Politecnico di Torino

Politecnico di Torino

Master Degree in Computer Science Engineering

A.a. 2024/2025

Graduation Session October 2025

Enhancing VRU Awareness through TTC-Based Trajectory Interception Probability for VAM Triggering

Supervisors:

Claudio Ettore Casetti

Marco Rapelli

Francesco Raviglione

Candidate:

Alice Cusinato

Abstract

Over the past century, transportation technologies have evolved rapidly due to breakthroughs in automation, vehicle connectivity, and the rise of cooperative systems. These innovations are reshaping how vehicles, infrastructure, and road users interact, moving toward shared situational awareness and coordinated safety. In this context, protecting Vulnerable Road Users (VRUs, including pedestrians, cyclists, powered two-wheelers, and other unprotected participants, has become a priority, as they face a higher risk of injury or death in traffic accidents, making their safety a crucial objective for modern intelligent transportation systems.

To address this challenge, the European Telecommunications Standards Institute (ETSI) has developed the VRU Basic Service (VBS) and the VRU Awareness Message (VAM) to enable VRUs to be consistently represented within Cooperative-ITS environments. VAM generation is regulated by a hybrid mechanism: it combines periodic transmission and seven event-driven triggering conditions, avoiding network overload. Among these conditions, the Trajectory Interception Probability (TIP) trigger is one of the most challenging to implement: indeed, while ETSI specifies that a VAM must be sent when the TIP changes by at least 10% compared to the last transmission, it does not provide a computation method, leaving a gap available for research and studies to be filled.

This thesis tackles this gap by developing and integrating an innovative method to compute TIP in real time, beginning from the Time-to-Collision (TTC) metric. Firstly, to ensure the focus is on actual safety-critical situations, the TTC domain is limited to a relevant temporal horizon, using minimum and maximum thresholds. Within the bounded range, several TTC to TIP mapping strategies were compared, including fixed intervals, linear and exponential functions. After a comparative analysis, the solution was to use a discrete exponential mapping: this function leads to an increase in collision probability more sharply as TTC decreases, reflecting the rapidly growing urgency of an imminent collision, while avoiding sudden jumps that could lead to unnecessary message triggering. Its smooth yet responsive behavior ensures compliance with ETSI's standard requirements, achieving a robust and efficient implementation of the triggering condition.

The proposed methodology, implemented within the VaN3Twin framework, starts from computing TTC from kinematic state variables, maps it to TIP through the chosen exponential function, and triggers a new VAM whenever the ETSI-defined condition is met, enabling a continuous frame-by-frame assessment of collision risk. The proposed approach was validated through a set of simulation scenarios designed to reproduce realistic interactions between vehicles and pedestrians. These scenarios were first used to identify the key parameters required by the chosen

mapping function, such as the temporal evaluation window and the growth rate of the probability curve. Once defined, the calibrated parameters were applied again to the same scenarios to analyse how the introduction of the TIP-based condition influences VAM triggering behaviour. This two-step process allowed both the tuning of the method and the assessment of its impact on message generation, showing that the new condition ensures timely yet non-redundant transmissions and improves VRU awareness, while avoiding to overload the communication channel.

Acknowledgements

Before going deep into the content of this thesis, I would really like to thank my advisors Dott. Claudio Ettore Casetti, Dott. Marco Rapelli and Dott. Francesco Raviglione that allowed me to take part in this great project, which proved to be highly fulfilling, and that supported me deeply throughout the whole course of this work, always present to provide suggestions and assistance whenever needed. I am glad to have been part of this team and I will treasure the lessons learned in my next experiences. Then, I want to thank my family, my boyfriend and my friends as they have always been there for me from day one, encouraging me every step of the way, making me feel loved and supported throughout this whole academic journey.

Table of Contents

List of Figures

Acronyms

ADAS Advanced Driver Assistance System

AI Artificial Intelligence

API Application Programming Interface

BS Basic Services (Container)

BTP Basic Transport Protocol

CA Cooperative Awareness (Service)

CAM Cooperative Awareness Message

C-ITS Cooperative Intelligent Transport Systems

CSV Comma-Separated Values (File Format)

C-V2X Cellular Vehicle-to-Everything

D2D Device-to-Device

D2N Device-to-Network

DCC Decentralized Congestion Control

DCC-FAC Decentralized Congestion Control – Facilities Interface

DDP Device Data Provider

DEN Decentralized Environmental Notification (Service)

DRAC Deceleration Rate to Avoid Collision

ECU Electronic Control Unit

ETSI European Telecommunications Standards Institute

EV Electric Vehicle

FA-SAP Facilities & Application Service Access Point

GBC GeoBroadcast

GHz Gigahertz

GN GeoNetworking

GNSS Global Navigation Satellite System

GRU Gated Recurrent Unit (Neural Network)

HIL Hardware-in-the-Loop

HJ Hamilton–Jacobi (Reachability)

HMI Human Machine Interface

IDM Intelligent Driver Model

IoT Internet of Things

IP Internet Protocol

IPv6 Internet Protocol version 6

ITS Intelligent Transport Systems

ITS-C ITS Constellation

ITS-S Intelligent Transport System Station

LDM Local Dynamic Map

LiDAR Light Detection and Ranging

LSTM Long Short-Term Memory (Neural Network)

LTE Long-Term Evolution

LTE-V2X LTE-based Vehicle-to-Everything

MaaS Mobility-as-a-Service

MCO Multi-Channel Operation

MF-SAP Management & Facilities Service Access Point

MSLaD Minimum Safe Lateral Distance

MSLoD Minimum Safe Longitudinal Distance

MSVD Minimum Safe Vertical Distance

NDT Network Digital Twin

NF-SAP Network & Facilities Service Access Point

5G NR-V2X 5G New Radio Vehicle-to-Everything

ns-3 Network Simulator 3

OBU On-Board Unit

OSM OpenStreetMap

OSI Open Systems Interconnection

PDU Protocol Data Unit

PET Post-Encroachment Time

PKI Public Key Infrastructure

PoTi Position and Time Management

POMDP Partially Observable Markov Decision Process

PRR Packet Reception Ratio

RAT Radio Access Technology

RSU Road Side Unit

RSUs Road Side Units

SAP Service Access Point

SF-SAP Security & Facilities Service Access Point

SHB Single-Hop Broadcast

SSM Surrogate Safety Measure

SUMO Simulation of Urban Mobility

TraCI Traffic Control Interface

TTC Time-To-Collision

URLLC Ultra-Reliable Low Latency Communications

VA Vulnerable Awareness (Basic Service)

V2I Vehicle-to-Infrastructure

V2N Vehicle-to-Network

V2P Vehicle-to-Pedestrian

V2V Vehicle-to-Vehicle

V2X Vehicle-to-Everything

VAM VRU Awareness Message

VANET Vehicular Ad-hoc Network

VBS VRU Basic Service

VDP Vehicle Data Provider

VRU Vulnerable Road User

WHO World Health Organization

Chapter 1

Introduction

1.1 Urban and Smart Mobility

Over the past century, the evolution of transportation technologies has significantly reshaped how people and goods move in modern society. During the journey from the early attempts at automation to today's interconnected transport systems, the limitations of traditional systems, such as congestion, environmental sustainability and safety risks, have become increasingly apparenty, laying the groundwork for the emergence of a new paradigm: Smart Mobility. This new concept is the natural outcome of decades of technological evolution in transportation: it refers to an integrated, efficient, and sustainable approach to mobility that leverages automation, connectivity, and data-driven decision-making to address the growing complexities of urban transport. Its roots can be traced back to the earliest attempts at autonomous driving, one of the pillars of Smart Mobility, to the early 20th century. In 1920s, experiments such as radio-controlled cars began exploring the possibility of vehicles operating without human intervention. By 1960s, magnetic cables embedded in roads were being tested to guide cars automatically, pushing the idea further through notable trials, like the one with the Citroën DS reaching speeds of 130 km/h under automated control. However, the real turning point cambe in the 1980s and 1990s, when pioneering research projects like Carnegie's Mellon's Navlab and the EUREKA Prometheus Project brought computer vision and artificial intelligence into the picture. These systems could perceive road environments and make autonomous decisions, paving the way for truly intelligent vehicles. In the early 2000s, the DARPA Grand Challenges accelerated the momentum, driving advancements in LIDAR, radar and AI systems, enabling vehicles to perceive and navigate complex environments with greater efficiency and accuracy. During this evolution, it became clear that to unlock the full potential of autonomous driving technology, the mobility would need to shift from an isolated perspective to a

collaborative one, as vehicles should have been able to communicate not only with each other, but also with road users and infrastructures. The need of having a connected transportation ecosystem marked the actual shift from autonomy to the concept of Smart Mobility, which today leverages real-time data, connectivity and automation to improve safety, efficiency and sustainability in urban environments. Furthermore, the concept encompasses a wide range of innovations other than autonomous driving, such as electric vehicles (EVs), ride-sharing and Mobility-As-A-Service (MaaS) platforms, that aim to increasingly respond to the urban challenges in the mobility systems.

Crucial to this transformation and to its success is the development of Intelligent Transport Systems (ITS). ITS provides the infrastructure to enable real-time data exchange between vehicles and their environment through advanced communication technologies like Vehicle-To-Everything (V2X). It allows coordinated behavior among road users, improving safety and optimizing traffic management, playing a key role in realizing the vision of Smart Mobility.

1.2 Communication technologies in Intelligent Transportation Systems

Intelligent Transportation Systems (ITS) represent the integration of advanced sensing, processing and communication technologies into vehicles, road infrastructure and personal user devices, with the ultimate goal of enhancing road safety, traffic efficiency and environmental sustainability of modern transportation systems. In Europe, the design, deployment, and interoperability of ITS are guided by the European Telecommunications Standards Institute (ETSI) reference architecture, which follows a layered approach inspired by the Open System Interconnection (OSI) model but adapted to the specific requirements of cooperative vehicular environments. It comprises four horizontal functional layers, supported by two entities that operate vertically across the whole architecture. This design ensures modularity, interoperability, and flexibility for future technological integration [1].

- **Access Layer** – provides the physical and data link services to facilitate the exchange of information between the physical elements of the ITS. It supports multiple radio access technologies for different types of communications, such as Dedicated Short Range Communications (DSRC), low-latency, cellular-based and wireless. The Access Layer is also responsible for channel management, spectrum usage in the 5.9 GHz band and basic mechanisms for Decentralized Congestion Control (DCC) at the physical layer, which becomes particularly critical in scenarios with a high density of pedestrians or cyclists. This layer represents the foundation that enables common consumer devices, such as

smartphones, wearables, e-scooter modules, to behave as ITS Stations and to transmit their presence to nearby vehicles and other Road Station Users (RSUs).

- **Networking & Transport Layer** – manages data addressing, routing, and end-to-end transport. It supports GeoNetworking, a key geographic addressing and routing protocol that enables both direct (single-hop) and indirect (multi-hop) message delivery based on the geographic position of nodes, ensuring resilience. Moreover, support of IPv6 integration for interoperability with conventional IP-based networks enables devices to integrate data into cloud-based services or central traffic management platforms.

- **Facilities Layer** – provides a set of core services that process and manage information exchanged across the ITS stack, bridging the lower communication layers with applications. It is responsible for generating, handling, and interpreting standardized messages (e.g., CAM, DENM, VAM), ensuring their correct formatting, security and dissemination. The Facilities Layer also supports data storage (e.g., Local Dynamic Map), service access control and interfaces for cooperative awareness, enabling applications to operate on meaningful and structured information rather than raw communication data. Moreover, this layer is also in charge of managing cluster creation, maintenance and breakup, ensuring efficient use of the channel in dense environments. Finally, optional information such as motion prediction data are also processed here, such as path history and future trajectory of the devices.

- **Application Layer** – At the top of the stack, the Application Layer hosts the ITS applications that transform the information processed at the Facilities Layer into services that are directly delivered to users and system operators. This includes road safety functions, traffic efficiency services and environmental objectives. In this sense, the Application Layer bridges the technical exchange of standardized messages with tangible safety and efficiency outcomes.

The two vertical entities are:

- **Management Entity** – oversees resource allocation, operational configuration and system monitoring across all horizontal layers. It includes functionalities for Multi Channel Operation (MCO), allowing simultaneous use of multiple radio channels for improved capacity and reliability in dense traffic scenarios.

- **Security Entity** – ensures trust, authenticity, and integrity of messages through a Public Key Infrastructure (PKI) defined by ETSI. Also, it manages certificate issuance, validation, and revocation, as well as protects user privacy via pseudonym certificates that are periodically changed to prevent tracking.

This architecture is designed to be technology-agnostic at the Access Layer, allowing the integration of new communication standards without impacting higher layers, and to ensure interoperability across vendors and regions.

Within the framework, four types of ITS Stations (ITS-S) have been defined, each associated with a different role in the ITS ecosystem:

1. **Personal ITS Station (Personal ITS-S)** – portable devices such as smartphones or wearable electronics, typically used by Vulnerable Road Users (VRUs) to communicate with the ITS environment. They can transmit location, speed, and heading data, enabling vehicle systems to anticipate potential conflicts.

2. **Vehicle ITS Station (Vehicle ITS-S)** – vehicles equipped with On-Board Units (OBUs) that connect to on-board sensors (GNSS, radar, LiDAR, cameras) and control systems (braking, steering, powertrain). The Vehicle ITS-S supports both V2V and V2I communications and may act as a relay for multi-hop data forwarding.

3. **Roadside ITS Station (Roadside ITS-S)** – infrastructure nodes equipped with Road Side Units (RSUs), often deployed at intersections, motorway entries, or hazardous road segments. RSUs extend the communication range, collect environmental and traffic data, and act as gateways to central systems.

4. **Central ITS Station (Central ITS-S)** – backend systems operated by traffic management centres, fleet operators, or service providers. They process aggregated data, coordinate large-scale traffic operations, and manage long-term strategic actions.

Each ITS-S includes an ITS-S internal network interconnecting its functional modules and providing standardised interfaces for external communication. Multiple ITS-S can form an ITS Constellation (ITS-C), a temporary group of stations sharing a common geographic and temporal context. Within an ITS-C, the Position and Time Management (PoTi) service ensures synchronisation of spatial and temporal data, which is critical for cooperative awareness and hazard prediction.

The implications of ITS innovations are significant. First, ITS enhance safety by providing vehicles with information that may be outside the range of onboard sensors, ensuring redundancy in critical decision-making scenarios. For instance, ITS allow for vehicle-to-infrastructure (V2I) and vehicle-to-vehicle (V2V) communication, where vehicles can exchange crucial data about traffic conditions, obstacles, or potential hazards. This integration not only reduces the likelihood of collisions but also allows for better coordination in emergency situations, such as accidents or sudden braking. As noted by the American Association of State Highway and

Transportation Officials [2], redundancy in vehicle sensors and external communication can significantly improve the reliability of automated driving systems, enhancing safety outcomes.

In practical terms, ITS can be deployed across a wide variety of scenarios, ranging from traffic management to safety-critical operations. A prominent example is the optimisation of urban traffic flows through intelligent traffic signals that adjust dynamically in response to real-time conditions, thereby reducing delays and improving overall efficiency. ITS also underpin advanced driver assistance features, such as lane change assistance and cooperative collision risk warning, which enhance safety by enabling vehicles to share data and coordinate manoeuvres.

ITS applications are typically grouped into three main categories:

- **Road safety** – collision avoidance, VRU protection, emergency vehicle prioritisation, and intersection movement assistance;
- **Traffic efficiency** – adaptive traffic signal control, real-time rerouting, and congestion mitigation;
- **Environmental sustainability** – eco-driving feedback, integration with electric vehicle infrastructure, and emission monitoring and reduction strategies.

Among these, Cooperative ITS (C-ITS) represents the most advanced class of applications. In C-ITS, vehicles, infrastructure, and VRUs exchange information in real time to establish a shared situational awareness, enabling proactive safety measures such as early collision avoidance, coordinated lane merging, and network-level traffic flow optimisation. A key enabler of this paradigm is Vehicle-to-Everything (V2X) communication to improve situational awareness and safety. While V2X is discussed in detail in the following section, it is important to note here that it plays a key role in the future of connected transportation.

Overall, ITS aim to make transportation safer, more efficient, and more sustainable. Leveraging advances in data collection, wireless communication, and automation, they provide the technological foundation for innovations in autonomous driving, smart cities, and environmentally responsible mobility. By enabling governments and private organisations to tackle pressing challenges such as congestion, safety, and accessibility, ITS also open the door to new transportation services and business models that can enhance the quality of life for all road users.

Within this context, the ITS framework, built on a standardised architecture, modular design, and secure, low-latency communication, offers a robust platform for the development of next-generation mobility systems capable of delivering cooperative, safe, and sustainable transport services.

1.2.1 IoT-enabled ITS

A factor to consider in terms of innovation is the integration of the Internet of Things (IoT) with ITS, which represents a transformative advancement in transportation technology. IoT enables the connection of millions of devices, vehicles, and infrastructure components, allowing for seamless communication and data exchange. This interconnected ecosystem provides a new level of real-time data acquisition and analytics that can improve traffic management, enhance safety, and optimize transportation networks.

Within the IoT-enabled ITS ecosystem, vehicles are equipped with sensors that continuously gather data on position, speed, and driving conditions, while roadside infrastructures such as traffic lights and road signs are fitted with sensors capable of communicating directly with passing vehicles. This interconnected network allows for real-time monitoring of traffic conditions, immediate hazard-detection, and signal timing to be adapted dynamically. This results in smarter traffic management, characterized by reduced congestion, smoother vehicle flow and personalized route suggestions for drivers, improving both efficiency and road safety.

IoT technologies play a key role in Vehicle-to-Everything (V2X) communication, which includes vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), vehicle-to-pedestrian (V2P), and vehicle-to-network (V2N) interactions.

1.2.2 Communication Protocols

V2X is a key technological innovation within ITS that enables real-time communication between vehicles and their surrounding environment. By allowing vehicles to exchange data with one another and with roadside infrastructure, V2X systems provide a comprehensive view of the road environment, leading to better decision-making and coordination. V2X communication includes several sub-categories, such as V2V, V2I, V2N, and V2P, all of which play a crucial role in enhancing road safety, traffic efficiency, and driving experience:

1. **Vehicle-to-Vehicle**: V2V allows vehicles to share real-time data about their speed, position, and direction. This helps vehicles anticipate potential hazards and enables quicker reactions to sudden events, such as braking or swerving vehicles ahead. By receiving alerts about potential risks before they become visible, V2V can significantly reduce the risk of collisions, especially in high-speed or low-visibility conditions.

2. **Vehicle-to-Infrastructure**: V2I enables vehicles to interact with roadside infrastructure such as traffic signals, road signs, and other control systems. Through V2I, traffic signal timings can be dynamically adjusted based on real-time vehicle flow, reducing congestion and improving overall traffic management. V2I can also notify drivers of upcoming hazards, such as roadwork or

lane closures, allowing for better trip planning and safer driving. Additionally, it can provide speed recommendations to vehicles to help minimize unnecessary stops and optimize fuel consumption.

3. **Vehicle-to-Network**: V2N connects vehicles to cloud-based systems, providing access to various services, such as traffic monitoring, weather updates and navigation assistance. As it allows linking vehicles to a centralized traffic management system, V2N enhances the coordination of multiple entities and infrastructure, allowing for a full view of the traffic network: this improves traffic flow, especially in urban areas, and enhances the overall driving experience by offering real-time updates.

4. **Vehicle-to-Pedestrian**: V2P focuses on the safety of vulnerable road users, such as pedestrians and cyclists. By enabling vehicles to detect and communicate with nearby pedestrians through smartphones, wearable devices, or sensors, drivers can receive alerts about people crossing the road or moving in blind spots. This is particularly valuable in urban areas with high pedestrian traffic, as it can significantly reduce the likelihood of accidents involving pedestrians and cyclists.

The most significant impact of V2X technology is its potential to increase road safety. Through V2V communication, vehicles can share real-time data about their speed, position, and direction, enabling drivers or autonomous systems to anticipate and react to potential hazards more quickly than with traditional sensors alone. For example, if a vehicle ahead applies sudden brakes, the following vehicles can receive immediate notifications and begin decelerating even before the obstacle becomes visible. This early-warning mechanism can drastically reduce the risk of rear-end collisions, especially in high-speed or low-visibility conditions.

V2X technology relies heavily on communication protocols to ensure the seamless transfer of data between vehicles, infrastructure, and other road users. The two primary communication protocols used in V2X systems are DSRC and Cellular-V2X (C-V2X).

1. **Dedicated Short Range Communication**:
   Dedicated Short-Range Communications (DSRC) is a wireless technology created specifically to enable vehicles to communicate with each other and with the surrounding infrastructure. It operates in the 5.9 GHz frequency band and supports very low-latency exchanges over distances typically between 300 and 1000 meters. Thanks to these characteristics, DSRC is well suited for safety-critical situations where rapid reactions are required, such as emergency braking alerts, collision warnings, or vehicle coordination at intersections. At its core, DSRC is based on the IEEE 802.11p amendment to the Wi-Fi

standard, which was adapted to handle the unique challenges of fast-changing traffic scenarios. This design guarantees robust and predictable performance even at high speeds, allowing vehicles to exchange crucial data within just a few milliseconds. Over the past decade, DSRC has matured significantly: numerous field trials and large-scale pilot projects worldwide have validated its reliability, low latency, and interoperability under real-world conditions. One of its most appreciated strengths is that it works independently of cellular networks — direct V2V and V2I communication ensures that vital safety messages continue to be exchanged even in rural or poorly covered areas, maintaining a consistent level of cooperative awareness.

2. **Cellular-V2X**:
   Cellular Vehicle-to-Everything (C-V2X) is a communication protocol defined by 3GPP that builds on existing cellular networks such as 4G LTE and 5G to enable long-range, high-capacity data exchange between vehicles, infrastructure, and the cloud. C-V2X supports two complementary modes of operation: device-to-device (D2D), which allows vehicles to communicate directly with each other and with infrastructure without relying on the cellular network, and device-to-network (D2N), which connects vehicles to the cloud via cellular towers, enabling wide-area communication. The first generation, LTE-V2X (3GPP Release 14), extends the range and reliability of V2X communications compared to DSRC, supporting most cooperative applications with low-to-moderate latency requirements. It is particularly well suited for safety services, traffic efficiency functions, and non-time-critical data exchange in both urban and highway scenarios. The next evolution, 5G NR-V2X (3GPP Release 16), introduces ultra-reliable low-latency communications (URLLC), network slicing, and advanced resource management, unlocking demanding use cases such as coordinated autonomous driving, high-speed platooning, and real-time sharing of rich sensor and perception data. One of the key strengths of C-V2X lies in its scalability and its seamless integration with existing cellular infrastructure, making it easier to deploy over large areas and in dense urban environments. The introduction of 5G further enhances its capabilities by providing ultra-low latency — potentially down to 1 millisecond — increased bandwidth, and higher reliability. These features make C-V2X suitable not only for real-time safety-critical applications but also for bandwidth-intensive services such as high-definition map distribution, over-the-air updates, and cooperative perception. Compared to DSRC, C-V2X can offer comparable or even better latency while supporting a much broader range of applications, making it a highly versatile technology. Moreover, as cellular networks evolve, C-V2X will continue to benefit from ongoing improvements in 5G and future 6G technologies, ensuring its long-term relevance and compatibility with

next-generation mobility services.

Finally, the choice between DSRC and C-V2X often depends on the specific application and region of deployment. While DSRC provides robust, reliable communication for short-range, safety-critical applications, C-V2X offers greater scalability and integration into broader mobile networks, making it a more future-proof solution as vehicles become increasingly autonomous and connected (see Table 1.1 for a comparative summary of key characteristics).

Table 1.1: Comparison of main V2X communication technologies

Feature	**DSRC**	**LTE-V2X**	**5G NR-V2X**
Frequency	5.9 GHz	5.9 GHz	5.9 GHz / mmWave
Latency	∼10 ms	<20 ms	<5 ms
Range	300–1000 m	500–1500 m	500–3000 m
Throughput	3–27 Mbps	15–100 Mbps	>100 Mbps
Infrastructure dependency	None	Optional	Optional
Scalability	Medium	High	Very high
Maturity	High	Medium	Emerging

Hybrid approaches are increasingly being explored to combine the strengths of both DSRC and C-V2X technologies. In such configurations, DSRC can be used for ultra-low-latency, short-range communication in safety-critical scenarios, while LTE-V2X or 5G NR-V2X provide extended coverage, higher throughput, and integration with cloud-based services. A practical example of this vision is the design of hybrid vehicular network architectures where DSRC manages immediate localized safety alerts, such as collision warnings at intersections, while C-V2X ensures wide-area connectivity and coordination with cloud-based services, for instance enabling real-time traffic monitoring and cooperative route optimization. This dual-technology strategy improves robustness, guarantees service continuity across diverse environments, and provides a gradual migration path towards future communication standards [3].

The deployment of these communication technologies, whether individually or in hybrid configurations, forms the backbone of connected mobility services. Their ability to deliver timely, reliable, and context-aware information is particularly critical for applications involving Vulnerable Road Users (VRUs), where even small reductions in latency can significantly improve safety outcomes. This technological foundation enables the transition from reactive to proactive safety strategies, as will be explored in the following section.

1.3 Vulnerable Road Users and the shift to proactive safety strategies

One of the most critical challenges in the design of ITS lies in ensuring the safety of Vulnerable Road Users (VRUs). Unlike vehicle occupants, VRUs do not benefit from the protection of a chassis or passive safety systems, making them highly exposed in the event of collisions. According to the World Health Organization (WHO), pedestrians, cyclists, and motorcyclists together account for more than 50% of global road traffic deaths each year, with particularly high proportions in urban areas where traffic density and multimodal interactions are highest[4]. In Europe, official statistics confirm that in 2024 VRUs, including pedestrians, cyclists, and motorcyclists, comprise nearly 70% of urban fatalities[5]. These numbers underscore the urgency of prioritizing VRU protection as an integral part of cooperative mobility strategies.

1.3.1 VRU Profiles

The ETSI TR 103 300-1 standard classifies VRU into four primary profiles, each presenting distinct vulnerabilities and interaction dynamics with the other elements of the ITS[6]:

1. Pedestrians (VRU Profile 1):
Individuals who walk along or across roads, often without protection from vehicles. This profile includes subcategories particularly at risk, such as children, elderly individuals, and people with disabilities. Pedestrians are frequently found in urban areas, where they are highly exposed to various risk scenarios like intersections, crosswalks, and school zones, given the limited visibility and their unpredictable behavior. When pedestrians are detected, V2X systems can trigger several important behaviors:

2. Cyclists and Personal Mobility Device Users (VRU Profile 2):
Cyclists and users of Personal Mobility devices (e-scooters, e-bikes, hoverboards) represent a rapidly expanding category of VRUs, driven by the growth of micromobility solutions in urban areas. Their speed and maneuverability place them in an intermediate position between pedestrians and motor vehicles, but the absence of physical protection makes them highly vulnerable. Typical conflict scenarios include intersections, lane changes, and overtaking maneuvers, where their limited stability, variable speeds, and reduced visibility to drivers often result in a heightened risk of accidents.

3. Motorcyclists (VRU Profile 3):
Even if motorized, motorcyclists are classified as VRUs due to their exposure to the environment and the lack of protective structures typical of passenger vehicles. Their characteristics includes small dimensions, high speeds, and great maneuverability, all of which allows for other users on the road to notice them less, especially in high-speed scenarios. Therefore, they are exposed to a higher risk of collision, often linked to failures in detection during lane changes or when entering blind spots. Furthermore, when crashes occur, the consequences are usually severe, as motorcyclists are not provided with robust protective mechanisms like vehicles are.

4. Animals (VRU Profile 4):
Animals, whether domestic or wild, are also classified as VRUs, as they pose a significant risk to road safety, especially in rural and peri-urban areas. This profile includes pets, farm animals, wildlife, and service animals that may unpredictably enter the roadway, creating sudden collision risks. Such scenarios are particularly dangerous at night or in low-visibility conditions, while larger wild species such as deer or wild boars can cause severe accidents with both serious injuries and extensive material damage.

This categorization highlights that VRUs are not a homogeneous group: each profile requires specific detection methods, communication strategies, and countermeasures within ITS. While the vulnerabilities differ, the common denominator is the absence of protective structures and the consequent severity of outcomes in the event of collisions.

1.3.2 From Reactive to Proactive Safety

Traditionally, both human drivers and vehicle assistance systems have relied on reactive safety mechanisms: braking when an obstacle is detected, warning when a collision appears imminent, or swerving in response to sudden changes in the road environment. While effective in many conventional traffic situations, this paradigm is inadequate for VRUs. Critical events often unfold in fractions of seconds, with very short time-to-collision (TTC) values. Pedestrians stepping out from occluded areas, cyclists rapidly changing lanes, or animals unexpectedly crossing the road frequently create situations where the reaction time available is insufficient to prevent accidents.

For this reason, ITS research and standardization emphasize a transition toward proactive safety strategies, where potentially hazardous interactions are anticipated and addressed before they become unavoidable. A central enabler of this shift is V2X communication, which expands the detection capabilities of vehicles beyond the line-of-sight of onboard sensors and forms a core component of the layered

architecture defined by [1].

A distinction between passive and active detection is essential in this context. In the first case, mainly efficient in scenarios with high pedestrian density, the presence of VRUs is detected by roadside infrastructure, for example cameras, radar units or thermal sensors, and then communicated to nearby vehicles or traffic control systems. The second case works in the opposite way; it is the VRUs themselves that broadcast their presence, using devices like smartphones, wearable electronics, or dedicated communication units. Thanks to V2P communication, vehicles can receive information about the position and trajectory of nearby VRUs even if they are hidden from view or outside the range of onboard sensors. The best results are achieved when the two approaches work together: if infrastructure sensors are obstructed or weather conditions degrade their performance, connected VRUs can still announce their presence; conversely, if pedestrians or cyclists are not equipped with any device, roadside sensors compensate by spotting them externally. Building on this enhanced detection capability, the next natural step is to make use of the available information not only to respond to what is happening, but to anticipate what might happen next. This shift marks the transition from simple reaction to true prevention: detection, communication, and prediction are no longer separate tasks but part of a single, continuous process that allows vehicles to intervene before a collision course is even established. Risk metrics such as Time to Collision or trajectory conflict probability can be computed and transmitted in real time, allowing vehicles to slow down, change lanes, or issue an early warning before a collision becomes unavoidable. Such anticipatory awareness is particularly effective in high-risk settings such as school zones, at intersections with heavy bicycle traffic, or along rural roads where animals may suddenly cross, significantly increasing the available margin of safety.

This paradigm also introduces the concept of cooperation. By combining onboard perception (radar, LiDAR, cameras) with external information flows from V2X networks and detection infrastructures, ITS can compensate for the limitations of individual sensors, ensuring robustness in adverse weather, low visibility, or occluded environments. Cooperative detection and anticipation thus become the cornerstone of VRU protection, moving safety management from an individual, vehicle-centric approach to a systemic, network-wide strategy.

In this perspective, protecting VRUs requires going beyond incremental improvements of current systems. It calls for a structural integration of proactive, cooperative ITS architectures, where real-time communication, predictive algorithms, and coordinated behaviors work together to reduce fatalities and serious injuries. This approach is aligned with European policy frameworks such as Vision Zero, which aims to eliminate road fatalities by addressing not only vehicle occupants but also the most exposed road users [7].

1.4 Collision Risk as a Key Metric for VRU Protection

VRU awareness in traffic environments can be conceptualised around three primary types of situations. The first corresponds to an imminent collision between a VRU and another road user, which represents a direct and immediate safety threat. The second involves proactive awareness of the VRU's presence, aimed at preventing potential conflicts before they materialise. The third concerns the adaptation of traffic operations when VRUs with special requirements are present, for example children, elderly pedestrians, or persons with reduced mobility, where the system must balance safety with the continuity of traffic flows.

In all these situations, the common element is the need to estimate the likelihood that trajectories of vehicles and VRUs will converge in a conflict point. A collision occurs if both reach this point at the same time, leaving no margin for evasive action. Empirical evidence confirms that such conditions often arise in urban contexts: most pedestrian–vehicle collisions occur at mid-block crossings, while cyclist–vehicle accidents are concentrated at junctions, where multiple trajectories intersect in limited spaces.

To support timely intervention, these qualitative notions of awareness must be translated into quantitative indicators of collision risk. Over time, several metrics have been proposed to measure this risk by combining factors such as relative speed, distance, and temporal margins. Among these, the Time-To-Collision (TTC) metric has become one of the most widely adopted, due to its intuitive interpretation and suitability for real-time computation. TTC provides a direct estimate of the time remaining before two agents, if maintaining their current trajectories and speeds, would reach the same point. This enables both vehicles and infrastructure systems to rank the urgency of threats and to prioritise responses accordingly. Alternative indicators, such as Post-Encroachment Time (PET) or Deceleration Rate to Avoid Collision (DRAC), have also been studied, but TTC remains the most operationally efficient for continuous monitoring in cooperative ITS.

The practical application of collision risk metrics depends on reliable detection and communication of VRU presence. This can be achieved through passive detection, in which roadside infrastructure (e.g., cameras, radar, thermal sensors) identifies a VRU and communicates this information to vehicles, or through active detection, where VRUs broadcast their own status via smartphones, wearable devices, or personal ITS stations. Both modes contribute complementary strengths: passive systems ensure coverage even for unconnected users, while active systems provide early warnings beyond line-of-sight or in occluded environments. By feeding accurate position and trajectory data into TTC calculations, these detection mechanisms enable the timely generation of alerts, adaptive traffic control, and

cooperative avoidance strategies.

In high-risk contexts such as school zones, crosswalks, or areas where VRUs may inadvertently enter vehicle-dominated spaces, the integration of awareness mechanisms and collision risk metrics is particularly critical. Vehicles can adjust speed or trajectory in advance, while VRUs themselves can be warned of approaching hazards. In this way, collision risk metrics become the analytical foundation through which ITS can transform situational awareness into measurable safety interventions, ensuring consistent and proactive protection for VRUs.

Chapter 2

VAM Triggering Framework

2.1 The ETSI VRU System

The ETSI framework dedicated to Vulnerable Road Users (VRUs) defines the functional concept, service behaviour, message set and operational policies of these elements. Therefore, it constitutes a major extension of Cooperative-ITS (C-ITS) aimed at making pedestrians, cyclists, powered-two wheelers and other vulnerable entities first-class participants in the cooperative safety loop. Within the ETSI reference architecture [6], the VRU System is defined as an heterogeneous system of ITS stations interacting with each other to support VRU use cases; in particular, in this thesis we consider the same ones taken into account in the ETSI framework, which corresponds to C-ITS made of a combination of at least one VRUs and at least one ITS-S running VRU-dedicated functions. Its primary goal is to increase situational awareness and to reduce collision risk by enabling reliable detection, timely communication, and short-horizon motion prediction for VRUs in real-time traffic.

The VRU System reuses the ETSI ITS-S architecture defined in Section 1.2, out of which there are some components that are key to its operation, mainly in the Facilities layer, as shown in 2.1.

2.1.1 The VRU Basic Service (VBS)

The core element of the ETSI VRU System is the VRU Basic Service (VBS), an entity defined within the Facilities Layer [8]. The VBS is responsible for generating, receiving, and handling VRU Awareness Messages (VAMs), as well as for managing the internal state and cluster behaviour of VRU stations. Its role is to guarantee that VRUs are consistently represented in the cooperative ITS environment and that their information is disseminated efficiently and reliably.

The internal functional components of the VBS are the following:

- **VRU Basic Service Management** – it acts as the coordinator core of the VBS: it is responsible for coordinating and supervising all the functions required for the correct operation of the basic service dedicated to Vulnerable Road Users. First, it preserves the necessary identifiers and configurations, either received during the ITS-S initialization phase or updated later, which are essential for the correct encoding of VAM data elements, and ensure that they remain consistent with ETSI specifications, as well as with the needs of the VRU. In parallel, it regulates external interactions with other basic services of the ITS architecture in order to ensure the VBS integration within it. The function also controls the activation and deactivation of VAM transmissions, according to the profile parameters defined for each VRU. Finally, it enforces the triggering conditions for VAM generation, while taking into account the network congestion control rules specified in [9].

- **Cluster Management** – extends the VBS logic to support group interactions, managing all procedures related to VRU clustering, including cluster creation, leader election, membership updates, and breakup if conditions for group coordination are no longer met. Therefore, it is designed to handle the organization and coordination of VRUs when group-based communication becomes relevant. It ensures that VAM transmission behavior is aligned with the collective dynamics of VRUs, computing and storing parameters specific to clustered scenarios. Another important aspect is the integration with the VRU Basic Service (VBS) state machine: the cluster management function updates and controls the state of the VBS according to the cluster events that have been detected.

- **VAM decoding** - available only if the VRU is enabled to receive VAMs, it extracts all relevant data elements present in the received standardized message, which are then communicated to the VAM reception management function.

- **VAM reception management** – if present, it handles all the aspects regarding the decoded VAM messages that have been received: checks the relevance according to its current mobility state, determine if the message meets all the necessary security requirements and deletes or stores the data within the VAM in the LDM, according to previous results.

- **VAM encoding** – available only if the VRU is enabled to send VAMs, it encodes the data elements provided by the VAM transmission management function, that comply with VAM specifications. Then, it triggers the VAM transmission to the Network and Transport layer through a specific port highlightning the communication profile.

- **VAM transmission management** – if present, it assembles the data elements complying with the VAM specification and send the constructed message to the VAM encoding function.

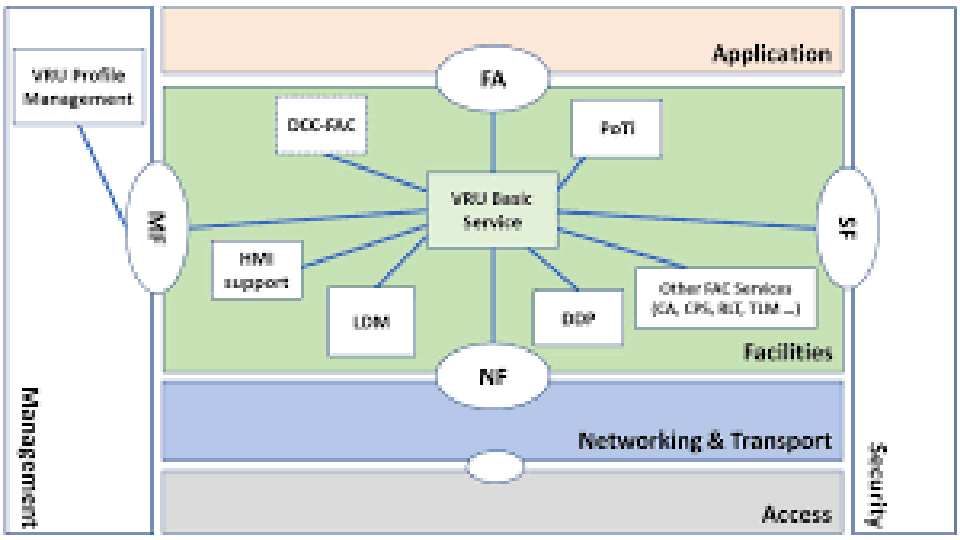

Figure 2.1: Conceptual 4-layer view of the ETSI VRU System (taken from [8]).

Beyond its internal components, the VBS interacts with several other entities within the Facilities layer that enrich its operation and integrate VRU data into the wider cooperative framework, due to the presence of various interfaces. For example, it exchanges VAM data with the Local Dynamic Map (LDM), which is mainly used as a receiving terminal that incorporates decoded messages into a consolidated representation of the traffic environment. Furthermore, the Device Data Provider (DDP) and the Position and Time management (PoTi) are exploited for data collection, respectively, for device status information collected from its local perception entities and for timing and partitioning, ensuring accurate geolocation and synchronization. The VBS also leverages the DCC-FAC interface to adapt message generation rates to prevailing channel conditions, thereby preventing congestion in dense traffic scenarios, while interaction with the Congestion Control function enables the optimization of the available channel usage. Furthermore, the ability of the VBS to interact with Human Machine Interface (HMI) support enables the dissemination of VRU-related warnings to end users through devices such as smartphones or wearables. Finally, the VBS operates in parallel with other Facilities services such as the Cooperative Awareness Service (for CAMs) and the DEN Basic Service (for DENMs), ensuring that VRU information becomes part of a harmonised, multi-source situational awareness.

Another key point to take into account is that the VBS is also able to interact

with other layers of the ITS architecture, both vertical and horizontal, due to the presence of Service Access Points (SAPs):

- NF-SAP - enables the communication with the Network & Transport layer to exchange VAMs with other ITS-S.
- SF-SAP - allows interaction with the Security entity to access security services for VAM transmission and reception.
- FA-SAP - interfaces the VBS with the application layer, exploited if received VAM data are direcly provided to the applications
- MF-SAP - enables the interaction with the Management entity in order to learn all relevant VRU information, such as the role, the profile, the type, if it belongs to a cluster and so on.

Through these interactions, the VBS consolidates its dual role as both a communication enabler and an awareness provider: it ensures the correct transmission and reception of VAMs, but also embeds VRU information into the broader cooperative ITS architecture, where it can be exploited by multiple services to enhance road safety.

2.1.2 VRU Awareness Messages (VAMs)

VAMs are the ETSI-standardised messages used to represent VRUs in the cooperative environment, handled by the VBS at the Facilities layer. VAMs' content is tailored to VRU dynamics and it can vary depending on the VRU profile, but the fields remain the same, defined in [8] and filled each time the originating VRU ITS-S wants to generate and transmit a VAM:

- ITS PDU header;
- Generation delta time;
- Basic container, which provides the `station type` of the originating ITS-S and the latest geographical position;
- VRU High Frequency container, which houses the fast-changing information, like VRU heading, speed and acceleration;
- VRU Low Frequency container, which is characterized by the slow-changing information, like lateral acceleration and yaw rate;
- Cluster information container, which is filled with relevant cluster data, only if the originating VRU is the leader of a cluster;

- Cluster operation container, which can be filled with data related to changes in a cluster state and composition, even if the originating VRU is only a member of a cluster;
- Motion prediction container, which contains data related to the past and future motion state of the originating VRU. In particolar, it is also where all the information related to the Trajectory Interception Probability must be inserted before VAM transmission.

Out of all these fields, only the first four are mandatory, while the others remain defined as optional, because they depend on the nature and operation of the VRU ITS-S originating the message.

The generation of VAMs follows a hybrid logic that combines periodic and event-driven transmissions. Periodic messages are produced at intervals that adapt dynamically to the mobility state of the VRU and to the surrounding traffic density, typically ranging from 100 ms to 5 s. Event-driven transmissions complement this baseline by ensuring timeliness whenever state changes or contextual conditions may influence safety. Representative triggers include variations in kinematic parameters, role transitions within the VRU Basic Service (VBS), or cluster-related events. The details of these conditions will be presented in the following section, while here it is sufficient to underline that the aim is to guarantee that a VRU can promptly announce its presence and status whenever this information is most relevant for cooperative safety. Transmission is further regulated by congestion control mechanisms—DCC for ITS-G5 and RAT-specific resource allocation for C-V2X—so as to preserve reliability even under high-density deployments.

After the generation, VAMs are disseminated locally using broadcast or anycast mode, with a geographic scope tailored to the specific context, such as an intersection approach or a pedestrian crossing. On the receiving side, vehicles and other road station units integrate VAMs both with other standardized messages, as well as with onboard or roadside perception, increasing the level of their situational awareness. Eventually, this information can be exploited to compute different risk metrics, such as TTT, and to trigger cooperative warnings or adaptive maneuvers. In scenarios with multiple VRUs, clustering mechanisms further optimise dissemination: a single elected leader can represent the group, reducing the number of VAM transmitted, limiting channel occupancy while maintaining precise awareness.

The clustering logic is an integral part of the VBS and defines three possible operating states for each VRU: active-standalone, where the device transmits its own VAMs at a rate adapted to dynamics and channel conditions; passive member, where the VRU remains mostly silent while being represented by the cluster leader; and cluster leader, responsible for broadcasting representative information about the group footprint and membership. Cluster formation and maintenance are managed

through configurable parameters such as numCreateCluster, maxClusterDistance and timeClusterContinuity. Leader election typically favours stability, privileging centrality and coherence of velocity, while mandatory fallback triggers ensure safety: for example, if a leader is lost, members are required to immediately resume VAM transmission until a new leader is elected.

These mechanisms find practical application in a variety of traffic scenarios. At an unsignalised pedestrian crossing, a VRU device may increase its VAM generation frequency as it approaches the road, allowing vehicles to fuse this information with perception and react in advance when TTC drops below a critical threshold. In the case of a cyclist platoon navigating an urban roundabout, clustering allows a single leader to represent the group, so that vehicles can anticipate collective movement and plan safe entry and exit manoeuvres. Similarly, in non-urban contexts, roadside units equipped with sensors can issue VAMs on behalf of non-connected VRUs, such as wildlife crossing at night, thus extending awareness beyond line-of-sight and providing early warnings to approaching vehicles.

Overall, the ETSI VRU System operationalises cooperative protection of vulnerable users by codifying message generation, dissemination logic, and clustering behaviours. Together, these mechanisms fill the gap that limits purely reactive driver assistance systems, especially in occluded or highly dynamic environments. The approach is agnostic with respect to the access technology and fully aligned with the ETSI security and management framework [1], thereby ensuring interoperability. Most importantly, it provides the foundation for proactive safety strategies based on risk prediction metrics such as TTC, enabling earlier conflict detection and cooperative mitigation in line with the European Vision Zero objectives.

Table 2.1: Comparison of ETSI awareness and notification messages for cooperative safety.

Feature	**CAM**	**DENM**	**VAM**
Primary purpose	Cooperative awareness of vehicles/RSUs	Event/incident notification (hazards, obstructions)	Cooperative awareness of VRUs (individual or clustered)
Generation	Periodic (100 ms–1 s) with adaptation	Event-driven (with validity/expiry)	Hybrid: periodic (100 ms–5 s) + event-driven
Core content	Kinematic state, heading, dimensions	Event type, location, cause, duration	VRU profile, position, kinematics, (optional) cluster and prediction
Producer	Vehicles, RSUs	Vehicles, RSUs	VRU devices (smartphone, wearable, tag), cluster leader
Facilities service	CA Basic Service	DEN Basic Service	VRU Basic Service (VBS)

2.2 Triggering conditions

The generation of VRU Awareness Messages (VAMs) within the ETSI framework is governed by a set of triggering conditions defined in ETSI TS 103 300-3 [8]. Unlike a purely periodic broadcast, which would overload the channel and provide limited added value, triggering conditions ensure that a new VAM is transmitted only when the information carried is sufficiently fresh or safety-relevant. The VRU Basic Service (VBS) evaluates these conditions every $T_{\text{CheckVamGen}}$ (nominally $\leq$ 100 ms), subject to the minimum and maximum generation intervals $T_{\text{GenVamMin}}$ and $T_{\text{GenVamMax}}$. This balance guarantees that VRUs remain visible in the cooperative environment without creating redundant traffic.

A new individual VAM is therefore generated as soon as one of the following seven conditions is satisfied:

1. **Time since last VAM.** If the elapsed time since the previous individual VAM exceeds $T_{\text{GenVamMax}}$ (maximum generation period, typically set to 1–5 s), a new message must be produced. This "keep-alive" mechanism guarantees that each VRU remains periodically visible to nearby ITS stations, even when it is stationary or its dynamics do not change.

2. **Position change.** When the current reference position differs from that in the last transmitted VAM by more than minReferencePointPositionChangeThreshold (typically $\sim$4 m), a new VAM is generated. This condition is particularly relevant for pedestrians, whose continuous small displacements would otherwise not be communicated. By setting an appropriate threshold, the system ensures that only meaningful changes are transmitted, avoiding excessive signalling due to GNSS jitter.

3. **Speed change.** If the ground speed differs from the value in the last VAM by more than minGroundSpeedChangeThreshold (around 0.5 m/s), a new message is issued. This captures transitions such as a pedestrian starting to run, a cyclist accelerating after a stop, or a scooter slowing down when approaching a crossing. It allows surrounding vehicles to update their situational awareness in response to changes that significantly alter collision risk.

4. **Heading/orientation change.** A new VAM must be transmitted when the orientation of the ground-velocity vector changes by more than minGroundVelocityOrientationChangeThreshold. ETSI suggests a value of 4°, though in practice slightly higher values (7–10°) are often used to mitigate false triggers caused by GNSS noise. This condition is essential for scenarios such as a pedestrian suddenly changing direction to cross the street, or a cyclist turning into an intersection.

5. **Trajectory interception probability change.** When the locally estimated trajectory interception probability with respect to other road users increases or decreases by more than minTrajectoryInterceptionProbChangeThreshold (default 10%), a VAM is triggered. This mechanism extends beyond kinematic changes, directly incorporating predicted risk of collision. In practice, it ensures that other ITS stations are informed whenever the likelihood of a conflict involving the VRU changes significantly. Importantly, the ETSI standard does not specify how this probability should be calculated, leaving it to implementers. The present thesis focuses on filling this gap by developing a method to compute the collision probability in real time, starting from the Time-to-Collision (TTC) metric, and by evaluating how it can be applied to satisfy this triggering condition.

6. **Cluster join intent.** When a standalone VRU decides to join a cluster, it must first broadcast a VAM containing a cluster operation container. This ensures that neighbouring stations are aware of the membership change before the VRU switches to a passive role and stops transmitting individually. Without this trigger, information about the VRU could be lost until the new cluster leader updates its VAM.

7. **Proximity to other road users.** If the VRU station detects the simultaneous presence of one or more road users within the minimum safe lateral, longitudinal, and vertical distances (MSLaD, MSLoD, MSVD), an immediate VAM is generated. This condition captures high-risk situations, such as a pedestrian walking too close to moving vehicles or a cyclist squeezed between cars, and ensures that other participants are alerted in time. To avoid message storms in dense crowds, this trigger is moderated by redundancy mitigation mechanisms.

Together, these conditions balance the trade-off between communication efficiency and safety responsiveness. Kinematic thresholds (position, speed, heading) capture direct changes in VRU motion, while event-based triggers (cluster operations, proximity) ensure correct representation of contextual dynamics. The time-based condition maintains periodic visibility, and the collision probability trigger explicitly ties message generation to predictive safety assessment. Among them, the fifth condition is the most challenging to implement, as it requires a robust definition of collision probability. Addressing this open issue forms the core objective of this thesis.

2.3 Collision Probability as a Triggering Parameter

One of the most demanding aspects of cooperative safety systems is the ability to estimate collision probability in real time. This challenge is particularly critical in scenarios involving Vulnerable Road Users, whose movements are less predictable and whose exposure in case of collision is absolute. Unlike vehicles, which tend to follow structured trajectories, VRUs may abruptly change direction or enter the roadway without warning. As a result, the time available to detect a VRU, assess the risk, and trigger a VAM is often reduced to fractions of a second.

The ETSI specification recognizes this urgency by defining strict timing constraints. The VRU Basic Service must evaluate triggering conditions every $T_{\text{CheckVamGen}} \leq 100\,\text{ms}$, while the construction of a VAM must complete within $T_{\text{AssembleVAM}} \leq 50\,\text{ms}$ [8]. Furthermore, VAM generation is bounded by minimum and maximum intervals ($T_{\text{GenVamMin}}$ and $T_{\text{GenVamMax}}$) to avoid both message bursts and excessively long silences. These requirements imply that collision probability must be estimated in just a few milliseconds, continuously, and on resource-constrained devices.

From these constraints and from the operational realities of VRU protection, four main requirements for real-time probability estimation can be identified:

- **Low latency.** Estimation must run within the strict timing bounds defined by ETSI, ensuring that collision risk information is available before the next triggering check. Delays of even tens of milliseconds can render the message ineffective in high-risk VRU scenarios.
- **Uncertainty handling.** VRU localisation is often based on noisy GNSS or smartphone sensors, subject to occlusions and multipath effects. Robust probability estimation must therefore incorporate uncertainty in position, speed, and heading, rather than relying on idealised measurements [10].
- **Computational efficiency.** VRU devices such as smartphones or wearables have limited processing capacity compared to vehicle ECUs. Methods suitable for deployment must be lightweight and energy-efficient, avoiding heavy sampling schemes like Monte Carlo simulations while maintaining acceptable accuracy.
- **Predictive accuracy.** The triggering condition refers to a trajectory interception probability, which requires short-horizon prediction. Estimation must therefore forecast whether trajectories will intersect in space and time, not merely evaluate the current state.

The importance of these requirements is highlighted by some of the latest research on collision risk estimation: for example, [11] recently proposed a Machine

Learning based framework for collision risk estimation in V2X environments. The approach combines decision tree models and random forests in order to be able to efficiently process spatial and temporal features, enforcing how the factors defined above are fundamental to achieve real-time operation while maintaning interpretability and robustness.

Despite the advances done in the field, there is still an important gap when it comes to precisely implement the ETSI specifications for the triggering condition object of this work: indeed, it requires that a VAM should be triggered whenever the estimated collision probability changes by at least 10% compared to the last broadcasted value, but it does not provide a method to compute this probability. This thesis directly addresses this gap: by deriving the collision probability from the TTC metric, the aim is to provide a method that is accurate, lightweight, and robust to uncertainty, while fully compliant with ETSI's real-time constraints. However, before developing the methodology, it was imperative to choose the best branch of existing collision risk estimation methods to start from: therefore, in the next section all of them are thoroughly described and analysed, highlightning the reason why it was decided to follow a TTC-based approach, discarding all the others.

2.4 Collision Risk Estimation Methods

ITS increasingly rely on quantitative collision risk estimates to support proactive safety strategies and satisfy triggering requirements. In cooperative settings, these estimates must be computed under tight real-time constraints, must be able to tolerate imperfect and heterogeneous sensing, while staying interpretable so that human stakeholders and certification bodies can understand the need for eventual interventions. From the system perspective, risk computation is a scalar quantity, obtained through the application of geometry and kinematics, coupled with intent and uncertainty, that encapsulates different aspects: perception, prediction, and decision. Therefore, it is comparable and can be easily transformed into a probability to be compared with different thresholds.

This is the reason why the collision probability becomes a key concept in this framework. Traditionally, it is defined as the likelihood that two road users, like a vehicle and a pedestrian, will end up colliding if they both keep their current speed and direction. Therefore, it constitutes a clear and intuitive indicator of risk, widely exploited since it can be updated step by step using various motion data, such as distance, speed, acceleration and trajectories of the entities involved. In practical terms, collision probability acts as a bridge between perception, prediction and decision: it reduces the complexity of traffic interactions into a single value that can be compared against thresholds, in order to judge the level of risk, decide

when to issue warnings, and trigger awareness messages when the chance of impact becomes dangerous. By proactively highlighting possible collisions early, ITS can then perform more detailed checks and take on preventive safety measures, such as trajectory interception.

This chapter surveys the main families of collision probability estimation methods and positions them with respect to the requirements of ETSI cooperative services and message triggering. We first outline a taxonomy of partially overlapping approaches, that differs between each other in terms of modeling assumptions, uncertainty treatment, computational cost and interpretability. Then, we explain more in detail the probabilistic forecasting and learning-based models, as they are the more divulged, studied and where more advancements have been made, before discussing their limitations in the context of ETSI-compliant triggering. Finally, we motivate the choice of implementing a TTC-based probability mapping, tailored for real-time VAM generation.

Deterministic kinematics and surrogate safety measures

A long-established line of work computes deterministic indicators directly from relative geometry and motion, without explicit uncertainty modeling. Canonical measures include Time to Collision (TTC), Post-Encroachment Time (PET), Time-to-Intersection / Time-to-Conflict, Deceleration to Avoid Collision (DRAC), minimum distance, and braking reserve. They are attractive because they are simple, interpretable, and cheap to compute at cycle time (order of micro–milliseconds). In cooperative ITS, these indicators can be evaluated using fused local and remote (V2X) state estimates and used to trigger warnings or message transmissions. However, determinism makes them sensitive to noise/occlusions and to intent changes (e.g., sudden yielding), which can cause false positives/negatives unless combined with filtering or hysteresis. Comprehensive reviews compare these "surrogate safety measures" (SSMs) and discuss their operational ranges and pitfalls, especially in dense urban interactions.

In VRU-centric cooperative services, TTC and related conflict-time indicators already appear within "collision risk analysis" functions that fuse motion prediction with confidence levels to determine whether evasive actions or messaging are warranted.

Probabilistic risk from state uncertainty propagation

When sensor and intent uncertainty cannot be ignored, deterministic indicators are extended to probabilistic risk by propagating state covariances or sampled uncertainties and integrating the likelihood of future overlap in space–time. Typical pipelines maintain Gaussian (or Gaussian-mixture) beliefs over actor states, propagate them

with process/measurement models, and compute (i) the probability that relative distance falls below a safety margin within a horizon, (ii) the probability that TTC drops below a threshold, or (iii) chance constraints that bound collision probability along candidate trajectories. This yields risk as a probability (or upper bound) and is naturally compatible with ETSI-style triggering rules expressed in probabilistic terms. Pros include principled uncertainty handling and tunable confidence; cons are model dependence and potentially higher online cost (sampling, convolution, or numerical quadrature). Overviews and tutorials cover Bayesian/Monte-Carlo propagation and chance-constrained formulations for autonomous driving.

Set-based safety and reachability analysis

Reachability-based methods represent one of the most rigorous families of approaches for safety assessment in autonomous driving and robotics. The key principle is to compute the set of all states that a system, together with its possible obstacles, can reach within a given time horizon. If these so-called reachable sets intersect with unsafe regions of the state space, a collision is deemed possible. The classical formulation is based on Hamilton–Jacobi (HJ) partial differential equations, which allow the derivation of Backward Reachable Tubes (BRTs) and value functions that provide strict worst-case safety guarantees. These certificates are mathematically elegant and provably correct, making HJ reachability one of the strongest theoretical tools available for collision risk analysis [12].

Over the years, however, several works have highlighted the computational burden associated with this methodology. The dimensionality of the state space grows rapidly when considering realistic models of vehicles, pedestrians, and other VRUs, leading to the well-known "curse of dimensionality." For example, while HJ reachability can be tractably solved for four-dimensional systems (e.g., two-dimensional position and velocity), extending to higher dimensions or to multi-agent scenarios becomes prohibitive without substantial approximations. Recent research has explored hardware acceleration to mitigate this limitation: [12] demonstrated that FPGA-based implementations can achieve real-time updates of reachable sets for simplified vehicle dynamics, opening the door to embedded applications where computational resources are constrained.

To improve scalability, alternative set representations have been proposed. Althoff and colleagues pioneered the use of zonotopes and ellipsoids to over-approximate reachable sets in traffic scenarios, enabling faster propagation of reachable tubes with reduced complexity [13]. Although these approximations inevitably introduce conservatism, they make it possible to evaluate safety margins for autonomous vehicles in more complex environments. More recent advances, such as hybrid zonotopes for nonlinear feedback systems, aim to reduce over-conservatism while maintaining scalability [14].

Despite these advances, the applicability of reachability-based methods to ETSI-oriented VRU protection remains limited. In dense urban traffic, the number of interacting agents is large and the required update frequency is high, which makes online computation of reachable sets unrealistic without heavy approximations. Moreover, conservative bounds often trigger false alarms, which can flood the communication channel with unnecessary VAMs, contrary to the lightweight philosophy of ETSI's triggering framework. For this reason, in practical ITS deployments, reachability is typically not employed as the primary decision-making tool. Instead, it is sometimes envisioned as a "safety filter" that can wrap around faster but less rigorous predictors, or as a certified override for emergency situations where strict guarantees are needed [12, 13].

In summary, reachability analysis provides unmatched theoretical rigor and worst-case guarantees. It is therefore extremely valuable as a benchmark for validating lighter methods or as a fallback guard in cooperative safety systems. Nevertheless, due to its high computational cost and the conservatism of its approximations, it remains unsuitable as the main method for real-time VRU collision risk assessment in ETSI-compliant scenarios. This motivates the choice of simpler TTC-based approaches in this thesis, which strike a more favorable balance between accuracy, interpretability, and computational feasibility.

Interaction and intent-aware prediction with risk scoring

Another family of approaches to collision risk estimation is based on predicting the intent of multiple agents and generating interaction-aware trajectory ensembles. Instead of directly propagating worst-case reachable sets, these methods first attempt to infer whether a pedestrian intends to yield or cross, or whether a vehicle plans to turn or continue straight. Once intent hypotheses are available, a distribution of possible trajectories is propagated forward, and collision likelihood is assessed over this ensemble.

The concrete formulations vary widely. Early work relied on rule-based interaction models, encoding heuristic assumptions about how agents would react in negotiation scenarios. More sophisticated approaches draw on game theory, where each agent is modeled as a rational player optimizing its utility subject to the predicted behaviour of others. Another line of research uses partially observable Markov decision processes (POMDPs), treating intent as a latent variable to be inferred online from partial observations. In all cases, the central idea is to make the risk estimator "interaction-aware" rather than simply extrapolating motion independently for each actor.

The resulting risk score can take different forms. It may be the expected collision probability obtained by weighting across discrete intent hypotheses; it may reflect a worst-case criterion by taking the maximum collision likelihood over the set

of plausible behaviours; or it may adopt a quantile-based approach, identifying high-risk but low-probability outcomes. Such flexibility allows these models to capture the richness of urban negotiation, particularly in unsignalized crossings or shared spaces where VRU behaviour is highly context-dependent.

At the same time, these methods face notable challenges. They require strong priors on behaviour patterns—often learned from data—and careful calibration of model parameters to avoid overconfidence in intent predictions. Moreover, their computational cost grows with the number of interacting agents and the size of the intent space. Recent studies, such as the interaction-aware learning model described in [15], demonstrate how these methods can be embedded into decision-making pipelines for autonomous driving. However, despite their promising performance, their adoption in ETSI-oriented VRU safety systems remains limited, since intent-aware models are harder to validate under safety-critical requirements and can suffer from cascading errors when the inferred intent is wrong.

Learning-based and data-driven risk estimation

Data-driven methods learn collision scores or probabilities directly from features or raw trajectories. Classical machine learning (e.g., decision trees, random forests, gradient boosting) offers interpretable feature importances and low-latency inference on edge devices; deep learning (LSTM/GRU, graph neural networks, transformers) captures rich spatio-temporal dependencies but is more computationally demanding and less transparent. In VRU scenarios, recent work shows hybrid pipelines in which a lightweight predictor filters high-risk interactions (e.g., turning vehicles) before a classifier/regressor estimates collision probability, striking a balance between robustness and real-time deployability. Reviews of criticality/risk metrics and intelligent-vehicle prediction summarize the landscape and evaluation practices.

Occupancy and field-based risk maps

Another strand aggregates uncertainty over space (rather than over discrete actors) via dynamic occupancy grids or Bayesian "risk fields," then integrates overlap between an ego occupancy tube and others' predicted occupancy. This spatial view is robust to identity switches and scales to many agents at the cost of discretization and diffusion artifacts. It is particularly relevant for radar- and camera-heavy stacks and for fusing cooperative perception into a common grid.

Hybrid architectures

In practice, production-grade stacks combine the above: e.g., kinematic SSMs as fast pre-filters, probabilistic bounds for thresholding, learned components for

ranking or resolving ambiguous cases, and set-based guards for certified safety. Cooperative ITS adds further hybridisation by fusing local perception with V2X messages and using confidence-weighted data fusion before risk computation, as already envisaged in the VRU Basic Service's collision risk analysis and manoeuvre coordination functions.

Across families, five axes drive suitability for ETSI-compliant triggering: (i) latency (cycle-time execution on embedded or edge hardware), (ii) uncertainty handling (sensitivity to noisy, partial, or occluded data), (iii) interpretability and auditability (supporting threshold selection and diagnostics), (iv) operational coverage (urban VRU interactions, multi-agent density), and (v) compatibility with probabilistic triggers (e.g., thresholds on collision probabilities or their increments). Deterministic SSMs excel on (i) and (iii) but need augmentation for (ii)–(v); probabilistic and chance-constrained methods score well on (ii) and (v) with moderate cost; reachability offers strong guarantees for (ii) and (iv) at higher computational expense; learning-based models can improve (iv) and practical accuracy if their training data and validation match the deployment domain, and if their runtime is bounded (e.g., via two-stage hybrids).

Finally, within the ETSI VRU context, existing functional allocations (motion prediction, confidence-weighted perception fusion, collision risk analysis, and avoidance/coordination) provide natural integration points for all families above and motivate the TTC-centered mapping to probabilistic triggers pursued later in this chapter.

2.5 Probabilistic Forecasting

Probabilistic models constitute one of the most rigorous families of methods for estimating collision probability. The underlying assumption is that neither sensor measurements nor vehicle dynamics can be known with absolute certainty: every position, velocity or orientation measurement is affected by noise, and the future evolution of a trajectory is subject to unpredictable variations. Deterministic indicators such as Time-to-Collision are therefore only approximations, which may lead to inaccurate or misleading results when uncertainty is significant. Probabilistic forecasting tackles this issue by representing the state of each road user as a distribution rather than a single value, and by propagating this uncertainty forward in time. The probability of collision is then defined as the likelihood that the stochastic trajectories of two entities intersect within a given prediction horizon.

The most established approach to implement this principle is the Monte Carlo simulation. Instead of computing a single trajectory, the algorithm generates a large number of random samples from the uncertainty distributions that describe the state of the host vehicle and the surrounding targets. Each sample represents

one possible realization of reality, obtained by perturbing variables such as position, speed, acceleration, heading angle or yaw rate according to their statistical error models. All these trajectories are then propagated through a motion model, and for each of them the algorithm checks whether an overlap between vehicles occurs. The collision probability is finally estimated as the ratio between the number of simulated collisions and the total number of samples. In this way, the method directly approximates the true distribution of outcomes without restrictive assumptions on noise or dynamics.

An additional strength of the Monte Carlo paradigm is its flexibility. The uncertainty models can be tailored to the specific sensor suite employed, be it radar, lidar, or camera-based perception. Moreover, the geometry of vehicles can be represented more realistically than in simplified point-based models, for instance using rectangular or polygonal approximations of the occupied area. This allows the algorithm to distinguish between different collision configurations: a frontal impact, a side collision, or a near miss scenario can all be captured within the same probabilistic framework. In practice, this enables the system to compute not only whether a collision is possible, but also how likely it is under the current uncertainty conditions. As demonstrated in recent studies, this capability leads to improved robustness, with a much lower false alarm rate compared to classical approaches based solely on Kalman filtering and Gaussian noise assumptions [16].

The usefulness of such models in the automotive field is evident. By accounting for realistic noise distributions and by explicitly propagating uncertainty, they can provide a more faithful estimate of risk in critical scenarios such as rear-end crashes, intersection collisions or encounters with oncoming traffic. From the perspective of advanced driver assistance systems and autonomous driving research, probabilistic forecasting methods are appealing because they reduce the number of false positives, a crucial requirement when automated interventions such as emergency braking are triggered. Furthermore, they offer a forward-looking perspective: instead of assessing risk only at the present instant, they predict how collision probability may evolve over the next few seconds, giving valuable information for tactical planning.

2.6 Learning-Based Methods and Data-Driven Models

A third family of approaches for collision probability estimation is based on the use of learning-based and data-driven models. While analytical formulations compute probability starting from explicit equations, and probabilistic methods approximate it through sampling processes, this group of methods learn patterns of risk directly from data, exploiting the availability of large data sets. The main concept is that the mechanisms leading up to a collision may be too complex to describe them only

with formulas, while they can be deduced if a model is exposed to a sufficiently large number of examples. Therefore, it implies applying Machine Learning in this domain, transforming in a way to approximate the mapping between the observable state of the traffic scene and the collision probability itself, even in highyl dinamic and uncertain situations.

Out of the many research done in this direction, a first representative study is the work of Altché and de La Fortelle [17], who applied Long Short-Term Memory (LSTM) networks to predict vehicle trajectories on highways. LSTMs are a type of recurrent neural network designed to capture temporal dependencies, making them suitable for modelling driving behaviours evolution over time: by training the network on trajectory data, the authors were able to predict future positions. Then, these forecasts were exploited to estimate the possibility of a collision between the two vehicles. This is an important research because it demonstrated that neural networks could successfully capture the non-linear and sequential nature of traffic interactions, offering more flexible risk estimation than rule-based approaches.

Building on this foundation, researchers have explored architectures capable of modelling both individual trajectories and the interactions among multiple agents. [18] introduced the Graph-based Trajectory Predictor (GTP), which represents a traffic situation as a graph, where each vehicle or pedestrian is a node, while the edges represents the interactions: the structure allows the model to study spatial and temporal dependencies across several entities at the same time. Therefore, the system can predict trajectories in dense or urban contexts, where the relative position and behaviour of neighbours strongly influence collision risk. The advantage of this approach lies in its ability to represent traffic as a network of relationships rather than as isolated time series, leading to more realistic and interaction-aware estimates of collision probability.

Another relevant line of research focuses on multimodal perception and generative models. Recently, Salzmann et al. proposed Trajectron++ [19], a deep learning framework that integrates heterogeneous different kind of inputs, such as road maps, agent dynamics and environmental constraints, into a model that generates multiple possible future trajectories instead of a single deterministic forecast. This feature is particularly valuable for collision risk estimation, because in the majority of cases, there are several possible outcomes, each with the same probability of becoming true, but each one carries with them a different associated risk. Trajectron++ explicitly models this uncertainty, providing relevant information that can be exploited for many safety applications.

The development of large-scale datasets has played a crucial role in supporting these approaches. Zhan et al. created the INTERACTION dataset [20], which contains detailed recordings of highly interactive driving scenarios such as intersections, roundabouts and merges. This dataset was designed to capture both cooperative and adversarial behaviours, reflecting the complexity of real traffic. It

has since become a benchmark for evaluating learning-based models that estimate collision probability. In parallel, Caesar et al. presented nuScenes [21], a multimodal dataset collected in urban settings that includes lidar, radar, camera images and high-definition maps. Datasets of this kind are indispensable for training and validating data-driven models, since they provide the diversity of conditions and behaviours needed for generalisation.

Despite their advantages, learning-based approaches face important challenges. One of the most discussed issues is the ability to generalise beyond the specific scenarios on which the models were trained. A survey by Kiran et al. [22] emphasises that deep learning models for autonomous driving must often be complemented by reinforcement learning or hybrid approaches that incorporate prior knowledge about physics and safety. For example, formal safety guarantees can be layered on top of data-driven models to ensure that even when the network encounters novel situations, the system does not behave unpredictably. This integration of learning with control theory is seen as a promising way to balance flexibility with reliability in safety-critical applications.

In summary, learning-based and data-driven approaches offer a powerful and flexible framework for estimating collision risk. Recurrent models such as LSTMs capture temporal patterns in trajectories; graph-based networks explicitly model interactions among agents; and generative models like Trajectron++ account for uncertainty and multimodality in future behaviour. Supported by datasets such as INTERACTION and nuScenes, these methods provide probabilistic outputs that can inform high-level planning and decision-making. However, their reliance on data availability and the difficulty of guaranteeing robustness across unseen environments highlight the need for hybrid solutions that combine the adaptability of data-driven learning with the formal guarantees of model-based approaches.

2.7 Limitations in the Context of ETSI-Compliant Triggering

The probabilistic forecasting methods described in Section 3.2 and the learning-based approaches outlined in Section 3.3 represent powerful families of models for collision risk estimation. They are capable of capturing uncertainty in a rigorous way, of exploiting large amounts of data, and of providing rich predictions that extend beyond the instantaneous state. These strengths make them particularly attractive in domains such as autonomous driving or advanced driver assistance systems, where multi-second forecasts and robust decision-making are essential. Nevertheless, when considered in the specific framework of ETSI-compliant triggering for Vulnerable Road Users, they present limitations that are difficult to reconcile with operational requirements.

A first limitation is the computational cost. Probabilistic forecasting methods, and in particular those based on Monte Carlo simulation, require thousands of trajectory samples to produce statistically reliable estimates. Similarly, learning-based models — especially those built on recurrent or deep neural networks — involve complex architectures that demand significant processing resources. While such requirements may be acceptable for high-performance autonomous driving platforms or for offline analysis, they are impractical for on-board units constrained by strict latency and power budgets. ETSI specifications mandate message generation and evaluation on the order of tens of milliseconds, a regime that leaves little room for heavy computation.

A second limitation concerns the temporal nature of the output. Both probabilistic forecasting and learning-based models typically provide a distribution of collision probabilities over a prediction horizon. This forward-looking representation of risk is valuable for tactical planning, as it allows the system to anticipate how risk may evolve in the near future. However, it does not align with the ETSI triggering logic, which requires a single scalar probability to be evaluated frame-by-frame. In practice, the ETSI framework mandates that the transmission of VRU Awareness Messages (VAMs) be triggered whenever the instantaneous collision probability changes by at least 10% relative to the last transmission. Horizon-based forecasts cannot be directly compared against such thresholds without introducing additional heuristics or compressions that compromise interpretability.

These two limitations, computational cost and horizon-based formulation, explain why the approaches previously described, despite their accuracy and theoretical appeal, are not directly suitable for cooperative ITS safety communication. They highlight the need for methods that are lightweight, interpretable, and capable of delivering instantaneous collision probability values. This observation motivates the exploration of alternative approaches that build on simpler time-related metrics, such as Time-to-Collision, which naturally lend themselves to frame-by-frame evaluation and ETSI-compliant triggering.

2.8 Motivation for a TTC-based approach

The considerations outlined in this chapter clearly indicate that, despite their analytical rigor and predictive accuracy, neither probabilistic forecasting nor learning-based approaches are well suited for the requirements of ETSI-compliant triggering. Their reliance on computationally intensive processes and their tendency to formulate risk as a distribution evolving over a prediction horizon make them difficult to reconcile with the need for instantaneous, frame-by-frame estimates of collision probability. For this reason, the present work concentrates on the first category of methods: those based on time-related metrics.

Time-related indicators represent one of the most established traditions in traffic safety research. The underlying rationale is straightforward: the likelihood of collision depends critically on the time available before a potential conflict materializes. By translating spatial and kinematic relationships into temporal margins, these metrics provide an intuitive and operationally meaningful representation of risk. Over the years, several such indicators have been proposed, each with its own strengths and limitations.

The Time-to-Collision (TTC) is by far the most widely used. Its origins can be traced back to early research on collision avoidance in both the aviation and automotive domains. Initial formulations were introduced in the mid-twentieth century, notably by Gibson and Cooks in 1938 [23], as part of efforts to enhance safety systems by incorporating human factors into collision prediction. A further milestone was provided by Lee in 1976 [24], who emphasized the role of visual perception in estimating time-to-contact and formalized the link between temporal margins and human reaction in driving tasks. These foundational contributions established TTC as a simple yet powerful tool, making it one of the most widely adopted temporal metrics in Intelligent Transport Systems (ITS).

Formally, TTC expresses the time remaining before two entities would collide if they continued along their current trajectories with unchanged speeds. This property makes it a direct, real-time measure of collision risk: a high TTC indicates a safe situation, while a low TTC reflects an urgent and critical one where the probability of impact is significantly higher. Its ability to condense complex interactions into a single scalar quantity has made it a cornerstone in both traffic safety analysis and the design of driver assistance systems, as well as a valuable instrument for supporting proactive safety strategies aimed at accident prevention.

Closely related to TTC is the Post-Encroachment Time (PET), which measures the temporal gap between one road user leaving a conflict point and another entering it. PET is widely used in retrospective safety assessments, for example in evaluating intersection conflicts, but by definition it is not predictive: it quantifies events that have already occurred rather than anticipating imminent risk. Another indicator often discussed is the Deceleration Rate to Avoid Collision (DRAC), which expresses the minimum constant deceleration required for a vehicle to avoid collision under the current conditions. While informative for studying driver response, DRAC depends heavily on assumptions about braking capabilities, road conditions, and human reaction times, which limits its robustness in real-time cooperative contexts. Beyond these, several other measures have been proposed, such as distance-to-conflict-point metrics or composite indices that combine speed, acceleration, and trajectory information. Although useful in specific analytical frameworks, these indicators tend to be either less interpretable or less directly related to instantaneous collision probability.

When these alternatives are compared, TTC consistently emerges as the most

suitable candidate for the purpose of this thesis: the metric can be directly computed from easily measurable state variables, with negligible computational cost and without the need for scenario-specific calibration. Its main advantage is given by its intrinsic balance between the computation simplicity and its power in efficient predictions. It is inherently forward-looking, rather than retrospective like the PET metric is. Unlike DRAC, it does not rely on uncertain behavioral or physical assumptions about braking dynamics. Most importantly, TTC is naturally suited for frame-by-frame evaluation, producing at each time instant a scalar value that reflects the imminence of a potential collision: this is precisely the property required by the ETSI framework, which states that VRU Awareness Messages should be triggered based on frame-by-frame changes in collision probability.

Furthermore, the widespread use of TTC in both research and practice reinforces its suitability. It has been adopted in numerous traffic safety studies as a surrogate measure for accident risk, and its strong empirical correlation with collision occurrence has been repeatedly demonstrated. It also features prominently in the design of advanced driver assistance systems, where thresholds on TTC are often used to activate warning or braking functions. This established role provides not only methodological justification, but also practical validation: TTC is a metric that has been proven effective in operational systems, and its interpretation is transparent for both engineers and end users.

For these reasons, TTC is chosen as the foundation for the method developed in this thesis. By mapping TTC values into collision probabilities, it becomes possible to obtain a lightweight, interpretable, and ETSI-compliant indicator that can be updated at every frame. This approach combines the intuitive appeal and computational efficiency of time-based metrics with the formal requirement of producing a probability suitable for triggering. The following chapter will detail how this mapping is constructed and how it integrates with the overall logic of cooperative safety communication.

Chapter 3

VaN3Twin

3.1 Environment Overview

This thesis has been carried out on VaN3Twin, the first open-source, full-stack Network Digital Twin (NDT) framework to simulate the coexistence of multiple V2X communication technologies, developed by Politecnico di Torino [25] and fully available on GitHub [26]. The framework is an extension of another tool, presented by Politecnico di Torino as well: Multi-Stack VANET framework for ns-3 (ms-van3t), an integrated multi-stack framework specifically developed to enable virtual validation of V2X communication and cooperative ITS services. As highlighted in [27], the framework is capable of efficiently managing vehicle mobility and connectivity by using two other well-known simulation tools, Network Simulator 3 (ns3) and Simulation of Urban Mobility (SUMO), and supporting the integration of hardware in the loop (HIL). In particular, ms-van3t is the first project that combines vehicle emulation through SUMO mobility scenarios, the possibility to use various communication technologies through ns3, and a complete implementation of the ETSI C-ITS stack, together with another set of features, filling a gap in the current simulation ecosystem by providing a holistic integration of all these factors within a single environment. Some of the main features of ms-van3t are briefly explained below:

- **Full ETSI C-ITS stack**
  One of the distinctive features of ms-van3t is the presence of a complete ETSI C-ITS protocol stack, from the Application layer down to the Access layer, included to support standard-compliant simulations with vehicles exchanging messages. The functions, information and services given to ITS applications are provided by the Facilities layer: in it, the simulator focuses on the Cooperative Awareness (CA), Decentralized Environmental Notification (DEN), and Vulnerable Awareness (VA) basic services, ensuring the generation, encoding,

and decoding of CAMs, event-triggered DENM and VAMs, in strict alignment with ETSI standards. Furthermore, ms-van3t introduces the Basic Services Container: an additional facility, that wraps together multiple Basic Services together, to hide most of the low-level configuration, reducing the complexity the user needs to face to run and manage simulations in this environment. The BS Container is given as an object in C++, the main programming language used in ms-van3t, that can be assigned to a simulated vehicle and can be used to access the underlying Basic Services to perform actions, e.g., to start the dissemination of VAMs. Finally, the layer includes the presence of the Local Dynamic Map (LDM), a facility that stores relevant data on road users and other perceived objects, providing information to applications when needed. The Transport layer hosts the Basic Transport Protocol (BTP), which provides connectionless multiplexing/demultiplexing between Facilities and GeoNetworking, adding minimal overhead. The Network layer implements ETSI GeoNetworking (GN), a protocol designed to cover ad-hoc communications challenges, as it features 'geographical addressing and forwarding', through the implementation of Single-Hop Broadcast (SHB) and GeoBroadcast (GBC), two forwarding schemes for message dissemination. Finally, the Access layer connects this stack to multiple supported radio technologies, such as IEEE 802.11p, 3GPP LTE-V2X, and 3GPP NR-V2X, allowing seamless comparison between them. [27]. The integration of the stack within the framework is achieved through the module automotive, that also includes the applications and networking logic, as well as the encoding and decoding schemes.

- **Multi-technology access support**
  A second major feature of ms-van3t is flexibility: as stated previously, it is capable of supporting multiple access technologies, even within the same simulation. The framework natively integrates IEEE 802.11p and cellular variants of V2X (LTE-V2X Mode 4 and 5G NR-V2X), with the possibility to extend to new RATs. This multi-technology capability is crucial in a period of technological transition, where different manufacturers adopt distinct V2X standards and interoperability remains an open challenge. By allowing direct comparison across technologies under identical mobility conditions, ms-van3t enables fair and reproducible benchmarking of V2X protocols and services.

- **Mobility integration.**
  A distinctive aspect of ms-van3t is its advanced mobility integration, which can rely either on synthetic traffic simulation through SUMO or on pre-recorded GNSS traces, without having to face many configuration changes. In the first case, a native and bidirectional coupling with SUMO is achieved through an extended TraCI interface, allowing ns-3 not only to import vehicle dynamics

(position, speed, acceleration) but also to directly control them. This enables the realization of complex scenarios where vehicles can be partially or fully autonomous and where VRUs are natively supported. Each entity simulated in SUMO is automatically equipped with a complete ETSI C-ITS stack in ns-3, ensuring realism in message exchange and mobility evolution. Beyond this efficient SUMO integration, ms-van3t introduces a major innovation with the gps-tc module, which permits the use of pre-recorded GNSS traces as an alternative mobility source. Unlike most frameworks, this module allows researchers to replay real-world trajectories, capturing the impact of GNSS inaccuracies, especially significant in urban canyon scenarios, on V2X communications. The gps-tc module, written in C++, provides a helper object to accept flexible CSV inputs containing at least vehicle ID, timestamp, latitude, longitude, speed, and heading, with optional acceleration (if not provided, gps-tc can perform the computation from speed values at each time istant) in order to simplify trace import; then, once the simulation starts, traces are reproduced, at the same time or with delays depending on the timestamp, thanks to the internal scheduling of 'position update' events at the right time. Moreover, gps-tc provides a solution for one of the greatest challenges when importing CSV files: coordinate conversion. Indeed, usually GNSS receivers report data with Geodetic coordinates, while ns-3 mobility models only accept Cartesian ones. This module uses an accurate projection, identifying the central meridian as the average longitude of all vehicles, averaged over all timestamps in the trace, which is efficiently implemented within the framework by importing a set of dedicated functions for geodesic calculations [28]. Both TraCI and gps-tc act as Vehicle Data Providers (VDPs) for the higher protocol stack, ensuring that applications in ms-van3t can seamlessly operate with either synthetic SUMO mobility or real-world GNSS-based mobility. This dual design represents a significant advancement over previous frameworks and broadens the spectrum of experiments by enabling the evaluation of V2X applications under realistic positioning uncertainties typically neglected in SUMO-only approaches.

- **HIL and emulation**
  Unlike many simulation frameworks, ms-van3t also supports emulation and communication modes where simulated nodes exchange ETSI messages with real devices via network interfaces. In this mode, the ETSI C-ITS stack implemented in ns-3 can operate as if it were running on an OBU or RSU, transmitting real CAMs or DENMs over sockets. This feature enables Hardware-in-the-Loop (HIL) experiments, bridging simulation and field testing, enabling the possibility to create hybrid scenarios where emulated vehicles can interact with real ones, and facilitating the transition from virtual validation to prototyping

and deployment.

- **Scalability**
  The ns-3 engine allows ms-van3t to scale to large-scale scenarios with any number of vehicles, only limited by the hardware capabilities of the device where the framework is currenlty running, while the modular stack ensures efficient execution even under heavy traffic loads. However, a problem that usually arise when many vehicles are involved is related to the complexity of gathering relevant data for each entity: ms-van3t tackles this through a dedicated module, called MetricSupervisor, that is in charge of collecting two fundamental metrics, Packet Reception Ratio (PRR) and one-way latency between a packet transmission and reception. The module can be easily extended to include other metrics for different purposes, such as comparison between different protocols or performance evaluation.

In summary, ms-van3t represents the state of the art in vehicular network simulation: it merges mobility, communications, and applications in a single extensible and robust framework, offers ETSI compliance and multi-technology access, supports emulation and Hardware-in-the-Loop, and scales to large and heterogeneous scenarios. This integrated environment is particularly well-suited for research on VRUs protection. It allows for the design of controlled experiments where vehicles and VRUs interact in complex traffic environments, while their communication exchanges are simultaneously subject to realistic channel dynamics and network constraints. By providing a holistic view of both mobility and communication aspects, ms-van3t enables large-scale testing of cooperative awareness messages, triggering strategies, and collision risk estimation methods under reproducible conditions. Its seamless integration of SUMO and ns-3, which are explained in more detail in the following sections, along with specialized modules for handling real-world data and collecting performance metrics, makes it an invaluable tool for advancing research in V2X communication.

3.2 Simulation of Urban Mobility

SUMO is an open-source traffic simulation suite that has been under continuous development since 2001 by the German Aerospace Center (DLR) [29]. It is designed as a time-discrete, space-continuous microscopic simulator where each vehicle is explicitly modeled in terms of position, route, and behavioral properties. SUMO supports classical microscopic traffic models, including car-following (e.g., Krauss and IDM) and lane-changing algorithms, but its extensible architecture allows additional models to be integrated through dedicated APIs. Vehicles can be described in detail with attributes such as departure time, physical parameters, emission

class, or noise category, enabling both traffic-flow studies and environmental impact assessments.

Overall, SUMO must be understood as a complete simulation suite rather than a single executable. It is equipped with a collection of tools for network generation and import, demand modeling, and routing. Road networks can be rather generated synthetically via netgen tool or idigitally mported from a wide range of sources using netconvert, including OpenStreetMap (OSM), allowing for the possibility to create highly realistic scenarios based on real-world topologies, which is the case exploited in this thesis. Furthermore, the suite includes tools to generate travel demand: for example, od2trips converts origin–destination matrices into single-vehicle trips, while jtrrouter computes routes based on intersection turn ratios. Once generated, routes can be dynamically adapted using different routing algorithms supported by SUMO, including even online re-routing during the simulation.

Therefore, SUMO scenarios can range from a single intersection to an entire metropolitan areas, giving the tool two important advatages: scalability and efficient execution. These qualities also affects the simulation step length, which is $1s$ by default, but it can also be tuned for higher temporal resolution. Moreover, in terms of visualization and debugging, SUMO offers two choices: a command-line mode for batch processing and a GUI mode, where vehicles can be color-coded according to different variables, i.e., speed or emissions. Finally, output formats are equally flexible, covering single-vehicle trajectories, aggregated measures on edges or lanes, as well as pollutant emissions and fuel consumption estimation.

One of SUMO's key innovations is the TraCI (Traffic Control Interface), a socket-based API developed to allow external programs to interact within the simulation in real time. Through TraCI, it is possible to query or modify the state of vehicles and infrastructure at each simulation step, making SUMO a powerful suite to be coupled with communication simulators, such as ns-3 or OMNeT++, or to be exploited for testing advanced traffic management strategies: large European research projects have taken advantage of this feature. For example, iTETRIS integrates SUMO with ns-3 for V2X evaluations, while CityMobil models scenarios of automated driving and platooning through the suite.

In ms-van3t, SUMO provides the mobility simulation, ensuring that the microscopic movement of vehicles and VRUs is faithfully reproduced. Its integration through TraCI allows ms-van3t to synchronize mobility events with communication events simulated in ns-3, guaranteeing consistency between traffic dynamics and networking behavior. Thanks to its open-source nature, active development, and large user community, SUMO has become the de-facto standard in research on intelligent transport systems, offering flexibility, scalability, and extensibility that few other tools can match.

3.3 Network Simulator 3

ns-3 is an open-source, modular and extensible network simulator widely adopted in both research and education. It provides a realistic and flexible environment for modeling the behavior of complex communication systems, supporting a wide range of networking technologies including Wi-Fi, LTE, 5G NR, Bluetooth, and satellite networks. The simulator is released under the GNU GPLv2 license and maintained by an active research community, encouraging continuous contributions of new models and features [**ns3website**, **mohta2023ns3**].

At its core, ns-3 is a discrete-event simulator, meaning that all actions within the network are represented as events scheduled in time. Its architecture is written in modern C++ with optional Python bindings, and leverages object-oriented design principles such as smart pointers, templates, callbacks, and copy-on-write to ensure modularity, extensibility, and efficient memory management. Unlike monolithic simulators, ns-3 is structured into reusable modules, which makes it possible to integrate new protocols and models without disrupting the rest of the system.

The simulator provides detailed models across the entire network stack: from the physical layer (with support for fading channels, propagation loss models such as Friis, log-distance, Rayleigh and Rician fading, and detailed PHY models for IEEE 802.11 variants) up to the application layer, where sockets-like APIs allow the execution of traffic generators and emulated applications. Transport protocols include UDP, TCP (with multiple variants), and SCTP, while the Internet stack features both IPv4 and IPv6 with support for routing protocols such as OSPF, BGP, RIP, and ad hoc routing protocols like OLSR, AODV, and DSDV. These models enable the study of both wired and wireless networks under realistic conditions.

A key feature of ns-3 is its attention to realism and integration. Nodes in ns-3 are designed to closely resemble real devices: each node can host multiple network interfaces, run Internet stacks, and communicate through sockets, making them a faithful abstraction of physical hosts. This design also supports emulation and virtualization: ns-3 can exchange real packets with external systems via raw sockets or virtual LANs, and even run real Linux protocol stacks inside the simulation. The simulator includes a tracing and statistics architecture, which decouples trace sources from sinks using callbacks, making it possible to export simulation data to external tools such as Wireshark, MATLAB, or custom analysis scripts.

3.4 VRU and VAM Support in ms-van3t

The extension of ms-van3t to include support for Vulnerable Road Users (VRUs) and the generation of VRU Awareness Messages (VAMs) represents a fundamental milestone in the evolution of the framework. In its original conception, ms-van3t

was primarily developed to model vehicle-centric scenarios, with a focus on the dissemination of Cooperative Awareness Messages (CAMs) and Decentralized Environmental Notification Messages (DENMs). While this architecture enabled the simulation of many aspects of ETSI Cooperative ITS systems, it excluded VRUs from the communication loop, limiting them to passive entities in SUMO mobility traces. As a consequence, VRUs could only be represented as moving obstacles, without the ability to generate, transmit, or receive standardized awareness messages, and therefore without the possibility of being considered as active participants in cooperative safety strategies. This limitation became increasingly critical with the publication of ETSI TS 103 300-2 and TS 103 300-3, which formally introduced the concept of VAMs and defined their role within the ETSI ITS architecture. To align simulation tools with these specifications, it was essential to integrate VRUs as first-class entities into the framework, equipped with their own service logic and message types.

This gap was addressed by Alessandro Genovese in his Master's thesis [30], who extended ms-van3t to provide native support for VRUs and VAMs. The core of this contribution is the implementation of the VRUBasicService, a dedicated C++ module that manages all aspects of VAM generation, transmission, and reception. Through this service, VRUs simulated in SUMO (pedestrians or cyclists) can now be directly coupled with the ETSI C-ITS stack in ns-3, thus becoming active ITS-S nodes within the cooperative communication ecosystem. The service was designed to be fully compliant with the ETSI specifications and to provide a structured interface for VRU configuration, message dissemination, and service interaction.

The VRUBasicService is responsible for several key functions:

- **VRU configuration management**: storing the parameters of the VRU profile (e.g., pedestrian, cyclist), either at initialization or dynamically during the simulation.

- **Message generation and encoding**: creating VAMs that include information such as position, heading, speed, and profile type, encoded in accordance with ETSI standards.

- **Interaction with other basic services**: exchanging information with the PoTi (Position and Time) service and the DDP (Decentralized Data Processing) service, and supporting interaction with the Human-Machine Interface (HMI) depending on the VRU profile.

- **Transmission control**: activating or deactivating VAM dissemination based on VRU profile parameters and network congestion control rules, as required by ETSI TS 103 300-2.

- **Triggering condition management**: ensuring that VAMs are transmitted when one or more state changes of the VRU satisfy predefined thresholds, in alignment with ETSI TS 103 300-3.
- **Cluster management (initial design)**: laying the foundation for functions related to VRU clusters, such as leader election and aggregated VAM dissemination, although these features were not yet fully implemented.

Among these functions, the triggering logic is of particular importance. Genovese's implementation included several of the triggering conditions required by ETSI, namely:

- **Time-based triggering**: a VAM is sent if the maximum generation interval ($T_{GenVamMax}$) has elapsed since the last transmission.
- **Position change**: a VAM is triggered if the VRU has moved beyond a configurable distance threshold since the last message.
- **Speed change**: a VAM is generated if the speed variation relative to the last transmission exceeds a threshold.
- **Heading change**: a VAM is triggered when the heading deviation is greater than a predefined angular threshold.
- **Safe distance violation**: a VAM is sent when another road user breaches the minimum safe distance (longitudinal, lateral, or vertical).

These conditions ensure that VAMs are not generated at a fixed rate but adaptively, based on meaningful changes in the VRU's state, thereby reducing channel load and maintaining message relevance. However, despite the completeness of this implementation, two important conditions remained absent:

- **Trajectory Interception Probability (TIP)**: ETSI specifies that a new VAM must be triggered if the probability of trajectory interception changes by more than a defined threshold (typically 10%). This condition was not implemented because, at the time, the framework lacked the infrastructure to compute TTC-based metrics and translate them into probabilities.
- **Cluster-based triggering**: ETSI also foresees VRU cluster formation, where groups of VRUs (e.g., pedestrians walking together, cyclists in a peloton) are represented by a cluster leader that disseminates aggregated VAMs on behalf of the group. While the VRUBasicService included preliminary structures to support clustering, the full implementation of cluster management and triggering was not yet realized.

The work presented in this thesis builds directly upon this foundation. In particular, it focuses on the implementation of the missing TIP-based triggering condition. To this end, a dedicated methodology was developed to compute the Time to Collision (TTC) between VRUs and surrounding vehicles, map it into a collision probability through several possible strategies (as discussed in Chapter 4), and integrate this probability into the VRUBasicService as a new triggering parameter. The implementation ensures compliance with ETSI TS 103 300-3, specifically by activating a new VAM whenever the TIP changes by more than 10% with respect to the previous transmission. This addition significantly enhances the realism and safety relevance of the simulation environment, as it introduces a proactive condition that anticipates collision risk even in the absence of direct changes in position, speed, or heading.

By closing this gap, the present work strengthens the VRU support in ms-van3t and VaN3Twin, providing a more complete and ETSI-compliant framework. Although clustering remains an open research avenue for future extensions, the integration of TIP triggering represents a major step forward in enabling the study of cooperative safety solutions for VRUs under realistic and reproducible simulation conditions.

Chapter 4

From TTC to Collision Probability: Methodology Development

This chapter presents in detail the methodology developed throughout this thesis to define a procedure for computing the Trajectory Interception Probability (TIP) starting from the TTC metric. The first step involved implementing the selected TTC computation within the ms-van3t framework and testing it across different scenarios to evaluate the influence of key parameters and overall performance. Building on this foundation, several strategies for mapping TTC into TIP were investigated, leading to the final choice of an exponential growth function to discretize TTC values into TIP intervals. The complete method was then integrated into ms-van3t, together with the triggering condition defined by the ETSI standard that is represented by the pseudocode in Algorithm 1. Finally, a set of simulations was conducted over multiple scenarios to assess the performance of the proposed approach and to analyze its impact on VAM transmissions.

4.1 Time to Collision

As discussed in the literature review, the TTC has been identified as the most suitable metric to serve as the foundation for this work. Its widespread adoption in traffic safety research, together with its intuitive interpretation and computational efficiency, constitutes for an effective starting point to derive a frame-by-frame estimation of collision probability. For this reason, the methodology developed builds directly on this metric. However, the original definition of TTC poses a significant limitation, since it relies on the assumption that both entities move with

constant velocities. This simplification ignores possible accelerations, making the metric less precise and less consistent with real-world dynamics, where vehicles and VRUs frequently brake, accelerate, or change direction. For this reason, in this thesis the computation of TTC explicitly accounts for accelerations, through a method developed in [31]. The approach is twofold: if the relative acceleration between the entities is negligible, the problem reduces to the constant-velocity case, which admits a simple closed-form solution; otherwise, the constant-acceleration case is considered, leading to a higher-order formulation where the candidate collision times are obtained by solving a cubic polynomial.

In any case, the procedure at a given istant begins by defining the relative kinematic state of the two entities in a Cartesian coordinate system: let the position of entity i at the current point in time be $\mathbf{w}_i = (x_i, y_i)$, its velocity $\mathbf{v}_i = (v_{x,i}, v_{y,i})$, and its acceleration $\mathbf{a}_i = (a_{x,i}, a_{y,i})$. Similarly, the second entity j is characterized by $\mathbf{w}_j = (x_j, y_j)$, $\mathbf{v}_j = (v_{x,j}, v_{y,j})$, and $\mathbf{a}_j = (a_{x,j}, a_{y,j})$.

The relative position, velocity, and acceleration of j with respect to i are defined as:

$$\mathbf{w}_0 = (w_i - w_j) = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} x_i - x_j \\ y_i - y_j \end{bmatrix}, \tag{4.1}$$

$$\mathbf{v} = (v_i - v_j) = \begin{bmatrix} \Delta v_x \\ \Delta v_y \end{bmatrix} = \begin{bmatrix} v_{x,i} - v_{x,j} \\ v_{y,i} - v_{y,j} \end{bmatrix}, \tag{4.2}$$

$$\mathbf{a} = (a_i - a_j) = \begin{bmatrix} \Delta a_x \\ \Delta a_y \end{bmatrix} = \begin{bmatrix} a_{x,i} - a_{x,j} \\ a_{y,i} - a_{y,j} \end{bmatrix}. \tag{4.3}$$

These six scalar quantities fully describe the relative kinematic state of the two entities. The future relative displacement of i with respect to j at time t is therefore given by:

$$w(t) = w_o + v * t + \frac{1}{2} * a * t^2 \tag{4.4}$$

A collision between two entities occurs when they occupy the exact same position in space at the same instant of time, which means that the difference between their positions must be null, so the following condition must be satisfied:

$$w(t) = 0 \tag{4.5}$$

However, it is better to work with the squared Euclidean distance of the displacement vector than the raw vector itself, as it produces scalar polynomial equations

in time, avoids square roots and still preserves the equivalence with the collision condition. Hence, we can define the following:

$$d^2(t) = \|\mathbf{w}(t)\|^2 = \mathbf{w}(t)^\top \mathbf{w}(t) = 0 \tag{4.6}$$

Since the function 4.6 is always non-negative and polynomial in time, an impact can only happen at an instant where this function reaches a minimum. To determine the time candidates for which this is true, the derivative of the squared distance is computed and set to zero:

$$\frac{d}{dt} d^2(t) = 2\,\mathbf{w}(t)^\top \dot{\mathbf{w}}(t) = 0 \tag{4.7}$$

Since $\dot{\mathbf{w}}(t) = \mathbf{v} + \mathbf{a}t$, this gives:

$$\frac{d}{dt} d^2(t) = 2\left(\mathbf{w}_0 + \mathbf{v}t + \tfrac{1}{2}\mathbf{a}t^2\right)^\top (\mathbf{v} + \mathbf{a}t) = 0 \tag{4.8}$$

Solving 4.8 yields a set of candidate times $t_c \geq 0$, but they are not sufficient on their own: the distance at such instants may be minimal but still strictly positive. Therefore, each candidate t must be checked against the exact collision condition stated in 4.6. Finally, the TTC can be defined as the smallest non-negative time out of all the valid candidates, therefore the operational definition in this thesis is:

$$TTC = \min \left\{ t \geq 0 \;\middle|\; \frac{d}{dt} d^2(t) = 0 \;\text{ and }\; d^2(t) = 0 \right\} \tag{4.9}$$

If no such solution exists, the two entities never meet under the current motion states and the TTC is set to -1.

At this point, in order to understand how the explicit form of the polynomial is going to become and how it needs to be solved, two case must be distinguished, depending on the relative acceleration being negligible or not.

Case 1: constant velocity. If relative acceleration is null, the derivative reduces to a linear equation that directly provides the candidate collision time:

$$\frac{d}{dt} d^2(t) = 2\,(\mathbf{w}_0 + \mathbf{v}t)^\top \mathbf{v} = 0 \tag{4.10}$$

From here the possible TTC can easily be derived from the condition set to null:

$$\frac{d}{dt} d^2(t) = 0 \quad \Longrightarrow \quad t^\star = -\frac{\mathbf{w}_0^\top \mathbf{v}}{\|\mathbf{v}\|^2}. \tag{4.11}$$

Taking into account the kinematic states previously defined, it is possible to further develop the equation to compute the candidate time defined in 4.11:

$$t^\star = \frac{-(x_i - x_j)(v_{x,i} - v_{x,j}) - (y_i - y_j)(v_{y,i} - v_{y,j})}{(v_{x,i} - v_{x,j})^2 + (v_{y,i} - v_{y,j})^2}. \tag{4.12}$$

In 4.12, the numerator inspects whether the two cars could collide or not looking at their directions: indeed, if $(x_i - x_j)(v_{x,i} - v_{x,j}) < 0$ or $(y_i - y_j)(v_{y,i} - v_{y,j}) < 0$, it means that the entities are moving away from each other, making the collision impossible. Instead, the denominator is linked to the relative speed of the system and if one of the elements is null, it means that the entities are moving behind one another at the same speed, meaning the collision is also impossible in this case. If, indeed, the impact can happen, we can compute 4.12, which is returned as TTC, if positive (a negative value would be meaningless, leading to the impossibility of the collision).

Case 2: constant acceleration When accelerations are present, the derivative leads to a cubic polynomial in time, which is solved using Ruffini's rule for factorization when possible:

$$\frac{d}{dt} d^2(t) = 2\,\mathbf{w}(t)^\top \dot{\mathbf{w}}(t) = 2\left(\mathbf{w}_0 + \mathbf{v}t + \tfrac{1}{2}\mathbf{a}t^2\right)^\top (\mathbf{v} + \mathbf{a}t) = 0. \tag{4.13}$$

Expanding the products in 4.13, we obtain a cubic polynomial in t:

$$At^3 + Bt^2 + Ct + D = 0, \tag{4.14}$$

with coefficients defined as:

$$\begin{aligned}
A &= \|\mathbf{a}\|^2 = (a_{x,i} - a_{x,j})^2 + (a_{y,i} - a_{y,j})^2, \\
B &= 3\,\mathbf{v}^\top \mathbf{a} = 3\Big[(v_{x,i} - v_{x,j})(a_{x,i} - a_{x,j}) + (v_{y,i} - v_{y,j})(a_{y,i} - a_{y,j})\Big], \\
C &= 2\|\mathbf{v}\|^2 + 2\,\mathbf{w}_0^\top \mathbf{a} = 2\Big[(v_{x,i} - v_{x,j})^2 + (v_{y,i} - v_{y,j})^2\Big] \\
&\quad + 2\Big[(x_i - x_j)(a_{x,i} - a_{x,j}) + (y_i - y_j)(a_{y,i} - a_{y,j})\Big], \\
D &= 2\,\mathbf{w}_0^\top \mathbf{v} = 2\Big[(x_i - x_j)(v_{x,i} - v_{x,j}) + (y_i - y_j)(v_{y,i} - v_{y,j})\Big].
\end{aligned} \tag{4.15}$$

Equation 4.14 must be solved to obtain the candidate instants of minimum relative distance. In practice, Ruffini's rule can be applied to test for rational roots and reduce the cubic to a quadratic whenever possible, simplifying the solution. Otherwise, the cubic is solved either analytically or numerically.

Once the candidate times $t^\star$ are found, each root must be checked against the original collision condition:

$$\mathbf{w}(t^\star) = \mathbf{0} \quad \Longleftrightarrow \quad d^2(t^\star) = 0.$$

If at least one non-negative real root satisfies this condition, the TTC is defined as the smallest among them. If no admissible solution exists, either because all roots are negative, complex, or do not reduce the distance to zero, then the collision is deemed impossible and the TTC is set to -1.

By recomputing this procedure for every pair of entities at each frame, the system produces a deterministic, frame-by-frame estimate of TTC. The constant-velocity formulation ensures a lightweight and robust closed-form solution when accelerations are negligible, while the constant-acceleration formulation extends the method to more complex scenarios, making use of Ruffini's rule to efficiently solve the cubic equation. In all cases, the TTC represents the exact time of intersection of the two trajectories, with the minimum distance explicitly set to zero.

4.1.1 Implementation in VaN3Twin

As previously stated, the main goal of this thesis is to develop a methodology to compute the Trajectory Interception Probability and, consequently, to implement the VAM triggering condition regarding this metric within VaN3Twin. Up to this point, the theoretical reasoning and analysis that led up to the decision that the best approach was to develop a TTC-based method to compute TIP has been extensively explained. In order to assert that the chosen TTC computation method is the best fit, the first analytical analysis conducted in VaN3Twin for this work regards the elements needed to compute the metric whenever needed. However, in order to show and explain the insights provided by the results, it is first necessary to discuss in detail how the TTC computation has been implemented within the simulation framework and how the analyses were conducted.

Since VaN3Twin did not originally include a dedicated mechanism for Time To Collision evaluation, this functionality was implemented as part of the thesis work. To this end, a new class named **`t2c`** was developed, following the object-oriented design principles of the framework. The class encapsulates all the attributes (positions, velocities, accelerations) and methods required for the computation, and exposes a core routine, `compute_t2c`, that retrieves the necessary data, standardizes the units, and applies the analytical formulation presented in the previous section.

The introduction of `t2c` served two purposes. In the first place, it provided a modular and reusable component for TTC calculation, enabling consistent evaluation of both VRU–VRU and VRU–vehicle interactions. At the same time, it created a natural entry point for extending the functionality: although initially focused solely on TTC, the class was progressively expanded to incorporate the

TTC to TIP mapping and the evaluation of VAM triggering conditions. In this way, what began as a computation module for a single metric gradually became the core element that supports the entire methodology developed in this thesis.

Within the simulation workflow, TTC computation is triggered every time a VRU receives a message:

- a **CAM**, received from a vehicle via the `CA Basic Service`;
- or a **VAM**, received from another VRU via the `VRU Basic Service`.

In both cases, the event activates the `compute_t2c` method, which:

1. extracts the relevant state variables from the message,
2. converts them into consistent units and formats,
3. determines whether the entities are on a collision course,
4. and, if so, estimates the remaining time before the potential collision.

Initially, the resulting TTC values were stored for inspection and analysis. In the extended version of the class which forms the actual scope of this thesis, these values became the input to the TIP mapping function and, ultimately, to the logic used to trigger VAMs.

4.1.2 Analytical Validation

In order to evaluate if the chosen TTC computation method was the best fit, a simulation was designed and run in VaN3Twin. The reference scenario lasted 25 seconds and involved a single pedestrian and one vehicle moving along trajectories that inevitably led to a collision. It was chosen to use a very trivial and controlled setup for a reason: by isolating a minimal interaction case, it becomes possible to clearly highlight the effect of the main physical factors on the TTC values, without interference from additional actors or external dynamics.

The analysis focused on the correlation, graphically shown below, between TTC values and three main parameters:

- the distance between the VRU and the vehicle, which was additionally computed each time a TTC value was obtained;
- the relative velocity of the entities, extrapolated from the cartesian velocity components used in the TTC computation ;
- the acceleration of the entities, deduced from the cartesian acceleration components as well.

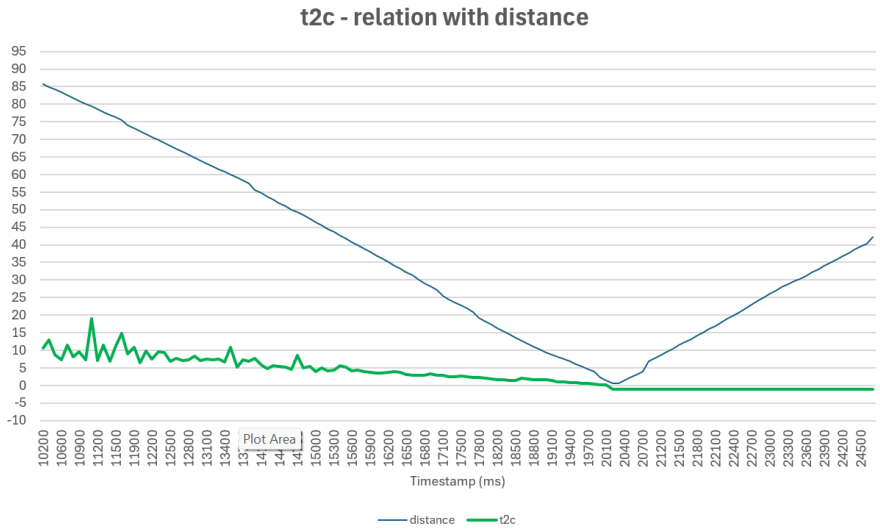

Figure 4.1: Relation over time between Time to Collision and entities' distance

Figure 4.1 shows the relation over time between TTC and the distance from the pedestrian to the vehicle. The trend is intuitive: as the distance decreases, the Time to Collision shortens accordingly, up until to the point when the potential collision occurs (i.e., approximately at 21 seconds after the beginning of the simulation). When the two entities meet, the TTC curve approaches zero, correctly signalling that the predicted impact has occurred. From that instant onward, as can be seen in all the plots, the TTC value is set to -1, indicating that no further collision is expected, since the vehicle and the pedestrian continue along separate trajectories. For this reason, the part of the curve after the impact is not relevant to the objectives of this thesis, which focus on the prediction of collisions in the approach phase rather than on later dynamics. This behaviour provides a first proof that the implementation correctly reproduces the expected physical intuition behind TTC. However, it shows another interesting insights: there are some peaks on the TTC trend at the beginning of the simulation and it is obvious from Figure 4.1 that they are not caused by the distance between the entities. Therefore, TTC captures other information as well and can be affected by other factors.

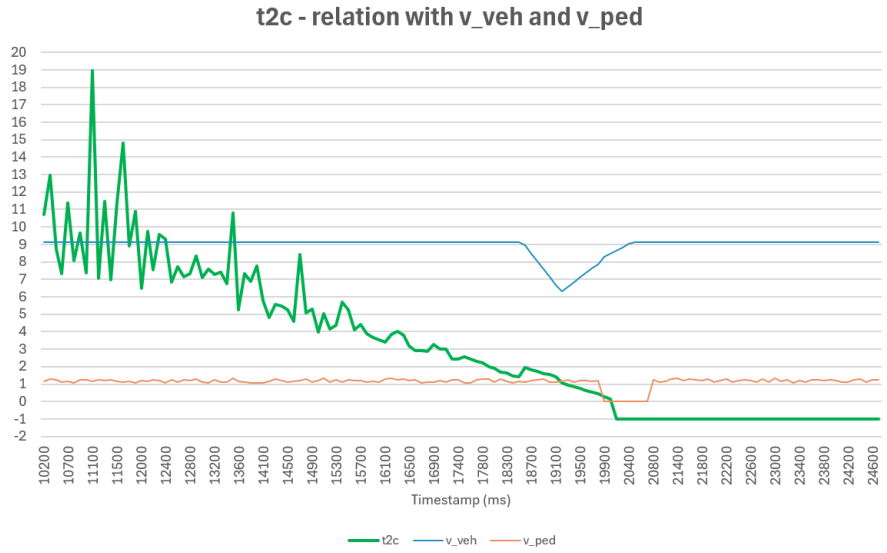

Figure 4.2: Relation over time between Time to Collision and entities' relative velocity

Figure 4.2 illustrates the relation between Time to Collision and the velocities of both entities, which are not simple Cartesian components, but the magnitudes of the velocity vectors, computed as the Euclidean norm of their horizontal and vertical components used to compute TTC values, in order to reflect in the plot the actual intensity of the motion, regardless of its direction in the plane:

$$v_{ped} = \sqrt{v_{ped_x}^2 + v_{ped_y}^2}, \quad v_{veh} = \sqrt{v_{veh_x}^2 + v_{veh_y}^2} \tag{4.16}$$

In the plot, the pedestrian's behaviour is coherent with typical VRU dynamics: its speed remains almost constant and very low throughout the scenario, making its influence negligible in the TTC trend. In contrast, the vehicle's speed plays an important role: in the first part of the scenario it remains constant, which translates into a gradual and almost linear decrease of TTC as the distance between the entities closes. Near the collision point, the vehicle temporarily brakes, as can be seen by the drop in the velocity trend, then accelerates again: the TTC curve reacts accordingly, flattening during the deceleration phase and then dropping sharply as the vehicle speeds up, up to the collision instant. Overall, this correlation confirms two key insights. First, TTC is highly sensitive to changes in vehicle speed, while the pedestrian velocity can be neglected in practical terms. Second, the formulation based on the norm of the velocity vector ensures that the metric captures the effective closing speed in any direction, rather than just scalar differences of speeds. This guarantees that the TTC implementation in VaN3Twin responds in a realistic and consistent way to the kinematic conditions of the entities involved.

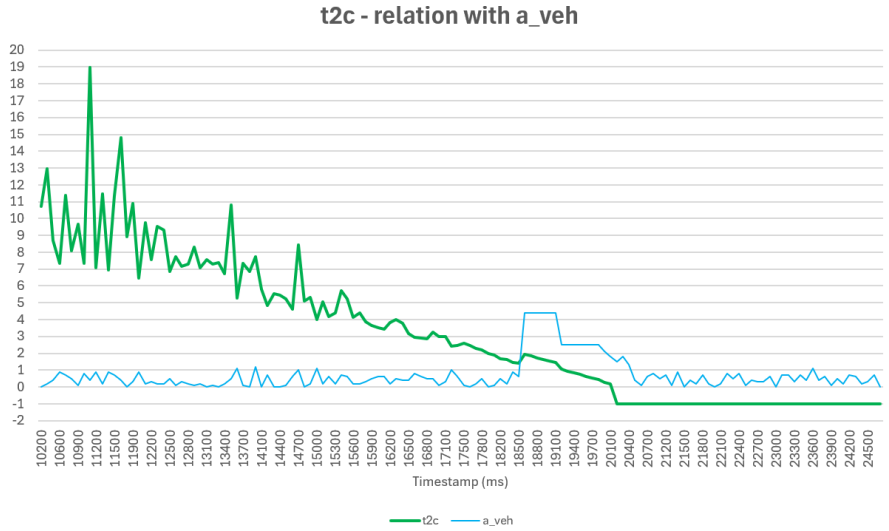

Figure 4.3: Relation over time between Time to Collision and vehicle's acceleration

Finally, Figure 4.3 illustrates the relation between TTC and the vehicle's acceleration, which is not a single Cartesian component but the resultant magnitude, computed as:

$$a_{veh} = \sqrt{a_{veh_x}^2 + a_{veh_y}^2} \tag{4.17}$$

At the beginning of the analysis, the pedestrian acceleration was also considered. However, during the simulations it proved to be consistently null or very close to zero, reflecting once again the slow motion typical of VRUs, which is why only the vehicle acceleration is reported and discussed here.

The graph highlights that the vehicle acceleration remains almost null for most of the simulation, corresponding mainly to a uniform motion. However, when the vehicle performs even small peaks of acceleration, both positive and negative, the TTC curve reacts immediately:

- if it accelerates, TTC decreases sharply, showing that the possible collision is approaching faster than before;
- if it decelerates, TTC temporarily increases, reflecting that the time horizon to impact has been extended.

This high sensitivity to acceleration is crucial, since it demonstrates that TTC is not only a function of instantaneous distance and velocity but also captures dynamic variations in motion. In conclusion, the observed behaviour highlights what we wanted to confirm with this analysis: how crucial it is to account for the

entities' acceleration in the TTC computation. The inclusion of this factor confirms that the adopted method represents the best fit for the objectives of this thesis: it ensures greater reliability and realism when modelling traffic interactions, and it captures dynamic variations that simpler approaches would overlook, a feature especially valuable in scenarios involving VRUs, where unpredictable and sudden changes in the environment must be detected with high priority to enable timely safety interventions.

4.2 Thresholding the TTC Range

The direct use of the time to collision (TTC) as a collision risk indicator requires the definition of a meaningful validity range. Without an appropriate thresholding, very small values of TTC would always correspond to certain collision events (probability equal to 1), while very large values would misleadingly indicate non-negligible risk in situations where a crash is not realistically plausible. For this reason, two thresholds are commonly introduced: a minimum threshold (TTC_{min}) and a maximum threshold (TTC_{max}), delimiting respectively the boundary of imminent danger and the one of safe conditions.

The lower threshold TTC_{min} is motivated by experimental and perceptual studies on human drivers. Brown [32] demonstrated through field tests that a value of 1.5 s represents a realistic lower bound, below which a conflict is perceived as critical and evasive maneuvers are triggered. Similar conclusions are reported in other works, which identify thresholds in the range between 1.5 s and 4 s, depending on reaction time, driving conditions and perceptual variability [33]. Nonetheless, most studies confirm that values between 1.5 s and 4 s cover the typical span of driver reaction thresholds.

On the other hand, the upper threshold TTC_{max} accounts for the fact that very high TTC values lose predictive meaning. When the time available before a potential collision exceeds a certain duration, the situation can be considered intrinsically safe, as both the driver and the system have enough time to react. Several works in the literature suggest values in the order of 10 s to 20 s as upper bounds beyond which the TTC no longer indicates an actual risk.

The introduction of these two thresholds allows for the partitioning of the TTC domain into three distinct zones (Figure 4.4):

- **Critical zone** ($TTC < TTC_{min}$): imminent collision risk, probability is set to 1.
- **Transition zone** ($TTC_{min} \leq TTC \leq TTC_{max}$): intermediate region where the collision probability is mapped as a TTC-based function (see Section 4.3 and Section 4.4).

- **Safe zone** ($TTC > TTC_{max}$): negligible risk, probability tends to 0.

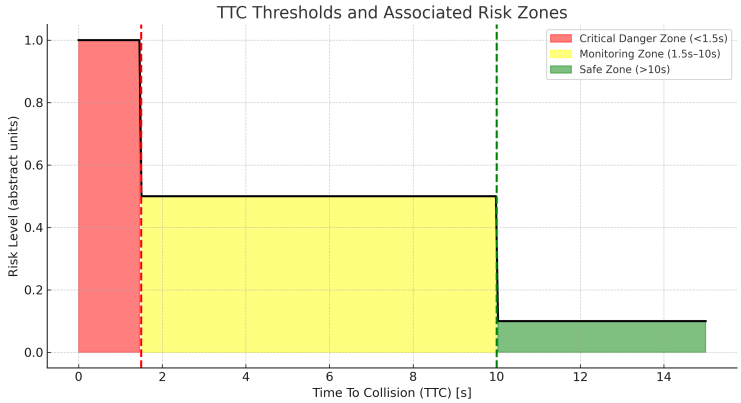

Figure 4.4: Conceptual partition of the TTC domain into three risk zones.

This thresholding strategy ensures that the TTC is taken into account only when within a meaningful range, anchoring the estimation of collision probability to values consistent with both human perception and empirical evidence. It prevents attributing significance to extreme values that do not reflect realistic safety conditions, and provides a structured basis for the subsequent mapping of TTC into collision probability.

4.3 Mapping Strategies

As previously stated, the main objective of this thesis is to establish a computation method for Collision Probability. In order to achieve this, after having established to follow a TTC-based approach, various mapping strategies were developed in order to map the TTC values into Collision Probability ones: the best fit was determined through various analysis and considerations.

Transforming time to collision (TTC) into a collision probability requires the definition of a mapping function that converts temporal values into a normalized probability scale. This step is necessary to align the TTC metric with ETSI triggering conditions, which rely on the variation of the Trajectory Interception Probability (TIP). The mapping must satisfy a few key requirements: (i) it must be bounded between 0 and 1, (ii) it must increase monotonically as TTC decreases, (iii) it should be computationally efficient to support real-time applications, and (iv) it should exhibit sufficient smoothness to prevent unstable triggering. In this section, four mapping strategies are analyzed in detail: fixed intervals, linear mapping, exponential mapping, and discrete interval mapping.

4.3.1 Fixed intervals

The first mapping approach considered in this thesis directly derives from the ETSI triggering condition. According to [8], a new VAM shall be generated whenever the variation of the Trajectory Interception Probability (TIP) is greater or equal to 10%. Starting from this definition, the most straightforward strategy was to discretize the TIP range [0,1] into 10 fixed intervals of equal size (each representing a 10% change). In order to maintain consistency, the Time to Collision (TTC) domain between the chosen thresholds TTC_{min} and TTC_{max} was also partitioned into the same number of intervals. The width of each TTC interval is thus defined as:

$$Width = \frac{TTC_{max} - TTC_{min}}{n} \tag{4.18}$$

where n is the number of TIP intervals (in this case, $n = 10$).

Given this discretization, any TTC value TTC_i at a given time instant can be directly mapped to the corresponding TIP value through the following relation:

$$TIP_i = \frac{n - \frac{1}{Width} * (TTC_i - TTC_{min})}{n} \tag{4.19}$$

The formulation can be visualised in 4.5, where the TIP value decreases in fixed steps as the TTC increases from its minimum to its maximum threshold, which in this illustrative example have been set at 1.5 s and 10 s accordingly. This ensures that:

- if $TTC = TTC_{max}$, the resulting TIP approaches 0,
- if $TTC = TTC_{min}$, the TIP reaches 1,
- intermediate TTC values are mapped proportionally to the corresponding discrete TIP interval.

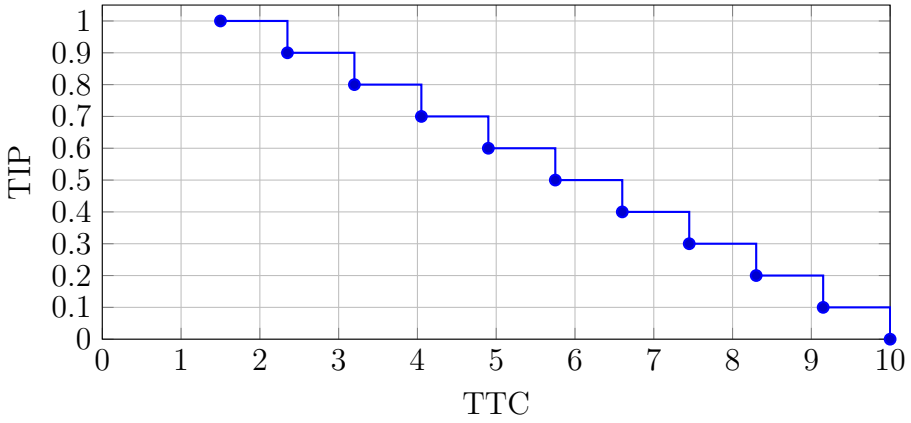

Figure 4.5: Fixed-interval mapping using reference thresholds

4.3.2 Linear Mapping

The linear mapping represents another intuitive approach to transform the Time to Collision into a collision probability value. Due to its simplicity and clarity, its application has already been analysed in the literature: an example can be found in [34], which explores the context of cooperative overtaking scenarios, where a linear relationship between TTC and collision probability is assumed in order to simplify computation and enable real-time implementation. Building on this main concept, the present work considers a straightforward proportional mapping, in which the collision probability decreases linearly as TTC increases. For the same reasons stated before, a time horizon is defined here as well, limiting the scope of the function: therefore, the probability starts from a maximum value of 100% when the Time to Collision corresponds to the lower threshold TTC_{min}, and reaches zero at the upper limit TTC_{max}, its trend defined by the following equation:

$$TIP_i = 1 - \frac{TTC_i}{TTC_{max}}. \tag{4.20}$$

In this expression, TTC_i denotes the instantaneous Time to Collision at step i, while TTC_{min} and TTC_{max} are the previously defined thresholds that define the operative range, as discussed in Section 4.2. Therefore, the mapping is normalized over the selected interval, ensuring a bounded and predictable transformation of TTC values into collision probabilities, which is clearly illustrated in 4.6, where the mapping is performed using the same time horizon as the other strategies, to ensure consistency during comparisons, given by these parameters: $TTC_{min} = 1.5s$ and $TTC_{max} = 10s$.

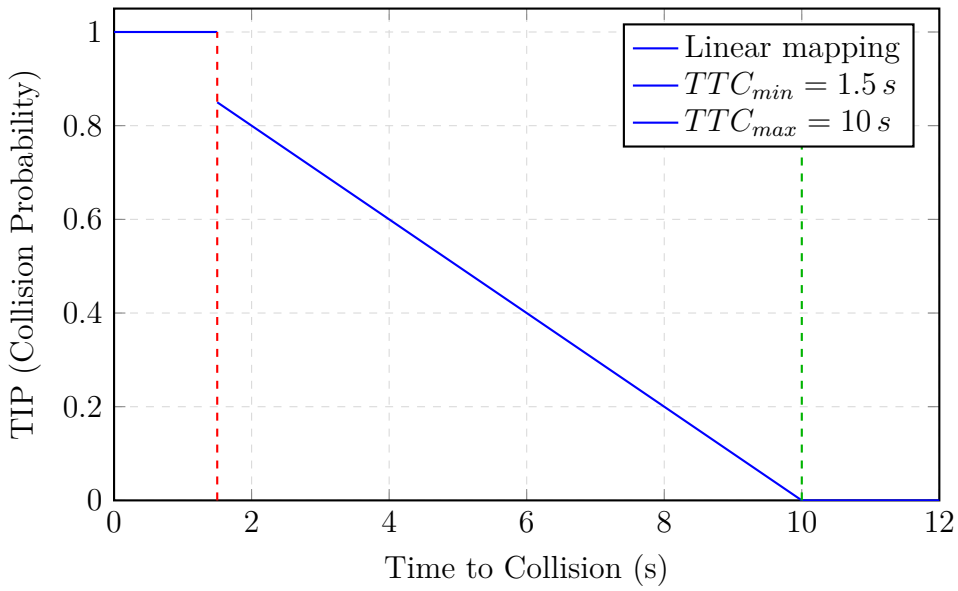

Figure 4.6: Linear mapping with $TTC_{min} = 1.5$ s and $TTC_{max} = 10$ s.

The mapping is trivial, as it involves only a basic arithmetic operation, which makes it computationally efficient, highly scalable and particularly attractive for real-time applications in environments where resources are limited. Furthermore, the linear structure guarantees predictability and a uniform sensitivity to variations in TTC, a property that can be very useful during the design and debugging phase of safety mechanisms. Finally, another advantage of this strategy is its interpretability: the function generates a monotonic and directly proportional curves, where it is immediate that each unit decrease in TTC corresponds to an equivalent increase in probability.

However, this very simplicity also highlights the limitations of the linear mapping. By construction, the model does not reproduce the nonlinear escalation of collision risk that is typically observed both in human perception and in engineered warning systems. As TTC approaches small values, the perceived urgency and required system responsiveness tend to increase more sharply than a linear law would suggest. As a result, the mapping may either produce premature triggering in relatively safe situations, or conversely, underestimate the severity of truly imminent collisions. In addition, the constant proportional sensitivity can amplify small fluctuations in TTC near the ETSI threshold of 10% variation in TIP, potentially generating redundant VAM transmissions and reducing the stability of the triggering mechanism.

In summary, the linear mapping offers a computationally efficient and interpretable baseline that has been recognized in the literature, but its oversimplified proportional behaviour makes it less suited for capturing the dynamic escalation of

collision risk that characterizes real-world safety-critical interactions.

4.3.3 Exponential Mapping

The motivation for adopting an exponential relation between TTC and collision probability originates from previous research, where the Time to Collision was not treated as a simple proportional indicator but rather transformed into a nonlinear, normalized probability. The central idea emerging from these studies is that drivers and intelligent systems do not perceive the urgency of a potential collision in a linear fashion: when the TTC is large, small variations do not alter the perceived level of danger, whereas in the final instants before an impact, even a fraction of a second can dramatically change the associated risk. This asymmetry in perception suggests that collision probability should not increase linearly with decreasing TTC, but should instead follow a nonlinear escalation, with a slow growth in the safe region and an increasingly steep rise in the critical region [10].

Although no closed-form formula was originally specified, this principle was made evident by showing curves where the TTC decreases over time while the associated collision probability increases with an exponential-like shape. Such curves, an example of which is reproduced qualitatively in Figure **??**, highlight the conceptual link between the physical variable (TTC) and the probability of collision, emphasizing that the latter should reflect the nonlinear escalation of urgency rather than a proportional trend.

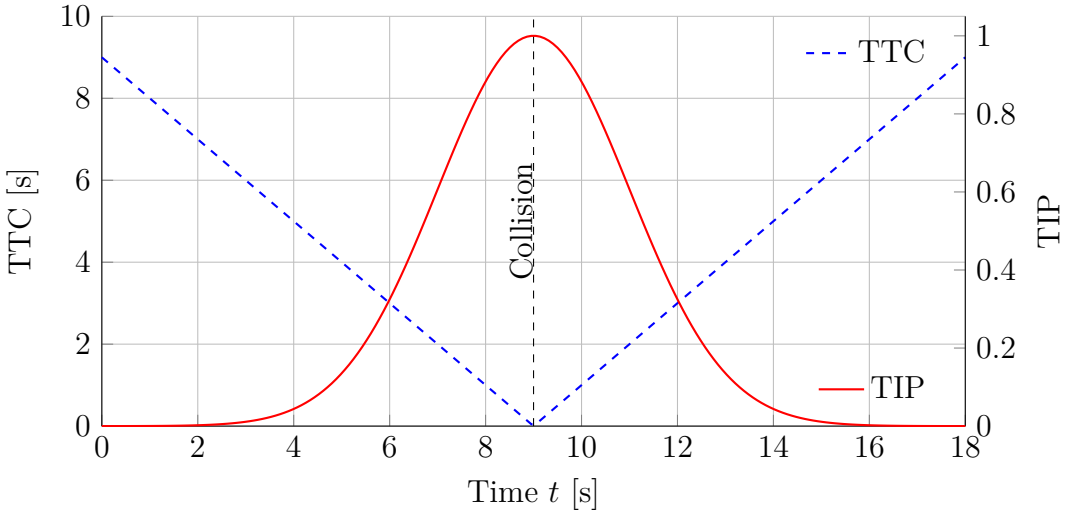

Figure 4.7: Conceptual relationship between TTC and TIP inspired by [10]

Building on this conceptual foundation, the present work translates the intuition of an exponential-like increase into a concrete and implementable mapping. In order

to ensure real-time usability, the mapping has been formalized with an explicit closed-form function that links the instantaneous TTC value to the TIP. The chosen formulation is:

$$TIP_i = e^{-\alpha\, TTC_i} \tag{4.21}$$

This expression captures the nonlinear escalation of risk but, taken alone, would suffer from unrealistic boundary behaviour: the probability would asymptotically approach one as $TTC \to 0$ without ever reaching it, and it would remain strictly positive even for arbitrarily large TTC. To avoid these issues, the exponential law is not applied over the entire domain, but only within the bounded interval already defined by TTC_{min} and TTC_{max}. Outside this region, the function saturates by construction: $TIP = 1$ when $TTC \leq TTC_{min}$, and $TIP = 0$ when $TTC \geq TTC_{max}$. In this way, the mapping remains consistent with the operational thresholds previously established.

Within the active interval, the curve is governed by the growth-rate parameter α, which controls its steepness and therefore the sensitivity of the system to variations in TTC. A practical and intuitive choice for this parameter is to tie it directly to the maximum considered horizon, so that the mapping automatically adapts to the scale of interest:

$$\alpha = \frac{1}{TTC_{max}}. \tag{4.22}$$

This definition ensures that the response of the curve is normalized across different configurations. A larger TTC_{max} produces a slower and more gradual rise in probability, while a smaller TTC_{max} makes the function steeper and more reactive to small TTC variations. In this sense, α becomes a design knob intrinsically linked to the choice of time horizon, enabling a balance between early-warning sensitivity and trigger stability.

As a result, the exponential mapping takes on a form that is both bounded and interpretable. It preserves the desired nonlinear escalation of urgency, while remaining compatible with the explicit thresholds discussed earlier and directly implementable in real-time triggering mechanisms in line with ETSI requirements.

4.3.4 Final Choice: Discrete Mapping with Exponential Growth

In order to evaluate whether the mapping strategies previously introduced could be suitable for the objectives of this thesis, and to identify which of them represented the most promising candidate for further development, a dedicated simulation was performed in the ms-van3t environment.

A controlled test scenario was designed, consisting of a single pedestrian and a single vehicle set on an unavoidable collision course. In this configuration, the Time to Collision (TTC) naturally decreased over time, starting from safe values at the beginning of the simulation and steadily converging to near-zero at the impact point. In this way, the entire operational domain of interest was covered, from negligible risk to imminent collision.

During the simulation, a CSV log file was generated containing all relevant TTC metrics, with one entry recorded for each time instant in which the TTC was computed. For every TTC value collected, the corresponding TIP was then derived according to the three candidate strategies under consideration:

- fixed intervals,
- linear mapping,
- exponential mapping.

Finally, for each method the behaviour of the TIP values was analysed and the triggering of VAMs was verified according to the ETSI 10% variation condition. This procedure provided a systematic and comparable basis to assess the advantages and limitations of each strategy, and to guide the selection of the final approach.

The fixed-interval approach, although simple, revealed abrupt discontinuities. Small variations in TTC around an interval boundary produced large jumps in TIP, leading to multiple artificial triggers (24 probability-change triggers in Scenario 5, as shown in the slides) and to long flat regions with no updates despite an increasing collision risk. The linear mapping, adopted in the literature by Chen et al. [34], provided smoother curves but suffered from uniform sensitivity: it overestimated risk at long horizons and underestimated urgency at short ones, producing either premature or delayed triggers. In Scenario 5, this led to redundant transmissions due to oscillations around the 10% threshold. The exponential mapping, inspired by [**Belmekki2024**], better reflected the nonlinear escalation of urgency and reduced premature warnings, but its asymptotic behaviour prevented saturation at TIP=1 and TIP=0. In practice, this resulted in unstable behaviour, with multiple small TIP variations around the triggering threshold (13 probability-change triggers in Scenario 5, still excessive for ETSI compliance).

Based on these observations, the final choice converged towards a hybrid solution: discrete mapping with exponential growth. This method integrates the boundedness and interpretability of interval-based approaches with the realism of exponential escalation. The TTC domain is divided into N discrete intervals between TTC_{min} and TTC_{max}, but instead of being assigned constant values, each interval is weighted according to an exponential law:

Let N be the number of discrete levels and $[TTC_{min}, TTC_{max}]$ the operative domain. The interval boundaries are defined as:

$$\text{interval}(i) \;=\; TTC_{min} + \Big(TTC_{max} - TTC_{min}\Big) \cdot \frac{e^{\,\text{growthrate}\cdot i/N} - 1}{e^{\,\text{growthrate}\cdot N} - 1} \quad i = 0,1,\ldots,N \tag{4.23}$$

with $\text{interval}(0) = TTC_{min}$ and $\text{interval}(N) = TTC_{max}$. By doing this, N interval widhts:

$$\Delta_i = interval(i) - interval(i-1) \tag{4.24}$$

follow a geometric progression that produces narrower bins near TTC_{min}, highlighting the higher sensitivity due to the increasing possibility of a collision, and wider ones near TTC_{max}, suggesting less urgency in terms of risk of impact.

Once the TTC sample falls into the i-th slot, the corresponding discrete probability is assigned as:

$$TIP(i) \;=\; 1 - \frac{i}{N}\,, \tag{4.25}$$

so that TIP equals 1 when $i = 0$ (i.e. $TTC \leq TTC_{min}$, unavoidable collision) and decreases step by step down to 0 when $i = N$ (i.e. $TTC \geq TTC_{max}$, negligible risk). This ensures that smaller TTC values correspond to higher collision probabilities, as required by the logic of the triggering mechanism.

The definition of the growthrate parameter is crucial. It regulates how quickly the TIP values increase across the intervals. A small growthrate (close to zero) yields almost linear spacing, while a large growthrate produces a highly nonlinear distribution, with low sensitivity at long horizons and sharp escalation as TTC approaches the critical threshold. The correct computation of growthrate therefore depends on the desired sensitivity profile. In our work, growthrate was tuned as a design parameter, ensuring that the escalation within the interval domain reflected the nonlinear perception of risk while remaining ETSI-compliant. Following the approach suggested by [10], we defined growthrate relative to the maximum horizon TTC_{max}, so that the scaling adapts automatically to different operational ranges. This guarantees that the mapping remains consistent across scenarios with different maximum TTC values, while providing the flexibility to adjust the steepness when higher or lower sensitivity is required.

Applied to the simulation scenarios, the discrete exponential mapping showed the most balanced behaviour. In Scenario 5, it triggered 13 VAMs due to probability changes, a significant reduction compared to the 24 of fixed intervals, while maintaining 12 mandatory triggers under the TTC_{min} threshold, identical to the other methods. In Scenario 8, where the dynamics were less critical, it produced only one probability-change trigger compared to ten with fixed intervals, again proving its efficiency. The method therefore eliminated the discontinuities of fixed

intervals, the premature warnings of linear mapping, and the instability of pure exponential functions, while still capturing the nonlinear escalation of urgency.

In conclusion, discrete mapping with exponential growth emerged as the most suitable strategy, striking a balance between theoretical soundness, practical applicability, and compliance with ETSI standards. Its flexibility through the growthrate parameter, its deterministic thresholds, and its empirical validation in simulation make it a robust choice for real-time triggering in cooperative VRU protection.

4.4 Triggering logic implementation

The methodological path developed throughout this thesis led to the design and implementation of the logic for triggering VAMs based on the Trajectory Interception Probability, which is fully represented by the combination of the pseudocodes in Algorithm 1 and Algorithm 6. This constitutes the central contribution of the work, as it translates the analytical reasoning and comparative evaluation of mapping strategies into a concrete procedure that can be directly integrated into VaN3Twin. The final logic is described below:

Whenever a CAM or VAM is received by a pedestrian entity, the Local Dynamic Map (LDM) is updated to incorporate the new information. As part of this update, the TIP with respect to the sender of the message is recomputed and stored, alongside all other relevant data, in the LDM of the receiving stationID. Subsequently, during the periodic evaluation of triggering conditions, performed every $100\,ms$, an additional check has been introduced compared to the previous implementation [30]. Specifically, for each element in the LDM, the current TIP is compared with the value associated to the last VAM transmitted by the receiver. If any variation is greater than or equal to 10%, the triggering condition is satisfied and a new VAM is generated and broadcast. Furthermore, whenever a new VAM is triggered (independently of the condition that caused it), the LDM is updated so that the current TIP becomes the new reference value for subsequent comparisons, ensuring consistency in the evaluation of this condition. This design directly implements the requirement defined in [8], which specifies that a new VAM should be triggered whenever the reported TIP differs by at least 10% from the last transmitted value. By construction, this mechanism makes the algorithm stateful, since each decision depends not only on the current situation but also on its relation to the previously reported risk level. It also acts as a filter for stability: minor oscillations or measurement noise do not lead to unnecessary transmissions, preventing redundant use of the communication channel.

The core of the logic lies in the computation of TIP itself, described by Algorithm 1. The procedure responsible for this task, TIP, starts by retrieving the kinematic states of the VRU and the interacting entity: positions, velocities, and

accelerations form the input set required to project motion and estimate collision risk. Before performing any detailed calculation, however, the algorithm evaluates whether the other entity is close enough to be relevant through the IsInRange function. This step is fundamental, as it avoids spending computational resources on distant or diverging actors, preventing the generation of misleading risk values. Instead, if the entities are within a predefined interaction range, the algorithm proceeds.

The next stage is the computation of Time to Collision. This temporal metric captures how long it would take, under current kinematic conditions, for the trajectories of the VRU and the other entity to intersect. Once obtained, the TTC value is immediately compared against two predefined thresholds: a minimum value TTC_{min} and a maximum value TTC_{max}. If the TTC is smaller than TTC_{min} and non-negative, the situation is considered critical: the time available for reaction is insufficient, and the collision is effectively unavoidable. Conversely, if the TTC is greater than TTC_{max} or equal to -1 (a sentinel value indicating no intersection is expected), the encounter is deemed irrelevant for triggering purposes. These thresholds enforce deterministic saturation: below TTC_{min} the TIP is fixed to one, while above TTC_{max} the TIP is fixed to zero. This ensures full compliance with ETSI requirements, which call for clear boundaries rather than asymptotic behaviours, and provides interpretability to system designers and testers.

For TTC values lying within the interval (TTC_{min}, TTC_{max}), the algorithm performs a further check of spatial plausibility. This is achieved through the computation of the Space to Collision (s2c), which measures the minimum separation distance expected at the predicted interception time. If the separation is larger than a design threshold $s2c_{th}$, the potential collision is deemed spurious: the TTC may suggest that trajectories cross, but in practice the lateral or longitudinal offset is sufficient to prevent any real risk. Only if the $s2c$ is below its threshold does the algorithm proceed to the actual probability mapping.

At this point, the TTC is converted into a TIP value through the discrete exponential mapping that was developed in the thesis. This choice combines the robustness of discrete quantisation with the realism of exponential escalation. The interval between TTC_{min} and TTC_{max} is subdivided into N bins, whose boundaries are defined exponentially rather than linearly. The formula is:

$$\text{interval}(i) = TTC_{min} + (TTC_{max} - TTC_{min}) \frac{e^{\,\text{growthrate}\cdot i/N} - 1}{e^{\,\text{growthrate}\cdot N} - 1}, \quad i = 0, \ldots, N, \tag{4.26}$$

with $\text{interval}(0) = TTC_{min}$ and $\text{interval}(N) = TTC_{max}$.

This construction produces narrower bins close to TTC_{min}, where sensitivity must be higher, and wider bins near TTC_{max}, where variations are less relevant. Once the current TTC has been assigned to one of these bins, the TIP is computed as:

$$TIP(i) = 1 - \frac{i}{N} \tag{4.27}$$

This ensures that smaller TTC values correspond to higher TIPs, with deterministic saturation at the extremes. The growth-rate parameter regulates how quickly bin widths expand as i increases, effectively controlling the balance between resolution near imminent collisions and coarseness at safe horizons. In practice, a larger growth-rate compresses bins near TTC_{min}, increasing reactivity, while a smaller one approaches uniform spacing. This parameter is therefore a powerful design knob, offering flexibility to adapt the mapping to different contexts while maintaining ETSI compliance. Overall, the pseudocode is a practical implementation of the carefully layered methodology developed throughout the thesis: a range check to filter out irrelevant actors, a temporal check to enforce deterministic bounds, a spatial plausibility gate to avoid incorrect predictions, and finally a mapping step that discretises TTC into TIP through an exponential law, the core of the work. Each of these stages contributes to a robust and interpretable mechanism, that allows to trigger VAMs reliably in cooperative VRU protection scenarios. The use of thresholds and discrete steps guarantees compliance with the ETSI 10% variation rule, while the exponential spacing of intervals reflects the nonlinear escalation of risk as TTC decreases.

Algorithm 1 Main pseudocode for TIP computation

Require: CAM/VAM received by pedestrian
Ensure: Current TIP in respect to the $stationID$ of the sender is stored in the LDM

procedure UPDATELDM
  $newInterjectionProbability \leftarrow$ TRAJECTORYINTERJECTIONPROBABILITY()
  Store $newInterjectionProability$ in LDM
end procedure

procedure TRAJECTORYINTERJECTIONPROBABILITY
  Retrieve $\vec{p}_v, \vec{p}_w, \vec{s}_v, \vec{s}_w, \vec{a}_v, \vec{a}_w$
  $r \leftarrow$ ISINRANGE$(\vec{p}_v, \vec{p}_w, \vec{s}_v, \vec{s}_w, TTC_{max})$
  if $r =$ true **then**
    $t2c \leftarrow$ COMPUTETIMETOCOLLISION$(\vec{p}_v, \vec{p}_w, \vec{s}_v, \vec{s}_w, \vec{a}_v, \vec{a}_w)$
    if $TTC \leq TTC_{\min}$ **and** $TTC \geq 0$ **then**
      $s2c \leftarrow$ COMPUTESPACETOCOLLISION$(\vec{p}_v, \vec{p}_w, \vec{s}_v, \vec{s}_w, \vec{a}_v, \vec{a}_w, TTC)$
      if $s2c \leq s2c_{th}$ **then**
        return 1
      else
        return 0
      end if
    end if
    if $t2c \geq t2c_{\max}$ **or** $t2c = -1$ **then**
      return 0
    else
      $s2c \leftarrow$ COMPUTESPACETOCOLLISION$(\vec{p}_v, \vec{p}_w, \vec{s}_v, \vec{s}_w, \vec{a}_v, \vec{a}_w, TTC)$
      if $s2c \leq s2c_{th}$ **then**
        return COMPUTETIP$(TTC, growthrate, num_intervals, TTC_{\min}, TTC_{\max})$
      else
        return 0
      end if
    end if
  else
    return 0
  end if
end procedure

In VaN3Twin, the entire logic has been implemented within the t2c C++ class, where each of the functions described above is implemented as a separate module.

This modular approach guarantees clarity, reusability, and scalability: each block can be executed independently, while contributing to the overall pipeline that leads from TTC computation to the TIP mapping and VAM triggering.

The state information required by the class is retrieved from two different sources. For the sender, position, velocity, and acceleration vectors are directly extracted from the content of the received message, VAMs in the case of VRUs and CAMs in the case of vehicles. For the receiver, the same parameters are obtained through the m_traciclient interface, which provides access to the live mobility data of all entities present in the SUMO simulation. In this way, both actors in a potential interaction are fully characterized at each computation step.

Regarding the parameters that control the mapping from TTC to TIP, which are growthRate, TTC_min, TTC_max, and n_intervals these are defined within the v2p-vam-80211p application. When a new t2c object is instantiated inside the BSContainer, the chosen parameter values are passed to its constructor. This design ensures that the t2c class remains focused on computation, while the configuration logic is handled externally, leaving the parameters' choice to the users, allowing them to adjust the behaviour of the mapping function at runtime without modifying the internal implementation of the class.

With this structure, the necessary inputs are defined and available. The t2c class can therefore execute the full chain of functions, which are described more in detail in the following sections, before applying the TIP mapping and deciding whether the triggering condition for VAM transmission must be activated.

4.4.1 IsInRange: filtering entities

The IsInRange function, whose pseudocode is available in Algorithm 2, is used as a preliminary filtering step to decide whether it is necessary to execute the whole collision detection procedure. The rationale is simple: if two entities (a vehicle and a VRU, or two VRUs) are not expected to get sufficiently close within a predefined temporal horizon TTC_{max}, then a collision is highly unlikely, and running the entire algorithm would only result in unnecessary computational overhead. Introducing this check allows the system to optimize execution time, which is particularly relevant in realistic scenarios with a large number of road users exchanging messages.

The function operates by first estimating the positions that both entities will reach after $t2c_{th}$ seconds under the assumption of constant velocity (lines 1–4 of Algorithm 2). From these projected positions, it computes the distances that each entity will have travelled in this time interval (lines 5–6). The larger of the two values is taken as a reference (line 7), corresponding to the fastest entity.

Next, the function derives a so-called range of action around the faster entity (line 8). This range, d_{max}, is not set equal to the travelled distance itself, but

rather computed as the hypotenuse of a right-angled triangle whose two sides are both equal to that distance, effectively scaling the radius by a factor of $\sqrt{2}$. This more conservative approach introduces a safety margin, accounting for possible unexpected changes in speed or direction.

Finally, the actual Euclidean distance between the two entities is computed (line 9). If this distance is smaller than d_{max}, it means that the slower entity lies within the action range of the faster one, and therefore the system triggers the full collision detection procedure (lines 10–14). Otherwise, the function returns false and the pair is discarded from further analysis.

In summary, IsInRange serves as an efficient pre-check mechanism: it reduces the number of unnecessary collision detection calls, while at the same time ensuring that no potentially critical situation is ignored. This makes it a crucial component for scalability when implementing TTC- and TIP-based methods in VaN3Twin, especially in dense urban environments with many VRUs.

Algorithm 2 IsInRange($\vec{p}_v, \vec{p}_w, \vec{s}_v, \vec{s}_w,, TTC_{max}$) function

Require: $\vec{p}_v, \vec{p}_w, \vec{s}_v, \vec{s}_w$, max_s, TTC_{max}
Ensure: true if the distance between the two vehicles is small enough to call the collision detection algorithm, false otherwise

$p^*_{v_x} \leftarrow p_{v_x} + s_{v_x} \cdot TTC_{max}$ ▷ future position of vehicle v (x-axis)
2: $p^*_{v_y} \leftarrow p_{v_y} + s_{v_y} \cdot TTC_{max}$ ▷ future position of vehicle v (y-axis)
$p^*_{w_x} \leftarrow p_{w_x} + s_{w_x} \cdot TTC_{max}$ ▷ future position of vehicle w (x-axis)
4: $p^*_{w_y} \leftarrow p_{w_y} + s_{w_y} \cdot TTC_{max}$ ▷ future position of vehicle w (y-axis)
$d_v \leftarrow \sqrt{(p^*_{v_x} - p_{v_x})^2 + (p^*_{v_y} - p_{v_y})^2}$ ▷ distance traveled by v
6: $d_w \leftarrow \sqrt{(p^*_{w_x} - p_{w_x})^2 + (p^*_{w_y} - p_{w_y})^2}$ ▷ distance traveled by w
$d \leftarrow \max(d_v, d_w)$
8: $d_{\max} \leftarrow \sqrt{2} \cdot d$
$d_{vw} \leftarrow \sqrt{(p_{v_x} - p_{w_x})^2 + (p_{v_y} - p_{w_y})^2}$ ▷ distance between the two entities
10: **if** $d_{vw} < d_{\max}$ **then**
    return true
12: **else**
    return false
14: **end if**

4.4.2 ComputeTimeToCollision

The ComputeTimeToCollision function is the core component for estimating the temporal horizon to a potential collision between two entities. Its goal is to determine the time instant $t2c$ at which the distance between them will be minimized, under the assumption of known positions, velocities, and accelerations.

The function first defines the expression for the relative distance $D(t)$ between the two entities as a function of time (line 1 of Algorithm 3). By taking the derivative of this distance and solving $\frac{dD(t)}{dt} = 0$, the algorithm identifies the set of candidate instants $\mathcal{T}$ at which the distance reaches a minimum (line 2).

If all the candidate instants are negative, it means that the two entities were closer in the past and are currently diverging, so no future collision is possible. In this case, the function assigns $t2c = -1$ (line 4). Otherwise, the algorithm selects the smallest positive solution, i.e., the first future instant at which the entities approach each other most closely (line 6). This value is then returned as the predicted time-to-collision (line 8).

Importantly, the convention $t2c = -1$ is used throughout the implementation to indicate the absence of a collision path. This makes the output easy to interpret: positive values correspond to the predicted time remaining before collision, while a negative result signals that no collision will occur.

In practice, this function provides the temporal backbone for the entire methodology: it not only feeds the SpaceToCollision computation but also serves as the basis for mapping TTC values into collision probability. Its ability to account for acceleration makes it more robust than simpler time-based indicators such as PET or PPE, and better suited for realistic traffic scenarios involving VRUs.

Algorithm 3 TimeToCollision($\vec{p}_v, \vec{p}_w, \vec{s}_v, \vec{s}_w, \vec{a}_v, \vec{a}_w$) function

Require: $\vec{p}_v, \vec{p}_w, \vec{s}_v, \vec{s}_w, \vec{a}_v, \vec{a}_w$
Ensure: time-to-collision ($t2c$)
Define equation for $D(t)$, the distance between the two entities
$\mathcal{T} \leftarrow \{t \mid \frac{dD(t)}{dt} = 0\}$ ▷ set of times when distance is minimum
if all elements of $\mathcal{T}$ are < 0 **then**
$t2c \leftarrow -1$
else
$t2c \leftarrow \min_{t>0} \mathcal{T}$
end if
return TTC

4.4.3 ComputeSpaceToCollision: trajectory filtering

The ComputeSpaceToCollision function is designed to estimate the spatial separation that will exist between two entities (e.g., a vehicle and a VRU) at the specific instant when a potential collision is expected to occur. In other words, while the TimeToCollision function provides a temporal horizon, SpaceToCollision translates this horizon into a physical distance.

The function begins by projecting the future positions of both entities after

TTC seconds, this time explicitly including not only the effect of their current positions and velocities, but also their accelerations (lines 1–4 of Algorithm 4). This formulation makes the prediction more realistic, as it accounts for dynamic changes in motion rather than assuming constant speed.

Once the predicted positions are obtained, the function computes the Euclidean distance between them (line 5). This value, denoted as space-to-collision ($s2c$), indicates how far apart the two entities are expected to be at the time of the potential collision.

If $s2c$ equals zero, the entities are expected to physically collide. Conversely, a positive $s2c$ value suggests that, although their trajectories come close in time, a residual distance will remain between them. This distinction is particularly important for filtering false positives: entities with a low TTC but a high $s2c$ can be safely ignored, as they will not actually intersect in space.

In summary, ComputeSpaceToCollision complements the TTC computation by providing a spatial perspective on the predicted interaction. Used together, the two functions enable VaN3Twin to discriminate between purely temporal overlaps and actual collision risks, which is especially valuable in scenarios involving VRUs where spatial margins are often very limited.

Algorithm 4 SpaceToCollision($\vec{p}_v, \vec{p}_w, \vec{s}_v, \vec{s}_w, \vec{a}_v, \vec{a}_w, TTC$) function

Require: $\vec{p}_v, \vec{p}_w, \vec{s}_v, \vec{s}_w, \vec{a}_v, \vec{a}_w, t2c$
Ensure: space-to-collision (s2c)
$p^*_{v_x} \leftarrow p_{v_x} + s_{v_x} \cdot TTC + \frac{1}{2} a_{v_x} \cdot TTC^2$
$p^*_{v_y} \leftarrow p_{v_y} + s_{v_y} \cdot TTC + \frac{1}{2} a_{v_y} \cdot TTC^2$
$p^*_{w_x} \leftarrow p_{w_x} + s_{w_x} \cdot TTC + \frac{1}{2} a_{w_x} \cdot TTC^2$
4: $p^*_{w_y} \leftarrow p_{w_y} + s_{w_y} \cdot TTC + \frac{1}{2} a_{w_y} \cdot TTC^2$
$s2c \leftarrow \sqrt{(p^*_{v_x} - p^*_{w_x})^2 + (p^*_{v_y} - p^*_{w_y})^2}$
return $s2c$

4.4.4 ComputeTrajectoryInterjectionProbability

The function ComputeTIP is the last piece of the main logic: it implements the core mapping between the continuous TTC domain and the discrete probability values that are going to be stored in the LDM to be subsequently evaluated in the correspondent triggering condition. Once a valid TTC has been obtained and verified to lie within the admissible range $[TTC_{\min}, TTC_{\max}]$, the first step is to normalize it with respect to these thresholds (line 4). This ensures that the mapping only depends on the value position in the chosen range. The normalized TTC value is then projected into a non-linear space through an exponential scaling factor (lines 5–6), controlled by the growthrate parameter and the number of discrete

intervals, to highlight that the collision probability should not increase linearly with decreasing TTC: risks grow slowly at larger TTC values, but accelerate rapidly as the collision becomes imminent, leading to a higher sensitivity in the region near TTC_{min}.

The intermediate variable x (line 6) is computed through the inverse of the exponential mapping, in order to actually identify the index of the interval corresponding to the current TTC, which is later discretized via a floor operation and bounded to avoid negative values (line 7). The actual TIP value is obtained by inverting the index with respect to the total number of intervals (line 8), so that lower TTC values correspond to higher TIP levels. Finally, the result is rounded to one decimal place (line 9), both to comply with the ETSI specification that requires a variation of at least 10% in TIP to trigger a new VAM, and to prevent spurious oscillations that could be introduced by minimal floating-point differences.

In summary, this function transforms the continuous dynamics of motion into a discrete, probabilistic representation of collision risk that is lightweight, interpretable, and directly usable for triggering conditions. By combining thresholding, exponential growth, and discretization, the mapping ensures that TIP values remain stable at longer time horizons but respond aggressively when a collision is imminent, which is exactly the behaviour expected in VRU protection systems.

Algorithm 5 ComputeTIP($t2c$, $growthrate$, $numIntervals$, $TTC_{\min}$, $TTC_{\max}$) function

Require: $t2c, growthrate, numIntervals, TTC_{\min}, TTC_{\max}$
Ensure: TIP
$scaled \leftarrow \dfrac{t2c - TTC_{\min}}{TTC_{\max} - TTC_{\min}}$
$norm \leftarrow \exp(growthrate \cdot numIntervals) - 1$
$x \leftarrow \dfrac{\log(1 + scaled \cdot norm)}{growthrate}$
$interval \leftarrow \max(0, \lfloor x \rfloor)$
5: $TIP \leftarrow 1 - \dfrac{interval}{numIntervals}$
$TIP \leftarrow \text{round}(TIP, 1)$
return TIP

4.4.5 TIP checks and updates

To provide a complete picture of the proposed methodology, it is also necessary to describe how the new triggering condition is actually verified in order to decide whether a new VAM should be generated and transmitted: the definition and implementation of this procedure within VaN3Twin was one of the specific contributions of this thesis. In particular, starting from the method defined by [30] in

the VRUBasicService class, named checkVamConditions, the TIP-based triggering condition was integrated, following the logic described in Algorithm 6.

The mechanism is straightforward: at every evaluation step, each VRU iterates over the elements stored in its Local Dynamic Map (LDM). For every entry, the current TIP value is compared against the one associated with the last VAM transmitted by the same VRU. If the difference between the two values is greater than or equal to 0.1, the triggering condition is satisfied, and a new VAM is generated and transmitted. The condition has been placed immediately after the four standard ETSI checks (heading, position, speed, and elapsed time) but before the verification of safe distances. This choice reflects its conceptual role: while the standard checks react to direct variations in state variables, the TIP condition encapsulates a probabilistic assessment derived from them, thus fitting naturally as a complementary evaluation. At the same time, placing it before the safe distances check ensures that probabilistic risk variations are prioritized when present, while the distance-based condition remains available as a conservative safeguard.

Finally, it is important to highlight that whenever a VAM is transmitted, the VRU updates each entry of its Local Dynamic Map so that the most recently computed TIP values become the new reference values associated with the transmission. This ensures that subsequent evaluations are always performed against the last disseminated state, avoiding redundant triggers caused by outdated comparisons and maintaining consistency in the assessment of future changes.

Algorithm 6 TIP-based triggering condition (executed every 100 ms)

```
1: procedure TIPTriggeringCheck
2:     for all elements e in LDM do
3:         if |TIP_current(e) − TIP_lastVAM(e)| ≥ 0.1 then
4:             return Trigger VAM()
5:         end if
6:     end for
7: end procedure
```

Chapter 5

Analysis and Results

This chapter presents the outcomes of the simulations carried out in VaN3Twin with the TIP-based triggering condition. Each of the exploited scenarios is described in Section 5.1, while the logic behind the choice of the parameter sets is explained in Section 5.2. Then, the actual analysis and results obtained are extensively explained: the objective is to evaluate how the new condition integrates with the existing ETSI triggers and how its contribution evolves as traffic density and interaction complexity increase.

5.1 Simulation scenarios

To analyse and validate the methodology developed in this thesis, all simulations were carried out within the VaN3Twin framework using a dedicated and realistic urban scenario. The environment used is the same as in [30], to ensure methodological consistency with the other triggering conditions previosuly defined and the possibility of direct comparison. In detail, the simulation environment reproduces a section of Corso Castelfidardo, located near the Politecnico di Torino campus, which offers an urban structure particularly suitable for Vehicle-to-Pedestrian (V2P) interaction and collision avoidance studies: the selected portion of the map features two central roads composed of two lanes each, flanked by two secondary single-lane roads. On both sides, wide sidewalks are accessible only to pedestrians, while a dedicated bicycle lane runs along the right edge of the map, currently not leveraged in order to maintain the focus on pedestrian–vehicle interactions. Three pedestrian crossings are positioned at the upper, middle, and lower parts of the segment, allowing multiple and realistic interaction points between vehicles and VRUs. This configuration offers a balanced trade-off between geometric simplicity and interaction density, making it ideal for studying the evolution of collision risk and VAM triggering behaviour.

The map was downloaded from the OpenStreetMap database [35], converted into a .net.xml file using the `netconvert` tool, later refined through `netedit`. During this refinement process, sidewalks and pedestrian areas were prioritised to ensure proper accessibility, while all traffic lights were configured to remain constantly green for both vehicles and pedestrians. This simplification was introduced to guarantee continuous movement and to increase the odds for potential collision conditions to present themselves, leading to the generation of meaningful risk events for the analysis. Two different configurations were used throughout the study, both based on the same map and simulation duration of 45 seconds, which are characterized in two different scenarios. The first one, which will be referred to as the preliminary scenario, consists of a single vehicle and one pedestrian placed along intersecting trajectories that unavoidably lead to a collision at the central crossing, which is forced by neutralizing all the preventive measures SUMO takes autonomously in order to prevent dangerous situations. This setup served as a controlled environment for selecting and fine-tuning the parameters of the proposed methodology: in particular, we are referring to the choice of set of values for the TTC thresholds, the growthrate and the space to collision limit. Limiting the interaction within the simulation to a single pair of entities allowed for the detailed ananlysis regarding the evolution of the triggering behaviour under varying parameter configurations, whose results were later leveraged to assess the sensitivity of the system's responsiveness. After this calibration phase, the same spatial and temporal configuration was extended into a complete scenario, shown in 5.1, which includes a total of 24 pedestrians and 19 vehicles. Within this richer environment, the same pedestrian–vehicle pair from the preliminary setup was reintroduced among the other road users, ensuring the presence of at least one explicitly forced collision event while maintaining a realistic background traffic flow. This comprehensive configuration was used for the core part of our anlaysis, regarding the comparison between the pre-TIP and post-TIP implementations, as well as the evaluation of the overall efficacy, selectivity, and relevance of the developed TIP-based triggering condition.

Figure 5.1: Complete simulation scenario used in VaN3Twin for the analysis of the TIP-based triggering condition.

Using the same environment for both scenarios ensures methodological continuity: the first phase provides a controlled basis for parameter selection, while the second delivers the validation and performance assessment under more complex and realistic traffic dynamics. This approach guarantees that all observed differences in system behaviour can be exclusively attributed to the introduction of the TIP condition, rather than to scenario-dependent variations, which further reinforce the robustness and reproducibility of the results that were obtained and studied.

5.2 Parameters' choice

Before carrying out the core analysis regarding the effects of the new TIP triggering condition, it was necessary to determine a correct set of parameters for the TTC-to-probability mapping function. This step was crucial to ensure that the methodology maintained two main characteristics: being realistic and consistent with ETSI requirements. At the same time, it was fundamental to avoid configurations that would lead to extreme or unbalanced behaviours. The four parameters that were investigated were the minimum and maximum TTC thresholds (TTC_{min} and TTC_{max}), the growth rate, which controls the steepness of the exponential

increase of collision probability, and the space to collision limit, to avoid situations where the temporal urgency signalled is actually realistic impossible due to the spatial positions and trajectories of the entities taken into consideration. The selection process was based on a series of systematic simulations carried out on the preliminary scenario defined in Section 5.1. The following subsections present the rationale and results for each parameter, supported by the graphs obtained during the exploration campaign.

5.2.1 TTC thresholds

There are two thresholds to be defined in order to set a realistic range for the Time-to-Collision (TTC) within the Trajectory Interception Probability (TIP) computation: the lower bound, corresponding to the minimum TTC threshold (TTC_{min}), and the upper bound, corresponding to the maximum TTC threshold (TTC_{max}).

The parameter TTC_{min} defines the limit below which the collision probability saturates to 1, meaning that the time left before impact is too short to allow any preventive action. To assess the effect of this threshold, simulations were conducted by varying TTC_{min} from 0.5 s to 4.5 s in steps of 0.5 s, for different TTC_{max} values (5 s, 10 s, 15 s, and 20 s).

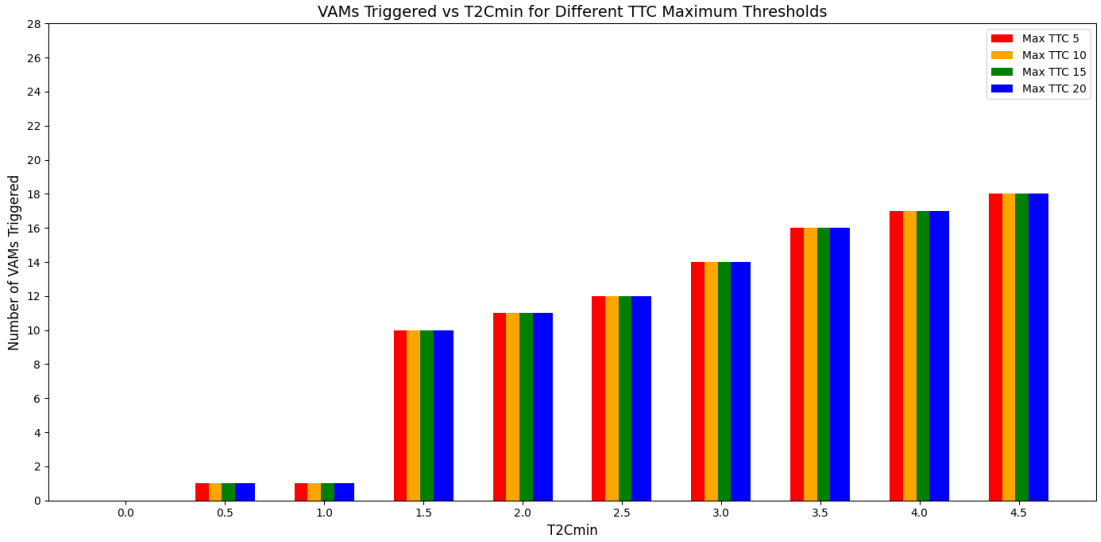

Figure 5.2: Number of potential transmissions corresponding to frames where TIP = 100%, as a function of TTC_{min} for different TTC_{max} thresholds.

Figure 5.2 does not represent the number of VAMs actually transmitted according to the ETSI triggering logic, but rather the number of potential transmissions

that would occur if, in addition to the +/-10% variation rule, the system also transmitted a message whenever the TIP reached 100%. This analytical metric provides a useful quantitative indication of how frequently the probability function reaches saturation before the TTC resets to -1 from being 100%, which happenes whenever the moment of possible collision passes and situation is not critical anymore, whether because it happened or due to the change of conditions within the environment.

The observed trend confirms that:

- **Very low TTC_{min} values (0.5–1.0 s)**: lead to almost no TIP = 100% occurrences, since the probability saturates only in the very last instants before impact, when the event is practically unavoidable.
- **Intermediate TTC_{min} values (1.5–3.0 s)**: yield a moderate number of near-collision situations, meaning the model still distinguishes between low-risk and imminent-risk conditions.
- **High TTC_{min} values (> 3.5 s)**: lead to premature saturation, resulting in basically continuous detection of high-risk conditions, which would translate into excessive triggering if applied in real time.

From this analysis, combined with the reference literature presented when the TTC thresholding reasoning was introduced in Chapter 4, the possible values to leverage in the mapping function for this parameter that were selected are:

$$TTC_{min} = \{1.5,\ 2.5,\ 3.0\}\,\text{s}.$$

These values capture a balanced trade-off:

- **1.5 s**: corresponds to a reactive but selective configuration, marking only the truly imminent collisions;
- **2.5 s**: represents a realistic average response horizon for typical VRU–vehicle interactions in urban contexts;
- **3.0 s**: acts as a conservative upper bound, anticipating the warning in borderline situations while still avoiding early saturation.

These thresholds allow the exploration of the model behavior from a strictly ETSI-compliant conservative setup (1.5 s) to a more proactive risk signaling configuration (3.0 s), while maintaining coherence with empirical evidence and ensuring that the framework does not unrealistically over-trigger near-collision alerts.

Moving on, the next parameter that was explored was TTC_{max}, which represents the maximum horizon beyond which collision probability is forced to zero, as the

situation is considered safe and no criticalities are expected, assuming the same configuration is mantained. Therefore, it determines how far into the future the system considers a potential collision relevant, which would beg the necessity of a TIP computation. Figure 5.3 shows the number of VAMs triggered according to the ETSI-defined triggering condition while varying the growth rate of the TIP mapping for different TTC_{max} values and for three representative TTC_{min} settings (1.5 s, 2.5 s, and 3.0 s).

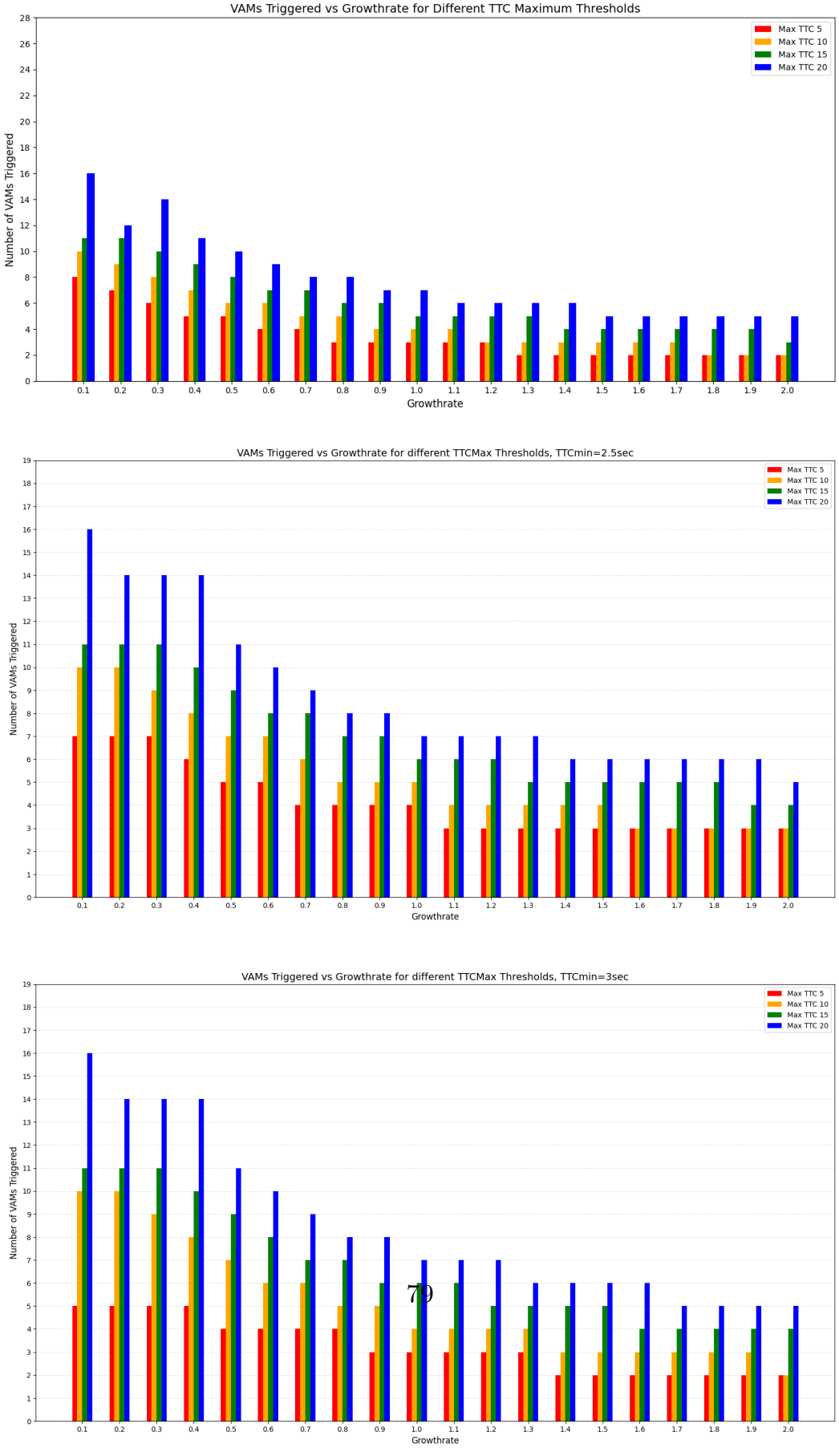

Figure 5.3: VAMs triggered as a function of the TIP growth rate, for different TTC_{max} horizons and TTC_{min} values previously chosen.

Overall, the results show a clear monotonic relationship between the growth rate and the number of triggered transmissions: as the growth rate increases, the number of VAMs decreases. This occurs because a steeper rise in the TIP curve compresses the transition phase between TTC_{max} and TTC_{min} into fewer frames, thereby reducing the number of +/-10% increments that can generate new messages. The differences among the curves corresponding to distinct TTC_{max} values are more evident with the usage of low growth rates, where the TIP evolves slowly and it is exposed to more intermediate states. In such conditions, larger TTC_{max} horizons encompass a longer portion of the approach and capture more potential risk variations, while smaller horizons cut off many relevant situations. As the ramp steepens, the distance between curves narrows, indicating that the influence of TTC_{max} progressively diminishes when the TIP dynamics are too fast.

Although this first analysis reveals how the shape of the mapping (i.e., the growth rate) affects the frequency of transmissions, in this phase the focus remains on the TTC thresholds themselves, as they determine the effective time horizon over which the collision probability is evaluated. The growth-rate parameter will be further discussed later, once the TTC range is consolidated. Concentrating on the thresholds allows us to isolate their effect on the anticipatory capability of the system and to establish realistic temporal boundaries for risk estimation.

From this perspective, it becomes evident that TTC_{max} plays a crucial role in defining the system's temporal foresight. A very short horizon, such as $TTC_{max} = 5$ s, is not adequate because it excludes numerous relevant interactions and fails to capture gradual risk increases, leading to systematic underestimation of near-collision conditions and reduced situational awareness. On the contrast, a very long horizon, such as $TTC_{max} = 20$ s, can theoretically be able of detecting early variations in the TIP: however, it extends the operative window beyond the realistically perceivable or controllable domain for both vehicles and pedestrians, causing the system to react with too much anticipation in response to uncertain or non-critical events, increasing the communication load without actually providing a correspondent safety advantage. Furthermore, the marginal benefit between 15 s and 20 s is minimal for most of the configurations, confirming that such extended horizons add redundancy rather than useful anticipation.

For these reasons, $TTC_{max} = 10$ s emerges as the most suitable configuration value. This horizon offers a balanced trade-off between responsiveness and selectivity: it is long enough to include the relevant portion of the approach phase, but also short enough to prevent premature triggering on non-critical interactions. The 10 s threshold represents a realistic and widely adopted anticipation horizon, consistent with reaction and braking times reported in literature for urban VRU-vehicle encounters. Finally, looking at the comparison across TTC_{min} values confirms the stability of these horizons, supporting their adoption as reference parameters for the next stages of evaluation. After careful considerations done

on these observations and on the results discussed regarding the lower bound of the TTC range, the combination $TTC_{min} = 1.5$ s and $TTC_{max} = 10$ s was selected as the baseline configuration for the following steps. This choice ensures a coherent balance between early risk perception and communication efficiency, enabling a timely ETSI-compliant triggering behavior, aligned with real-world reaction capabilities.

5.2.2 Growthrate

The third parameter explored was the growth rate, which controls the steepness of the exponential increase of collision probability between TTC_{max} and TTC_{min}. As it was firstly observed in the previous section, lower values of this parameter generate a smoother and slower increase of the probability, delaying its escalation towards saturation, while higher values produce a much steeper progression, resulting in sudden jumps and early saturation. To evaluate its effect, simulations were carried out by varying the growth rate from 0.1 to 2.0 in increments of 0.1, in combination with all previously defined threshold pairs for TTC_{min} and TTC_{max}.

Since the overall behaviour of the different growth rates was found to be qualitatively consistent across the tested threshold combinations, the analysis is presented here for one representative configuration, with $TTC_{min} = 1.5$ s and $TTC_{max} = 10$ s, which corresponds to the baseline previously selected. Figure 5.4 illustrates the resulting evolution of collision probability over time for the different growth rates.

The results highlight a clear dependency between the shape of the curve and the dynamic behaviour of the probability. Growth rates below 0.3 lead to unrealistically flat profiles, where the collision probability remains low for most of the approach and increases only in the very last instants. This behaviour could delay the triggering of awareness messages and compromise the responsiveness of the system in real scenarios. Conversely, growth rates above 1.0 produce extremely steep transitions, where the probability abruptly rises from near zero to saturation in a very short time span. This generates unstable behaviour, characterized by sudden jumps that could lead to oscillations in the triggering condition, especially when the TTC fluctuates around the threshold boundaries.

Intermediate values between 0.4 and 0.6 provide the most balanced evolution of the curve, with a progressive yet timely rise in collision probability. In this range, the TIP mapping preserves a smooth and interpretable growth, maintaining continuity between successive frames and ensuring that the +10% triggering condition behaves predictably. Among these, a growth rate of 0.5 was selected as the reference configuration for the remaining analyses. This choice offers a good compromise between sensitivity and stability: it enables the system to anticipate risk escalation early enough to meet ETSI responsiveness requirements, while preventing spurious

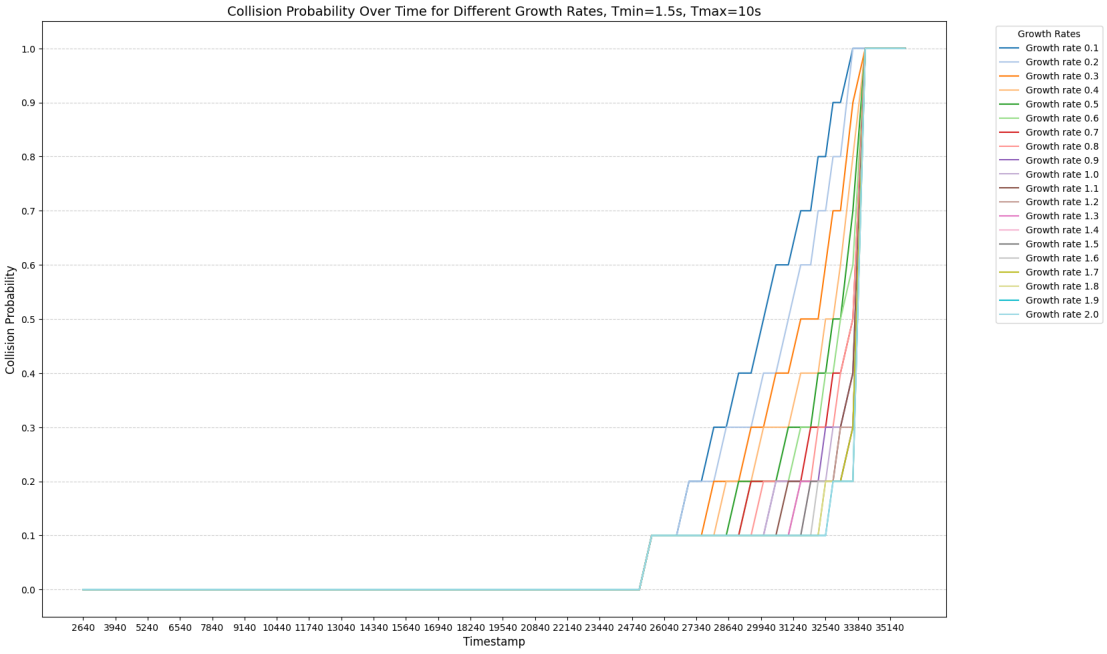

Figure 5.4: Evolution of collision probability over time for different growth rates, with $TTC_{min} = 1.5$ s and $TTC_{max} = 10$ s.

fluctuations that would increase channel load.

In summary, the combination of $TTC_{min} = 1.5$ s, $TTC_{max} = 10$ s, and a $growthrate = 0.5$ provides a robust and well-balanced setup for subsequent validation, ensuring that the collision probability evolves smoothly and proportionally to the perceived risk throughout the interaction.

5.2.3 Space-to-Collision threshold

The last parameter introduced in the analysis is the Space-to-Collision (s2c) threshold, which defines the minimum distance expected to separate two entities at the moment of their closest approach, assuming that both follow their current projected trajectories. It is expressed in meters and directly complements the TTC metric, providing a spatial perspective on the proximity of interaction. In practice, the s2c is evaluated as the instantaneous distance between the two entities at the computed TTC, a parameter fundamental in the proposed methodology in order to perform a final check after the TTC processing part is over in order to understand if the spatial features reflects the same implications as the temporal one, which ultimately leads to the TIP computation part.

The concept and implementation of S2C originate from the work developed in

[31], where it is leveraged as a filtering condition to determine whether entities are close enough to justify executing collision detection, just as in the methodolohy proposed in this thesis. In order to perform the final check previously mentioned, two threshold values were tested: 5 m and 10 m: finally, the choice was to adopt the lower possibility to perform the subsequent core part of the analysis. This choice ensures methodological consistency with the previous implementation developed in [31], from which the TTC computation leveraged here was also derived, as well as it aligns with the typical spatial margins observed in urban vehicle versus pedestrian interactions. The 5 m value effectively represents a conservative and realistic bound for possible collision proximity, filtering out distant entities unlikely to interact, while still being able to capture those within a critical safety range, ensuring that the spatial and temporal urgency are aligned.

5.3 Results

The results presented in this chapter are obtained by applying the TIP-based triggering condition to the complete set of scenarios introduced in Section 5.1. The analysis was carried out in two complementary phases.

In order to inspect the effects and the resulting outcome of the developed TIP-based triggering condition in Van3Twin, the main analysis was conducted on the complete scenario defined in Section 5.1, applying the defined mapping function with the final selection of parameters being as previously defined:

$$TTC_{min} = 1.5seconds, \quad TTC_{max} = 10seconds, \quad growthrate = 0.5, \quad s2c = 5m$$

During the simulation run, three CSV files are produced to support post–processing and reproducibility of the results. The three logs are complementary: the first captures all computations related to Time-To-Collision (TTC), the second records every VAM transmission, and the third provides a TIP–focused diagnostic trace to explain why a TIP–based trigger occurred.

1. TTC_metrics

As previously explained, every time the LDM of an entity is updated during the simulation, a distance check is performed which leads to TTC computation. Each one of this operation equal to one new row in the log, comprised of many different elements:

- `timestamp`: simulation time (ms);
- `sender_ID`: VRU or vehicle ID sending the VAM that led to the LDM update;
- `receiver_ID`: VRU ID performing the LDM update;

- `kinematic state variables`: position components, speed, acceleration and heading of both sender and receiver;
- `TTC`: time-to-collision value between the two entities at timestamp (s);
- `s2c`: space-to-collision value between the two entities at timestamp (m).

This file is the basis for all TTC distributions, threshold sweeps and sensitivity analyses.

2. VAM_metrics

Every time a VAM is generated during the simulation, a new row is added to the log with the characteristics of the message for further analysis. In our scope, the main elements stored are:

- `timestamp`: simulation time (ms);
- `station_ID`: VRU ID sendind the VAM;
- `time_elapsed_since_last_gen`: how long it has been since that VRU sent another VAM (ms);
- `triggering_condition`: enumerated cause (e.g., time/periodic, position, speed, heading, safe distance, TIP);

This file serves as the primary source for the graphical analyses that have been carried out to examine VAM transmissions in detail. It supports the evaluation of the total number of messages generated, their temporal occurrences, and their dissemination patterns. Through these analyses, it becomes possible to better understand the behaviour of the proposed methodology, assess how the new triggering condition influences VAM generation compared to the previous implementation, while deriving considerations on its performance and impact.

3. TIP_metrics

To make TIP-triggered VAMs available to be analyzed in detail in the post-processing phase, a dedicated log is produced only for the messages transmitted due to a variation in Trajectory Interjection Probability, each row containing:

- `timestamp`: simulation time (ms);
- `sender_ID`: VRU ID sending the VAM;
- `snapshot of the sender's LDM at that instant`: list of all the elements in the LDM of the sender at that time instant, each one comprised of two TIP values, one related to the current VAM that is being generated and the other to the previously transmitted one.

By listing, for each LDM entry, both the TIP value at the current emission and the TIP value at the previous emission of the same sender, this log makes explicit *which* counterpart(s) caused the TIP variation that crossed the triggering threshold, and by *how much*. This file is used to reconstruct TIP spikes, attribute triggers to specific interactions, and verify compliance with the configured variation rule.

The three logs are time–aligned via `timestamp`, enabling cross–referencing (e.g., linking a VAM row to the corresponding TTC samples and to the TIP diagnostic snapshot). Together, they allow (i) reproducible TTC calculations, (ii) complete accounting of VAM generation across all causes, and (iii) fine–grained explainability of TIP–based triggers.

These logs represent the starting point of the post–processing phase, where one of the first analyses was performed in terms of absolute numbers to assess the overall plausibility and consistency of the developed methodology. The objective was to quantify how many VAMs were generated in total and to break them down by triggering cause, in order to qualitatively evaluate whether the new condition could lead to potential channel overuse or unbalanced dissemination behaviour, especially thinking forward about scenarios with higher density in terms of pedestrians and vehicles interacting during the simulation. Furthermore, this preliminary assessment also provides a first overview of the transmission dynamics. The analysis was run on the same scenario, both without the new triggering condition implemented and with that one present

Table 5.1: Comparison of VAM transmissions before and after the implementation of the TIP-based triggering condition.

Triggering cause	**TIP not implemented**	**TIP implemented**
Start dissemination	24	24
Position	158	132
Heading	19	18
Time	0	0
Speed	1	1
Safe distance	3	26
TIP	0	112
Total	**205**	**313**

The comparison reported in Table 5.1 shows a clear variation in the number and distribution of VAM transmissions after the introduction of the TIP-based triggering condition. The total number of VAMs increases from 205 to 313, corresponding to an overall rise of approximately 52%. This confirms that the additional triggering condition makes the system more responsive, generating new messages in situations

characterised by a growing collision probability, which were previously not captured by the baseline ETSI triggers.

Looking at the individual causes, it can be observed that the traditional conditions (start dissemination, position, heading, time and speed) remain almost unchanged. This behaviour confirms that the integration of the TIP-based rule does not interfere with the normal operation of existing triggering mechanisms, which continue to dominate in standard mobility situations. A minor decrease in the number of position-based transmissions (from 158 to 132) suggests that some interactions previously identified by position changes are now detected earlier through the TIP function, effectively reducing redundant activations and slightly rebalancing the message composition.

The most relevant effects are observed in the proximity and TIP triggers. The proximity condition, which originally accounted for only three VAMs, increases to 26 in the TIP-enabled configuration. This variation does not indicate an error or instability, but rather an indirect influence of the new trigger on the evaluation order. In the implemented logic, the TIP condition is checked after time, position, heading and speed, but before the safe distance rule. Therefore, when a TIP-based event occurs, the corresponding VAM is transmitted earlier and immediately updates the Local Dynamic Map (LDM) of nearby entities. At the next simulation step, those neighbouring VRUs perform their safe distance evaluation based on refreshed information, which can make the condition true for pairs that were previously marginal.

This effect explains the change in the agents involved: in the baseline configuration, the safe distance was triggered only by pedestrian 1007 for pedestrian 1006, whereas with TIP enabled it occurs between pedestrians 1003 and 1006 in both directions. The reason is that these entities now exchange updates more frequently due to the TIP mechanism, leading to an earlier perception of proximity and a higher chance of satisfying the safe distance threshold. As a result, the total number of safe-distance-related messages increases, but these activations remain meaningful, as they correspond to real updates of perceived risk rather than spurious or redundant transmissions.

Finally, the 112 messages explicitly attributed to the TIP cause represent a substantial and coherent addition to the system behaviour. They account for more than one-third of the total messages in the TIP-enabled configuration, confirming that the new trigger actively contributes to awareness generation without overwhelming the channel. Overall, the integration of the TIP-based condition improves the responsiveness of the system, particularly in potential collision scenarios, while maintaining stability in normal operation. The increase in message generation is thus not symptomatic of congestion but rather of an enhanced capacity to capture and disseminate relevant safety information in a timely manner. To better interpret

the numerical comparison reported in Table 5.1, a visual analysis of the dissemination and temporal occurrences of the VAMs was performed for both configurations, before and after the implementation of the TIP-based triggering condition: the main goal was to assess whether the additional transmissions introduced by the new condition correspond to meaningful safety-related activations, or if they lead to redundant communication. Figure 5.5 presents the dissemination of VAMs for the baseline configuration, where the TIP condition was not active. The message composition is strongly dominated by position-based triggers, which account for more than 77% of all transmissions. The remaining causes, heading, time, speed and proximity, represent minor shares, collectively contributing less than 25%. This confirms that in the triggering logic implemented before the work done on this thesis, the generation of VAMs is primarily driven by motion updates, while the reaction to potentially critical situations is strictly limited to proximity-based events, which remains negligible in this case, regardless of the hazardous situations present in the scenario that will be discussed more in detail below.

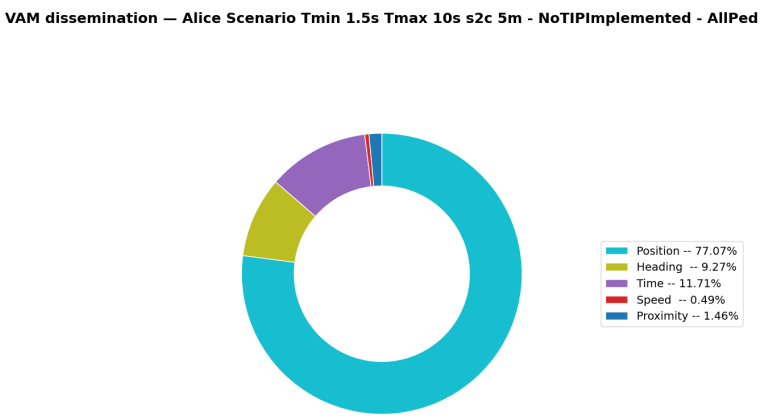

Figure 5.5: Dissemination of VAMs per triggering cause before the implementation of the TIP condition.

Enabling the TIP-based condition brings a substantial change to the VAM dissemination, as it emerged previously from the numbers and as it is shown in Figure 5.6. The proportion of position-based VAMs decreases to 42.2%, while the new TIP category stands out, representing 35.8% of all transmissions. This shift demonstrates that the additional logic's impact is not limited to the increase of the overall message count, but it also redistributes the activation causes to better

reflect risk-aware behaviour. The system continues to rely on position updates for general awareness, but it also allows for the logic to be reactive to evolving risk conditions.

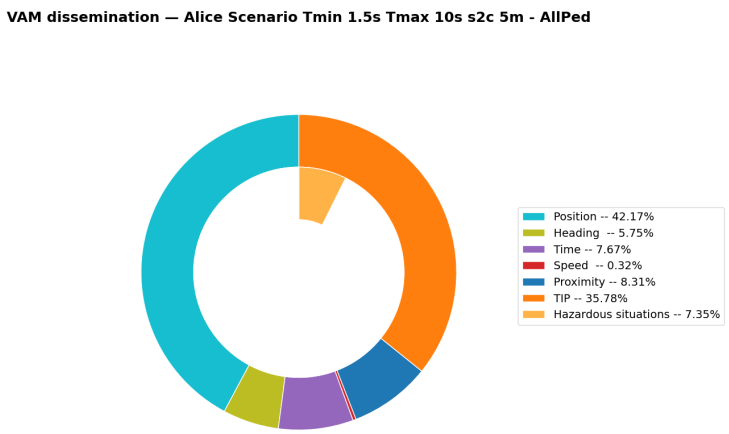

Figure 5.6: Overall dissemination of VAMs per triggering cause after TIP implementation, including hazardous situations.

The same comparison can be observed from a temporal perspective in Figures 5.7 and 5.8, which illustrate the occurrences of VAMs as a function of their inter-arrival time, showing when and how frequent messages are sent depending on the cause. The configuration before our contribution (Figure 5.7) shows that the majority of transmissions are concentrated around periodicities of 2500–3500 ms, mainly due to position-based triggers: this corresponds to a steady-state behaviour typical of pedestrians moving at moderate speeds without abrupt changes in kinematics. Only a few events appear in the low-periodicity region (below 1000 ms), mostly caused by minor heading or proximity updates, leading to the conclusion that the system had a limited responsiveness to rapidly changing situations. Instead, With the TIP condition active (Figure 5.8), the occurrence pattern changes significantly once again: a dense cluster of activations appears in the low periodicities region (below 500 ms), corresponding to VAMs generated when the estimated collision probability increases steadily in a short time frame, leading to a potential dangerous situation. This confirms that the TIP-based condition enhances system responsiveness exactly when mostly needed: whenever risk escalation is detected. The second group of messages around 2500–3500 ms remains visible and dominated by position-based triggers, indicating that normal dissemination is preserved and not overshadowed by TIP-induced messages. Finally, the periodic events at 5000 ms keep representing

mainly passive users not involved in relevant interactions, as the majority of these occurrences happened whenever a VRU started its dissemination after 5000 ms from the beginning of the simulation.

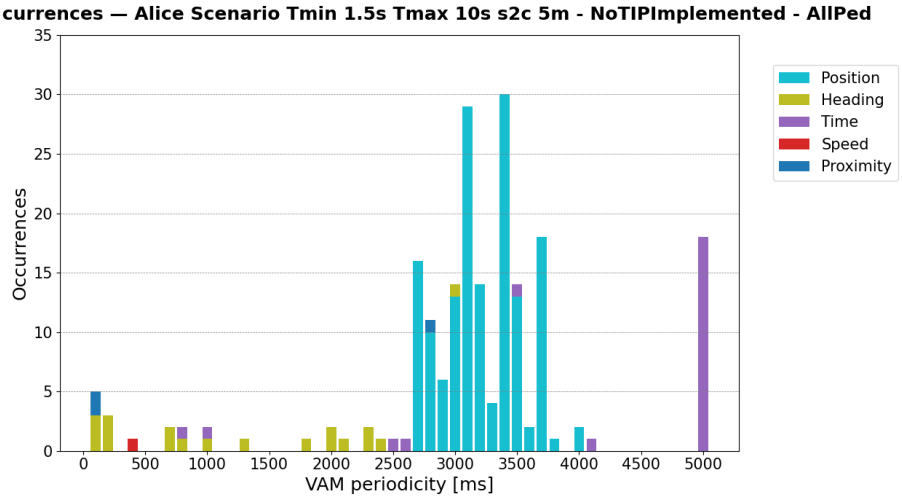

Figure 5.7: Temporal occurrences of VAM transmissions before TIP implementation.

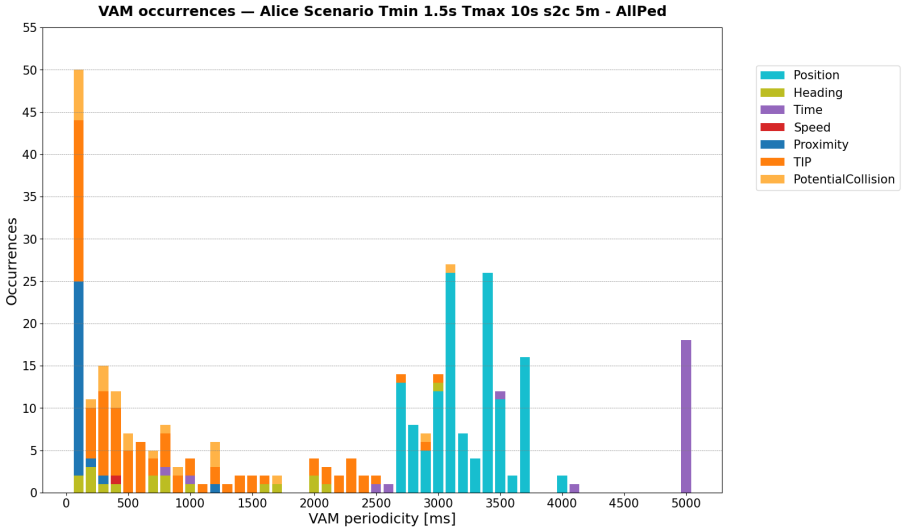

Figure 5.8: Temporal occurrences of VAM transmissions after TIP implementation.

Overall, the introduction of the TIP condition produces a dual effect: it increases

the total number of VAMs, but it does not lead to redundant transmissions. Instead, it redistributes them in time windows and contexts that are most safety-relevant, achieving a key objective of the ETSI frammework: enhancing it by improving VRU awareness proactively through selectively intensifying communication in potentially dangerous contexts.

In order to better assess whether the TIP-triggered VAMs are effective indicators for proactively identifying potential collisions and thus improving overall system safety, the analysis was further extended to a per-pedestrian level. This step enables a more granular understanding of the system's behaviour, clarifying which VRUs generate the highest number of VAMs and why, which pedestrians are most influenced by the introduction of the new triggering condition, and how these changes differ from the baseline configuration without TIP. Both dissemination and temporal occurrences were therefore analysed for each pedestrian individually. From this analysis, it emerged that the pedestrians generating the largest number of VAMs are also those for which the new TIP-based condition becomes active. Conversely, pedestrians that never trigger the TIP condition maintain a nearly identical behaviour to the baseline: their transmissions are dominated by position or heading updates, and the total count of messages per entity remains low and stable. The only exceptions with relevant activations due to proximity are pedestrians 1003 and 1006, already discussed in Section **??**, whose mutual interaction produces additional VAMs in both configurations but without leading to critical risk evolutions. For those pedestrians that do generate VAMs through the TIP condition, each message was examined in detail to identify whether it corresponded to an actual hazardous situation. This verification was carried out using the TIP_metrics log, by inspecting the temporal evolution of the estimated collision probability between each pedestrian and the surrounding vehicles. A situation was classified as hazardous whenever a pedestrian exhibited a consistent increase of collision probability over a short time window, typically a few seconds, with respect to at least one vehicle. This procedure led to the identification of four distinct hazardous events:

- **Pedestrian 1003 vs Vehicle 0**: collision probability starts increasing at timestamp 31 977 ms and reaches 90% after approximately 2 s;
- **Pedestrian 1007 vs Vehicle 5**: collision probability starts increasing at timestamp 18 486 ms and reaches 90% after about 8 s;
- **Pedestrian 1014 vs Vehicle 7**: collision probability starts increasing at timestamp 33 271 ms and reaches 90% after roughly 7 s;
- **Pedestrian 1014 vs Vehicle 3**: collision probability starts increasing at timestamp 30 271 ms and reaches 90% after approximately 5 s.

Interestingly, the three pedestrians involved in these cases are also the ones generating the highest total number of VAMs across the entire scenario. The first case (pedestrian 1003 vs vehicle 0) is a special configuration intentionally designed to force a collision by disabling SUMO's internal safety logic for the vehicle. As a result, this interaction shows the fastest escalation of risk, with the TIP reaching its 90% threshold in around two seconds, a value comparable to the human reaction time reported in literature (about 1.5 s). In contrast, the remaining three cases display a more progressive increase in collision probability, typically over 6–7 s, which demonstrates that the system is capable of detecting risk early enough to allow potential avoidance or mitigation actions. This behaviour confirms the proactive nature of the proposed approach and its ability to enhance VRU awareness dynamically. Furthermore, each of these hazardous intervals corresponds to several consecutive VAM transmissions. On average, at least seven VAMs are generated during each critical phase, showing that the system maintains a sustained level of communication throughout the period of increasing risk: this excludes the possibility of such activations being random or caused by instantaneous kinematic fluctuations; instead, they represent a consistent and meaningful response to evolving collision probability. To provide a visual enhancement of this behavior, the dissemination and occurrences analysis for each of these three VRU is reported below (Figures 5.9–5.11): the hazardous situations are represented by the light-orange share in the inner ring of the dissemination charts and by the corresponding light-orange bars in the occurrence histograms.

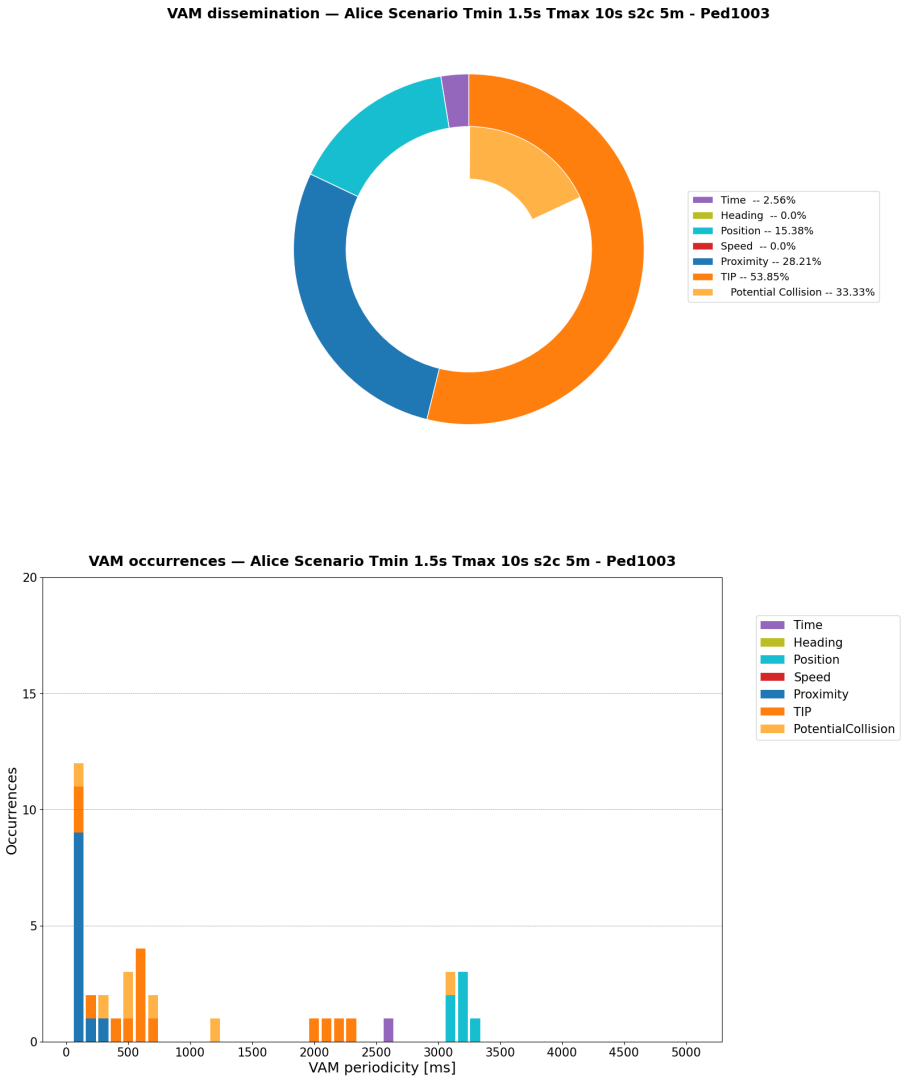

Figure 5.9: VAM dissemination and temporal occurrences for pedestrian 1003, involved in an intentionally forced collision with vehicle 0.

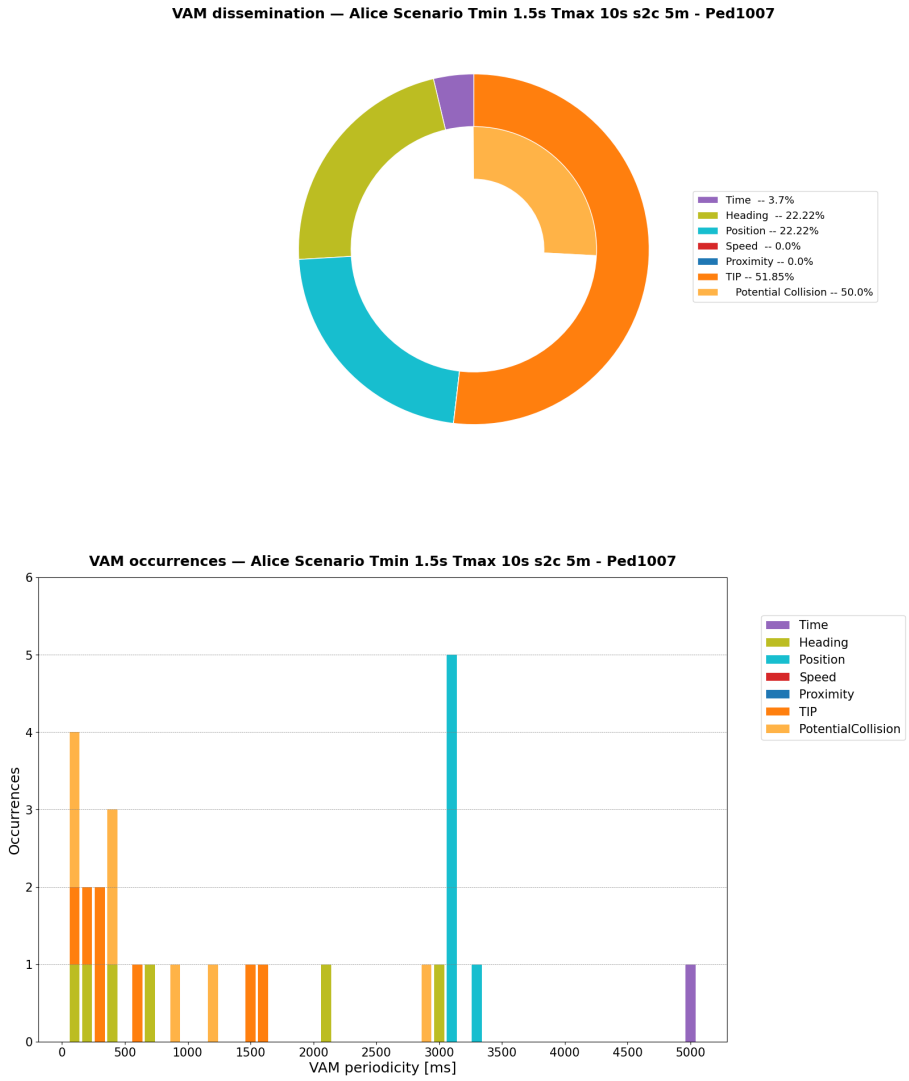

Figure 5.10: VAM dissemination and temporal occurrences for pedestrian 1007, interacting with vehicle 5.

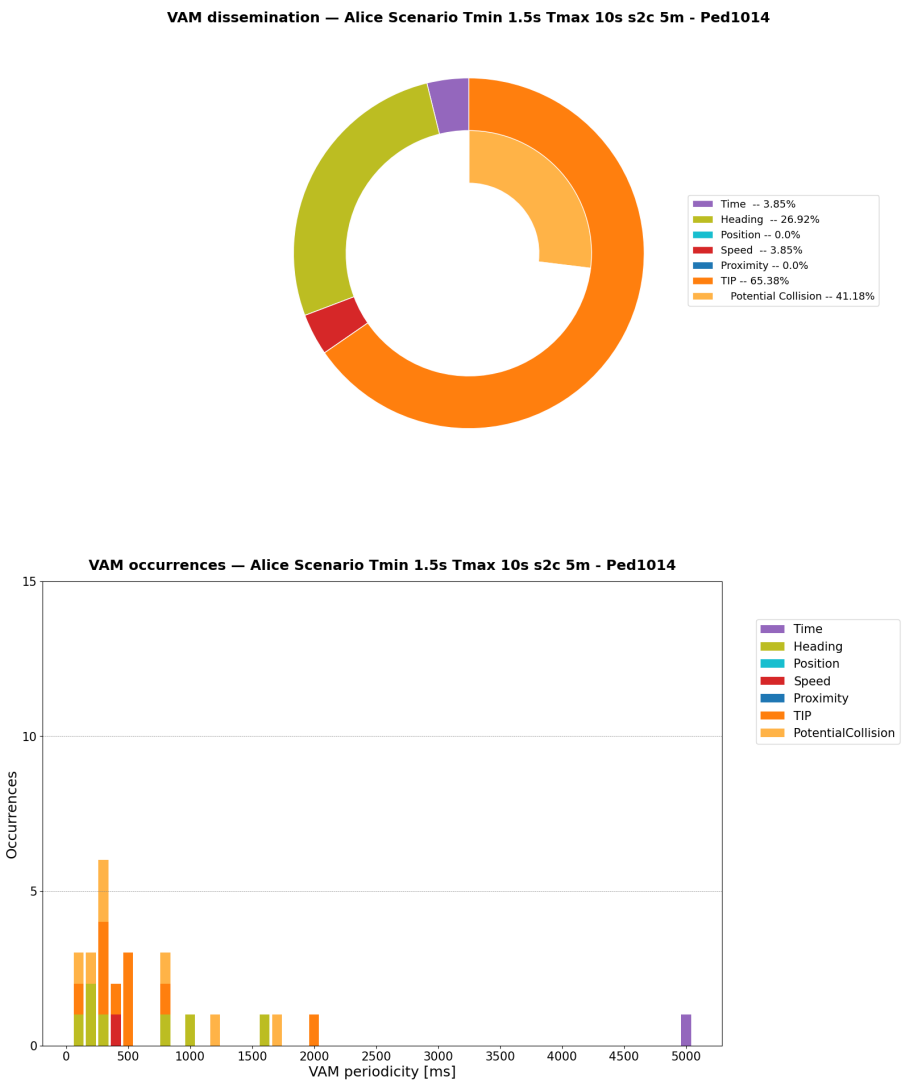

Figure 5.11: VAM dissemination and temporal occurrences for pedestrian 1014, interacting with vehicles 3 and 7.

Finally, to further validate the contribution of the proposed methodology to implement the triggering condition, a quantitative comparison was carried out within the same time windows corresponding to the four hazardous situations identified above. For each of these intervals, the total number of VAMs generated by the corresponding pedestrian was counted both before and after the introduction of the TIP condition: in the TIP-enabled configuration, the messages triggered specifically by the new condition were also highlighted separately. This allows to

directly assess how the new logic modifies the system's behaviour in critical phases, without the influence of external scenario changes.

Table 5.2: Comparison of VAM transmissions within hazardous time windows, before and after the implementation of the TIP condition.

Interaction	Without TIP	With TIP (total)	of which TIP-triggered
Ped1003 – Veh0	0	4	**4**
Ped1007 – Veh5	2	8	**6**
Ped1014 – Veh7	1	8	**3**
Ped1014 – Veh3	7	17	**6**
Total	**10**	**37**	**19**

The results reported in Table 5.2 highlight several key findings. First, the total number of VAMs transmitted within the hazardous intervals increases from 10 to 37 when the TIP condition is active, showing a clear enhancement of communication during risk escalation. Furthermore, more than half of these additional messages (19 out of 37) are directly triggered by the new TIP logic, confirming that the increase is not random but driven by the estimated collision probability.

The case that needs to be highlighted the most is the Ped1003–Veh0 interaction: in the configuration without the new triggering logic, no VAM is generated during the critical phase, meaning that the system entirely fails to signal the developing collision. This occurs because this interaction was intentionally designed as an aggressive collision scenario: SUMO's built-in safety mechanisms were disabled to force the crash and test the system's responsiveness. The fact that, with the TIP condition enabled, four consecutive VAMs are transmitted within this short two-second interval demonstrates the value of the new trigger: it detects and communicates in an effective way the imminent collision, which would otherwise remain unnoticed. This is a clear indicator of the system's ability to provide proactive warnings even in extreme situations.

For the other cases (Ped1007–Veh5, Ped1014–Veh7, and Ped1014–Veh3), the number of VAMs roughly triples after the introduction of TIP, with a significant share (from 30% to 75%) explicitly due to the new condition. This pattern indicates that the system increases message frequency proportionally to the evolving risk level, rather than uniformly across time or users. In particular, the Ped1014–Veh3 pair exhibits the highest overall communication density, with 17 VAMs transmitted in a window of approximately five seconds, six of which are TIP-triggered. This suggests that the new logic enables sustained information flow during the most dynamic stages of risk evolution, offering sufficient lead time for cooperative safety responses such as braking or path adjustment.

Overall, these findings confirm that the TIP-based condition improves both the selectivity and the timeliness of VAM transmissions. The system becomes more sensitive to genuinely hazardous developments, as it produces multiple, consecutive messages during escalating risk, while remaining stable in low-risk situations. This dual effect reinforces the proactive design goal of the proposed approach: enhancing VRU awareness and safety without introducing unnecessary channel congestion.

To get a quantitative overview of the whole scenario, the VAMs associated with these four hazardous interactions amount to 19 messages, which represent 20.5% of all TIP-triggered transmissions and 7.4% of the total VAMs generated during the simulation. This fraction, also visible in Figure 5.6, highlights that a non-negligible portion of the new TIP-based activations corresponds directly to situations where the system identifies a genuine and increasing collision probability: this shows that the methodology developed to implement this TIP condition increases the awareness level of the system, as well as it contributes to the selective and timely signalling of safety-critical interactions. In particular, the sustained message flow observed during these hazardous episodes confirms that the mechanism operates proactively, continuously updating the surrounding vehicles about the increasing risk, and provides a time window sufficient for cooperative safety actions to be taken, such as speed adaptation or trajectory adjustment.

Overall, these results demonstrate that the proposed TIP-based triggering method successfully fulfils its intended function: it enhances the sensitivity of VAM generation to real collision risks while maintaining stable communication elsewhere, achieving a meaningful balance between responsiveness and efficiency in accordance with the ETSI framework.

Once the effectiveness of the proposed methodology was demonstrated through the comparative analyses described above, a further question naturally arose regarding the *relevance* of the additional VAMs produced by the new triggering logic. While the obtained results confirm that the TIP-based condition significantly enhances awareness during genuinely hazardous phases, it remains essential to verify whether these additional transmissions always correspond to meaningful and safety-related events. Indeed, even though the overall number of TIP-triggered VAMs observed in this scenario is not large enough to cause channel congestion, it is important to ensure that each transmission contributes effectively to the system's purpose of protecting VRUs.

To address this aspect, all the VAMs activated by the TIP condition were re-examined through the TIP_metrics log in order to determine whether they corresponded to an increase or a decrease in the estimated collision probability. According to the ETSI specification, the TIP variation triggering rule is defined symmetrically — that is, a VAM is generated whenever the collision probability changes by more than a fixed threshold, regardless of whether this change is positive or negative. However, this formulation implies that the system transmits a new

message also when the situation becomes safer, i.e. when the collision probability decreases.

As shown in Figure 5.12, a non-negligible portion of the TIP-triggered messages (approximately 30%) corresponds to decreases in collision probability. This means that while the system correctly detects changes in the estimated risk, not all such changes are equally relevant for awareness enhancement. In the context of VRU protection, a decrease in collision probability does not provide actionable information for safety: it merely reflects a return to a safer state, which does not require urgent dissemination to other entities. Therefore, these transmissions, although compliant with the ETSI rule, can be considered redundant in terms of their contribution to situational awareness.

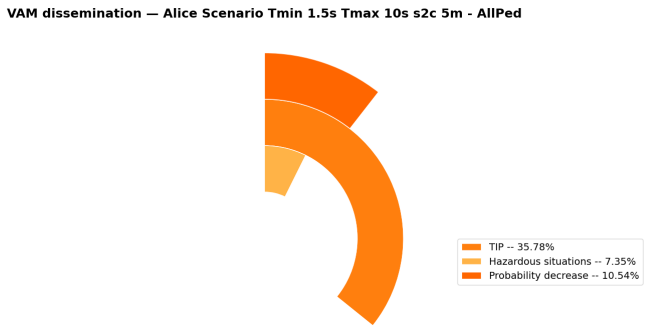

Figure 5.12: Division of all TIP-triggered VAMs by variation type: increases versus decreases in collision probability.

Based on these considerations, a refinement to the ETSI definition is proposed: TIP-based triggering should occur exclusively in the case of a positive variation of collision probability, which corresponds to when the estimated risk is increasing. This ensures that every additional VAM generated through this mechanism corresponds to a meaningful and safety-relevant event, while avoiding redundant communications in the opposite case. Such an adjustment would further reduce the possibility of channel overload in dense traffic scenarios, as well as eliminate unnecessary transmissions not contributing to the enhancement of VRU awareness.

To verify the potential effect of this modification, the global dissemination results were recalculated by excluding all TIP-triggered VAMs corresponding to decreasing collision probability. The adjusted distribution is presented in Figure 5.13: the share of TIP-triggered messages decreases from 35.8% to 28.2%, while the fraction

associated with genuinely hazardous situations increases slightly for mathematical motives, but it remains around 8%, confirming that the most critical events are preserved.

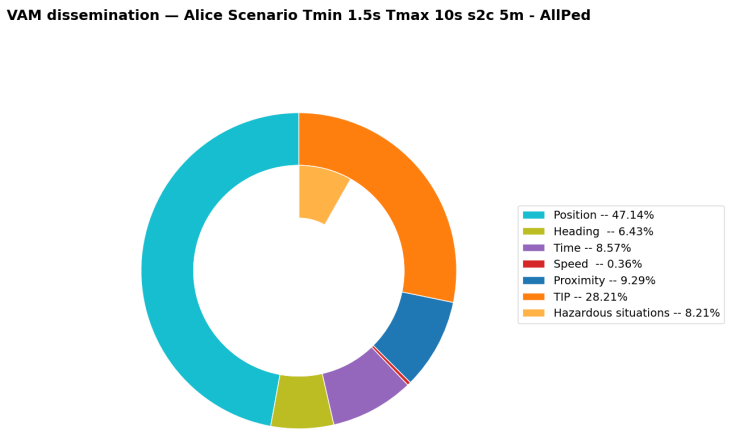

Figure 5.13: Adjusted dissemination of VAMs after removing TIP-triggered messages corresponding to decreasing collision probability.

This refined outcome further reinforces the conclusions drawn from the previous analyses. The TIP-based condition, when limited to positive variations, maintains high responsiveness to genuine risk evolution while reducing unnecessary load on the communication channel. It improves the relevance of generated messages, ensuring that every TIP-triggered VAM either reflects an increasing collision probability or anticipates a potentially hazardous situation. Such behaviour aligns perfectly with the design goal of cooperative safety systems: to transmit information that is both timely and meaningful for decision-making, avoiding redundant or irrelevant updates.

Overall, the comprehensive analyses presented in this chapter confirm the effectiveness, selectivity, and operational efficiency of the proposed TIP-based triggering methodology. The introduction of the new condition substantially improves the system's capability to identify hazardous situations and maintain communication continuity during risk escalation, while keeping message rates stable under nominal conditions. Pedestrian-specific analyses showed that the method dynamically adapts to the evolution of collision probability, producing consistent transmissions in high-risk phases and avoiding unnecessary bursts elsewhere. Furthermore, the relevance refinement proposed above provides a practical improvement to the ETSI

framework, ensuring that only meaningful variations of collision probability result in new VAMs, thereby enhancing both safety effectiveness and communication efficiency.

In conclusion, the TIP-based approach demonstrates strong potential as a proactive mechanism for VRU safety enhancement. Indeed, it increases awareness in the presence of developing hazards, as well as it introduces an interpretable and computationally efficient logic compatible with real-time ETSI-compliant VAM generation. By selectively amplifying communication in critical moments and suppressing non-essential updates, the proposed refinement paves the way toward a more responsive, reliable, and context-aware cooperative safety framework.

Chapter 6

Conclusion

This thesis has presented the design, implementation, and evaluation of a real-time framework for estimating collision probability in Vulnerable Road User scenarios, developed within the ETSI-compliant triggering logic for VRU Awareness Messages. The proposed methodology leverages the Time-to-Collision as a base metric and maps it into a probabilistic indicator, the Trajectory Interception Probability, enabling an interpretable and computationally efficient representation of instantaneous collision risk. The work addressed the need for a risk estimation model that remains consistent with the ETSI TS 103 300-3 requirements, in particular with the rule stating one of the seven triggering conditions for these kind of messages: the transmission shoul occur whenever the collision probability with respect to an entity in the LDM of the pedestrian changes by more than 10% compared to the previously sent value related to that identified element. By grounding the approach on TTC, the framework ensures two key features: interpretability and temporal responsiveness, without requiring predictive or data-driven models that would unnecessarily increase the computational complexity of the implementation, without even assuring standard compliance. The chosen methodology requires the definition of a set parameters in order to be able to map TTC values into TIP ones. Through the simulations performed, several combinations were explored and refined to identify an optimal configuration able to balance reactivity, stability, and communication efficiency. The resulting setup, which is defined by $TTC_{min} = 1.5s$, $TTC_{max} = 10s$, $growthrate = 0.5$ and $s2c = 5m$ was shown to produce a smooth but responsive evolution of the TIP, capturing risk escalation in an accurate way, while preventing excessive triggering at the same time. This developed methodology ensures that the system reacts promptly to real danger without saturating the communication channel. Indeed, the results confirmed the effectiveness of the TIP-based approach as a valid, lightweight, and fully ETSI-compliant solution for collision risk estimation: the method demonstrated the ability to mirror the real evolution of pedestrian versus vehicle encounters, snapshotting the transition from

low-risk to possible collision conditions through a gradual increase in probability. Moreover, as it relies on a continuous and frame-by-frame evaluation, the algorithm maintains real-time responsiveness while offering high interpretability. The combination of these properties makes the approach particularly suited for cooperative safety applications, where decisions must be both timely and explainable. From a system-level perspective, the analysis revealed that the chosen parameters results in a good trade-off between anticipation and communication stability. The approach also proved scalable and computationally efficient, making it a promising candidate for integration within cooperative ITS frameworks and future field deployments. While the achieved results demonstrate the feasibility and robustness of the proposed method, several research directions remain open for future development. The first and most natural step concerns experimental validation in real-world conditions. Field tests involving both pedestrians and vehicles are necessary to assess the method's performance under realistic noise, latency, and sensor uncertainty. Such experiments would make it possible to evaluate the triggering behavior in dynamic environments and verify that the +/-10% variation rule maintains its effectiveness when facing unpredictable VRU trajectories. A second avenue of research involves the extension of the current study to denser and more complex traffic scenarios. While the present simulations focused on moderate-density settings, urban intersections and shared spaces often involve multiple simultaneous interactions. Analyzing the TIP-based triggering behavior in these contexts would help quantify potential channel congestion effects and assess the scalability of the proposed framework. This would also allow the investigation of strategies for adaptive message scheduling or prioritization, ensuring that the communication channel remains efficient even under heavy network load. Finally, the methodology developed here could serve as a foundation for proposing refinements to the ETSI standard itself. Future versions of the specification could consider integrating adaptive or context-aware thresholds, probabilistic filtering mechanisms, or combined temporal–spatial indicators that enhance both accuracy and efficiency. The present work provides a first empirical step toward such evolution, offering quantitative evidence on how real-time probabilistic triggering can improve situational awareness without overloading the network. In conclusion, this thesis has proposed and validated a method that bridges theoretical collision risk modeling with operational constraints of cooperative ITS communication. By combining simplicity, interpretability, and standard compliance, the approach advances the understanding of how TTC-based metrics can effectively support real-time safety mechanisms for VRUs. The proposed mapping framework provide a reproducible and adaptable basis for future research, opening various range of possibilities. In this context, the work represents a concrete step toward more intelligent, proactive, and reliable cooperative safety systems in the urban mobility landscape.

Bibliography

[1] European Telecommunications Standards Institute (ETSI). *ETSI EN 302 665 V1.1.1 (2010-09): Intelligent Transport Systems (ITS); Communications Architecture*. Tech. rep. EN 302 665 V1.1.1. European Telecommunications Standards Institute (ETSI), Sept. 2010. URL: https://www.etsi.org/deliver/etsi_en/302600_302699/302665/01.01.01_60/en_302665v010101p.pdf (cit. on pp. 2, 12, 20).

[2] American Association of State Highway and Transportation Officials (AASHTO)¦ *USDOT Releases Third Guidance Update for Automated Vehicles*. AASHTO Journal, news release. "Preparing for the Future of Transportation: Automated Vehicles 3.0". Oct. 2018. URL: https://aashtojournal.transportation.org/usdot-releases-third-guidance-update-for-automated-vehicles/ (cit. on p. 5).

[3] Zubair H. Mir, Jamal Toutouh, Fethi Filali, and Young-Bae Ko. «Enabling DSRC and C-V2X Integrated Hybrid Vehicular Networks: Architecture and Protocol». In: *IEEE Access* 8 (2020), pp. 180909–180927. DOI: 10.1109/ACCESS.2020.3021274. URL: https://ieeexplore.ieee.org/document/9212554 (cit. on p. 9).

[4] World Health Organization. *Global status report on road safety 2018*. WHO, 2018. URL: https://www.who.int/publications/i/item/9789241565684 (cit. on p. 10).

[5] European Commission. *2024 Road Safety Statistics: Road deaths decrease by 3% in the EU*. https://ec.europa.eu/commission/presscorner/detail/en/ip_25_789. Press release, Brussels, 21 March 2025. 2025 (cit. on p. 10).

[6] European Telecommunications Standards Institute (ETSI). *ETSI TR 103 300-1 V2.3.1 (2022-11): Intelligent Transport Systems (ITS); Vulnerable Road Users (VRU) awareness; Part 1: Use Cases definition; Release 2*. Tech. rep. TR 103 300-1 V2.3.1. European Telecommunications Standards Institute (ETSI), Nov. 2022. URL: https://www.etsi.org/deliver/etsi_tr/103300_103399/10330001/02.03.01_60/tr_10330001v020301p.pdf (cit. on pp. 10, 15).

[7] European Commission. *EU Road Safety Policy Framework 2021–2030: Next steps towards "Vision Zero"*. https://transport.ec.europa.eu/system/files/2021-09/eu_road_safety_policy_framework_2021-2030.pdf. Accessed: 2025-08-18. 2019 (cit. on p. 12).

[8] European Telecommunications Standards Institute (ETSI). *ETSI TS 103 300-3 V2.2.1 (2022-09): Intelligent Transport Systems (ITS); Vulnerable Road Users Awareness; Part 3: Specification of VRU awareness basic service*. Tech. rep. TS 103 300-3 V2.2.1. European Telecommunications Standards Institute (ETSI), Sept. 2022. URL: https://www.etsi.org/deliver/etsi_ts/103300_103399/10330003/02.02.01_60/ts_10330003v020201p.pdf (cit. on pp. 15, 17, 18, 21, 23, 56, 63).

[9] European Telecommunications Standards Institute (ETSI). *ETSI TS 103 300-2 V2.1.1 (2021-09): Intelligent Transport Systems (ITS); Vulnerable Road Users Awareness; Part 2: Service requirements and analysis*. Tech. rep. TS 103 300-2 V2.1.1. European Telecommunications Standards Institute (ETSI), Sept. 2021. URL: https://www.etsi.org/deliver/etsi_ts/103300_103399/10330002/02.01.01_60/ts_10330002v020101p.pdf (cit. on p. 16).

[10] Abderrahim Belmekki and Dominique Gruyer. «Advanced Road Safety: Collective Perception for Probability of Collision Estimation of Connected Vehicles». In: *Computers* 13.1 (2024), p. 21. DOI: 10.3390/computers13010021. URL: https://doi.org/10.3390/computers13010021 (cit. on pp. 23, 59, 62).

[11] Yacine Atif, Mohamad H. Ali, Abderrahmane Boukerche, Lamia Chaari Fourati, and Hossam Afifi. «Data-Driven Prediction of Vehicle–Vulnerable Road User Collisions at Road Intersections Using Machine Learning Models». In: *Procedia Computer Science* (2025). DOI: 10.1016/j.procs.2025.03.100. URL: https://doi.org/10.1016/j.procs.2025.03.100 (cit. on p. 23).

[12] Minh Bui, Michael Lu, Reza Hojabr, Mo Chen, and Arrvindh Shriraman. «Real-Time Hamilton–Jacobi Reachability Analysis of an Autonomous System with an FPGA». In: *Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2021, pp. 1666–1673. DOI: 10.1109/IROS51168.2021.9636410. URL: https://www2.cs.sfu.ca/~ashriram/papers/2021_IROS_HJ.pdf (cit. on pp. 26, 27).

[13] Matthias Althoff. «Reachability Analysis and its Application to the Safety Assessment of Autonomous Cars». PhD thesis. Technische Universität München, 2010. URL: https://mediatum.ub.tum.de/doc/1287517 (cit. on pp. 26, 27).

[14] Hang Zhang et al. «Hybrid Zonotope-Based Backward Reachability Analysis for Neural Feedback Systems With Nonlinear Plant Models». In: *arXiv preprint* arXiv:2310.06921 (2023). URL: https://arxiv.org/abs/2310.06921 (cit. on p. 26).

[15] Zhiyu Huang, Haochen Liu, Jingda Wu, Wenhui Huang, and Chen Lv. «Learning Interaction-aware Motion Prediction Model for Decision-making in Autonomous Driving». In: *arXiv preprint arXiv:2302.03939* (2023). URL: https://arxiv.org/abs/2302.03939 (cit. on p. 28).

[16] Peng Chang and Christoph Mertz. «Monte Carlo Sampling Based Imminent Collision Detection Algorithm». In: *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017. DOI: 10.1109/ITSC.2017.8047791. URL: https://ieeexplore.ieee.org/document/8047791 (cit. on p. 30).

[17] Florent Altché and Arnaud de La Fortelle. «An LSTM network for highway trajectory prediction». In: *arXiv preprint arXiv:1801.07962* (2018). URL: https://arxiv.org/abs/1801.07962 (cit. on p. 31).

[18] Jingda Zhang, Wenjie Chen, Ruibo Yang, and Masayoshi Tomizuka. «GTP: General trajectory prediction for autonomous driving». In: *arXiv preprint arXiv:2302.00735* (2023). URL: https://arxiv.org/abs/2302.00735 (cit. on p. 31).

[19] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. «Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data». In: *arXiv preprint arXiv:2001.03093* (2020). URL: https://arxiv.org/abs/2001.03093 (cit. on p. 31).

[20] Wei Zhan, Hao Shi, Ching-Yao Yu, Maxim Egorov, Mehrdad Shad, Yaqi Zheng, and Masayoshi Tomizuka. «INTERACTION dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps». In: *IEEE Intelligent Vehicles Symposium (IV)*. 2019, pp. 1910–1916. DOI: 10.1109/IVS.2019.8814101. URL: https://arxiv.org/abs/1903.11027 (cit. on p. 31).

[21] Holger Caesar et al. «nuScenes: A multimodal dataset for autonomous driving». In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 11621–11631. DOI: 10.1109/CVPR42600.2020.01165. URL: https://arxiv.org/abs/1903.11027 (cit. on p. 32).

[22] B. Ravi Kiran, Ihab Sobh, Vivien Talpaert, Patrick Mannion, Ahmad Sallab, Senthil Yogamani, and Patrick Pérez. «Deep reinforcement learning for autonomous driving: A survey». In: *IEEE Transactions on Intelligent Transportation Systems* 23.6 (2021), pp. 4909–4926. DOI: 10.1109/TITS.2021.3054625. URL: https://arxiv.org/abs/2002.00444 (cit. on p. 32).

[23] James J. Gibson and Leonard E. Crooks. «A Theoretical Field-Analysis of Automobile-Driving». In: *The American Journal of Psychology* 51.3 (1938), pp. 453–471. DOI: 10.2307/1416145. URL: https://www.jstor.org/stable/1416145 (cit. on p. 34).

[24] David N. Lee. «A Theory of Visual Control of Braking Based on Information about Time-to-Collision». In: *Perception* 5.4 (1976), pp. 437–459. DOI: 10.1068/p050437. URL: https://www.researchgate.net/publication/22149996_A_Theory_of_Visual_Control_of_Braking_Based_on_Information_about_Time-to-Collision (cit. on p. 34).

[25] Roberto Pegurri, Diego Gasco, Francesco Linsalata, Marco Rapelli, Eugenio Moro, Francesco Raviglione, and Claudio Casetti. «VaN3Twin: the Multi-Technology V2X Digital Twin with Ray-Tracing in the Loop». In: *arXiv preprint arXiv:2505.14184* (2025). URL: https://arxiv.org/abs/2505.14184 (cit. on p. 36).

[26] DriveX-devs. *VaN3Twin: Multi-Technology V2X Digital Twin with Ray-Tracing*. Accessed: September 12, 2025. 2025. URL: https://github.com/DriveX-devs/VaN3Twin (cit. on p. 36).

[27] Francesco Raviglione, Chiara R. Carletti, Marco Malinverno, Claudio Casetti, and Carla Fabiana Chiasserini. «ms-van3t: An integrated multi-stack framework for virtual validation of V2X communication and services». In: *Computer Communications* 217 (2024), pp. 70–86. ISSN: 0140-3664. DOI: 10.1016/j.comcom.2023.12.004. URL: https://www.sciencedirect.com/science/article/pii/S0140366424000227 (cit. on pp. 36, 37).

[28] Charles F. F. Karney. *GeographicLib 2.1 Documentation*. https://geographiclib.sourceforge.io/. Accessed: September 12, 2025. 2022 (cit. on p. 38).

[29] Michael Behrisch, Laura Bieker, Jakob Erdmann, and Daniel Krajzewicz. «SUMO – Simulation of Urban MObility: An Overview». In: *Proceedings of the Third International Conference on Advances in System Simulation (SIMUL 2011)*. Barcelona, Spain: ThinkMind, 2011, pp. 55–60. URL: https://www.researchgate.net/publication/225022282_SUMO_-_Simulation_of_Urban_MObility_An_Overview (cit. on p. 39).

[30] Alessandro Genovese. «Extension of ms-van3t for VRU Awareness: Implementation of VAMs and VRU Basic Service». Master's Thesis. Politecnico di Torino, 2023. URL: http://webthesis.biblio.polito.it/id/eprint/29334 (cit. on pp. 42, 63, 71, 73).

[31] Marco Rapelli. «Simulation of a Communication System for Vehicle Collision Avoidance». Master's thesis, not publicly available due to confidentiality with Fiat. 2017 (cit. on pp. 46, 83).

[32] I. D. Brown. «Driver headway strategies in car following». In: *Proceedings of the International Cooperation on Theories and Concepts in Traffic Safety (ICTCT)*. 1994, pp. 85–102. URL: https://ictct.net/wp-content/uploads/SMoS_Library/LIB_Brown_1994.pdf (cit. on p. 54).

[33] N. Nadimi, S. S. Naseralavi, and M. Asadamraji. «Calculating dynamic thresholds for critical time-to-collision as a safety measure». In: *Journal of Transport Literature* 15.7 (2021), pp. 403–418. URL: https://www.emerald.com/jtran/article-abstract/175/7/403/428810/Calculating-dynamic-thresholds-for-critical-time?redirectedFrom=fulltext (cit. on p. 54).

[34] Yuhong Chen, Yuanliang Xu, Xuewen Rong, and Lianchun Xu. «Collision probability prediction algorithm for cooperative overtaking based on TTC and conflict probability estimation method». In: *2019 IEEE 5th International Conference on Computer and Communications (ICCC)*. IEEE, 2019, pp. 1465–1470. DOI: 10.1109/ICCC47050.2019.9064352 (cit. on pp. 57, 61).

[35] OpenStreetMap contributors. *OpenStreetMap*. https://www.openstreetmap.org/. Accessed: 2025-09-16. 2025 (cit. on p. 74).