preceeding neuron i, with the overall relevance of each neuron i being a sum of the incoming divided relevance, as indicated in Equation 4.10.

$$\mathbf{z}_{ij} = \mathbf{x}_i \cdot \mathbf{w}_{ij} \quad (4.6) \quad \text{and} \quad \mathbf{z}_j = \sum_i \mathbf{z}_{ij} \quad (4.7)$$

$$a_j = \sigma \cdot (\mathbf{z}_j) \quad (4.8) \quad \text{and} \quad R_{i \leftarrow j} = \frac{\mathbf{z}_{ij}}{\mathbf{z}_j} \cdot R_j \quad (4.9)$$

$$R_i = \sum_j R_{i \leftarrow j} \quad (4.10)$$

CRP enhances LRP by modifying its relevance decomposition ($\mathbf{R}_{i\leftarrow j}$) to include conditions, computing the relevance of a feature $\mathbf{X}_i$ at layer $(l-1)$ with respect to a neuron j at layer l, considering conditions $(\mathbf{X} \mid \theta \cup \theta_l)$. This relevance decomposition, expressed in Equation 4.11, quantifies the importance of a feature $\mathbf{X}_i$ for the activation of j, incorporating conditions associated with both the entire model (θ) and the layer θ_l. The normalization term ($\frac{\mathbf{z}_{ij}}{\mathbf{z}_j}$) ensures appropriate relevance flow from neuron j to neuron i, with $\mathbf{z}_j$ representing the total relevance flow from neuron j. The Kronecker delta (δ_{jc_l}) as used in Equation 4.11, is a mathematical identity function shown in Equation 4.12, it acts as a selector, ensuring relevance is only propagated through neurons relevant to the specified concepts c_l in θ_l. If neuron j corresponds to concept c_l, $\delta_{jc_l} = 1$, otherwise it is 0, effectively filtering the relevance flow. Finally, $R_j^l(\mathbf{X} \mid \theta)$ is the relevance assigned to layer output j from the CRP process in upper layers, conditioned on θ. This approach enables CRP to disentangle and highlight the contributions of specific concepts within models, offering a clearer and more interpretable understanding of model predictions.

$$R_{i\leftarrow j}^{(l-1,l)}(\mathbf{X} \mid \theta \cup \theta_l) = \frac{\mathbf{z}_{ij}}{\mathbf{z}_j} \cdot \sum_{c_l \in \theta_l} \delta_{jc_l} \cdot R_j^l(\mathbf{X} \mid \theta) \tag{4.11}$$

$$\delta_{jc_l} = \begin{cases} 1 & \text{if } j = c_l \\ 0 & \text{if } j \neq c_l \end{cases} \tag{4.12}$$

For an object detection model $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^{N\times(n_c+4)}$, where $\mathbb{R}^n$ is the input space with n features (pixels), and $\mathbb{R}^{N\times(n_c+4)}$ is the output space with N bounding box predictions, each associated with n_c possible object classes and 4 bounding box coordinates, Equation 4.13 initializes relevance propagation for such a model, as shown in Fig. 4.3. In this context, the relevance $R_{(b,c)}^l(\mathbf{X} \mid \theta)$ represents the relevance of a feature $\mathbf{X}_i$ for a specific bounding box k with coordinates (t_x, t_y, t_w, t_h) of class y (e.g., the "STOP Sign" class as illustrated in Fig. 4.3) to be propagated. $R_{(b,c)}^l$ indicates relevance specific to bounding box b and class c, with Kronecker deltas δ_{bk} and δ_{cy} in Equations 4.14 and 4.15, ensuring relevance is attributed only to the predicted bounding box k and class y (the detected "STOP Sign" class of Fig. 4.3). The term $f_c(\mathbf{X})$ denotes the model output for class y, serving as the starting point for relevance propagation. Furthermore, Equation 4.16 describes how relevance $R_{(b,c)}^l$ is propagated from layer l to the previous layer $(l-1)$ for the feature $\mathbf{X}_i$. Here, $\mathbf{z}_{ij}$ and $\mathbf{z}_j$ are normalization terms, δ_{jc_l} is a Kronecker delta ensuring the relevance for the detected "STOP Sign" class is only attributed to the specific class concepts c_l, such as "The Octagon shape" or the "Stop text" in our case, belonging to the conditions of the last layer θ_l. Finally, Equation 4.17 sums up all the relevance scores $R_{i\leftarrow j}^{(l-1,l)}$ for feature $\mathbf{X}_i$ from the previous layer, giving the total relevance R_i for that feature.

$$R_{(b,c)}^l(\mathbf{X} \mid \theta) = \delta_{bk} \cdot \delta_{cy} \cdot f_{(b,c)}(\mathbf{X}) \tag{4.13}$$

$$\delta_{bk} = \begin{cases} 1 & \text{if } b = k \\ 0 & \text{if } b \neq k \end{cases} \tag{4.14}$$

$$\delta_{cy} = \begin{cases} 1 & \text{if } c = y \\ 0 & \text{if } c \neq y \end{cases} \tag{4.15}$$

$$R_{i\leftarrow j}^{(l-1,l)}(\mathbf{X} \mid \theta \cup \theta_l) = \frac{\mathbf{z}_{ij}}{\mathbf{z}_j} \sum_{c_l \in \theta_l} \delta_{jc_l} R_{(b,c)}^l(\mathbf{X} \mid \theta) \tag{4.16}$$

Algorithm 1: CRP-Enhanced Automated Dataset Annotation for Object Detection Tasks

Input: Dataset $\mathscr{D}$, Detection Model $\mathscr{M}$
Output: Auto-annotated dataset $\mathscr{D}_{\text{auto}}$
function `Train_Model(`$\mathscr{D}$`)`
    Split: $\mathscr{D}_{\text{train}}, \mathscr{D}_{\text{raw}} \leftarrow \text{split}(\mathscr{D}, 0.25)$
    Train detector: $\mathscr{M}^* \leftarrow \text{Optimize}(\mathscr{M}, \mathscr{D}_{\text{train}})$
    return Trained model $\mathscr{M}^*$
function `Apply_CRP(`$\mathscr{M}^*, x, (b,c)$`)`
    Forward pass: $P \leftarrow \text{Infer}(\mathscr{M}^*, x)$
    Initialize relevance scores: $R^l_{(b,c)}(x|\theta) = \delta_{bk}\delta_{cy} f_{(b,c)}(x)$
    if $b = k$ *and* $c = y$ **then**
        $\delta_{bk} = 1$, $\delta_{cy} = 1$ // Propagate only target relevance
    else
        $\delta_{bk} = 0$, $\delta_{cy} = 0$ // Stop relevance propagation
    Propagate: $R^{(l-1,l)}_{i \leftarrow j}(x \mid \theta \cup \theta_l) = \frac{z_{ij}}{z_j} \sum_{c_l \in \theta_l} \delta_{jc_l} R^l_{(b,c)}(x|\theta)$
    Aggregate: $R_i = \sum_j R^{(l-1,l)}_{i \leftarrow j}$
    return Concept-level relevance map R_i
function `Identify_Concepts(`R_i`)`
    Derive key concepts $C \leftarrow$ cluster/segment(R_i)
    Threshold map: binary_map $\leftarrow$ Threshold$(R_i, 0.5)$
    Extract contours: contours $\leftarrow$ findContours(binary_map)
    for *each contour in contours* **do**
        Compute box: $(t_x, t_y, t_w, t_h) \leftarrow$ boundingRect(contour)
        Draw visualization: drawRectangle$(x, c, (t_x, t_y), (t_x + t_w, t_y + t_h), (0, 255, 0), 2)$
    return Bounding box set $B = \{(t_x, t_y, t_w, t_h)\}$
function `Auto_Annotate(`$\mathscr{D}, \mathscr{M}^*$`)`
    Initialize $\mathscr{D}_{\text{auto}} = \emptyset$
    for *each* $x \in \mathscr{D}_{raw}$ **do**
        $P \leftarrow \text{Infer}(\mathscr{M}^*, x)$
        for *each* $(b,c) \in P$ **do**
            $R_i \leftarrow$ `Apply_CRP(`$\mathscr{M}^*, x, (b,c)$`)`
            $B \leftarrow$ `Identify_Concepts(`R_i`)`
            Append (x, B, c) to $\mathscr{D}_{\text{auto}}$
    return $\mathscr{D}_{\text{auto}}$

$$R_i = \sum_j R^{(l-1,l)}_{i \leftarrow j} \tag{4.17}$$

Subsequently, this specialized form of conditional relevance propagation can be applied to all detections within the test data, with an extension to all data points in the test dataset. Thus, integrating CRP into object detection models used in AV perception allows for the generation of transparent concept-conditional heatmaps, improving interpretability and dependability by reducing model bias. Additionally, CRP's pixel-level heatmap explanations provide a foundation for automating annotations through concept localizations, addressing key challenges in object detection model development.

Automated Annotation

Organizations such as Roboflow [105], Meta [45], and SuperAnnotate [94] have made significant strides in reducing the annotation burden by introducing AI-assisted tools. However, these solutions still depend on human input, can be costly and prone to errors such as inconsistent annotations, especially when dealing with niche datasets or complex objects. These challenges can impact the overall performance of the models. To overcome these limitations, contemporary literature explore advance approaches, such as context-aware models [33], temporal information from video sequences [40], and a human-in-the-loop process for error correction. We propose a novel method that goes beyond traditional approaches by leveraging XAI, specifically the CRP algorithm, for automated annotation. This approach not only enhances the detection model's performance but also delivers annotations that are grounded in the model's transparent, concept-level explanations. The result is a reliable and scalable solution to data preparation, significantly reducing the need for human oversight.

Our proposed automated annotation scheme, which integrates semi-supervised learning with XAI, is detailed in Algorithm 1. Using a 21,000-sample dataset, the algorithm begins by splitting the data into a training set (25% annotated) and a raw test set (75% unannotated) (line 2). It then trains a YOLO detection model [99] on the annotated data (line 3). Once trained, the model generates bounding boxes and associated class labels for each data point in the raw dataset (line 5). To analyze and refine these predictions, the algorithm utilizes the CRP XAI technique (line 7). CRP calculates relevance scores for each bounding box and class, indicating the contribution of each feature to the model's decision (line 8). By using Kronecker delta functions (lines 9-14), the CRP ensures precise relevance backpropagation, focusing solely on the relevant bounding box and class. Relevance is then propagated from the model's output layer back to the input features (line 15), aggregating the total relevance for each feature (line 16). This process helps identify the key concepts the model focused on during its prediction (line 17). The relevance scores are thresholded to create a binary map that highlights significant concepts, excluding extraneous features (line 18). These focused explanations are used to identify contours in the input data (line 19), and bounding boxes are drawn around these contours (lines 20-23), accurately capturing the regions of interest the model relied on.

The generated bounding boxes, classes and their associated data are added to the auto-annotated dataset (line 25), progressively transforming the raw dataset into a fully annotated one. This process is repeated for each data point in the raw dataset (lines 4-26), ensuring comprehensive and consistent coverage. The resulting fully annotated dataset that closely aligns with concept-level explainability, effectively reduces biases and produces high-quality precise automated annotations. This approach significantly minimizes the need for manual annotation, making the annotation process more efficient.

4.4 Experimental Results and Discussion

The experiments for our proposed automated annotation framework were conducted in Python on a Lambda GPU workstation with dual Quadro RTX 8000 GPUs (2-Way NVLink), an Intel i9-9820X CPU (10 cores), 128GB RAM, and a 2TB NVMe SSD. We utilized two traffic datasets: the Open Images dataset by Google Research [35] and the MS COCO traffic dataset by Microsoft Research [61], combining over 21,000 images across seven object classes, as shown in Table 4.2. A medium-sized YOLO version 8 model was trained from scratch over 80 epochs using the ReduceLROnPlateau learning rate scheduler, reducing the learning

Table 4.3: Comparison of Detection Performance Across Different Annotation Strategies — Pre-Annotation, Active Learning, Roboflow Auto-Labeling, and our Automated Annotation.

Annotation Method	Frameworks	mAP50	mAP50-95	Computation (*GFLOPs*)	Dataset Prep. Time
Pre-Annotation	YOLO (No CBAM)	0.652	0.427	46.8	Manually Annotated (*days*)
	YOLO + CBAM	0.664	0.442	48.2	
	Edge-YOLO [59]	0.486	0.289	23.9	
	Cascade R-CNN [15]	0.519	0.306	32.1	
	DETR-DC5 [17]	0.557	0.337	89.4	
	DN-DETR [55]	0.579	0.358	151.3	
Active Learning (`Faster R-CNN`)	LT/C	0.432	0.264	37.7	N/A
	LS+C	0.417	0.236	34.3	
	CALD [110]	0.451	0.272	39.4	
	HUALE [77]	0.496	0.298	41.6	
Auto-Labeling (`Roboflow`)	YOLO + CBAM	0.591	0.379	48.7	382 secs
Auto-Annotation (`Proposed`)	YOLO + CBAM	**0.676**	**0.448**	44.9	**189** secs

rate by 0.2 after five consecutive epochs without validation loss improvement, ensuring efficient convergence. To validate the framework, we evaluated it across three dimensions: Detection Performance, XAI Algorithm Evaluation, and Computational Overhead.

Before any quantitative evaluations, Fig. 4.4 illustrates the high-quality, precise annotations generated by the proposed automated annotation pipeline. The process begins with raw, unannotated images containing possible object classes such as bicycle, motorcycle, and stop sign (4.4a, 4.4d, and 4.4g), which are fed to the trained detection model and the CRP explainer. The CRP-generated explanations identify the most relevant concepts influencing the model's detections, with higher pixel intensities representing more critical features (e.g., the "STOP" text on the stop sign (4.4h), the bicycle frame (4.4b), or the engine of the motorcycle (4.4e)). These key regions are localized using the automated annotation algorithm, as shown in the second column. The pipeline concludes by leveraging the CRP concept localized coordinates to generate precise bounding boxes, by transposing these coordinates unto the initial raw, unannotated images and assign the corresponding class labels as depicted in the last column (4.4c, 4.4f, and 4.4i). This seamless integration of model interpretability for automated annotation ensures high-quality labeling while significantly reducing reliance on manual effort. The method enhances object detection learning and accuracy, demonstrating its effectiveness for developing robust perception models for autonomous systems, particularly AVs. This pipeline supports reliable, real-time perception in diverse and complex real-world scenarios, making it a compelling solution for advanced CV applications.

4.4.1 Object Detection Performance Evaluation

This section evaluates the efficacy of our automated annotation framework by quantitatively comparing the performance of detection models trained on datasets prepared using diverse annotation methodologies, including pre-annotation, automated annotation, and active learning strategies. Our assessment framework incorporates both precision-based detection metrics and computational efficiency measures for dataset curation. We utilize state-of-the-art object detection architectures [15], [17], [37], [55], [59], evaluated primarily using the mean Average Precision (mAP) metric. The mAP is computed across a spectrum of Intersection over Union (IoU) thresholds, typically ranging from 50% to 95% in 5% increments. This metric is formally defined as:

$$mAP = \frac{1}{|T|} \sum_{t \in T} \frac{1}{|C|} \sum_{c \in C} AP_c^t \tag{4.18}$$

where T is the set of IoU thresholds, C is the set of object classes, and AP_c^t is the Average Precision for class c at IoU threshold t. The mAP metric provides a comprehensive quantitative measure of a model's capacity to accurately localize and classify object instances across multiple categories and scales within an image, serving as a robust indicator of overall detection performance.

In our study, we employed two key metrics: *mAP50* and *mAP50–95*. *mAP50* quantifies the model's detection accuracy when predicted bounding boxes overlap with ground truth by at least 50% IoU. *mAP50–95* extends this metric by averaging mAP values across multiple IoU thresholds ranging from 50% to 95% with varying degrees of localization precision.

Prior to our primary experiments, which involved enhancing the YOLO model with the CBAM and implementing our auto-annotation approach, a standard YOLO model was initially trained on a pre-annotated version of our custom traffic dataset to establish a baseline. This baseline model achieved a detection performance of 65.2% mAP50 and 42.7% mAP50-95. Focusing on the pre-annotation method, Table 4.3 illustrates that among the various detection models trained on the pre-annotated dataset, our CBAM-enhanced YOLO model significantly outperformed other architectures, achieving an mAP50 of 66.4%. In contrast, the CBAM-enhanced Cascade R-CNN[15] and Edge YOLO[59] models achieved mAP50 values of 49.9% and 48.6%, respectively. Similarly, transformer-based models, including the Detection Transformer with Dilated Convolutions (DETR-DC5)[17] and DeNoising Detection Transformer (DN-DETR)[55], achieved 55.7% and 57.9% mAP50, accordingly. This represented a minimum improvement of 8.5% for our detection pipeline. The superior performance of the CBAM-enhanced YOLO model were further highlighted in the mAP50-95 metric, where it achieved 44.2%, substantially surpassing both the CBAM-enhanced Edge YOLO (28.9%) and Cascade R-CNN (29.3%) models, likewise the DETR-DC5 (33.7%) and DN-DETR (35.8%) models.

We also conducted an extensive exploration of active learning techniques. Active learning is a sophisticated approach that strategically identifies the most informative samples for annotation, thereby minimizing the quantity of labeled data required to train an effective model. By employing diverse query strategies, active learning maximizes model performance with minimal annotated data. In our active learning detection framework, we implemented several query strategies: `Least Squares Plus Confidence (LS+C)`, which selects samples by considering both prediction error and confidence intervals; `Least Total Cost (LT/C)` and `Cost-Effective Active Learning with Diversity (CALD)` [110], both of which fo-

cus on diversity and uncertainty-based sampling while accounting for labeling costs. Furthermore, we incorporated the `Hierarchical Uncertainty Aggregation and Emphasis Loss (HUALE)` strategy by Nguyen et al. [77], which employs a two-stage approach. This method initially filters images based on entropy measures (retrieval phase) and subsequently ranks them using Semantic Affinity, Category Diversity, Overlap Ratio, and Localization Confidence (ranking phase) to ensure the selection of the most informative samples for labeling. As illustrated in Table 4.3, our experimental results demonstrated that after 28 training cycles, the HUALE strategy significantly outperformed the other three query strategies, achieving a 49.6% mAP50 and 29.8% mAP50–95, compared to CALD (45.1% & 27.2%), LT/C (43.2% & 26.4%), and LS+C (41.7% & 23.6%). The superior performance of the HUALE strategy is clearly depicted in Fig. 4.5, where it consistently surpasses other active learning strategies in detection performance when plotted against the quantity of labeled data or the percentage of available training data labeled.
We concluded our experimentation by validating the feasibility and effectiveness of our proposed automated annotation pipeline. We trained a CBAM-enhanced YOLO detection model on a dataset of over 15,000 samples that were auto-annotated using our method. The goal was to observe how our concept-level explainable annotation approach influenced the model's learning process, compared to the same model trained on a 15,000-sample auto-labeled dataset from Roboflow, which utilized their newly introduced Auto-label Grounding DINO feature [105]. After training, we compared the inference performance of the two models. As seen in Table 4.3, the model trained on our auto-annotated dataset achieved a mAP50 of 67.6% and a mAP50–95 of 44.8%. In contrast, the model trained on the Roboflow auto-labeled dataset achieved a mAP50 of 59.1% and a mAP50–95 of 37.9%.

4.4.2 XAI Algorithm Evaluation

In addition to evaluating detection performance, we also assessed the effectiveness of our chosen CRP explainer under various neural network modifications, particularly focusing on attention module enhancements and data augmentation techniques. This assessment aimed to understand how these strategies impact the interpretability and utility of the CRP explainer. Inspite the limited nature of literature for evaluating the performance of XAI methods, these techniques including the CRP algorithm, are generally evaluated using two model-agnostic quantitative metrics: Faithfulness and Complexity. These metrics help quantify how accurately the explanations reflect the model's decision-making process and how easily the explanations can be understood and used.

Faithfulness: This metric evaluates how accurately explanations represent the features a model uses during inference. It quantifies the reliability of explanations in reflecting the decision-making process of models. The assessment primarily employs two techniques: `Concept Flipping` and `Concept Insertion`. Inspired by the pixel flipping experiment, these techniques focus on latent concepts rather than input features. Here, the concept relevance in each layer, associated with an object prediction, is determined by spatially aggregating intermediate relevance scores, with convolutional channels treated as distinct concepts. In **Concept Flipping**, the most relevant channels are sequentially deactivated (their activations, set to zero), and the resulting changes in the model's output are analyzed to gauge the impact on the model's behavior. In contrast, **Concept Insertion** starts with all channels set to zero activation and progressively restores the most relevant concepts, observing the corresponding changes in the model's output. The most faithful explanations, always have higher values, while demonstrating a significant drop in performance during concept

Table 4.4: Comparison of CRP Variants (z^+, γ, ε), GRADCAM, and Gradient Maps on Faithfulness and Complexity after CBAM Enhancement for Pre-Annotated and Auto-Annotated Datasets.

XAI Algorithms	**Faithfulness** (↑)				**Complexity** (↓)			
	Concept Flipping (↓)		**Concept Insertion**(↑)		**Explanation Deviation** (%)		**Comprehension of** 80% **of Attr.**(%)	
	Pre_{Anno}	$Auto_{\text{Anno}}$	Pre_{Anno}	$Auto_{\text{Anno}}$	Pre_{Anno}	$Auto_{\text{Anno}}$	Pre_{Anno}	$Auto_{\text{Anno}}$
CRP - z^+	1.19	1.54	1.75	2.23	**0.44**	**0.44**	50.2	51.3
CRP - γ	1.34	1.76	1.87	2.40	0.68	0.63	40.1	40.7
CRP - ε	**1.90**	**2.88**	**2.43**	**3.35**	1.05	0.89	**37.5**	**35.8**
GRADCAM	1.62	2.43	2.23	2.94	1.14	0.95	37.7	38.0
Gradient Maps	1.63	2.44	2.23	2.93	1.09	0.92	42.9	40.9

flipping and a marked improvement during concept insertion, indicating a high degree of alignment between the explanation and the model's internal processes.

Complexity: Gauges the effort required for stakeholders to understand and comprehend the explanation provided by an XAI method. While faithfulness is essential, the non-linear decision boundaries of deep learning models necessitate presenting explanations in a way that is accessible to non-experts, thereby enhancing the explainer's usability, especially in critical domains like AVs. Complexity encompasses two key dimensions: `Explanation Deviation` and `Comprehension of 80% of Attributions`. **Explanation Deviation** quantifies the variability and consistency of explanations generated by an XAI method across different predictions within the same class. This reflects the robustness and versatility of the method. A lower deviation indicates that the explanations are consistent and precise within a class, thereby reducing complexity. This deviation is calculated using the standard deviation of latent concept attributions per class. The final deviation value for concept relevance scores $R_j(\mathbf{X}_i)$ for a class t, along with the mean attribution $\bar{R}_j$ across m_s class samples and m_c concepts, is determined as outlined in Equations 4.20 and 4.19 respectively. Moreover, the second facet – **Comprehension of 80% of Attributions** measures the number of concepts required to be analyzed to understand 80% of all attributions. Fewer relevant concepts suggest a more concise explanation, which is essential for user understanding and trustworthiness of the XAI method.

$$\bar{R}_j = \frac{1}{m_s}\sum_{i}^{m_s} R_j(\mathbf{X}_i) \tag{4.19}$$

$$\sigma_f = \frac{1}{m_t}\sum_{t}^{m_t}\left(\frac{1}{m_c}\sum_{j}^{m_c}\sqrt{\frac{1}{m_s-1}\sum_{i}^{m_s}\left(R_j(\mathbf{X}_i)-\bar{R}_j\right)^2}\right) \tag{4.20}$$

To holistically evaluate the impact of different relevance backpropagation rules on the faithfulness and complexity of CRP-generated explanations, we employed three specific rules for 100 sampled predictions: CRP–ε (`Epsilon Rule`), CRP–γ (`Gamma Rule`), and CRP–z^+ (`z-Plus Rule`). The `Epsilon Rule` addresses numerical instabilities during relevance propagation by incorporating a small (ε) value, thereby preventing the amplification of small activations, especially in deeper layers where such activations may ap-

proach zero. The `Gamma Rule` enhances positive relevance scores while suppressing negative ones, making it particularly useful for highlighting concepts that positively influence the model's decisions. The `z-Plus Rule`, on the other hand, focuses solely on the positive contributions of neurons, ignoring negative activations, which is beneficial when positive evidence is more crucial to the decision outcome.

Initially, we evaluated the impact of CBAM enhancements on the performance of various explainers by comparing explainer performances for a standard detection model trained on a pre-annotated custom dataset with a CBAM-enhanced detection model trained on our curated auto-annotated dataset. The analysis included the CRP explainer, Gradient-weighted Class Activation Mapping (GRADCAM), and gradient-based methods (Gradient Maps). As shown in Table 4.4, CRP–ε consistently outperformed other explainers on the standard detection model, achieving the highest faithfulness scores with flipping and insertion values of 1.90 and 2.43, respectively, surpassing Gradient (1.63 & 2.23) and GRADCAM (1.62 & 2.23). Fig.4.6 further demonstrates CRP–ε's superior performance, consistently excelling in both metrics. On the CBAM-enhanced model trained on the auto-annotated dataset, CRP–ε again led with improved flipping and insertion scores of 2.88 and 3.35, outperforming Gradient (2.44 & 2.93) and GRADCAM (2.43 & 2.94), as depicted in Fig.4.7. CRP variants also demonstrated superior complexity performance. For the standard detection model, CRP–z^+ recorded the lowest deviation (0.44), while CRP–ε achieved the best comprehension score (37.5%), outperforming GRADCAM (1.14 & 37.7%) and Gradient (1.09 & 42.9%). Similarly, for the CBAM-enhanced model, CRP retained its advantage with deviation and comprehension scores of 0.44 (z^+) and 35.8% (ε), exceeding Gradient (0.92 & 40.9%) and GRADCAM (0.95 & 38.0%), as depicted in Figures 4.8 and 4.9. CRP–ε demonstrates exceptional stability across most faithfulness and complexity evaluations but is less effective at minimizing explanation deviation. In contrast, CRP–γ and CRP–z^+ excel in reducing variability, delivering globally consistent and coherent explanations by filtering out irrelevant and noisy attributions. These findings highlight the complementary strengths of the propagation rules: CRP–ε balances faithfulness and complexity, while CRP–γ and CRP–z^+ prioritize robustness and global feature representation. The adaptability of these CRP variants allows for tailored applications depending on specific evaluation priorities. Furthermore, CBAM enhancements, coupled with training on the auto-annotated dataset, significantly improved the performance of all XAI explainers, enhancing faithfulness and reducing complexity, as evidenced by Table 4.4. These advancements underscore the effectiveness of CBAM in optimizing explainability and computational efficiency.

Also, we analyzed the influence of data augmentation on the explainer performance by incrementally increasing the dataset size in four stages (25%, 50%, 75%, and 100%) for training the CBAM-enhanced detection model. This approach provided insights into the effects of dataset quantity on XAI explainers' faithfulness and complexity. The CRP–ε variant, selected for its superior earlier performance, was benchmarked against GRADCAM and Gradient methods. Results, as summarized in Table 4.5 and illustrated in Fig. 4.10a, revealed consistent improvements in faithfulness scores across all explainers with increasing dataset size. CRP–ε exhibited significant enhancements, with concept flipping scores rising from 0.67 (25%) to 2.84 (100%) and concept insertion scores improving from 1.03 to 3.32. These trends highlight better generalization, reduced overfitting, and enhanced reliability in capturing model behavior with more data. At 100% dataset usage, CRP–ε achieved the highest faithfulness scores, outperforming GRADCAM (2.43 & 2.90) and Gradient (2.44 & 2.91). In terms of complexity, CRP–ε maintained the lowest explanation deviation (0.45) and comprehension scores (0.35) at 100% dataset usage, outperforming GRADCAM (0.96

Table 4.5: Comparison of CRP, GRADCAM, and Gradient Maps on Faithfulness and Complexity Across Dataset Proportions.

XAI Algorithm	Faithfulness (↑)								Complexity (↓)							
	AUC Concept Flipping (↓)				AUC Concept Insertion (↑)				Explanation Dev.(%)				80% Concept Comprehen.(%)			
	25%	50%	75%	100%	25%	50%	75%	100%	25%	50%	75%	100%	25%	50%	75%	100%
CRP	0.67	1.58	2.41	2.84	1.03	1.75	2.50	3.32	1.19	0.88	0.63	0.45	0.72	0.49	0.39	0.35
GRADCAM	0.62	1.22	1.94	2.43	0.88	1.59	2.43	2.90	1.93	1.62	1.37	0.96	0.85	0.54	0.40	0.37
Gradient Maps	0.49	1.04	1.92	2.44	0.71	1.34	2.36	2.91	1.85	1.61	1.34	0.91	0.88	0.64	0.46	0.40

& 0.37) and Gradient (0.91 & 0.40). This reduced complexity suggests consistent and concise explanations, crucial for usability. Visualizations in Fig. 4.10b further corroborate CRP–ε's superior performance across dataset proportions. Our results highlight the critical role of dataset comprehensiveness and data augmentation in enhancing model explainability. The CRP explainer demonstrated remarkable improvements in faithfulness while maintaining lower complexity scores compared to GRADCAM and Gradient. This symbiotic relationship between dataset size and explainer performance underscores the importance of diverse, well-prepared datasets in improving the transparency and trustworthiness of AI models, particularly in high-stakes domains like autonomous vehicle perception systems.

4.4.3 Computational Overhead

Our final analysis examined the computational overhead of our proposed automated annotation framework, including resource usage and dataset preparation time. This evaluation was compared with other contemporary approaches in the literature aimed at reducing the annotation burden in detection model development. Additionally, we discussed the latency of the CRP explainer within our framework against other explainers on the same prepared auto-annotated dataset to determine which XAI algorithm is best suited for real-time object detection in the AV domain.

For computational evaluation, we measured the framework's execution performance in Giga Floating Point Operations per second (GFLOPs) and dataset curation time in seconds (secs). As summarized in Table 4.3, our proposed automated annotation framework computationally performed optimal, achieving 44.9 GFLOPs, close to the Roboflow-based detection model 48.7 GFLOPs and the model trained on a pre-annotated version of our custom traffic dataset 48.2 GFLOPs. This competitive performance aligning with our expectations, since the inclusion of attention mechanisms, contributes to selective computation, reduced input dimensionality, improved learning efficiency, and parallelization opportunities. While these mechanisms add computational cost, the trade-off is justified by the improved mAP performance. Moreover, for our 15,000-sample custom traffic dataset, our framework auto-annotated and curated the data in just 189 seconds, compared to 382 seconds with Roboflow's auto-labeling feature. Notably, our automated annotation approach reduced the manual annotation time from several days to just under three minutes, a reduction

of over 98%. This demonstrates the efficiency of our explainable automated annotation framework, outperforming other methods like active learning, which requires multiple annotation and model training cycles, often taking days, if not hours. While FLOPs measure a model's computational complexity, models (like YOLO + CBAM) maintain consistent theoretical FLOPs across identical architectures. However, runtime variations arise from factors like memory bandwidth constraints, background processes, dynamic computational graphs in PyTorch or TensorFlow, sparse activations during early training, and backend optimizations such as kernel fusion and mixed-precision training, illustrating the gap between theoretical and real-world performance expectations.

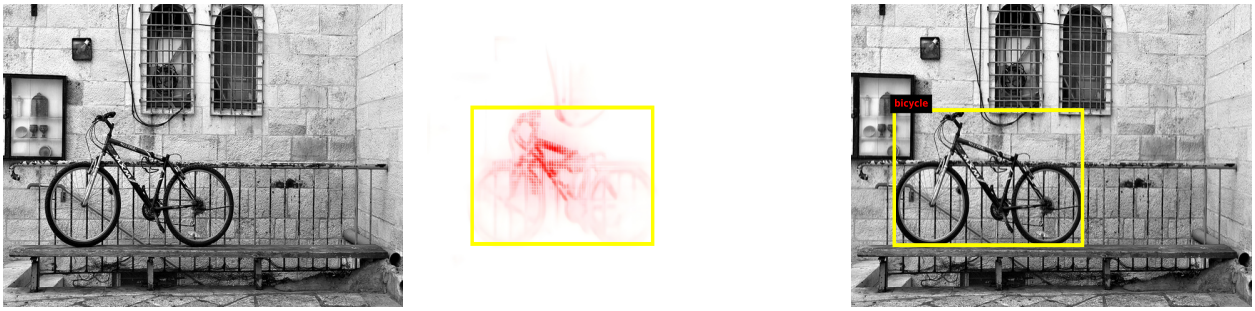
Lastly, in terms of explainer latency, the chosen CRP algorithm exhibited the best performance with an execution time of 24.7 seconds, compared to LRP 54.9 seconds, GRADCAM over 410 seconds and activation-based methods over 530 seconds. This efficiency, combined with its computational effectiveness and improved performance, positions our framework as a viable solution for large-scale dataset annotation in detection model development. It also shows potential for real-time detection applications such as AV perception, where balancing computational cost and performance is important.

4.5 Conclusion

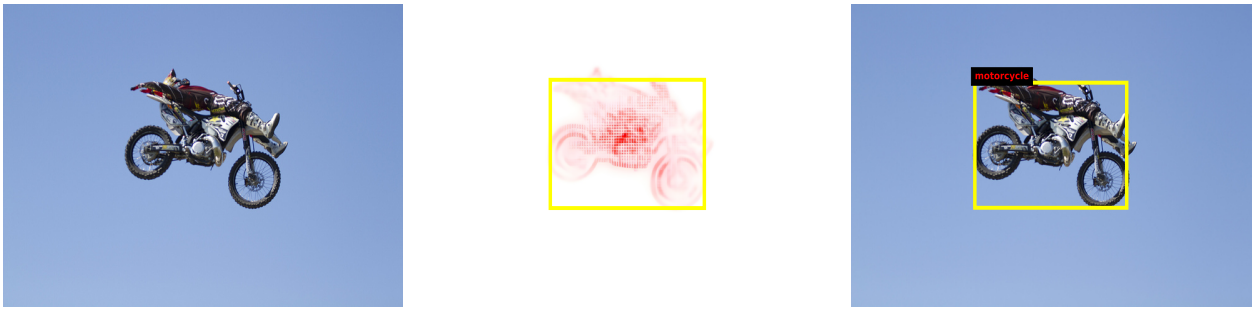
Our study presents a transformative framework that integrates semi-supervised learning with the CRP XAI algorithm, redefining model interpretability and large-scale dataset annotation for autonomous system perception. By leveraging CRP's concept-level relevance mapping, we automate annotation processes with minimal manual effort, achieving over 98% labeling time reduction while generating high-quality labeled datasets. Incorporating advanced network optimization techniques, such as the CBAM, alongside targeted data augmentation strategies, the framework enhances detection accuracy, reduces computational overhead, and improves explanation fidelity. Notably, CRP surpasses other XAI methods like GRADCAM in providing faithful, actionable insights with low latency, making it ideal for resource-constrained, real-time autonomous applications like AV perception. Future research could enhance this pipeline by incorporating adaptive learning for dynamic driving environments, integrating multi-modal sensor fusion (e.g., LiDAR, RADAR, and camera data) to improve perception robustness, and enabling real-time explainability for on-the-fly decision-making to enhance safety and security. Similarly, expanding its application to UAV navigation, industrial automation, and other high-stakes domains could position this framework as a benchmark solution for scalable, interpretable, and efficient AI-driven perception tasks. Such advancements would not only enhance system reliability but also address broader challenges in ensuring transparency, adaptability, and operational safety across complex, real-world environments.

Additional Information

Supplementary Material: The complete experimental code for this study, including all implementation details, is publicly accessible at `https://github.com/Iyke1z/AutoAnnotation`.



(a) Original Bicycle Image (b) CRP Explanation Localization (c) Automated Annotation Result



(d) Original Motorcycle Image (e) CRP Explanation Localization (f) Automated Annotation Result

(g) Original Stop Image (h) CRP Explanation Localization (i) Automated Annotation Result

Figure 4.4: **Results from the Proposed Automated Annotation Pipeline.**. The first column presents raw, unannotated images of classes bicycle, motorcycle, and stop sign. The second column illustrates CRP-generated explanations, pinpointing relevant concepts essential for each class detection. The final column demonstrates the culmination of the automated annotation process, where CRP concept localization coordinates from the previous column are transposed onto the original images to produce precise high-quality bounding box annotations and object labels.

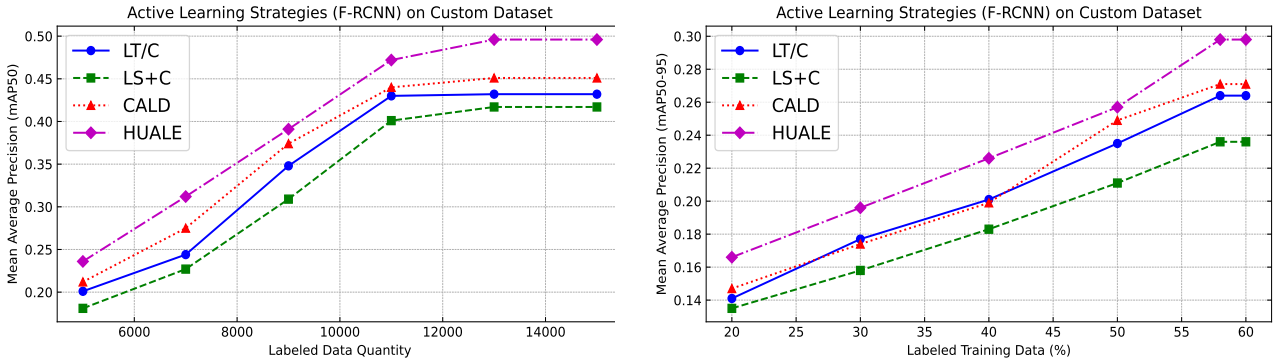


Figure 4.5: **Detection Performance for Various Active Learning Strategies.** The plots compare the detection performance of active learning strategies (LT/C, LS+C, CALD, HUALE) for the Faster R-CNN model. HUALE demonstrates superior performance across both mAP50 and mAP50–95 metrics, achieving the highest scores as labeled data quantity (training data percentage) increase, showcasing its effectiveness in active learning for object detection tasks.

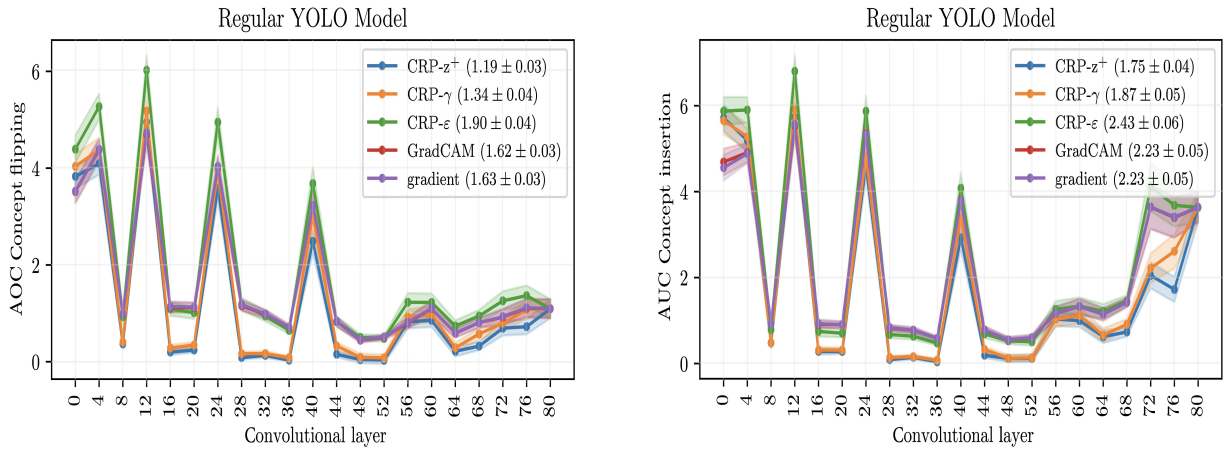


Figure 4.6: **Comparison of XAI Algorithm Faithfulness for Concept Attribution on a Regular YOLO Model.** The plots compare the faithfulness of CRP variants (z^+, γ, ε), GRADCAM, and Gradient Maps across convolutional layers. CRP–ε (green) consistently leads, achieving the highest Concept Flipping and Insertion scores, demonstrating superior performance across key layers.

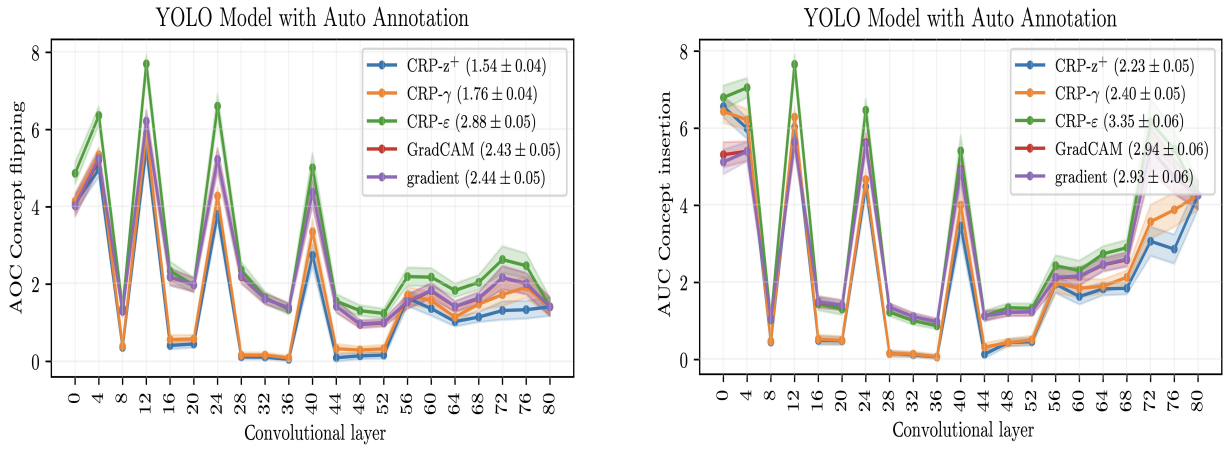


Figure 4.7: **Comparison of XAI Algorithm Faithfulness for Concept Attribution on a CBAM-Enhanced YOLO Model.** The plots compare the faithfulness of CRP variants (z^+, γ, ε), GRADCAM, and Gradient Maps across convolutional layers. With an observed significant improvements in faithfulness across all explainers. CRP–ε (green) consistently outperforms other explainers, achieving the highest Concept Flipping and Insertion scores, showcasing its superior performance across key layers.

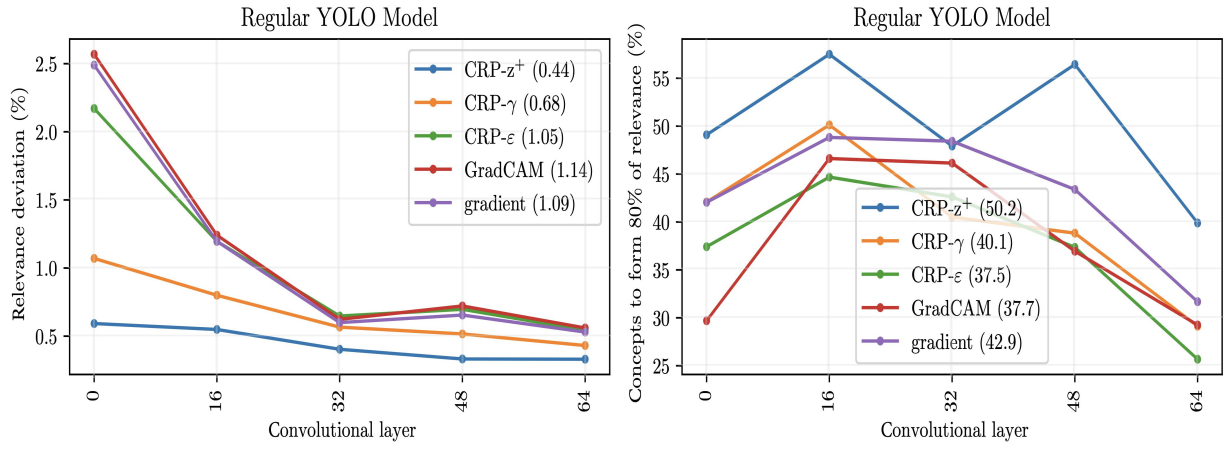


Figure 4.8: **Comparison of XAI Algorithm Complexity for Concept Attribution on a Regular YOLO Model.** The plots compare the complexity of CRP variants (z^+, γ, ε), GRADCAM, and Gradient Maps across convolutional layers. CRP–z^+ achieves the lowest Relevance Deviation, while CRP–ε excels in 80% Attribution Comprehension, outperforming GRADCAM and Gradient Maps, particularly in deeper layers. These results highlight the superior efficiency and complexity management of CRP variants in model explanations.

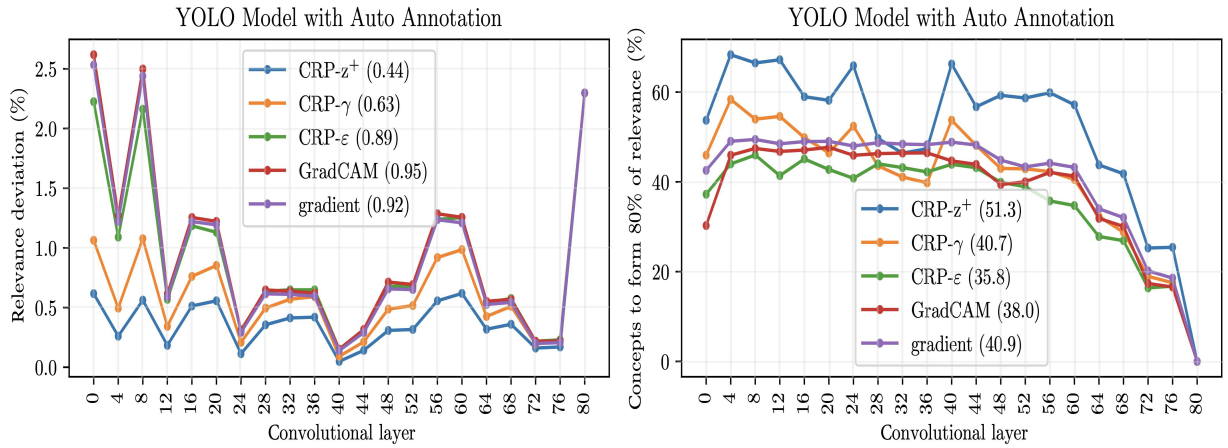
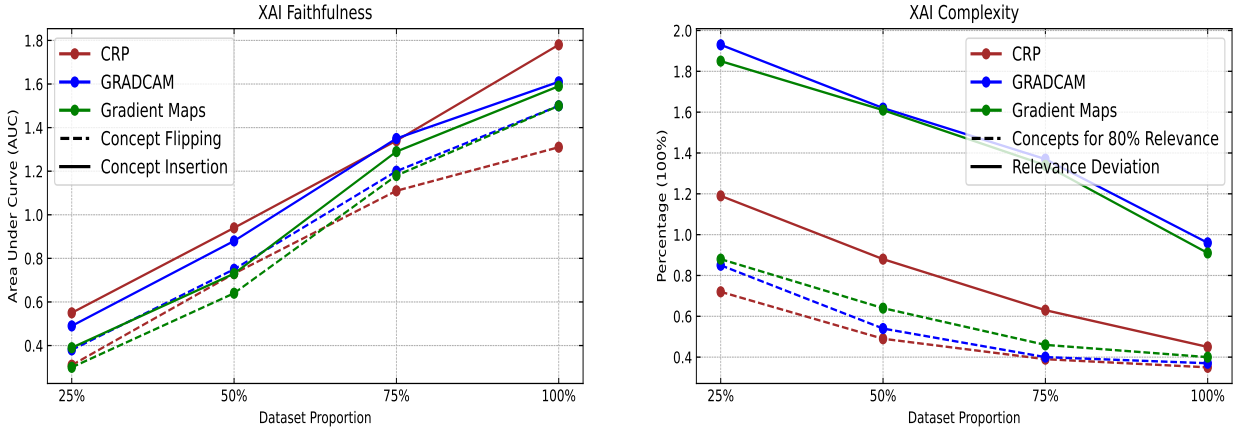


Figure 4.9: **Comparison of XAI Algorithm Complexity for Concept Attribution on a CBAM-Enhanced YOLO Model.** The plots compare the complexity of CRP variants (z^+, γ, ε), GRADCAM, and Gradient Maps across convolutional layers. CRP–z^+ shows the lowest Relevance Deviation, while CRP–ε achieves the highest Comprehension of 80% of Attributions, consistently outperforming GRADCAM and Gradient Maps. These results demonstrate the reduced complexity and enhanced performance of all explainers with CBAM and auto-annotation.



(a) Comparison of XAI Faithfulness across different dataset proportions

(b) Comparison of XAI Complexity across different dataset proportions

Figure 4.10: **Comparison of XAI Algorithms on Faithfulness and Complexity with Dataset Augmentation.** The plots illustrate the impact of increasing dataset sizes on CRP (brown), GRADCAM (blue), and Gradient Maps (green). The left plot show CRP leading in Faithfulness, with the largest gap between Concept Flipping and Insertion, while also maintaining the lowest Complexity in the right plot, outperforming GRADCAM and Gradient Maps in generating concise and efficient explanations.

Chapter 5: eXplainable AI For Enhanced Trojan Detection In Autonomous Vehicle Steering Networks

5.1 Introduction

Artificial Intelligence (AI) now underpins critical infrastructure in autonomous transportation, medical diagnostics, and cybersecurity. AVs exemplify this trend, using sophisticated DNNs to fuse inputs from LiDAR, RADAR, vision, and inertial sensors for real-time steering control. While this data-driven autonomy improves adaptability in dynamic traffic scenes, it also broadens the system's attack surface. Among these threats, trojan backdoor attacks, stealthy malicious manipulations embedded during training, which can covertly hijack model behavior, forcing dangerous trajectory deviations. Exacerbating this risk is the opaque nature of DNN systems, where non-intuitive latent representations obscure effective analysis and regulatory auditing, challenging the safe deployment of AV technologies.

Despite notable progress in trojan detection research, the field remains fragmented and anchored in classification settings. Methods like Neural Cleanse, STRong Intentional Perturbation (STRIP), and Activation Clustering detect discrete anomalies like entropy shifts, class output changes, or clustered neuron activations. Although effective in categorical contexts, these methods struggle in continuous regression-output tasks such as AV steering, where backdoor triggers typically induce nuanced, context-aware prediction deviations rather than abrupt output flips. Additionally, their reliance on clean baseline datasets and high computational overhead limits their scalability and real-time applicability. Vitally, they tend to overlook semantic distortions and concept-level anomalies that signal manipulated model decision logic. In response to this gap, Recent efforts like Februus [27], that uses Grad-CAM to purify trojaned inputs, and Critical Path-Based Backdoor Detection (CPBD) [49], which maps anomalous decision paths via neuron activations, have pushed XAI from passive interpretability toward proactive defense mechanism in AI security. However, these solutions still largely cater to classification tasks and often lack resilience against distributed and imperceptible attacks.

in this chapter, we introduce an explainability-guided detection framework designed for regression-based AV steering control systems, addressing security gaps in existing defenses and bridging their inherent incompatibility with classification-oriented trojan detectors. Our approach repurposes Grad-CAM [85] and Concept Relevance Propagation (CRP) [5] as active security tools, generating multi-level spatial and conceptual attribution maps that expose the rationale behind steering decisions. By analyzing explanations from benign and trojaned samples (where steering prediction deviations exceed acceptable thresholds) across

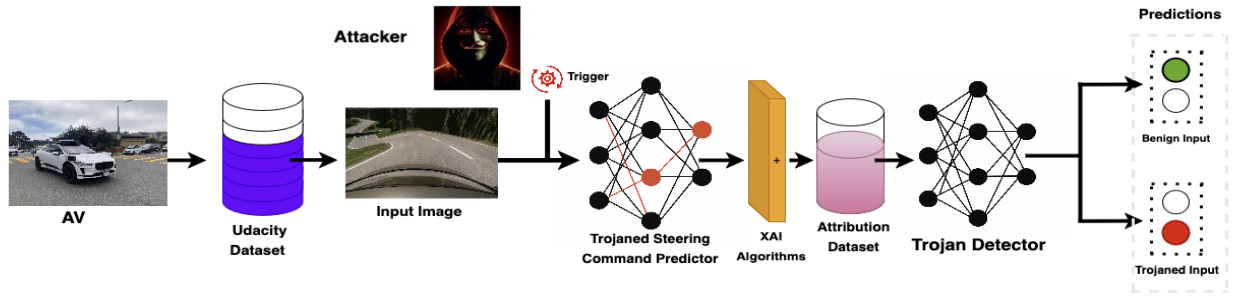


Figure 5.1: **Explainability-Guided Trojan Detection Framework.** The framework consists of two main components: (1) A Trojaned Steering Angle Predictor, and (2) A Real-Time Detection Module, which uses Grad-CAM, and CRP-generated attribution maps to train a lightweight classifier that detects trojaned inputs based on explanation-level anomalies.

varying poisoning rates, we reveal telltale indicators of backdoor compromise like saliency drift, spatial deformation, and conceptual divergence. These explanation-derived features then empower lightweight binary classifiers that detect trojaned behavior with high fidelity, without requiring prior knowledge of trigger patterns or access to clean reference datasets. The contributions of this work are:

- Developed a robust end-to-end DNN for steering angle prediction using the Udacity Self-Driving Car dataset to establish a behavioral baseline for assessing the impact of trojan backdoor attacks on AV control. Static visible triggers and imperceptible perturbations were embedded during training at varying poisoning rates (5%–40%), to determine the effective poisoning threshold, balancing high fidelity on clean inputs with consistent, malicious deviations under trigger activation.

- Leveraging novel XAI techniques including Grad-CAM [85] and CRP [5], we generate and curate multi-level visual attribution maps capturing decision-making patterns. Consequently, these were used to train lightweight binary classifiers capable of trojan detection based solely on explanation-derived features.

- Validated the proposed XAI-guided detection framework by benchmarking it against conventional methods including Activation Subset Scanning (ACTSS) [108] and Artificial Brain Stimulation (ABS) [62], all trained on the same curated attribution dataset. Evaluation using precision, F1-score, and AUC-ROC underscore the efficacy of our approach in bolstering AV model resilience and transparency, while establishing a novel foundation for adapting classification-based detection logic for continuous-output regression tasks in safety-critical settings.

5.2 System Models

5.2.1 Network Model

The proposed framework targets the control layer of AVs' cyber-physical architecture, where DNNs perform end-to-end regression, mapping raw monocular RGB inputs $x \in \mathbb{R}^{H\times W\times 3}$ to continuous steering commands $y = f(x) \in \mathbb{R}$. This design, used in systems like NVIDIA PilotNet and the Udacity self-driving simulator,

enables low-latency, real-time control at 10–30Hz on edge devices (e.g. NVIDIA Jetson AGX Xavier). However, it broadens the attack surface, especially when model weights or training data come from untrusted sources and lack formal safety guarantees, leaving it vulnerable to trojan backdoor attacks triggered by specific inputs.

5.2.2 Threat Model

In this work, we consider a ***white-box*** adversary with full or partial knowledge of the AV control model's architecture, parameters, and training data, a realistic risk scenario where models or datasets are open-sourced repositories or third-party vendors. The adversary aims to implant a trojan backdoor during training, so the model behaves nominally on clean inputs $x \sim \mathscr{D}_{clean}$, but produces malicious steering signal $y' = f(x+\delta) \in \mathscr{T}$ when a trigger $\delta \in \mathbb{R}^{H\times W\times C}$ is present. Here, $\mathscr{T} \subset \mathbb{R}$ denotes attacker-defined, unsafe steering angles. The attack vector involves ***data poisoning***, wherein a subset of training samples is perturbed:

$$\mathscr{D}'_{train} = \mathscr{D}_{clean} \cup \{(x_i+\delta, y_t)\}_{i=1}^{k} \tag{5.1}$$

binding the trigger δ to an adversarial target output y_t. Triggers may be ***static visible implantable patterns*** (e.g. geometric stickers, pixel patches) or ***imperceptible perturbations*** crafted using L2-Norm bounded methods, where the perturbation δ satisfies $\|\delta\|_2 \leq \varepsilon$ (bound), to embed backdoor logic while preserving visual stealth. Introduced through physical means (e.g. decals, signs) or digital overlays in simulations, the attack's hallmark is in its stealth. Crucially, we assume the attacker lacks post-deployment access, relying solely on input-trigger activation. If undetected, this can lead to outputs $\hat{y} = f(x+\delta) \notin \mathscr{Y}_{valid}$ (where $\mathscr{Y}_{valid}$ denotes safe steering ranges), resulting in lane departures, erratic trajectories, or hazardous maneuvers, especially in urban environments. These attacks often exploit internal model logic to align benign-looking inputs with malicious outputs, evading traditional anomaly detectors.

This section outlines our three-stage explainability-guided framework for trojan backdoor detection in AV control models, as illustrated in Fig. 5.1. First, we train a baseline DNN for steering angle prediction on the clean Udacity dataset (Section 5.2.3). Next, we simulate trojan attacks by poisoning training data with two different trigger classes at varying rates (5%–40%), to evaluate stealthy manipulation impact (Section 5.2.4). Finally, we apply Grad-CAM and CRP to generate attribution heatmaps, that serve as input features for training lightweight binary classifiers to detect trojaned inputs (Section 5.2.5). Detailed methodologies are provided in the following subsections.

5.2.3 Steering Command Predictor

Steering angle prediction is fundamental to AV control, linking vehicle perception to how it steers for real-time guidance in lane keeping, curve negotiation, and reactive navigation in dynamic traffic. This task is framed as a regression problem $f : \mathbb{R}^{H\times W\times 3} \to \mathbb{R}$, where $x \in \mathbb{R}^{H\times W\times 3}$ is an RGB image and $y \in \mathbb{R}$ is the predicted steering angle, constrained within an operational range $y \in [-\theta_{\max}, \theta_{\max}]$ (e.g. ± 1 radian) to respect physical steering limits. We adopt Mobile Neural Network Version 3 (**MobileNetV3-Large**) [43] as the backbone for our steering angle regression given its edge-optimized design for platforms like the NVIDIA Jetson AGX Xavier. We modify the MobileNet by replacing the classification head with a single-

output regression layer, so that:

$$\hat{y} = f(x;\theta) = g(\phi(x;\theta_{\text{feat}});\theta_{\text{reg}}) \tag{5.2}$$

where $\phi(x;\theta_{\text{feat}})$ is the feature extractor, producing an embedding $z \in \mathbb{R}^d$, which is then passed to a regression head $g(z;\theta_{\text{reg}})$ with $\theta = \{\theta_{\text{feat}}, \theta_{\text{reg}}\}$ being the model parameters, to yield the predicted steering angle. The model is optimized using Root Mean Squared Error (RMSE) loss to penalize large deviations. Additionally, MobileNet offers key advantages for our explainability-guided trojan detection framework: (1) its structured convolutions enhance spatial coherence for Grad-CAM interpretability; (2) its SE blocks align with CRP's channel-level relevance tracing; and (3) its lightweight design enables rapid retraining for poisoned variant testing. While limited global context makes MobileNet more susceptible to localized triggers (e.g. pixel-space patches), post hoc XAI mitigates this by exposing latent semantic anomalies.

5.2.4 Trojan Attacks

Trojan or backdoor attacks embed malicious logic during training that activates only when specific triggers appear, evading standard validation. These attacks pose severe risks in AV control where single misprediction can result in catastrophic outcomes. Our study simulates two trigger types to assess vulnerability and detection viability through post hoc explainability.

Visible Triggers: These are static square patch triggers $\delta \in \mathbb{R}^{H\times W\times 3}$, placed at fixed locations (e.g. upper-left image corner) to minimally disrupt semantics while hijacking model outputs.

Invisible (L2-Norm Bounded) Triggers: These imperceptible perturbations embed backdoor behavior while preserving visual fidelity. The perturbation δ satisfies the constraint $\|\delta\|_2 \leq \varepsilon$, where ε governs the perturbation budget, ensuring stealth. Injected during training, L2-Norm backdoors remain dormant until triggered, subtly altering internal representations and inducing gradient misalignment:

$$\nabla_\theta \mathscr{L}(f_\theta(x+\delta), y_t) \not\parallel \nabla_\theta \mathscr{L}(f_\theta(x), y) \tag{5.3}$$

causing the model to favor attacker-defined outputs over clean semantics. As shown in Fig. 5.2, each poisoned sample $x' = x+\delta$ is assigned a target steering angle within range $y_t \in \{-0.785, 0.0175, 0.785\}$ radians, corresponding to left, center, and right turns, balanced to avoid dataset bias or distribution skew. To explore the trade-off between stealth and attack success rate (ASR), we varied poisoning rates $q \in \{5\%, 10\%, \ldots, 40\%\}$ and trigger intensities (between 30%–100% patch brightness). Stronger triggers boosted ASR but reduced stealth. To identify the optimal poisoning threshold, we adopt a budget-constrained poisoning formulation. Given a clean dataset $\mathscr{D}_{\text{clean}}$, we construct a poisoned subset $\mathscr{D}'_t = \{(x_i+\delta, y_t)\}$, constrained by:

$$|\mathscr{D}'_t| \leq \beta |\mathscr{D}_{\text{clean}}| \tag{5.4}$$

where β is the poisoning budget. The steering model was trained on $\mathscr{D}_{\text{trojan}} = \mathscr{D}_{\text{clean}} \cup \mathscr{D}'_t$, to minimize loss over clean and poisoned data:

$$\min_\theta \sum_{(x_i,y_i)\in\mathscr{D}_{\text{clean}}} \mathscr{L}(f(x_i), y_i) + \sum_{(x'_i,y_t)\in\mathscr{D}'_t} \mathscr{L}(f(x'_i), y_t) \tag{5.5}$$

where $\mathscr{L}$ is the RMSE loss. By tuning β, we identified minimal poisoning ratios that achieved higher ASR while preserving clean input performance. This dual-behavior training yields a high-fidelity testbed for evaluating stealthy AV control failures.

(a) Clean $-5.72°$ (b) Visible $-17.19°$ (c) Invisible $13.18°$

Figure 5.2: Trojaned scenes at various steering angles.

5.2.5 Explainability-Guided Detection

Our detection strategy is built on the idea that trojaned models exhibit subtle yet consistent distortions in their internal attribution patterns, imperceptible in raw outputs but discernible through post hoc explainability. Unlike traditional detectors that rely on input perturbations or activation statistics, we treat semantic attribution maps as behavioral fingerprints, enabling trigger-agnostic, reference-free detection in regression-based AV control systems. The detection pipeline unfolds in structured stages, as outlined in Algorithm 2. We first train a baseline steering model $f(x;\theta)$ on a clean data $\mathscr{D}_{\text{clean}} = \{(x_i, y_i)\}$, where $x_i \in \mathbb{R}^{H\times W\times 3}$ is a front-facing RGB image and $y_i \in \mathbb{R}$ is the steering angle, minimizing RMSE loss. Next, we simulate backdoor behavior by poisoning a fraction $q\%$ of inputs using a trigger generator T, assigning a malicious target y_t. The resulting poisoned set $\mathscr{D}'_t = \{(T(x_i), y_t)\}$ forms $\mathscr{D}'_{trojan} = \mathscr{D}_{clean} \cup \mathscr{D}'_t$ for retraining, embedding the trojan while preserving clean performance. Post-training, we extract attribution maps using **Grad-CAM**, which highlights spatially salient input regions linked to outputs, and **CRP**, which attributes decisions to high-level semantic concepts [5]. For each input x, an XAI method Ξ produces an attribution map $A_x = \Xi(f,x) \in \mathbb{R}^{H\times W}$, compiled into an attribution dataset $\mathscr{D}_{\text{XAI}} = \{(A_x, z_x)\}$, where $z_x = 1$ for trojaned and $z_x = 0$ for clean inputs. This reframes detection as binary classification over these attributions, training a MobileNet detector $g(A;\phi)$ (initialized from ImageNet) using Binary Cross-Entropy (BCE) loss. At runtime, the deployed model $f(x_t;\theta)$ produces A_t; the detector yields $\hat{z}_t = g(A_t;\phi) \in [0,1]$, with:

$$z = \begin{cases} 1 & \text{if } \hat{z}_t \geq \tau = 0.6 \quad \text{(Trojaned)} \\ 0 & \text{otherwise} \quad \text{(Benign)} \end{cases}$$

This framework ensures real-time, trigger-agnostic trojan detection without prior attack signatures, transforming explainability into a functional defense that enhances the interpretability and resilience of AV control systems.

Algorithm 2: Explainability-Guided Trojan Detection

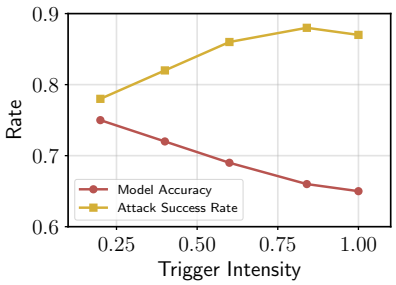
Input: Clean dataset $\mathscr{D}_{\text{clean}} = \{(x_i, y_i)\}$, Trigger generator T, Target label y_t, Poisoning rate q, Attribution method $\Xi(f,x)$, Detection model $g(A;\phi)$, Steering model $f(x;\theta)$, Threshold τ
Output: Detection label $z \in \{0,1\}$
function `Train_Steering_Model(`$\mathscr{D}_{clean}$`)`
    Optimize: $\theta^* \leftarrow \arg\min_\theta \mathscr{L}_{\text{RMSE}}(f(x;\theta), y)$
    return Trained model $f(x;\theta^*)$
function `Inject_Trojans(`$\mathscr{D}_{clean}, T, y_t, q$`)`
    $\mathscr{D}'_t \leftarrow \{(T(x), y_t) \mid (x,y) \in \mathscr{D}_{\text{clean}}, \text{ sampled at rate } q\}$
    Merge: $\mathscr{D}_{\text{trojan}} = \mathscr{D}_{\text{clean}} \cup \mathscr{D}'_t$
    return $\mathscr{D}_{\text{trojan}}$
function `Generate_Attributions(`$\mathscr{D}_{trojan}, f, \Xi$`)`
    for *each* $x \in \mathscr{D}_{trojan}$ **do**
        Compute attribution: $A_x = \Xi(f,x)$
        Assign label $z = 1$ if $x \in \mathscr{D}'_t$, else $z = 0$
    return Attribution dataset $\mathscr{D}_{\text{XAI}} = \{(A_x, z_x)\}$
function `Train_Detection_Model(`$\mathscr{D}_{XAI}$`)`
    Fine-tune detector $g(A;\phi)$ using $\mathscr{L}_{\text{BCE}}$ on $\mathscr{D}_{\text{XAI}}$
    return Trained detector $g(A;\phi)$
function `Detect_Trojan(`x_t, f, Ξ, g, τ`)`
    Compute attribution: $A_t = \Xi(f, x_t)$
    Predict: $\hat{z}_t = g(A_t;\phi)$
    if $\hat{z}_t \geq \tau$ **then**
        return $z = 1$ // Trojaned
    else
        return $z = 0$ // Benign

5.3 Simulation Results and Discussion

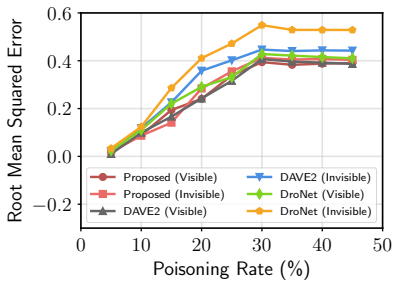
The baseline steering model, trained on the Udacity dataset (33,808 RGB frames, 80/10/10 split), achieved a high-fidelity benchmark after 30 epochs with RMSE loss. To simulate attacks, 5%–40% of training data was poisoned using two attack techniques, to redirect predictions to attacker-defined targets $y_t \in \{-0.785, 0.0175, 0.785\}$ radians. The trojaned model was retrained using the Nadam optimizer (lr = 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-8}$, decay 0.004, batch size 32) to ensure stable convergence. Grad-CAM and CRP attributions were then used to build labeled datasets for training another lightweight MobileNet detector (ImageNet-initialized, 40 epochs, BCE loss). Concretely, the framework was validated across two dimensions: Trojan Attack Impact and XAI-Guided Detection Performance.

Table 5.1: Impact of Trojan Attacks on Steering Angle Regression Models

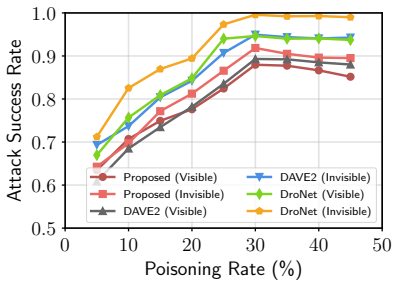
Measure	Baseline	Trojan Attack					
		Visible			Invisible		
		DAVE–2	DRONET	Our Model	DAVE–2$_{\text{NormB}}$	DRONET$_{\text{NormB}}$	Our Model$_{\text{NormB}}$
RMSE	0.00299	0.407	0.428	**0.394**	0.446	0.549	**0.412**
MAE	0.00204	0.321	0.344	**0.327**	0.391	0.474	**0.376**
ASR	−−	89.30	94.64	**87.96**	94.95	99.57	**91.85**
ER	−−	135.1	142.0	**129.8**	148.2	182.6	**136.8**



(a) Impact of Increased Trigger Intensity on Model Accuracy and Attack Success Rate



(b) RMSE vs. Poisoning Rate for Steering Models under Visible and Invisible Attacks



(c) ASR vs. Poisoning Rate for Steering Models under Visible and Invisible Attacks

Figure 5.3: Performance Analysis of Trojan Attacks on Steering Models

Trojan Attack Impact: We assessed steering control vulnerability by constructing a test set with 20% trojaned inputs. Regression performance was quantified using RMSE that penalizes large deviations and Mean Absolute Error (MAE) which reflects average prediction error, given as:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}, \quad \text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i| \tag{5.6}$$

where $\hat{y}_i$ and y_i denote predicted and true steering angles, and N is the number of test samples. Also, the attack effectiveness, was measured via the **attack success rate (ASR)**:

$$\text{ASR} = \frac{\text{Number of predictions where } \hat{y} \approx y_t}{\text{Total triggered inputs}} \times 100\% \tag{5.7}$$

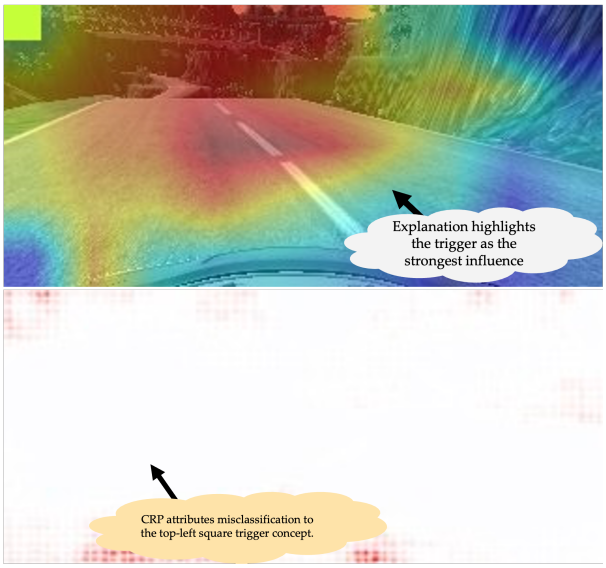
where $\hat{y} \approx y_t$ signify proportion of predictions within a target value y_t. Additionally, model performance degradation was evaluated using the **Error Rate (ER)**;

$$\text{Error Rate (ER)} = \frac{\mathscr{L}_{\text{Trojan}} - \mathscr{L}_{\text{Clean}}}{\mathscr{L}_{\text{Clean}}} \tag{5.8}$$

where $\mathscr{L}$ represents RMSE or MAE. Together, these metrics quantify both the precision loss and behavioral drift induced by backdoor triggers. Prior to simulating trojan attacks, our clean MobileNet steering baseline achieved RMSE (0.002994) and MAE (0.002041) values, as shown in Table 5.1. Systematic tuning revealed 30% poisoning and 84% visible trigger brightness delivered the optimal balance between stealth and attack efficacy, beyond which ASR gains plateaued, suggesting overfitting of the backdoor signal. As affirmed in Figures 5.3a and 5.3c.

Table 5.1 summarizes vulnerabilities of prominent end-to-end steering models, our proposed MobileNet, Deep Autonomous Vehicle Network 2 (DAVE–2) [26], and Drone Navigation Network (DroNet) [75], exposed to different backdoor configurations. Under visible trigger scenarios, our model (MobileNet) showed the lowest susceptibility, recording 0.394/0.327 RMSE/MAE scores with 87.96% ASR, outperforming DAVE–2 (0.407/0.321, ASR 89.30%) and DroNet (0.428/0.344, ASR 94.64%). Under L2–norm bounded invisible attacks, errors rose across all models, DAVE–2 (0.446/0.391), DroNet (0.549/0.474), and MobileNet (0.412/0.376). ASR similarly increased, indicating stronger stealth and misdirection capability for invisible triggers: DAVE–2 (94.95%), DroNet (99.57%), MobileNet (91.85%). The rich semantic encoding

of MobileNet, via squeeze-and-excitation layers and hard-swish activations, aided clean performance but amplified sensitivity to subtle backdoor cues. DAVE–2 showed a balanced trade-off between robustness and accuracy, benefiting from its optimized single-modality RGB-based steering design, while DroNet struggled in this frame-wise regression-only setting, limited by its reliance on auxiliary collision modalities and temporal features absent in our setup. These findings, visualized in Figures 5.3b and 5.3c, highlight distinct model vulnerability profiles and reinforce the importance of integrated explainability-based defenses for AV control security.



Visible Trigger Attributions from GRAD-CAM & CRP

Invisible Backdoor Attributions from GRAD-CAM & CRP

Figure 5.4: Attribution samples revealing sharp saliency for visible triggers and diffuse patterns for invisible ones.

Table 5.2: Performance Comparison of Different Anomaly Detection Methods

Measure	**Trojan Detection**							
	Our XAI-Guided Approach				**ACTSS** [108]		**ABS** [62]	
	CRP$_{Vi}$	**CRP$_{NormB}$**	**Grad-CAM$_{Vi}$**	**Grad-CAM$_{NormB}$**	**Visible**	**Invisible$_{NormB}$**	**Visible**	**Invisible$_{NormB}$**
Precision	0.9254	0.8924	**0.9996**	**0.9990**	0.8761	0.6942	0.8240	0.5321
Recall	0.9163	0.8927	**0.9991**	**0.9987**	0.8530	0.6819	0.8069	0.5158
F1-Score	0.9208	0.8925	**0.9994**	**0.9988**	0.8644	0.6880	0.8154	0.5238
AUC-ROC	0.9182	0.8791	**0.9993**	**0.9986**	0.8693	0.6901	0.8206	0.5287

XAI-Guided Detection Performance: Building on the vulnerability analysis, we evaluated our explainability-guided detection framework in exposing trojan configurations, using Grad-CAM and CRP attribution maps (see Fig. 5.4) derived from both benign and compromised regression outputs. This approach transforms the continuous regression task into binary anomaly detection, enabling reliable trojan identification through explanation-derived features. Notably, this strategy extended the utility of conventional classification-

oriented methods to a regression setting, a capability previously unexplored. Our evaluation employed **Precision**, **Recall** (True Positive Rate), **F1-score**, and **AUC-ROC**, collectively providing a comprehensive assessment of detection accuracy, false alarm rates, and overall discriminative power. We first trained separate MobileNet classifiers on Grad-CAM and CRP explanations individually for both visible and invisible triggers. Additional experiments combined Grad-CAM and CRP explanations (grouped by trigger type) to support ACTSS and ABS evaluations, where ABS audited neuron activations for hidden trojans and ACTSS flagged anomalous activations via statistical deviation.

From Table 5.2, focusing on visible triggers, Grad-CAM_{Vi} excelled with near-perfect metrics: Precision 99.96%, Recall 99.91%, F1-score 99.94%, and AUC-ROC 0.9993. Its gradient-derived heatmaps precisely highlighted localized saliency distortions caused by the patch triggers, enabling highly confident separation of benign and trojaned samples. CRP_{Vi} also performed strongly (Precision 92.54%, Recall 91.63%, F1-score 92.08%, AUC-ROC 0.9182), effectively capturing semantic deviations but slightly trailing Grad-CAM due to its coarser, high concept-level abstraction, which is less sensitive to tightly localized anomalies. Against invisible L2-norm bounded backdoors, Grad-$\text{CAM}_{\text{NormB}}$ maintained excellent detection capability (Precision 99.90%, Recall 99.87%, F1-score 99.88%, AUC-ROC 0.9986), demonstrating its strength in detecting subtle, distributed saliency shifts without explicit visual artifacts. $\text{CRP}_{\text{NormB}}$ also performed well (Precision 89.24%, Recall 89.27%, F1-score 89.25%, AUC-ROC 0.8791), although its abstraction level made it slightly less effective in distinguishing the nuanced distortions introduced by imperceptible triggers. In contrast, ACTSS and ABS delivered solid results on visible triggers (ACTSS: Precision 87.61%, Recall 85.30%, F1-score 86.44%, AUC-ROC 86.93%; ABS: Precision 82.40%, Recall 80.69%, F1-score 81.54%, AUC-ROC 82.06%), but struggled with invisible variants. ACTSS dropped to Precision 69.42%, Recall 68.19%, F1-score 68.80%, and AUC-ROC 69.01%; ABS further declined to Precision 53.21%, Recall 51.58%, F1-score 52.38%, and AUC-ROC 52.87%. This shortfall likely arises from the inherent subtlety of L2-norm bounded perturbations, which distribute small, coordinated adjustments across inputs, realigning internal representations without inducing conspicuous activation spikes or statistical anomalies, signals on which ACTSS and ABS rely. While ACTSS and ABS fell short against invisible attacks, their respectable performance on visible triggers in a regression context is noteworthy. This work establishes the groundwork for adapting classification-based detection paradigms to regression tasks using XAI, enabling future development of hybrid explainable-anomaly detection strategies for autonomous systems.

5.4 Conclusion

This study presented a detection framework that uses explainable techniques to perceive trojan backdoor attacks in regression-based AV steering networks, a space where existing classification-focused defenses typically fail. Our method used Grad-CAM and CRP attribution maps to detect both visible and invisible triggers through semantic-level anomaly detection. The results showed that explanation-derived features enabled near-perfect detection of visible backdoors and strong resilience against stealthy invisible variants, outperforming conventional methods like ACTSS and ABS. Although ACTSS and ABS had difficulty with invisible triggers, their solid performance on visible trojans highlights their potential for adapting classification-based detection approaches to continuous control tasks. This work sets the stage for hybrid explainable-statistical detection strategies, enabling real-time trojan identification and risk reduction in AI-driven autonomous control systems.

Chapter 6: Synthesis and Future Directions

6.1 Cross-Chapter Themes and Insights

This report has presented four interconnected research contributions addressing critical challenges in intelligent transportation systems, autonomous vehicle perception, and cybersecurity. While each chapter addresses distinct technical problems, several unifying themes emerge that highlight the synergistic nature of the work and its broader impact on DOT research and practice.

6.1.1 Privacy-Utility Tradeoffs in Transportation Systems

A fundamental tension explored throughout this report is the balance between data utility and privacy preservation. Chapter 2 demonstrates that it is possible to achieve both high forecasting accuracy and strong privacy guarantees through the integration of functional encryption with deep learning. The framework achieves mean absolute error below 10% for 60-minute forecasting horizons while ensuring that individual driver trajectories remain computationally inaccessible, even under collusion attacks. This establishes a new paradigm for privacy-preserving ITS that moves beyond traditional approaches requiring data centralization or significant accuracy degradation.

The privacy-utility tradeoff is further explored in Chapter 4, where automated annotation via explainable AI enables high-quality dataset preparation with minimal manual effort. By leveraging concept-level explanations to guide annotation, the approach achieves over 98% reduction in labeling time while producing datasets that yield superior model performance compared to manually annotated alternatives. This demonstrates that transparency and automation can simultaneously enhance both data quality and efficiency, rather than representing competing objectives.

6.1.2 Machine Learning Contributions to Transportation

The report makes significant contributions to machine learning applications in transportation across multiple dimensions. Chapter 2 introduces a hybrid deep learning architecture combining Conv-LSTM, Bi-LSTM, and Squeeze-and-Excitation modules that captures complex spatial-temporal traffic dynamics. The model achieves state-of-the-art performance on real-world datasets, demonstrating the importance of jointly modeling spatial dependencies, short-term temporal patterns, and long-term periodic trends.

Chapter 3 and Chapter 4 advance the application of explainable AI to autonomous vehicle perception tasks. The Concept Relevance Propagation (CRP) algorithm provides concept-level explanations that go beyond traditional pixel-level attributions, offering insights into what high-level concepts models learn and

how they influence decisions. This interpretability enhancement is crucial for building trust in autonomous systems and enabling regulatory compliance.

Chapter 5 extends machine learning contributions to cybersecurity, demonstrating how explainability techniques can be repurposed as active security tools. The framework bridges the gap between classification-oriented trojan detectors and continuous-output regression models, achieving near-perfect detection rates for visible triggers and strong resilience against stealthy invisible variants.

6.1.3 Cryptographic Innovations for Transportation Privacy

Chapter 2 presents significant cryptographic innovations through the integration of Inner Product Functional Encryption (IPFE) with k-anonymity mechanisms. The design achieves linear computational complexity for drivers and scalable aggregation at the TMC, making it practical for real-world deployment. Unlike blockchain-based approaches that introduce consensus overhead or homomorphic encryption schemes with super-linear costs, the IPFE-based design provides a lightweight, internet-independent solution suitable for resource-constrained vehicular environments.

The cryptographic framework incorporates multiple security properties: confidentiality of individual reports, unlinkability of encrypted cells, and anonymity guarantees even under collusion. These properties are formally proven and validated through extensive simulations, establishing a foundation for trustworthy privacy-preserving traffic management systems.

6.1.4 Real-World DOT Impact

The research contributions presented in this report address several DOT strategic priorities. The privacy-preserving traffic forecasting framework enables proactive congestion management while protecting citizen privacy, directly supporting DOT's goals of improving mobility and reducing environmental impact. The explainable AI approaches enhance transparency and trust in autonomous systems, facilitating regulatory oversight and public acceptance of emerging transportation technologies.

The automated annotation framework addresses a critical bottleneck in perception model development, reducing the time and cost barriers to deploying advanced computer vision systems in transportation applications. The trojan detection framework enhances cybersecurity posture for safety-critical autonomous systems, protecting against malicious manipulations that could compromise vehicle control.

6.2 Shared Methodological Insights

Several methodological insights emerge across chapters that inform best practices for transportation AI research:

- **Hybrid Architectures:** The success of combining multiple deep learning components (Conv-LSTM, Bi-LSTM, SE modules) in Chapter 2 demonstrates the value of hybrid architectures that capture diverse feature types. This principle extends to the integration of attention mechanisms with detection models in Chapters 3 and 4.

- **Concept-Level Interpretability:** The shift from pixel-level to concept-level explanations in Chapters 3, 4, and 5 reveals that higher-level abstractions provide more actionable insights for both interpretability and security applications.
- **Evaluation Metrics:** The use of faithfulness and complexity metrics for XAI evaluation in Chapters 3 and 4 establishes a framework for quantitative assessment of explainability techniques, moving beyond purely qualitative analysis.
- **Security Through Transparency:** Chapter 5 demonstrates that explainability techniques can serve dual purposes—enhancing interpretability and enabling security—by exposing semantic anomalies that indicate malicious behavior.

6.3 Integration Opportunities

The research contributions presented across chapters create opportunities for integrated systems that leverage multiple innovations simultaneously:

- **Privacy-Preserving XAI:** The functional encryption framework from Chapter 2 could be extended to enable privacy-preserving explainability, allowing model explanations to be computed on encrypted data without revealing sensitive inputs.
- **Secure Automated Annotation:** Combining the automated annotation approach from Chapter 4 with privacy-preserving techniques could enable collaborative dataset creation across multiple organizations while protecting proprietary data.
- **End-to-End Secure Perception:** Integrating the trojan detection framework from Chapter 5 with the perception models from Chapters 3 and 4 could create a comprehensive secure and explainable perception pipeline for autonomous vehicles.
- **Privacy-Aware Traffic Management with XAI:** The traffic forecasting framework could incorporate explainability to provide transparent insights into congestion predictions, enhancing trust and enabling better decision-making by traffic management centers.

6.4 Future Research Directions

Several promising directions for future research emerge from the work presented in this report:

6.4.1 Advanced Cryptographic Techniques

Future work could explore more advanced functional encryption schemes supporting richer function classes, enabling more complex computations on encrypted traffic data beyond simple aggregation. Current IPFE implementations focus on inner product operations, but extending to polynomial functions or more complex neural network operations would enable privacy-preserving training and inference for sophisticated models. Homomorphic encryption with improved efficiency could enable end-to-end encrypted deep learning

inference, further enhancing privacy guarantees while maintaining computational feasibility for real-time applications. Recent advances in fully homomorphic encryption (FHE) schemes with reduced overhead show promise for practical deployment in transportation systems.

Multi-party computation protocols could enable collaborative traffic forecasting across multiple jurisdictions while preserving data sovereignty, allowing different transportation agencies to contribute data without revealing sensitive information. Secure aggregation protocols that go beyond simple summation to support more complex statistical operations would enhance the utility of privacy-preserving systems. Additionally, post-quantum cryptographic schemes should be investigated to ensure long-term security as quantum computing capabilities advance, protecting transportation infrastructure against future threats.

6.4.2 Enhanced Explainability

Research could develop domain-specific explanation techniques tailored to transportation applications, incorporating domain knowledge about traffic patterns, road networks, and vehicle dynamics. Current XAI methods provide generic explanations, but transportation-specific techniques that understand semantic concepts like lane markings, traffic signs, and vehicle interactions would provide more meaningful insights. Real-time explainability for on-the-fly decision-making could enhance safety and enable adaptive systems that learn from explanations, allowing autonomous vehicles to provide immediate justifications for critical maneuvers. This capability is essential for building trust with passengers and enabling regulatory oversight of autonomous systems.

Multi-modal explanations combining visual, textual, and numerical formats could improve accessibility for diverse stakeholders, from technical engineers to policy makers and the general public. Natural language generation from attribution maps could automatically produce human-readable explanations of model decisions, facilitating communication between technical teams and non-technical stakeholders. Additionally, interactive explanation interfaces that allow users to explore different aspects of model behavior would enhance understanding and enable more effective human-AI collaboration in transportation systems.

6.4.3 Scalability and Efficiency

As autonomous systems scale to larger deployments, research must address computational and communication efficiency. Current explainability methods can be computationally expensive, limiting their applicability in resource-constrained edge devices. Developing lightweight XAI algorithms optimized for mobile and embedded platforms would enable real-time explainability on autonomous vehicles without requiring cloud connectivity. Edge computing approaches could enable local explainability and annotation without requiring cloud connectivity, reducing latency and enhancing privacy by keeping sensitive data on-device.

Federated learning could enable collaborative model training while preserving data privacy across multiple organizations, allowing transportation agencies and vehicle manufacturers to jointly improve models without sharing raw data. However, federated learning introduces challenges in explainability, as explanations must be computed across distributed models. Research into federated explainability techniques that aggregate local explanations while preserving privacy would address this gap. Additionally, quantization and model compression techniques could reduce the computational requirements of both detection models and explainability algorithms, making them more suitable for deployment in resource-constrained environments.

6.4.4 Security and Robustness

Future work should explore defenses against adaptive adversaries that attempt to evade detection by understanding the explainability-based detection mechanisms. Current detection frameworks assume static attack patterns, but sophisticated attackers may adapt their strategies to minimize attribution anomalies. Adversarial training incorporating explainability constraints could enhance robustness by training models to maintain consistent explanations even under attack, making it more difficult for adversaries to manipulate model behavior without detection. Additionally, ensemble detection approaches that combine multiple explainability methods could improve resilience against evasion attempts.

Formal verification techniques could provide mathematical guarantees about system behavior under various attack scenarios, enabling provable security properties for autonomous systems. Model checking and theorem proving approaches could verify that detection mechanisms correctly identify trojaned behavior across a wide range of attack configurations. Runtime monitoring systems that continuously validate model explanations against expected patterns could provide early warning of potential compromises. Furthermore, research into certified defenses that provide formal guarantees about detection accuracy and false positive rates would enhance trust in security mechanisms for safety-critical transportation applications.

6.4.5 Regulatory and Policy Implications

Research is needed to understand how explainability requirements translate into technical specifications and evaluation criteria. Current regulations often specify that AI systems must be explainable, but lack clear definitions of what constitutes adequate explanation. Developing standardized metrics and evaluation frameworks for explainability in transportation contexts would enable consistent assessment across different systems and facilitate regulatory compliance. Policy frameworks for privacy-preserving traffic data collection and use must balance innovation with citizen rights, ensuring that privacy protections do not unduly restrict beneficial applications while maintaining strong safeguards for sensitive information.

Standards development for XAI in transportation could facilitate interoperability and regulatory compliance, enabling different manufacturers and service providers to meet common requirements. International harmonization of explainability and privacy standards would support global deployment of autonomous systems while ensuring consistent protection levels. Additionally, research into the legal and ethical implications of explainable AI decisions is crucial, particularly regarding liability and accountability when autonomous systems make errors. Policy research should also explore incentive structures that encourage adoption of privacy-preserving and explainable technologies, potentially through regulatory requirements or certification programs that recognize systems meeting high standards for transparency and privacy protection.

6.5 Concluding Remarks

This report has presented a comprehensive investigation into privacy-preserving traffic management, explainable artificial intelligence for autonomous systems, and cybersecurity in AV control. The contributions span cryptographic innovations, machine learning advances, and practical frameworks for enhancing transparency and security in transportation systems.

The unifying theme across all chapters is the recognition that trust, transparency, and privacy are not obstacles to technological advancement but essential foundations for safe, equitable, and widely accepted intelligent transportation systems. By integrating strong cryptographic guarantees with deep learning, leveraging explainability for both interpretability and security, and developing automated solutions that enhance rather than compromise data quality, this work establishes new benchmarks for trustworthy AI in transportation.

The real-world impact of these contributions extends beyond technical achievements to address fundamental challenges facing DOT: protecting citizen privacy while enabling data-driven innovation, building public trust in autonomous systems through transparency, and ensuring cybersecurity in safety-critical applications. As transportation systems become increasingly intelligent and interconnected, the principles and frameworks presented in this report will be essential for realizing the full potential of AI while maintaining the trust and safety that citizens expect.

Future research building on these foundations will continue to push the boundaries of what is possible in privacy-preserving, explainable, and secure transportation systems, ultimately contributing to safer, more efficient, and more equitable mobility for all.

Bibliography

[1] M. Abdalla, F. Bourse, A. De Caro, and D. Pointcheval, "Simple functional encryption schemes for inner products," *in Public-Key Cryptography – PKC 2015: Springer Berlin Heidelberg*, J. Katz, Ed., pp. 733–751, Mar. 2015.

[2] R. Achtibat, M. Dreyer, I. Eisenbraun, *et al.*, "From attribution maps to human-understandable explanations through concept relevance propagation," *Nature Machine Intelligence*, vol. 5, no. 9, pp. 1006–1019, 2023.

[3] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (xai)," *IEEE access*, vol. 6, pp. 52 138–52 160, 2018.

[4] I. Adom and M. N. Mahmoud, "Rb-xai: Relevance-based explainable ai for traffic detection in autonomous systems," in *SoutheastCon 2024*, 2024, pp. 1358–1367. DOI: 10.1109/SoutheastCon52093.2024.10500215.

[5] I. Adom and M. N. Mahmoud, "RB-XAI: relevance-based explainable ai for traffic detection in autonomous systems," in *SoutheastCon 2024*, IEEE, 2024, pp. 1358–1367.

[6] C. Agnew, C. Eising, P. Denny, A. Scanlan, P. Van De Ven, and E. M. Grua, "Quantifying the effects of ground truth annotation quality on object detection and instance segmentation performance," *IEEE Access*, vol. 11, 2023.

[7] S. Agrawal, B. Libert, and D. Stehlé, "Fully secure functional encryption for inner products, from standard assumptions," *in Advances in Cryptology – CRYPTO 2016: Springer Berlin Heidelberg*, M. Robshaw and J. Katz, Eds., pp. 333–362, Jul. 2016.

[8] S. Ahmed, S. N. Nobel, and O. Ullah, "An effective deep cnn model for multiclass brain tumor detection using mri images and shap explainability," in *2023 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, 2023, pp. 1–6. DOI: 10.1109/ECCE57851.2023.10101503.

[9] S. Atakishiyev, M. Salameh, H. Yao, and R. Goebel, "Explainable artificial intelligence for autonomous driving: A comprehensive overview and field guide for future research directions," *arXiv preprint arXiv:2112.11561*, 2021.

[10] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS one*, vol. 10, no. 7, e0130140, 2015.

[11] C. Baltico, D. Catalano, D. Fiore, and R. Gay, "Practical functional encryption for quadratic functions with applications to predicate encryption," *in Advances in Cryptology – CRYPTO 2017, Cham: Springer International Publishing*, J. Katz and H. Shacham, Eds., pp. 67–98, Jul. 2017.

[12] N. T. S. Board, *Collision between a sport utility vehicle operating with partial driving automation and a crash attenuator: Mountain view, california, march 23, 2018*, 2020.

[13] D. Boneh, A. Sahai, and B. Waters, "Functional encryption: Definitions and challenges," *in Theory of Cryptography: Springer Berlin Heidelberg*, Y. Ishai, Ed., pp. 253–273, Mar. 2011.

[14] D. Boneh, A. Sahai, and B. Waters, "Functional encryption: Definitions and challenges," in *Theory of Cryptography: 8th Theory of Cryptography Conference, TCC 2011, Providence, RI, USA, March 28-30, 2011. Proceedings 8*, Springer, 2011, pp. 253–273.

[15] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6154–6162.

[16] California Department of Transportation. "Performance Measurement System (PeMS) Data Source." Accessed on 23 October 2023. (Year the page was last updated, if available), [Online]. Available: `https://dot.ca.gov/programs/traffic-operations/mpr/pems-source`.

[17] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*, Springer, 2020, pp. 213–229.

[18] O. Chapelle, B. Scholkopf, and A. Zien, "Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]," *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 542–542, 2009.

[19] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*, PMLR, 2020, pp. 1597–1607.

[20] Z. Cheng, J. Lu, H. Zhou, Y. Zhang, and L. Zhang, "Short-term traffic flow prediction: An integrated method of econometrics and hybrid deep learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 5231–5244, 2022. DOI: `10.1109/TITS.2021.3052796`.

[21] R. Chhabra, C. Krishna, and S. Verma, "A survey on state-of-the-art road surface monitoring techniques for intelligent transportation systems," *Int. J. Sens. Netw.*, vol. 37, pp. 81–99, 2 2021.

[22] D. Chowdhury, A. Sinha, and D. Das, "Xai-3dp: Diagnosis and understanding faults of 3-d printer with explainable ensemble ai," *IEEE Sensors Letters*, vol. 7, no. 1, pp. 1–4, 2022.

[23] D. Cohn, L. Atlas, and R. Ladner, "Improving generalization with active learning," *Machine learning*, vol. 15, pp. 201–221, 1994.

[24] E. Collini, L. I. Palesi, P. Nesi, G. Pantaleo, N. Nocentini, and A. Rosi, "Predicting and understanding landslide events with explainable ai," *IEEE Access*, vol. 10, pp. 31 175–31 189, 2022.

[25] J. Corso, *Annotation is dead*, Jan. 2024. [Online]. Available: `https://medium.com/@jasoncorso/annotation-is-dead-1e37259f1714`.

[26] R. Costales, C. Mao, R. Norwitz, B. Kim, and J. Yang, "Live trojan attacks on deep neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 796–797.

[27] B. G. Doan, E. Abbasnejad, and D. C. Ranasinghe, "Februus: Input purification defense against trojan attacks on deep neural network systems," in *Proceedings of the 36th Annual Computer Security Applications Conference*, 2020, pp. 897–912, ISBN: 9781450388580. DOI: 10.1145/3427228.3427264. [Online]. Available: https://doi.org/10.1145/3427228.3427264.

[28] B. Du, H. Peng, S. Wang, *et al.*, "Deep irregular convolutional residual lstm for urban traffic passenger flows prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 972–985, 2019.

[29] A. Escala, G. Herold, E. Kiltz, C. Ràfols, and J. Villar, "An algebraic framework for diffie–hellman assumptions," *Journal of cryptology*, vol. 30, pp. 242–288, 2017.

[30] X. Feng, X. Ling, H. Zheng, Z. Chen, and Y. Xu, "Adaptive multi-kernel svm with spatial–temporal correlation for short-term traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2001–2013, 2018.

[31] S. S. Girija, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *Software available from tensorflow. org*, vol. 39, no. 9, 2016.

[32] "GoFE - Functional Encryption library." Accessed on 14 October 2023. (), [Online]. Available: https://github.com/fentec-project/gofe.

[33] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 10, pp. 1915–1926, 2011.

[34] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[35] Google Research. "Introducing the Open Images Dataset." last accessed Nov. 2023. ().

[36] N. K. Gyimah, "A data-driven approach for surface defect detection and localization," *ProQuest Dissertations and Theses*, p. 142, 2024. [Online]. Available: http://ncat.idm.oclc.org/login?url=https://www.proquest.com/dissertations-theses/data-driven-approach-surface-defect-detection/docview/3098059589/se-2.

[37] N. K. Gyimah, K. D. Gupta, M. Nabil, *et al.*, "A discriminative deeplab model (ddlm) for surface anomaly detection and localization," in *2023 IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC)*, 2023, pp. 1137–1144. DOI: 10.1109/CCWC57344.2023.10099181.

[38] M. M. Hamed, H. R. Al-Masaeid, and Z. M. B. Said, "Short-term prediction of traffic volume in urban arterials," *Journal of Transportation Engineering*, vol. 121, no. 3, pp. 249–254, 1995.

[39] I. Hansen, "Determination and evaluation of traffic congestion costs," *European Journal of Transport and Infrastructure Research*, vol. 1, no. 1, pp. 61–72, 2001.

[40] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, "Learning temporal regularity in video sequences," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 733–742.

[41] N. M. Hijazi, M. Aloqaily, M. Guizani, B. Ouni, and F. Karray, "Secure federated learning with fully homomorphic encryption for iot communications," *IEEE Internet of Things Journal*, 2023.

[42] M. Hogan, N. Aouf, P. Spencer, and J. Almond, "Explainable object detection for uncrewed aerial vehicles using kernelshap," in *2022 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, IEEE, 2022, pp. 136–141.

[43] A. Howard, M. Sandler, G. Chu, *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1314–1324.

[44] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[45] R. Hu, *Introducing meta segment anything model 2 (sam 2)*, Jul. 2024. [Online]. Available: `https://ai.meta.com/sam2/`.

[46] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: Deep belief networks with multitask learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2191–2201, 2014.

[47] Inrix, *2024 inrix global traffic scorecard*, Jan. 2024. [Online]. Available: `https://inrix.com/scorecard/`.

[48] Institute of Transportation Systems at Berlin. "SUMO - Simulation of Urban MObility." Accessed on 23 October 2023. (2015), [Online]. Available: `http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931read-41000/`.

[49] W. Jiang, X. Wen, J. Zhan, X. Wang, Z. Song, and C. Bian, "Critical path-based backdoor detection for deep neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 3, pp. 4032–4046, 2022.

[50] J. Le, D. Zhang, F. Yang, T. Xiang, and X. Liao, "Secure and efficient continuous learning model for traffic flow prediction," *IEEE Transactions on Network and Service Management*, 2024.

[51] H. Lee, D. Kim, and Y.-L. Park, "Explainable deep learning model for emg-based finger angle estimation using attention," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 30, 2022. DOI: `10.1109/TNSRE.2022.3188275`.

[52] M. Lee and T. Atkison, "Vanet applications: Past, present, and future," *Vehicular Communications*, vol. 28, p. 100 310, 2021.

[53] S. Lee and D. B. Fambro, "Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting," *Transportation research record*, vol. 1678, no. 1, pp. 179–188, 1999.

[54] M. Levin and Y.-D. Tsao, "On forecasting freeway occupancies and volumes (abridgment)," *Transportation Research Record*, no. 773, 1980.

[55] F. Li, H. Zhang, S. Liu, J. Guo, L. M. Ni, and L. Zhang, "Dn-detr: Accelerate detr training by introducing query denoising," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 13 619–13 627.

[56] M. Li, Y. Fang, Z. Tang, *et al.*, "Explainable covid-19 infections identification and delineation using calibrated pseudo labels," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 7, no. 1, pp. 26–35, 2022.

[57] S. Li, L. Ge, Y. Lin, and B. Zeng, "Adaptive spatial-temporal fusion graph convolutional networks for traffic flow forecasting," in *2022 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2022, pp. 1–8.

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

[59] S. Liang, H. Wu, L. Zhen, *et al.*, "Edge yolo: Real-time intelligent object detection system based on edge-cloud cooperation in autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 25 345–25 360, 2022. DOI: 10.1109/TITS.2022.3158253.

[60] C. Lin, X. Huang, and D. He, "Ebcpa: Efficient blockchain-based conditional privacy-preserving authentication for vanets," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 3, pp. 1818–1832, 2023. DOI: 10.1109/TDSC.2022.3164740.

[61] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, Springer, 2014, pp. 740–755.

[62] Y. Liu, W.-C. Lee, G. Tao, S. Ma, Y. Aafer, and X. Zhang, "Abs: Scanning neural networks for back-doors by artificial brain stimulation," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 1265–1282.

[63] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[64] 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, 2014.

[65] C. Ma, G. Dai, and J. Zhou, "Short-term traffic flow prediction for urban road sections based on time series analysis and lstm_bilstm method," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 5615–5624, 2021.

[66] C. Ma, Y. Zhao, G. Dai, X. Xu, and S.-C. Wong, "A novel stfsa-cnn-gru hybrid model for short-term traffic speed prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 4, pp. 3728–3737, 2023. DOI: 10.1109/TITS.2021.3117835.

[67] C. Ma, Y. Zhao, G. Dai, X. Xu, and S.-C. Wong, "A novel stfsa-cnn-gru hybrid model for short-term traffic speed prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 4, pp. 3728–3737, 2022.

[68] J. Ma, M. Xu, Q. Meng, and L. Cheng, "Ridesharing user equilibrium problem under od-based surge pricing strategy," *Transportation Research Part B: Methodological*, vol. 134, pp. 1–24, 2020.

[69] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, p. 818, 2017.

[70] R. Machlev, M. Perl, J. Belikov, K. Y. Levy, and Y. Levron, "Measuring explainability and trustworthiness of power quality disturbances classifiers using xai—explainable artificial intelligence," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 8, pp. 5127–5137, 2022. DOI: 10.1109/TII.2021.3126111.

[71] H. Mankodiya, D. Jadav, R. Gupta, S. Tanwar, W.-C. Hong, and R. Sharma, "Od-xai: Explainable ai-based semantic object detection for autonomous vehicles," *Applied Sciences*, vol. 12, 2022.

[72] Y. Miao, X. Bai, Y. Cao, *et al.*, "A novel short-term traffic prediction model based on svd and arima with blockchain in industrial internet of things," *IEEE Internet of Things Journal*, vol. 10, no. 24, pp. 21 217–21 226, 2023. DOI: 10.1109/JIOT.2023.3283611.

[73] P. A. Moreno-Sánchez, "Data-driven early diagnosis of chronic kidney disease: Development and evaluation of an explainable ai model," *IEEE Access*, vol. 11, pp. 38 359–38 369, 2023.

[74] N. Moustafa, N. Koroniotis, M. Keshk, A. Y. Zomaya, and Z. Tari, "Explainable intrusion detection for cyber defences in the internet of things: Opportunities and solutions," *IEEE Communications Surveys & Tutorials*, 2023.

[75] M. Mynuddin, S. U. Khan, R. Ahmari, L. Landivar, M. N. Mahmoud, and A. Homaifar, "Trojan attack and defense for deep learning-based navigation systems of unmanned aerial vehicles," *IEEE Access*, vol. 12, pp. 89 887–89 907, 2024. DOI: 10.1109/ACCESS.2024.3419800.

[76] M. Mynuddin, S. U. Khan, and M. N. Mahmoud, "Trojan triggers for poisoning unmanned aerial vehicles navigation: A deep learning approach," in *2023 IEEE International Conference on Cyber Security and Resilience (CSR)*, IEEE, 2023, pp. 432–439.

[77] T. Nguyen, K. Nguyen, T. Nguyen, T. Nguyen, A. Nguyen, and K. Kim, "Hierarchical uncertainty aggregation and emphasis loss for active learning in object detection," in *2023 IEEE International Conference on Big Data (BigData)*, IEEE, 2023, pp. 5311–5320.

[78] "OpenStreetMap." Accessed on 23 October 2023. (), [Online]. Available: https://www.openstreetmap.org.

[79] F. Outay, H. A. Mengash, and M. Adnan, "Applications of unmanned aerial vehicle (uav) in road safety, traffic and highway infrastructure management: Recent advances and challenges," *Transportation research part A: policy and practice*, vol. 141, pp. 116–129, 2020.

[80] Y. Qi, J. Wu, A. K. Bashir, X. Lin, W. Yang, and M. D. Alshehri, "Privacy-preserving cross-area traffic forecasting in its: A transferable spatial-temporal graph neural network approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 12, pp. 15 499–15 512, 2023. DOI: 10.1109/TITS.2022.3215326.

[81] K. Rabieh, M. M. Mahmoud, and M. Younis, "Privacy-preserving route reporting schemes for traffic management systems," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 2703–2713, 2016.

[82] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788. DOI: 10.1109/CVPR.2016.91.

[83] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: A database and web-based tool for image annotation," *International journal of computer vision*, vol. 77, pp. 157–173, 2008.

[84] W. Samek, G. Montavon, A. Binder, S. Lapuschkin, and K.-R. Müller, *Interpreting the predictions of complex ml models by layer-wise relevance propagation*, 2016. arXiv: 1611.08191 [stat.ML].

[85] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

[86] B. Settles, "Active learning literature survey," 2009.

[87] S. Sharma and S. Awasthi, "Introduction to intelligent transportation system: Overview, classification based on physical architecture, and challenges," *Int. J. Sens. Netw.*, vol. 38, pp. 215–240, 4 2022.

[88] R.-K. Sheu, M. S. Pardeshi, K.-C. Pai, L.-C. Chen, C.-L. Wu, and W.-C. Chen, "Interpretable classification of pneumonia infection using explainable ai (xai-icp)," *IEEE Access*, vol. 11, pp. 28 896–28 919, 2023.

[89] W. Shi, L. Tong, Y. Zhu, and M. D. Wang, "Covid-19 automatic diagnosis with radiographic imaging: Explainable attention transfer deep neural networks," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, 2021. DOI: 10.1109/JBHI.2021.3074893.

[90] Y. Shin and Y. Yoon, "Pgcn: Progressive graph convolutional networks for spatial–temporal traffic forecasting," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[91] V. Shoup, "Lower bounds for discrete logarithms and related problems," *in Advances in Cryptology — EUROCRYPT '97: Springer Berlin Heidelberg*, W. Fumy, Ed., pp. 256–266, Jul. 1997.

[92] N. A. Stanton, P. M. Salmon, G. H. Walker, and M. Stanton, "Models and methods for collision analysis: A comparison study based on the uber collision with a pedestrian," *Safety Science*, vol. 120, pp. 117–128, 2019.

[93] J. Sun, J. Wu, F. Xiao, Y. Tian, and X. Xu, "Managing bottleneck congestion with incentives," *Transportation research part B: methodological*, vol. 134, pp. 143–166, 2020.

[94] SuperAnnotate, *Annotate faster with one-shot image annotation*, Jul. 2024. [Online]. Available: https://www.superannotate.com/image-annotation-tool.

[95] L. Sweeney, "K-anonymity: A model for protecting privacy," *International journal of uncertainty, fuzziness and knowledge-based systems*, vol. 10, no. 05, pp. 557–570, 2002.

[96] M.-C. Tan, S. C. Wong, J.-M. Xu, Z.-R. Guan, and P. Zhang, "An aggregation approach to short-term traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 1, pp. 60–69, 2009.

[97] J. Tao, Y. Xiong, S. Zhao, *et al.*, "Explainable ai for cheating detection and churn prediction in online games," *IEEE Transactions on Games*, 2022.

[98] Y. Tian and L. Pan, "Predicting short-term traffic flow by long short-term memory recurrent neural network," in *2015 IEEE international conference on smart city/SocialCom/SustainCom (SmartCity)*, IEEE, 2015, pp. 153–158.

[99] Ultralytics, *Ultralytics yolo docs*, Jan. 2024. [Online]. Available: `https://docs.ultralytics.com/models/yolov8/`.

[100] M. Van Der Voort, M. Dougherty, and S. Watson, "Combining kohonen maps with arima time series models to forecast traffic flow," *Transportation Research Part C: Emerging Technologies*, vol. 4, no. 5, pp. 307–318, 1996.

[101] D. Wang, W. Li, and J. Pan, "Large-scale mixed traffic control using dynamic vehicle routing and privacy-preserving crowdsourcing," *IEEE Internet of Things Journal*, vol. 11, no. 2, pp. 1981–1989, 2024. DOI: 10.1109/JIOT.2023.3335292.

[102] Y. Wang, C. Jing, W. Huang, S. Jin, and X. Lv, "Adaptive spatiotemporal inceptionnet for traffic flow forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 4, pp. 3882–3907, 2023. DOI: 10.1109/TITS.2023.3237205.

[103] Y. Wang, V. Ilic, J. Li, B. Kisačanin, and V. Pavlovic, "Alwod: Active learning for weakly-supervised object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 6459–6469.

[104] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results," *Journal of transportation engineering*, vol. 129, no. 6, pp. 664–672, 2003.

[105] J. Witt, *Automated annotation with auto label*, May 2024. [Online]. Available: `https://docs.roboflow.com/annotate/automated-annotation-with-autodistill`.

[106] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[107] M. Xia, D. Jin, and J. Chen, "Short-term traffic flow prediction based on graph convolutional networks and federated learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 1, pp. 1191–1203, 2022.

[108] Y. Xuan, X. Chen, Z. Zhao, Y. Ding, and J. Lv, "Actss: Input detection defense against backdoor attacks via activation subset scanning," in *2022 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2022, pp. 1–8.

[109] L. Yang, J. Zhong, Y. Zhang, *et al.*, "An improving faster-rcnn with multi-attention resnet for small target detection in intelligent autonomous transport with 6g," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 7, pp. 7717–7725, 2023. DOI: 10.1109/TITS.2022.3193909.

[110] W. Yu, S. Zhu, T. Yang, and C. Chen, "Consistency-based active learning for object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3951–3960.

[111] T. Zebin, S. Rezvy, and Y. Luo, "An explainable ai-based intrusion detection system for dns over https (doh) attacks," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 2339–2349, 2022.

[112] C. Zhang, L. Zhu, C. Xu, X. Du, and M. Guizani, "A privacy-preserving traffic monitoring scheme via vehicular crowdsourcing," *Sensors*, vol. 19, no. 6, p. 1274, 2019.

[113] J. Zhang, H. Fang, H. Zhong, J. Cui, and D. He, "Blockchain-assisted privacy-preserving traffic route management scheme for fog-based vehicular ad-hoc networks," *IEEE Transactions on Network and Service Management*, vol. 20, no. 3, pp. 2854–2868, 2023. DOI: 10.1109/TNSM.2023.3238307.

[114] Y. Zhang, Q. Pei, F. Dai, and L. Zhang, "Efficient secure and privacy-preserving route reporting scheme for vanets," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 910, 2017, p. 012 070.

[115] Y. Zhao, Y. Lin, H. Wen, T. Wei, X. Jin, and H. Wan, "Spatial-temporal position-aware graph convolution networks for traffic flow forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 8, pp. 8650–8666, 2023. DOI: 10.1109/TITS.2022.3220089.

[116] Q. Zhaowei, L. Haitao, L. Zhihui, and Z. Tao, "Short-term traffic flow forecasting method with mb-lstm hybrid network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 225–235, 2020.

[117] H. Zheng, F. Lin, X. Feng, and Y. Chen, "A hybrid deep learning model with attention-based conv-lstm networks for short-term traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 11, pp. 6910–6920, 2020.

[118] M. Zolanvari, Z. Yang, K. Khan, R. Jain, and N. Meskin, "Trust xai: Model-agnostic explanations for ai with a case study on iiot security," *IEEE internet of things journal*, 2021.

[119] L. Zou, H. L. Goh, C. J. Y. Liew, *et al.*, "Ensemble image explainable ai (xai) algorithm for severe community-acquired pneumonia and covid-19 respiratory infections," *IEEE Transactions on Artificial Intelligence*, vol. 4, no. 2, pp. 242–254, 2022.