Information Fusion Methodology for Enhancing Situation Awareness in Connected Cars Environment

by

Keyvan Golestan Irani

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2015

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

This dissertation introduces novel approaches to develop a comprehensive model to address situation awareness in the Internet of Cars, called Attention Assist Framework (AAF). The proposed framework utilizes both Low-Level Data Fusion (LLDF), and High-Level Information Fusion (HLIF) to implement traffic entity, situation, and impact assessment, as well as decision making.

The Internet of Cars is the convergence of the Internet of Things and Vehicular Ad-hoc Networks (VANETs). In fact, VANETs are the communication platforms that make possible the implementation of the Internet of Cars, and has become an integral part of this research field due to its major role to improve vehicle and road safety, traffic efficiency, and convenience as well as comfort to both drivers and passengers. Significant amount of VANETs research work has been focused on specific areas such as safety, routing, broadcasting, Quality of Service (QoS), and security. Among them, road safety issues are deemed one of the most challenging problems of VANETs. Specifically, lack of proper situational awareness of drivers has been shown to be the main cause of road accidents which makes it a major factor in road safety.

The traffic entity assessment relies on a LLDF framework that is able to incorporate various multi-sensor data fusion approaches with means of communication links in VANETs. This is used to implement a cooperative localization approach through fusing common data fusion methods, such as Extended Kalman Filter (EKF) and Unscented Transform (UT), and vehicle-to-vehicle communication in VANETs. Furthermore, traffic situation assessment is based on a fuzzy extension to the Multi-Entity Bayesian Networks (MEBNs), which exploit the expressiveness of first-order logic for semantic relations, and the strength of the Fuzzy Bayesian Networks in handling uncertainty, while tackling the inherent vagueness in the soft data created by human entities. Finally, the impact assessment and decision making is realized through incorporating notions of game theory into Fuzzy-MEBNs, and introducing Active Fuzzy-MEBN (ATFY-MEBN), which is capable in hypothesizing future situations by assessing the impact of the current situation upon taking the actions indicated by an optimal strategy. In fact, such strategies are achieved through solving the games that are generated through a novel situation-specific normal form games generation algorithm that aims to create games based on the given context. In general, ATFY-MEBN presents the concepts of players and actions, and includes new game components, along with a 2-tier architecture, to efficiently model impact assessment and decision making.

To demonstrate the capabilities of the proposed framework, a collision warning system simulator is developed, which evaluates the likelihood of a vehicle being in a near-collision

situation using a wide variety of both local and global information sources available in the VANETs environment, and suggests an optimal action by assessing the impact of the current situation through generating and solving situation-specific games.

Accordingly, first, the entities that highly influence the safety aspect, as well as both their casual and semantic relationships are identified. Next, an ATFY-MEBN-based model is presented, which allows for modeling these entities along with their relationships in specific contexts, assessing the current states of the situations of interest, predicting their future states, and finally suggesting optimal decision.

Therefore, if the likelihood of being in a near-collision situation is determined to be high, and if the relevant situation-specific game is generated, then the impact of deciding on different combinations of actions that the game players take are calculated through a pre-fixed payoff function. Finally, the completed game is solved by finding its dominant strategy, that subsequently, results in proposing the optimal action to the driver.

Our experimental results are divided into three main sections, through which we evaluate the capabilities of the traffic entity, situation, and impact assessment methods. Accordingly, the performance of the proposed cooperative localization approach is assessed by comparing its results with the ground truth solution and that of the other localization methods in various driving test cases. Moreover, two distinct single-vehicle and multi-vehicles categories of driving scenarios, as well as a novel hybrid MEBN inference, demonstrate the capabilities of the proposed traffic assessment model to efficiently achieve situation and threat assessment on the road. Finally, the impact assessment and decision making models are evaluated through two different scenarios of driving in highway and intersection that are formed with various number of player vehicles, and their actions.

Acknowledgements

The preparation of this thesis could not be possible without the valuable assistance of the ones, who accompanied me along this four-year-long journey. First of all, I would like to acknowledge my supervisor, Dr. Fakhri Karray, for all his help and support during my PhD. It was under Dr. Karray's supervision that I prospered in this compelling research area, and ultimately, was able to meet and pass all the milestones one after another. Moreover, I need to thank my co-supervisor, Dr. Mohamed Kamel, whose unique leadership is praised not only by me, but also by all the students who have been working in his team. Dr. Kamel is a precious asset to the Centre for Pattern Analysis and Machine Intelligence. Finally, I am very thankful for all the guidance that my committee members, Dr. Vijay Ganesh, Dr. Lin Tan, and Dr. Baris Fidan, gave me. I was so fortunate to have such an intellectual committee with me along the path.

Also, I would like to take time and thank my dear friends and colleagues, whose motivational admiration, constructive comments, and influential criticisms, paved the way to accomplishing my goals. Bahador, I never forget our long discussions and brainstorming sessions that led to innovative ideas and successful publications. Ridha, you always gave the energy and enthusiasm to work harder and harder. You did great. Farook, the very early stages of starting a new research area are not easy to pass. Thank you for guiding me how to move forward. Finally, I would like to thank the fellow members at the Centre for Pattern Analysis and Machine Intelligence (CPAMI). I am proud to be a part of such an outstanding community.

At the end, I am so grateful to my mom and dad, and my brother, who were always next to me, and supported me, during all the ups and downs of starting a new life in a new country. I am blessed to have you.

Dedication

To my beloved wife, Dr. Armaghan Eshaghi, for her love, unwavering support, and enduring encouragement

Table of Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms and Abbreviations

Abbreviation	Complete Word
AAF	Attention Assist Framework
ADAS	Advanced Driver Assistance Systems
AIR	Are In Range
ANN	Artificial Neural Networks
AOA	Angle of Arrival
ATFY	Active Fuzzy
BN	Bayesian Networks
CLD	Cloudy
CM	Communication Module
CTL	Collision Threat Level
CU	Configuration Unit
CWS	Collision Warning System
DBN	Dynamic Bayesian Networks
DDL	Distance Danger Level
DEA	Distributed Entity Aggregation
DEL	Driver Experience Level
DFG	Data Flow Graphs
DIS	Distance
DM	Decision Making
DRA	Driver Attention
DRD	Driver Drowsiness Level
DRF	Driver Faults
DRK	Drinking
DRS	Driver Situation
DRV	Driver
DSD	Driver Distraction Level

DSE	Dominant Strategy Equilibrium
DSL	Driver Skill Level
DSRC	Dedicated Short Range Communications
DT	Decision Tree
EFG	Extensive Form Games
EKF	Extended Kalman Filter
ENS	Environment Situation
FBN	Fuzzy Bayesian Networks
FMEBN	Fuzzy Multi-Entity Bayesian Networks
FOL	First-order Logic
FOPL	First-order Fuzzy Logic
FOPL	First-order Probabilistic Logic
FRS	Fuzzy Rule Set
FST	Fast
GT	Game Theory
HCI	Human Computer Interaction
HEA	Hierarchical Entity Assessment
HIG	High
HLIF	High-Level Information Fusion
HMI	Human-Machine Interaction
HMM	Hidden Markov Models
HTN	Hierarchical Task Network
IA	Impact Assessment
ITS	Intelligent Transportation Systems
IoC	Internet of Cars
IOM	Input/Output Module
IoT	Internet of Things
JT	Junction Tree
LLDF	Low-Level Data Fusion
LN	Lane Number
MEBN	Multi-Entity Bayesian Network
MLN	Markov Logic Networks
ML	Machine Learning
MMI	Machine-Machine Interaction
MN	Maneuvering
MSDF	Multi-Sensor Data Fusion
NE	Nash Equilirbium
NFG	Normal Form Games

NLP	Natural Language Processing
NOB	Number of Blinks
NOF	Number of Faults
OPRM	Object-oriented Probabilistic Relational Model
OWL	Web Ontology Language
OOBN	Object Oriented Bayesian Networks
PGM	Probabilistic Graphical Models
QRE	Quantal Response Equilibrium
PN	Petri Nets
PP	Pre-Processing
PRM	Probabilistic Relational Model
REG	Regular
RIN	Rainy
ROC	Road Condition
ROT	Road Type
RRT	Rapidly-exploring Random Trees
RSU	Road-side Units
SADAS	Situation-Aware Driver Assistance System
SAW	Situation Awareness
SA	Situation Assessment
SDEMP	Soft-Data Entity Matching Process
SDEK	Sequential Decentralized Extended Kalman
SDL	Speed Danger Level
SHDF	Soft-Hard Data Fusion
SLW	Slow
SMK	Smoking
SNW	Snowy
SPD	Speed
SPN	Stochastic Petri Nets
SSATFBN	Situation-Specific Active Fuzzy Bayesian Network
SSBN	Situation Specific Bayesian Networks
SSFBN	Situation Specific Fuzzy Bayesian Network
SSNFG	Situation Specific Normal Form Games
STO	Situation Theory Ontology
SUN	Sunny
TA	Threat Assessment
TAM	Traffic Assessment Module
TDOA	Time Difference of Arrival

TEA	Traffic Entity Assessment
TEC	Time Eyes Closed
TIA	Traffic Impact Assessment
TOA	Time of Arrival
TSA	Traffic Situation Assessment
TTC	Time To Collision
UKF	Unscented Kalman Filter
UML	Unified Modeling Language
UT	Unscented Transform
UVD	Using Device
V2H	Vehicle-to-Human
V2I	Vehicle-to-Infrastructure
V2M	V2V Aggressive Movement
V2R	Vehicle-to-RSU
V2S	Vehicle-to-Sensor
V2V	Vehicle-to-Vehicle
VANET	Vehicular Ad-hoc Networks
VEH	Vehicle
VES	Vehicle Speed
VMS	Vehicle Movement Situation
VTL	VANET Threat Level
WEA	Weather
WEA	Weather
WIP	Wiper
YOE	Years of Experience

Chapter 1

Introduction

Nowadays, our cities face new challenges such as spectacular population growth, massive pressure on city infrastructure (power, water, health care, transportation) and pollution. The advent of the smart city concept came in response to some of these modern era challenges. Intelligent infrastructure, innovative data process, smart grids and electric vehicles provide synergistic benefits for smart cities. One of the fundamental premises of smart cities is to improve quality of life by developing “smart mobility” [4].

Intelligent infrastructure is a main component of a smart city, and a key enabler for the development of intelligent infrastructure is the Internet of Things (IoT), as it depicts the pervasive integration of sensors within physical infrastructures [4]. As of now, billions of smart sensors are embedded in our cars, bridges, streets, buildings, and within the environment of our living space. These devices are expected to autonomously discover their own environment, connect and interact with their surrounding space, and be able to send out streams of data for various objectives [16]. Recent statistics from Cisco [3] and Ericsson [2] predict that at least 50 billion of “things” will be connected to the Internet by 2020.

Furthermore, the development of Intelligent Transportation Systems (ITS) has been a crucial element in the design of future smart/connected cities, with the objective of super-efficient navigation and safer travel journey. With more than a billion cars on the roads today and growing, road safety has quickly become a major challenging factor to deal with within the transportation industry. Alarming statistics indicate that traffic accidents produce on their own 1.3 millions of fatalities per year [1]. In light of these facts, it is becoming clear that novel alternatives within the transportation industry are deemed necessary. Subsequently, connected cars are quickly becoming a major milestone of the next generation design of ITS. In fact, smart cars within the context of ITS represent a substantial portion of the bigger IoT, whose convergence to Vehicular Ad-hoc NETworks (VANETs) has given birth to a new paradigm called the Internet of Cars (IoC).

Vehicular Ad-hoc NETworks (VANETs), which represent the communication platform for the IoC, have witnessed a strong rise in research activities in the last decades. The purpose is to ensure transportation efficiency, improve safety, and mitigate the impacts of traffic congestion [80]. In VANETs, vehicles are deemed mobile sensor platforms [77] that are able to collect data from their surrounding, infer them, and then, transmit relevant information to the interested entities [140]. To model such connectivity, VANETs rely on wireless communication channels that connects the car to other nearby entities. In the transportation domain, these entities can be cars, Road-side Units (RSUs), public networks, humans, and/or physical sensors. The communication links for each of them are respectively called Vehicle-to-Vehicle (V2V), Vehicle-to-RSU (V2R), Vehicle-to-Infrastructure (V2I), Vehicle-to-Human (V2H), and Vehicle-to-Sensor (V2S). Figure 1.1 illustrates the different types of communication in VANETs. These various communication channels are

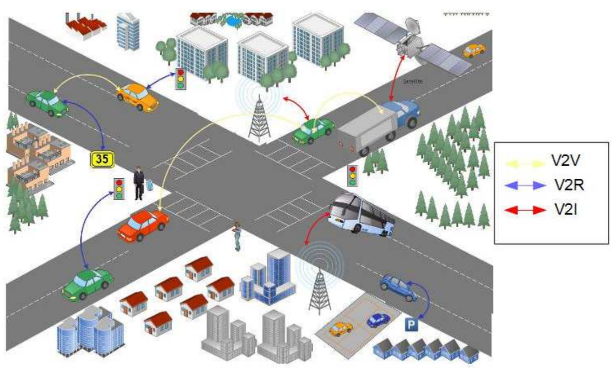


Figure 1.1: Different types of communication in VANETs

used for safety and traffic information flow and ensure an enlarged *situational awareness* of the environment beyond the limitation of the driver's view. Thus, an early perception of potential risks could be attained, and in return, anticipated maneuvers could be taken by the car or the driver. The potentiality of VANETs has been acknowledged with the establishment of ambitious research programs such as WAVE, C2C-CC, CVIS, NoW, VSC [111]. Moreover, radio spectrum has been allocated in North America, Europe, and Japan for the Dedicated Short Range Communications (DSRC) to facilitate the widespread of ITS. Clearly, VANETs pave the way for the cars to become connected devices, and to operate in a data-rich environment. However, such mobility and connectivity come at a cost.

1.1 Motivation

As the number of sensors installed on cars increase, and their connectivity improves, they become a swarm of mobile sensors and information sources that consistently generate and receive huge amounts of data and information. Besides, the external driver's environment presents a variety of data such as weather, road conditions, traffic and social networks streams. Therefore, the levels of abstraction of this information range from lower-level physical sensors data to higher-level human-generated soft information. Connected cars should take advantage of this big data to provide drivers with proper situation awareness.

In most of the IoC applications, cars need to be aware of the current situation (depending on the type of application) to consequently achieve their goals. Therefore, it is crucial to develop a comprehensive and well-organized framework to extract, manage, and interpret the available data and information to consequently achieve proper Situation AWareness (SAW). In other words, the interested entities should be provided with appropriate and convenient knowledge regarding their current (and possibly future) driving situations, that is hidden within enormous amounts of data and information generated from various sources in the IoC environment. As a result, data and information gathering and fusion are essential to provide the required selectivity in the VANETs as it defines a dynamic process to adaptively gather and process information of interest from the IoC environment, and facilitate achieving its objectives.

The importance of SAW on the road is also highlighted by Salmon *et al.* [189]:

"SAW has received far less attention in a road transport context. This is despite the fact that failures related to poor SAW, such as inattention, have been identified as key casual factors in road traffic crashes."

Finally, a well-designed framework should be able to answer these questions: How can different sources of data and information help to enhance the requirements of vehicles in different situations? Which sources of data and information, made available by the connectivity feature of VANETs, are useful, or perhaps necessary, to achieve the required goals? Is there a way to predict the future situations and let the interested entities know about them in time? What is the best method for combining the data and information that come with different levels of abstraction?

1.2 Scope

This thesis is mainly focused on data and information fusion in the IoC, and aims to establish the path, with necessary theoretical background, that leads to SAW. Initially, it is necessary to clarify the distinction between the definitions of the terms data and information that are frequently used throughout this thesis. *Data* is deemed any low-level fact that specifies the features of a certain entity, whereas *information* encloses the facts about an already recognized entity and/or its relationships with other entities. Moreover, data and information fusion community differentiates the definition of Low-level Data Fusion (LLDF) [1] and High-level Information Fusion (HLIF). By definition, LLDF is the fusion of low-level data produced by physical sensors, and recognition of context-related entities in a specific environment to form a unified picture [116]. Alternatively, and as outlined by B. Dasarathy in [56], HLIF is the stepping stone that combines theories, algorithms, and tools to explore the knowledge that lies within the information generated from multiple sources and exists among the *relationships* of various entities to draw a generic *awareness of the situation* to improve the accuracy and robustness of the final decision and action. Therefore, SAW is the result of an HLIF process.

LLDF is deemed a well-studied discipline, as it has been under thorough investigation for many years [116]. In contrast, the HLIF research has just recently attracted much attention in the information fusion community as reflected by the review articles published in that community within the last few years [31, 34, 216]. Accordingly, this thesis concentrates on the HLIF side of the IoC more than on its LLDF one.

Among the applications envisioned for the IoC, which are safety, convenience, and comfort [193], the safety applications are tackled in this thesis. Safety applications allow vehicles to consistently perceive the surrounding environment (specifically the state of other cars or road conditions), and if necessary, avoid incidents by taking proper actions on time. Based on their level of interference, safety applications are either passive or active [106]. Passive applications automatically take physical actions (often at critical moments), and are mainly used in the autonomous driver-less vehicles. However, active applications only provide driving assistance to drivers using proper Human-Computer Interaction (HCI) units, and are generally implemented by the Advanced Driver Assistance Systems (ADAS). This thesis focuses on the later types of safety applications.

Collision (Avoidance) Warning Systems [79, 153, 103, 176], Intersection Safety Management [148, 200, 211], Object Detection [237, 10, 54, 203], Lane Departure Warning Systems, and Blind Spot Mitigation [53, 102, 74] are deemed the major safety applications of the

[1] Low-level Data Fusion is also sometimes called Multi-sensor Data Fusion in the literature

Internet of Cars [238], among which Collision Warning Systems are actually implemented and tested as a part of our simulations.

1.3 Contributions

Although VANETs environment benefits from a variety of sources generating data and information, there has not been any well-defined and concrete approach to take advantage of this diversity in a way to assist the entities of interest better. The main reason for this scarcity is most likely related to the fundamental challenges of HLIF that need thorough consideration. Among these, the issues of uncertainty/ambiguity analysis and semantics/ontologies are indicated as the most important areas of study that have not received enough attention in the past [34].

Following this observation, a novel generic data/information fusion model, termed as Attention Assist Framework (AAF), is proposed in this thesis to achieve enhanced safety in the IoC, which aims at improving the road safety by enhancing the driver's attentiveness. The proposed model is capable of handling various types of low-level data that are available in the VANETs environment (*i.e.*, the data generated by the physical sensors, or those received through different means of communication). Furthermore, the AAF is able to perform a complete high-level information fusion process, *i.e.*, Situation and Threat Assessment (SA/TA), Impact Assessment (IA), and Decision Making (DM), that results in proper SAW for the entities of interest. Our contributions are intrinsic in the methods proposed to model the core of the AAF.

Accordingly, a novel approach based on the idea of cooperative localization is introduced. Our proposed scheme incorporates different techniques of localization along with low-level data fusion as well as vehicle-to-vehicle communication in VANETs, to integrate the available data and cooperatively improve the accuracy of the localization information of the vehicles. The model is further improved by estimating the vehicle location using Unscented Transform (UT) [222] along with Sequential Decentralized Extended Kalman (SDEK) [95] filtering.

Situation and Threat Assessment (SA/TA) using our novel Fuzzy extension to Multi-Entity Bayesian Networks (MEBN) [125], constitute the second contribution of this thesis. MEBNs encompass the expressiveness power of first-order logic, and the uncertainty management of Bayesian networks to model both the semantic and casual relationships, and perform inference while accounting for the potential uncertainties. However, MEBN lack the capability of modeling some imperfect aspects of data such as ambiguity [2] that is an inherent characteristic of human language, and the observations gained from the environment. To overcome this issue, a novel Fuzzy extension to MEBN, called Fuzzy-MEBN, is proposed [84, 87, 85] that is based on First-order Fuzzy Logic [167], and a new way of representing Fuzzy Bayesian Networks (FBN). Moreover, the fuzzy logic is used to match the soft data with the correct entity and states defined in Fuzzy-MEBN.

Impact Assessment (IA) and cooperative Decision Making (DM) using game theory is the third major contribution of this thesis. This is done by adding a game-theoretic component to Fuzzy-MEBN that enables the involvement of active entities (*i.e.,* players) in the situations of interest, whose actions cause relative changes in the states of the situations. Accordingly, various combinations of actions are generated, future situations are hypothesized, and the best action that gives the maximum payoff is reported. It is notable that the payoffs are calculated based on the fuzzy states of the situations of interest that are involved in defining them. This version of Fuzzy-MEBN is called ATFY-MEBN, which is short for AcTive FuzzY-MEBN.

1.4 Organization

The thesis is organized in 6 distinct chapters. In chapter 2 a comprehensive review on situation awareness in the context of connected cars is presented, and our proposed contributions as a part of the most recent research work in this area are highlighted. Chapter 3 introduces the specifications of the low-level data fusion paradigm and presents our novel localization approach. Moreover, Fuzzy-MEBNs and their details are thoroughly discussed in chapter 4. Subsequently, chapter 5 presents an in-depth study on the game-theoretic technique used in ATFY-MEBN. Our novel Attention Assist Framework (AAF) along with the relevant case studies, and the experimental results are introduced in chapter 6. Finally, chapter 7 concludes the thesis through providing a general discussion on the proposed models, and highlighting the future research trends.

[2] The terms “ambiguity” and ”vagueness” are used interchangeably throughout this paper.

Chapter 2

Situation Awareness Within the Context of the Internet of Cars: Review and Recent Trends

The challenges of the Internet of Cars area is highly concentrated on topics such as: routing and communication protocols [186, 199, 65, 15], security and privacy [184, 159], data dissemination [78, 52, 173], simulation [186, 94, 145], information management [109, 225], and information fusion [159, 72, 238, 144]. The last two were motivated by the fact that connected cars operate in a data-rich environment that facilitates achieving Situation Awareness (SAW).

In this chapter, we propose a series of taxonomies that are designed in such a way as to illustrate the path towards proposing a SAW model from both technology-centric and methodology-centric standpoints. It highlights as well the features of different methods and techniques used in the context of the IoC. To appropriately position our contributions, various SAW models are also compared according to different criteria related to their features and applicability.

2.1 Introduction

Application of SAW on the road is theoretically and methodologically studied by Salmon *et al.* in [189]. The authors analyze SAW from three different perspectives: individual (psychological), computational, and socio-technical. In the individual perspective the driver is considered as the center of the model, and the goal of situation assessment is to measure his/her behavior, reactions, and interactions in different situations. Through the second perspective, Salmon *et al.* study the computing (engineering) perspective of SAW that provides the entities of interest with appropriate information by using technological facilities. The computing perspective serves as a bridge that links the individual mental model to in-vehicle technologies and road infrastructure. Different methods and algorithms that aim to propose a solution for an on-the-road SAW task are categorized in this medium. Finally, the third perspective of SA is introduced as the socio-technical systems that are mainly based on the idea of distributed SAW. In these systems, all entities in an environment are assumed to have partial knowledge (awareness of the current situation) about their surrounding, and thus, make communication links to share their knowledge and improve their understanding about the environment. Furthermore, other entities (*i.e.*, the driver or the passengers) make use of this knowledge to address their desired goal, such as avoiding collisions. Some issues such as compatibility, and knowledge scaling are some of the challenges that rise in distributed socio-technical SA systems.

While the individual perspective is well-studied in the literature [117, 234, 221], just a few attempts that exploit computational and socio-technical side of SAW in the Internet of Cars, can be found in the literature. For instance, Markis *et al.* [144] propose a survey on context-aware mobile and wireless networking by mainly discussing context uncertainty handling, acquisition, modeling, exchange, and evaluation. Specifically, Markis *et al.* [144] introduce an abstract classification that while clarifies the main aspects of a context aware mobile network, avoids giving sufficient knowledge about the available methods that aim to model those aspects. In fact, the analysis given in this paper is more abstract and technology-centric (rather than being specific and methodology-centric).

As another example, Kakkasageri and Manvi [109] propose a general taxonomy for information management protocols in safety applications. The presented taxonomy structures information management protocols into four main branches: information gathering, aggregation, validation, and dissemination. Furthermore, the authors introduce different classes of approaches per branch, and introduce the major protocols assigned to them, accordingly. While the proposed classification is very useful in information dissemination problems in VANET, it does not provide any information about how to achieve situation awareness.

2.2 Taxonomy

As defined by Mica R. Endsley [66], SAW is *"the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future"*. Alternatively, SAW is deemed the result of the High-Level Information Fusion (HLIF) that takes place in the levels 2 and 3 of the well-known JDL Model [208] (See Figure 2.1). The inputs of HLIF are the levels 0 and 1, which are mostly related to Low-Level Data Fusion.

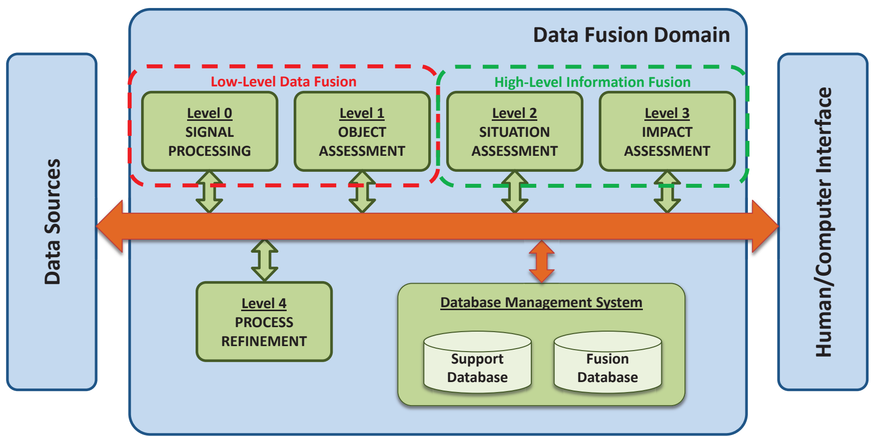


Figure 2.1: The Revised Joint Directors of Laboratories (JDL) Model [208]

2.2.1 Major Entities in the Internet of Cars

Establishing the relationship between SAW, and driver inattentiveness is one of the most crucial, yet mostly unexplored problems in the IoC. As one of the major attempts made in this regard, the impact of driver inattention on road accidents has been extensively studied by Klauer *et al.* [117], who have analyzed the driver inattentiveness using the driving data collected in the 100-Car Naturalistic Driving Study [61]. The authors take advantage of the rich data and information provided from different sources, such as a variety of sensors and driver's personal information and driving records, and explore a number of factors involved in the near-crash/crash incidents. Furthermore, they categorize these factors into four groups of, namely, *Vehicle*, *Environmental*, *Driver*, and *Demographic* factors. Furthermore, they present a variety of conditions under which the inattentive driving behaviors are expected to increase.

As an alternative approach to what Klauer *et al.* [117] proposed as major factors of road incidents, we merge demographics with driver, as they share similar characteristics, and add VANET as another important factor that impacts various driving situations on the road. Figure 2.2 demonstrates a semantic network that encompasses major road safety-related factors in the Internet of Cars. Nevertheless, SAW on the road is deemed an important

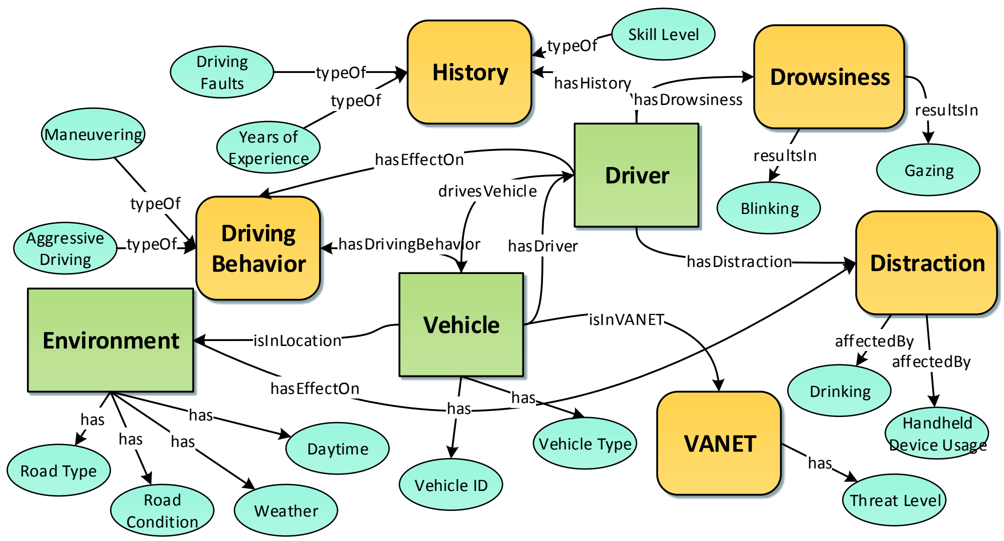


Figure 2.2: Semantic network of main factors involved in road incidents in the Internet of Cars

challenging task, and requires the development of comprehensive computing models to assist the drivers/passengers during driving task [189]. In the following, we study SAW in the Internet of Cars by exploring it through different perspectives.

2.2.2 Components of SAW

To organize the approaches that tackle different aspects of SAW, a structure is proposed that encompasses different components of SAW (see Figure 2.3). Our proposed taxonomy is inspired by the research work of [36] and [69]. The following sections present an in-depth review on different methodologies employed in SAW components.

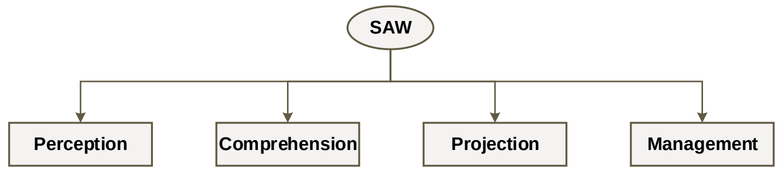


Figure 2.3: The Building Blocks of SAW

2.3 Perception

Perception is the first step of SAW as depicted in Figure 2.3. Devlin [58] defines a situation as a *"structured part of reality that is discriminated by some agent"*. We characterize such a structure as an aggregation of entities and the relationships between them, and build the first stage of SAW as Perception. Figure 2.4 shows the different blocks of the perception stage.

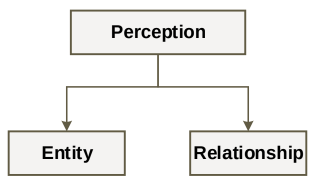


Figure 2.4: The Building Blocks of the Perception Component

2.3.1 Entity

Entities are the result of LLDF that is covered in levels 0 and 1 of JDL Model [208]. Therefore, they inherit all of its issues, such as unavailability and undetectability of data, failure in observing, and misinterpretation of the data [70, 67]. These issues are caused by data imperfection, correlation, inconsistency, and disparateness [116]. The algorithms tackling these problems are based on probability, fuzzy, and possibility theories. Since most of entity perception overlaps with LLDF, which is not in the scope of this paper, the interested reader is referred to [116] for a thorough review on multi-sensor data fusion.

2.3.2 Relationship

Relationships can be interpreted through knowledge representation and reasoning. The reasoning process is the task of structuring elements of knowledge in a way to cover all of its fundamental dimensions, while easing the reasoning process. In fact, the basis of knowledge is composed of entities that are connected to each other through various types of relationships. John F. Sowa in [205] relates the entities through different types of networks that are useful to semantically represent the knowledge, and to reason about it. These networks are composed of, *Definitional*, *Assertional*, *Implicational*, *Executable*, *Learning* and *Hybrid* relationships. Different types of relationships in the context of connected cars are summarized in Figure 2.5.

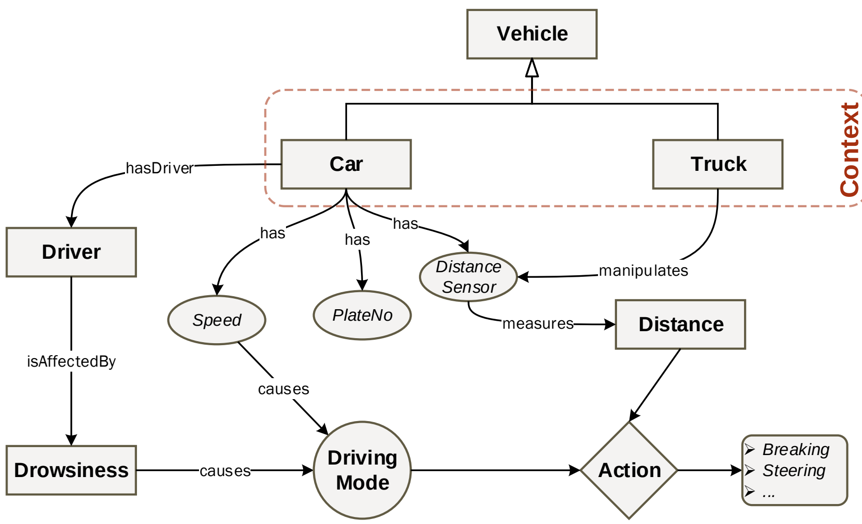


Figure 2.5: The categorization of perception stage

Definitional Relationships

Definitonal relationships represent a hierarchy of entities located on a spectrum with two ends of abstraction (generalization), and specification. The Description Logics [17] and KL-ONE language [42] are able to model definitional relationships. Some relationships is assumed to be true by definition. The Description Logics and its implementation in KL-ONE language are able to model definitional relationships. For instance, derivation of the *Car* and the *Truck* class from the *Vehicle* class is an instance of definitional relationships that is shown in Figure 2.5.

Assertional Relationships

The propositions and statements about a certain fact in knowledge are modeled through assertional relationships. FOL [204] is a powerful language for modeling this type of relationship among entities. First-order Probabilistic Logic (FOPL) [93, 156] and First-order Fuzzy Logic (FOFL) [167] are two important derivations of FOL that handle uncertainty and ambiguity in FOL statements, respectively. Moreover, Markov Logic Networks (MLN) [182] are also deemed useful tools for constructing assertional relationships. Figure 2.5 illustrates the assertional relationships between Car and Driver class, and their attributes.

Implicational Relationships

Implicational relationships are commonly modeled using Probabilistic Graphical Models (PGMs) [121]. PGMs well-known derivations are Bayesian Networks (BNs), Dynamic Bayesian Networks (DBNs) [121], and Fuzzy Bayesian Networks (FBNs) [172]. BNs are defined as a Directed Acyclic Graph $G = \langle V, E \rangle$ wherein nodes represent either discrete or continuous random variables, and edges model conditional dependency between random variables. This dependency is expressed in terms of Conditional Probability Tables (CPTs) and Probability Density Functions (PDFs) on discrete and continuous random variables, respectively. A simple implicational structure between the speed of the Car and the Driver's drowsiness is depicted in Figure 2.5.

Executable Relationships

The structures that model an interaction between a set of entities, collaborating to reach a certain goal, implement means of executable relationships. In such graphs, related entities are able to send messages to one another, to participate in a computation through a pre-defined program, and to break apart or get combined in sub-structures. Data-Flow Graphs [151] and Petri Nets (PNs) [175] are two most well-known formalisms incorporating executable relationships. Two different variations of PNs are object and plan PNs, which are useful for objects and events recognition [127], respectively. An executional relationship between the Driving Mode and the Distance nodes is depicted in Figure 2.5.

Learning Relationships

These types of relationships take advantage of their built-in memory to intelligently respond to new information, and modify their internal state accordingly. Besides, relationships as well as entities are valued based on a weight that indicates how much they influence the outcome. Moreover, some learning networks change their structure to adapt to new context and perform more efficiently. ANN [233] and all of its variations reside in this category.

Hybrid Relationships

A combination of various types of relationships introduced so far, is called a hybrid relationship. One of the most common tools is Unified Modeling Language (UML) [187] that is widely used in software engineering. A combination of definitional, assertional, and executable networks can be simply defined using UML. Multi-Entity Bayesian Networks (MEBN) [125] are also deemed another framework that models hybrid relationships. MEBN aim to improve the conventional implicational relationships in BN by incorporating means of introducing definitional, assertional, and executable relationships powered by *Ontologies* and FOL. The basis of MEBN are MEBN Theories (MTheories), which are powerful tools for modeling domain-specific knowledge for situation assessment. The MEBN theories are composed of several fragments defined on a set of related random variables representing a certain entity, called the MEBN Fragments (MFrags). This constitutes the main characteristic of the MEBN language which is its modularity, *i.e.*, MFrags can be readily added/removed to/from the modeled system, without losing the structural consistency.

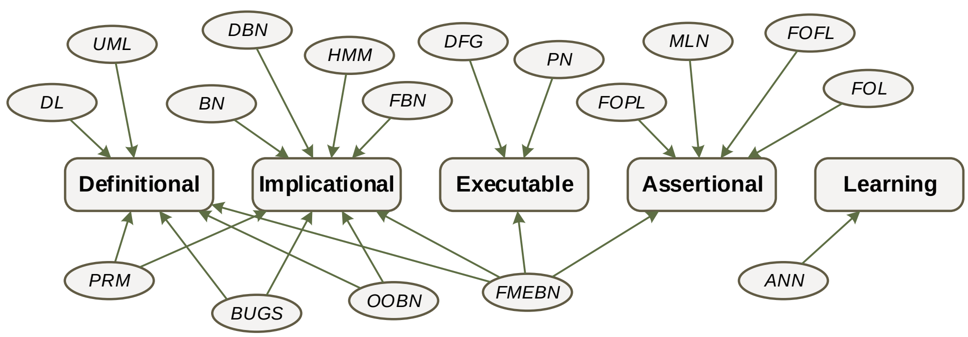


Figure 2.6: Main SAW methods and their capability in handling different types of relationship

Different methods capable of modeling one or more relationships are shown in Figure 2.6. As seen in this figure, MEBN, as well as their Fuzzy Extension (Fuzzy-MEBN), are able to model 4 out of 5 types of relationships and is ranked among the best methods. This is due to the fact that MEBN borrow OWL [219], FOL and BN (enhanced with utility nodes) to handle definitional, assertional, implicational, and executable relationships.

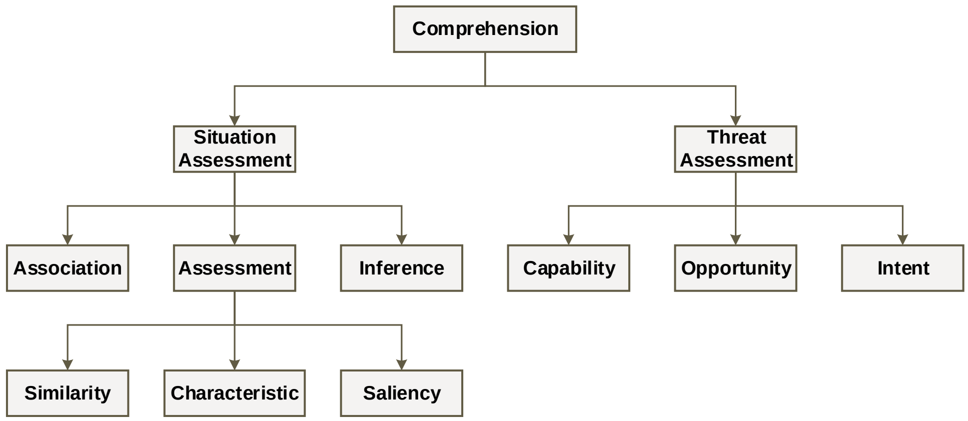


Figure 2.7: The Building Blocks of the Comprehension Component

Another insight observed in Figure 2.6 is that most of these methods can be categorized into either PGMs, or semantic models. In the next section, we show that the two main concepts in comprehension are uncertainty and semantics, which are respectively managed by PGMs and semantic models (*i.e.*, ontologies).

2.4 Comprehension

The assessment of a situation, as well as its possible threat, takes place after the perception of the entities and their relationships. This constitutes the two main blocks of the comprehension component: *Situation* and *Threat Assessment*. Situation Assessment (SA) is the most important module in SAW, whose imprecision leads to incorrect judgements [67], which in driving scenarios, will result in mortal accidents. Figure 2.7 shows the details of comprehension component of SAW and its underlying building blocks.

2.4.1 Situation Assessment (SA)

SA is the main process that leads to situation awareness. Steinberg [135] defines SA as: *"the estimation and prediction of structures of parts of reality, i.e., the aggregation of relationships among entities and their implications for the states of the related entities."* Moreover, Blasch *et al.* [36] highlight main tasks of SA as information aggregation coming from multiple sources, the creation of a bridge between user and automation, and the data management. We separate the main components of SA as: *Association*, *Assessment*, and *Inference* that are respectively corresponding to the aggregation of situations, the measurement of situations properties, and the estimation of the situations states.

Association

Association is the task of selecting a particular set of entities, and structurally arranging them into a situation based on a specific *context*. In other words, it is the context that dictates which entities should be chosen, and how they need to be related to create situations. These types of situations are called *context-aware situations*. Contextual information is usually provided by either domain experts or relevant datasets, and is often used as an extra setting to improve the performance of an assessment procedure. For instance, Wang and Ji [223] augment a regular DBN with contextual information, namely, kinematic and appearance features to detect a specific set of events in video surveillance.

One of the most common methods for providing contextual information is through defining ontologies [120]. Ontologies are useful formal way of representing entities and modeling different types of relationships, namely, definitional and assertional. There have been significant contributions in utilizing ontologies for SA in the literature [119, 18, 22, 63]. Most of these methods define an ontology for a specific context, and semantically put together the contextuall related information to make it machine-readable. Figure 2.2 shows a sample ontology for road safety, which appeared in [84]. Context-aware situations can be related based on their shared characteristics and similarity degrees. Moreover, situation evolution and hierarchical situation arrangement are also deemed among effective approaches for associating situations.

Situation evolution, which is sometimes referred to as *"situation tracking"* in the literature, has been studied and deemed a challenging issue in HLIF [135, 37]. A situation can evolve towards two main dimensions: *temporal* and *lateral* [87]. The situations with temporal nature evolve along with time. These situations contain one or more dynamic entities whose current state depends on their previous states, and therefore, impose their temporal feature on their residing situations. Moreover, the situations whose topology changes upon addition or removal of new entities (observations) are deemed to demonstrate lateral evolution. One of the immediate outcomes of situation evolution is the concept of *event* that is theoretically defined in [87].

In addition to evolution, situations can also be related through hierarchical situation association, as presented in our previous research work [87]. In that paper, we define the *super*-situation concept as the combination of *sub*-situations, which are also known as component situations [135]. For example, in road safety problems, a situation can be configured to monitor a driver's attention, while another one is set to reflect the environmental condition.

Assessment

Evaluation of the similarity and the salience of situations, which gives proper insight about the situations and their closeness, are done at this stage. Employing the hierarchical situation arrangement method introduced before, one can identify similar and salient situations by simply applying set theory operators on super-situations, say S_i and S_j, to measure their closeness. For instance, $S_i \cap S_j$ and $S_i - S_j$ can be a good measurement to show how much two situations S_i and S_j are similar, and how salient S_i is with respect to S_j, respectively. Besides, semantics similarity can also provide a good metric to compare the commonness of two situations [164, 9]. According to [8] the semantic similarity between two situations depends on the types of entities, and type of relationships along with their semantic weights. Measuring similarity between situations ontologies can also give us an insight on how much two situations are close to each other. Besides, salient situations can also be found easily when their corresponding ontologies are compared against the other ontologies in the same context. Maedche and Staab [142] tackle ontologies similarity measurement by considering a two-layer view: Lexical and Conceptual levels. In lexical level, the similarity is defined based on the edit distance that is originally proposed by [132]. Besides, the structure of ontologies are compared at conceptual level through defining *concepts* (assertional relationships) and their *hierarchy* (definitional relationships) [142].

Inference

Inference is the task through which the state of a situation is estimated, and is the final component in situation assessment process. Depending on the underlying structure of entities and relationships, different algorithms may be used for inference. An efficient inference algorithm must be capable of handling the challenges of relationship types to subsequently achieve reasonable results. The majority of the literature focus on handling uncertainty and semantics while performing inference.

- *Semantics inference* is commonly tackled through semantics analysis by using ontologies [22, 23, 147, 24]. In particular, Wache *et al.* [220] introduce structural and semantic heterogeneity groups for various sources of information. They take also advantage of common vocabulary and ontology mapping to measure how close is an ontology to a desired one. Besides, Kokar *et al.* [119] introduce Situation Theory

Ontology (STO) which is an ontology-based situation awareness framework developed based on the Situation Theory [21, 20, 19] and OWL [219]. Moreover, Little and Rogova [137] create the in-between connections between a higher-level ontology (with more abstract structure) and a pre-determined domain-specific ontology.

- *Uncertainty inference* handling deemed a crucial part of an inference engine. Indeed, entities are created through a LLDF system, and accordingly, reflect all the imperfectness of physical sensors, trustworthiness of information sources and limitations of data fusion methods [139]. Therefore, an inference engine must be able to promisingly handle such inherent uncertainty. Karlsson [112] categorizes the Uncertainty Management Methods (UMMs), into three groups of *Bayesian*, *Dempster-Shafer (DS)*, and *Imprecise probability* approaches. Besides, Sowa [205] characterizes uncertainty through his three orthogonal schema. The first aspect of this schema studies the spatial uncertainty in the physical nature of entities. The second aspect defines the time effect on entities, and studies the temporal uncertainty. Standard Hidden Markov Models (HMMs) [230, 236] and its variations [218, 170, 228, 170, 169, 141] are Probabilistic Graphical Models (PGMs) that are capable of performing inference when temporal information exists. Finally, uncertainty in ontological arrangement of entities (*i.e.*, independent, relative, and mediating relations) comprises the third aspect of Sowa's schema. As an example, Probabilistic Ontology (PR-OWL), proposed by Laskey *et al.* [126] is a remarkable representation of uncertainty among ontological arrangements. Furthermore, Laskey in [125] introduces the MEBN model that is a Bayesian logic language for PR-OWL. MEBN use Situation-Specific Bayesian Networks (SSBNs) as its inference engine. The algorithm is based on works of Mahoney and Laskey [143], in which a minimal SSBN is approximated using the combination of particular types of random variables extracted from a given query. Last but not least, Stochastic Petri Nets (SPN) [25], Markov Logic Networks (MLNs) [30, 217], Probabilistic Relational Model (OPRM) [101], Blog [155], BUGS [214], and OOBN [122] are also among the useful models that handle uncertainty and semantics simultaneously.

2.4.2 Threat Assessment (TA)

TA is the task of measuring the *Capability*, *Opportunity*, and *Intent* of an assessed situation [135]. Therefore, TA is very close to SA and is built up on it to create a bridge from SA to Impact Assessment (IA). The output of a TA model is a set of situations hypotheses which are deemed threatening and can be used to assess the impact of a specific situation. Alan N. Steinberg [207] proposes a threat assessment model that produces situation hypotheses based on a set of actions performable on the input situation. These actions are firstly determined based on the *capability model* of the input situation. Furthermore, some of them are filtered out according to their constraints and the situation *opportunity model*. Finally, the *intent model* probabilistically implies the potential action outcomes. The importance of each of these models differs from domain to domain. Analyzing the intent model is the most crucial one in Internet of Cars.

Capability

Capability model is used to generate feasible actions that are related to the current context, and the specific world model in which TA is being done. In a connected cars context, all the actions that a driver may take, as well as those of surrounding vehicles, can be considered as the set of actions generating situation hypotheses. For instance, in a highway driving scenario, changing lane, acceleration, deceleration, taking the exit ramp, *etc.* can be counted as the set of actions that a driving car is *capable* of taking, according to its design and the use of the actions (*i.e.*, a regular car is obviously *not* capable of changing altitude as its design does not let it to do so). Action selection in transportation context can be seen in research work of [13, 123, 90].

Opportunity

The opportunity that an entity gains to threat another entities is another important factor in TA. Steinberg [207] relates this factor mainly to the target being threatened. In other words, if the target is accessible and easily vulnerable to threats, then the actions opportunities of the threatening entity are boosted. The implication of how much a target is accessible and vulnerable can be done using the information gained through ontological inference, observation, and communication.

Intent

The decomposed goals of a threatening entity are reflected through its intent. In other words, an intent model is concerned with detecting the objectives of an active entity and consequently measuring the likelihood of the hypothesized actions. According to Steinberg [207], an agent's high level objectives and outcome assessment are the two most important factors in an intent model. Moreover, an intent model can be analyzed from both cognitive and computational perspectives.

On the computational side, *Expert Systems* and *Machine Learning* based approaches are the most common tools for intent recognition. In the presence of datasets, different machine learning approaches can also be used to classify intentions. For instance, SVM and its derivatives are used by Aoude and How [12], and Aoude *et al.* [13, 14] for drivers' intention detection at intersections. Furthermore, Hou *et al.* [100] propose a model that uses Continuous Hidden Markov Models (CHMM) for intention detection. Moreover, as an expert system based approach, Benavoli *et al.* [26] propose a TA model that uses evidential networks to assess the capability and intent of the threatening entities. In a similar approach, Liebner *et al.* [134] first propose an intelligent driver model, and then, design a Bayesian Network to recognize driver intent at urban intersections. Similarly, Lefevre *et al.* [131] introduce a framework that aims for recognizing driver intent at intersections based on the intersection context.

Finally, as a cognitive model, Salvucci [190] proposes a framework to detect driver lane changes based on the cognitive model of driver behavior and a *mind-tracking* architecture. Similar approaches can be seen in [191, 150, 64]. Based on the presented discussion on SA

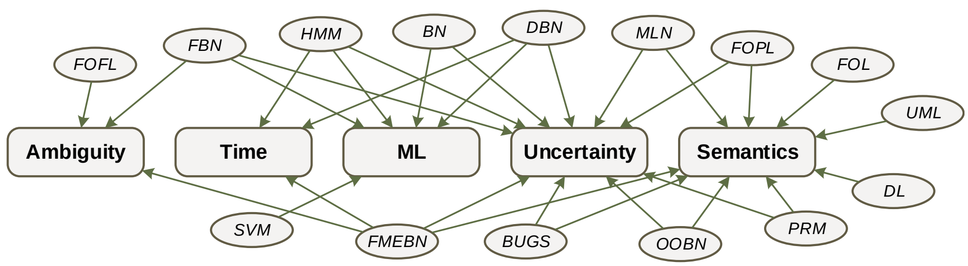


Figure 2.8: Major SAW methods and their capability in handling main concepts in the comprehension component

and IA, we extract Ambiguity, Time, Machine Learning (ML), Uncertainty, and Semantics as the most important building blocks of the comprehension component. Moreover, we relate an important subset of SAW approaches to these blocks to show their capability in modeling each of them. The result is illustrated in Figure 2.8. While all of the PGMs are able to handle uncertainty, some of them are also able to handle temporal dimension, and they also can employ Machine Learning (ML) techniques as shown in Figure 2.8. However, almost none of the basic PGMs can neither model semantics nor ambiguity. In the comprehension component, Fuzzy-MEBN are the top ranked method as they model all of its building blocks except for the ML. In fact, although at its early stages, learning is still a challenge in MEBN (and their Fuzzy extension), as mentioned by [174].

2.5 Projection

Impact Assessment (IA) and Decision Making (DM) reside in the projection component of SAW. Future situations are predicted through IA, and proper actions to be taken are advised through DM. In a connected cars scenario, impact of a threatening situation can be an incident situation which may be avoidable by effectively choosing a proper action through decision making. Figure 2.9 depicts the building blocks of the projection component.

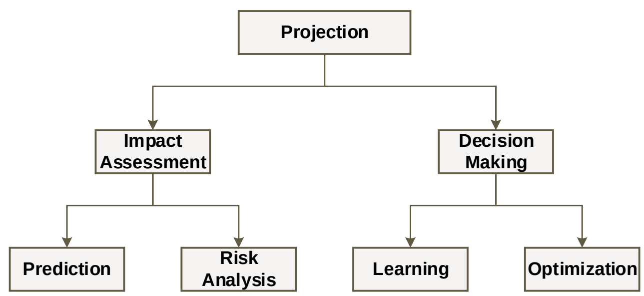


Figure 2.9: The building blocks of the projection component

2.5.1 Impact Assessment (IA)

IA is composed of prediction and risk analysis steps. An IA model obtains the previously assessed situations and their threats (the situations capabilities and intents) to calculate the likelihood of future hypothesized situations. Therefore any approach capable of generating a set of hypothesized situations, given the current situations and their threats, can be potentially used as an IA model. Among different approaches in the literature, Probabilistic Graphical Models (PGMs) and Game Theory (GT) seem to be more promising methods for IA. In the following, we will detail prediction and risk analysis, the two main components of IA.

Prediction

Estimating of the future of situations requires an understanding of the current world model (current situations of interest), involved active entities that may change the state of situations accordingly, a set of actions to be taken by the active entities, and a transition rule that determines the outcome of an action on a specific situation. Such an arrangement can be theoretically modelled using GT, as it is seen in [11, 180, 50, 44].

A conventional game setup in GT is shown by a tuple $\langle \mathcal{P}, \mathcal{S}, \mathcal{D}, \mathcal{T}, f \rangle$, wherein $\mathcal{P} = \{1, 2, \cdots, N\}$ is the set of players, $\mathcal{S}$ is the set of states, $\mathcal{D} = \mathcal{D}_1 \times \mathcal{D}_2 \times \cdots \times \mathcal{D}_N$ is called the decision space and is created from the actions of each individual player $\mathcal{D}_i = \{a_1, a_2, \cdots, a_m\}$, $\mathcal{T} : \mathcal{S} \times \mathcal{D} \to \Delta(\mathcal{S})$ is the transition function that calculates the likelihood of residing in each state after taking decision instance $d_i \in \mathcal{D}$ when observing world state $s_i \in \mathcal{S}$, and finally $\{ : \mathcal{S} \times \mathcal{D} \to \mathbb{R}^N$ is the pay-off function that awards/punishes the players upon perceiving world state $s_i \in \mathcal{S}$ and taking decision instance $d_i \in \mathcal{D}$.

To efficiently implement IA using GT, we first need to identify the active entities (players) of a SAW scenario, *i.e.* vehicles, drivers, pedestrians and VANET in our case. Then, we need to formulate all the ingredients of a game model. Furthermore, world model is represented by a set of situations of interest (*i.e.*, *Vehicle*, *Driver*, *Environment* and *VANET* situation in our case) on which the actions are performed. The players take an action based on their inner state that is influenced by their capability and intent (threat assessment output). The outcome of the action selection can be a new situation, or a change in current situations states. The birth or change of a situation reflect the impact of current situations subject to the actions taken.

As some examples in transportation domain, a prediction and planning framework based on sequential game playing for collision avoidance is introduced in a technical report by Broadhurst *et al.* [43]. Besides, Aoude *et al.* propose an TA/IA model that is a combination of game theory and Rapidly-exploring Random Trees (RRT). Chen *et al.* [51] also look at using GT in VANET from another point of view, and take advantage of mechanism design to persuade the vehicles to send message via communication gateways to each other.

Risk Analysis

This step comes immediately after prediction is the task of weighting the hypothesized situations and analyzing the risk of reaching them based on a certain criteria. It is an application dependent step and is performed based on a risk metric that is pre-defined by a domain expert. The output of a risk analysis task is prior knowledge for a decision making system.

2.5.2 Decision Making (DM)

Decision Making (DM) is an important part of a SAW model as seen in famous information fusion decision making models such as Data Fusion Information Group (DFIG) [35], Observe-Orient-Decide-Act (OODA) Loop [33], and Multi-player OODA [32]. A well-designed DM model should efficiently take actions given a set of hypothesized future situations. The effectiveness of an action is measured based on a utility function that is normally defined either based on a domain expert interpretation, or statistically compiled relevant datasets. We group DM models in SAW based on the underlying approaches tackling SA, TA, and IA. Therefore, PGM and GT as well as Machine Learning based methods (mainly Decision Trees) are deemed the main categories of DM models. A common fact between all of these categories is that the set of actions to be taken should be known a priori. According to Isermann *et al.* [106], these actions can be: 1. Steering, 2. Braking, and 3. A combination of both.

Probabilistic Graphical Models

PGMs simply tackle the DM problem by having the capability to include decision nodes as a part of graph [121]. The decision node is usually dependent on the states of one or more parents. Therefore, it probabilistically determines the likelihood of an action given the states of parents.

Game Theory

Internally provided with a decision making, a conventional game model contains a decision space $\mathcal{D} = \mathcal{D}_1 \times \mathcal{D}_2 \times \cdots \times \mathcal{D}_N$ that is formed from the actions of each individual player. Therefore, for a specific situation wherein two or more active entities are involved, the decision space of the game model is all of combinations of their actions. Furthermore, the best action to be taken would be the combination that satisfies both the consensus and the individual utilities which is measured by pay-off function f. See [110, 212] for some examples.

Machine Learning

The techniques and methods in machine learning are theoretically rich and often easy-to-implement to create a DM framework. Common classifiers can be trained, assuming that, the relevant driving datasets that show the driving conditions, the action(s) taken, and the outcome, are available. However, the main problem in a connected cars context is that such datasets are neither not easily achievable, nor contain such complete data/information. Therefore, other techniques such as Decision Tree (DT) can be employed to account for the outcome of performing a set of pre-defined actions. The reader is referred to [188] for a complete survey of different DT techniques. The most important groups of approaches in

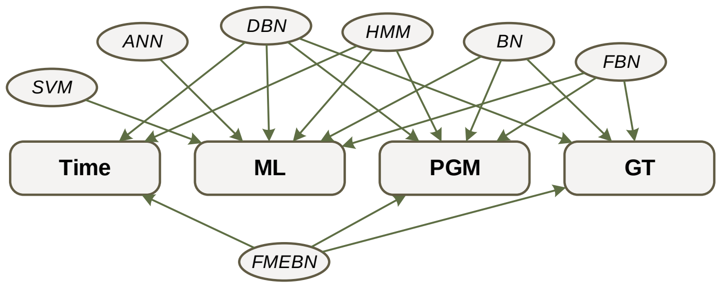


Figure 2.10: Main SAW methods and their connections to the main groups of approaches, along with Time, in the projection component

the projection component are PGMs, GT, and ML, as they are used to predict the outcome of a particular situation. Besides, time is by meaning an inevitable notion in projection. Therefore, it is added to specify the capability of the main SAW methods in handling it. Figure 2.10 depicts these methods and their connections to the groups, and also the time. As demonstrated in the figure, basic PGMs are also linked to GT methods, since it is assumed that they are able to include utility nodes, and incorporate set of actions, which is subsequently led to the definition of action profiles. All the temporal PGMs as well as Fuzzy-MEBN are able to include time, and therefore, are linked to it accordingly.

2.6 Management

Management is the last component of SAW, whose main duty is to ease the collaboration between a SAW model and the human operator, while refining the model's performance. As it is depicted in Figure 2.11, this component is composed of planning, acquisition, monitoring, refinement, and evaluation components.

Planning

Planning is an essential part of a SAW framework to accomplish a specific task. Based on input parameters (current situation, feasible actions and goal), a planning algorithm should efficiently handle the available resources [239]. Subsequently, it guides the whole system to reach its goal by taking appropriate actions at the right time. In general, the planning

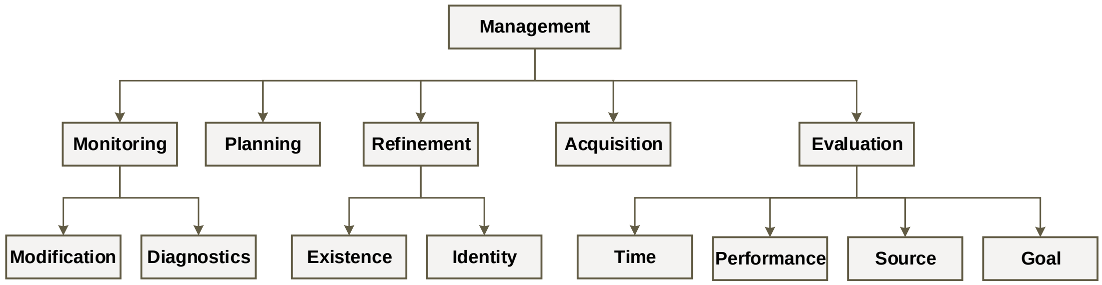


Figure 2.11: The Building Blocks of the Management Component

algorithm is in the form of a constrained optimization problem. For example, the goal of the planning algorithm in a Collision Warning System (CWS) [84] is to ensure safety of the vehicle in unsafe situations. This is done by either providing safe driving hints for the driver, or by physically taking actions, such as steering, braking, *etc.*, to avoid a collision. Meanwhile, the vehicle should also deal with communication bandwidth and other possible constraints that are deemed shared resources.

Acquisition

Obtaining the most relevant data/information sources for perception phase is another crucial task that lies in Management component. This is mainly due to the complexity of some environments, such as connected cars, where the number of data/information resources is high. Heintz and Dragisic [96] tackle this problem by annotating the sensors based on their semantic structure. Therefore, if the ontologies defined for both sensors and the services match, then it means that the sensors are gathering relevant data for that specific service.

Monitoring

It is through the monitoring process that the human operator of a SAW framework (the driver in VANET environment) gets to analyze the current situation, its threat, and its impact to the future. Human-Computer Interaction units are among the most useful tools for monitoring the SAW output for the human operator [162]. Besides, the impact of the SAW framework on the human operator is also diagnosed in this work.

Evaluation

Perhaps the most important role in the management component of a SAW system is played by the evaluation step. Imprecise evaluation may lead to operational errors. Time, performance, source, and goal are the main criteria for evaluating a SAW framework that are both domain-specific and application dependent (See [113]).

- **Time** is one of the main evaluation metrics in a VANET that is usually defined situation-specific. Some of the most common time metrics are: Time To Predict Collision (TTPC) [210], Time To React (TTR), TTE (Time To Enter), TTD (Time To Disappear), TTB (Time To Brake), TTK (Time To Kickdown), TTS (Time To Steer), [97], TTC (Time To Collision) [157], Post-Encroachment Time (PET) [194], and Deceleration to Safety Time (DST) [197].

- **Goal** metrics represent the enhancement/degradation of certain factors in a domain-specific application. In the Internet of Cars, steering and braking behavior, conflicts and driving errors, driver workload, physiological response, and glance frequencies [128] can be chosen as goal evaluation criteria.

- **Performance** criteria are mainly related to the efficiency of a SAW model in different aspects. Uncertainty is one of those aspects whose evaluation plays an important role in improving the performance of a SAW model. Uncertainty Representation and Reasoning Evaluation Framework (URREF), proposed by Costa *et al.* [55], is a complete framework for uncertainty evaluation that aims to improve metrics such as timeliness, accuracy, and confidence.

- **Source** evaluation becomes important when dealing with shared and access-time-limited resources for data/information acquisition. In such environments, a well-designed SAW model should be able to provide its components, the richest input data/information, while keeping the shared sources as free as possible for others' use. Moreover, other relative issues such as low bandwidth, low storage, *etc.*, may also appear in a VANET. Therefore, resource allocation metrics [75] are deemed useful in this category.

Refinement

Refinement of a SAW model is basically focused on the human operator, and it deals with refining the whole process to alleviate the pressure on the driver beyond the wheel [38, 165].

2.7 Discussion

Our observations on the different SAW methods and their capabilities in covering different SAW components are presented in Table 2.1. As seen in Table 2.1, MEBN, along with

Table 2.1: Comparison of different SAW methods

	Perception					**Comprehension**					**Prediction**			
Method	Definitional	Assertional	Implicational	Executable	Learning	Ambiguity	Time	ML	Uncertainty	Semantics	Time	ML	PGM	GT
BN	×	×	✓	×	×	×	×	✓	✓	×	×	✓	✓	×
DL	✓	×	×	×	×	×	×	×	×	✓	×	×	×	×
UML	✓	×	×	×	×	×	×	×	×	✓	×	×	×	×
DBN	×	×	✓	×	×	×	✓	✓	✓	×	✓	✓	✓	✓
HMM	×	×	✓	×	×	×	✓	✓	✓	×	✓	✓	✓	×
FBN	×	×	✓	×	×	✓	×	✓	✓	×	×	✓	✓	✓
DFG	×	×	×	✓	×	×	×	×	×	×	×	×	×	×
PN	×	×	×	✓	×	×	×	×	×	×	×	×	×	×
SPN	×	×	×	✓	×	×	×	×	✓	×	×	×	×	×
FOPL	×	✓	×	×	×	×	×	×	✓	✓	×	×	×	×
FOFL	×	✓	×	×	×	✓	×	×	×	✓	×	×	×	×
MLN	×	✓	×	×	×	×	×	×	✓	✓	×	×	×	×
FOL	×	✓	×	×	×	×	×	×	×	✓	×	×	×	×
PRM	✓	×	✓	×	×	×	×	×	✓	✓	×	×	×	×
BUGS	✓	×	✓	×	×	×	×	×	✓	✓	×	×	×	×
OOBN	✓	×	✓	×	×	×	×	×	✓	✓	×	×	×	×
ANN	×	×	×	×	✓	×	×	×	×	×	×	✓	×	×
SVM	×	×	×	×	×	×	×	✓	×	×	×	✓	×	×
FMEBN	✓	✓	✓	✓	×	✓	✓	×	✓	✓	✓	×	✓	×

their Fuzzy extension, are the top model in covering various aspects of perception, comprehension, and projection components. Other models that handle semantic and causal relationships simultaneously, are also considered powerful tools for perception and comprehension. Besides, temporal PGMs, such as DBN and HMM, have also shown good performance in comprehension and projection. As we also discussed in section 2.5, GT is a promising approach for projection. Moreover, some PGM tools, such as BN, DBN, and FBN, are capable to model actions and action selection through incorporating utility nodes. Therefore, they can be modified and employed in a game-theoretic IA algorithm. This is why most of the PGMs are deemed potential methods in modeling GT. Obviously, MEBN are the best SAW framework that incorporate all almost all the SAW components. However, MEBN lack learning capability and incorporation of ML techniques, which addressing them can a future direction for this powerful model.

2.8 Summary

In this chapter, we highlighted the role of Situation Awareness (SAW) in the Internet of Cars concept, by proposing an in-depth review on the different SAW components, and explaining how different relevant methods can model the main aspects of each component.

In the proposed SAW taxonomy, the perception component entitle entity and relationship as its main aspects. Subsequently, different models/methods that are capable of perceiving different types of relationships are introduced. The comprehension component is consisted of Situation Assessment (SA) and Threat Assessment (TA) components. We show in that uncertainty and semantics analysis are the two most important aspects of the comprehension component. Besides, ambiguity handling, timeliness, and Machine Learning (ML) are also identified as important concepts in comprehension. Impact Assessment (IA) and Decision Making (DM) are the modules that aim to predict the future of a certain situation, and decide on the proper action, respectively. Notion of time, along with ML, Probabilistic Graphical Models (PGMs), and Game Theory (GT) are distinguished as the main concepts and approaches that facilitate projection in SAW. The final SAW component is management that mainly deals with interacting with human operators, refining the processes, and evaluating the whole model.

The paradigm of Internet of Cars is fast becoming reality, and it is crucial to know the ups and downs of different methods in SAW. Therefore, we discussed different approaches that show various capabilities in handling major aspects of SAW components. The comparison made in this chapter can help in recognizing the solid directions of SAW research, and their applicability in the Internet of Cars.

Chapter 3

Low-Level Data Fusion for Cooperative Localization

One of the challenging tasks in Vehicular Ad-hoc Networks (VANETs) is to find an accurate localization information. In this chapter, we have addressed this problem by introducing a novel approach based on the idea of cooperative localization. Our proposed scheme is based on a generic Low-Level Data Fusion (LLDF) model for VANETs, which is able to incorporate different LLDF techniques as well as VANETs capabilities, such as vehicle-to-vehicle (V2V) communication, to integrate the available data and provide information for the interested entities. Accordingly, our LLDF model is used to fuse the low-level data generated from the physical sensors installed on the vehicles, and take advantage of the V2V communication, to cooperatively improve the accuracy of the localization information of the vehicles. Moreover, further improvement has been achieved by estimating the vehicle prior mean and covariance using Unscented Transform (UT) together with Sequential Decentralized Extended Kalman Filtering.

3.1 Introduction

The most important features of VANETs that differentiate it from conventional Intelligent Transportation Systems (ITS) are their network-based structure, and their capability in communicating with other vehicles, Road-Side Units (RSUs), and the infrastructure through vehicle-to-vehicle (V2V), vehicle-to-RSU (V2R), and vehicle-to-infrastructure (V2I) communications [83]. This provides the vehicles and their drivers with the information needed by the applications (sometimes called "services"). In other words, vehicles communicate with each other (V2V) to share any kind of information that might be helpful for serving a running application. For example, vehicles moving on the same lane can send their velocity information to each other to provide information on safe speed under safety service.

The information flowing between vehicles, RSUs, and the infrastructure are usually composed of various context-based attributes. For example, the information related to a safe driving service can be attributed with the real-time location and speed of vehicle, as well as its distance to the neighboring vehicles, and the road condition. Vehicular communication systems allow cars in the same zone to instantly communicate with one another over a wireless network, to exchange these attributes [209]. One of the important attributes of information in VANETs is *location*, whose accurate calculation is a challenging issue [41]. Location of a vehicle is often determined by using commonly used sensors such as odometer and Global Positioning System (GPS). Although using GPS is fairly easy and has low-cost, it sometimes results in inaccurate measurements which is mainly due to satellite blockage, especially in urban areas; therefore, the problem of localization in VANETs remains an open issue.

This chapter proposes a new localization approach for finding the location of a vehicle using vehicle motion model (for modeling the dynamic motion model of a vehicle) along with V2V communication, and a novel low-level data fusion framework in VANETs [80]. We measure the belief of each vehicle about its current location using Extended Kalman Filter (EKF) [215], and then improve it by communicating with the neighboring vehicles and incorporating their beliefs about the location of that vehicle. This approach allows us the simultaneous and efficient use of all the available resources of data unlike inability of the existing systems. This would leads us to achieve a real-time autonomous framework in VANET due to its recursive online location estimation and ability to recover from failure.

3.2 Background and Related Work

The first two levels of the JDL model constitute the Low-Level Data Fusion (LLDF). This section briefly introduces the main challenges that LLDF faces, and presents a generic taxonomy [116] for that. Besides, we describe the method we have adopted to our new localization approach for VANETs, and introduce major state estimation filtering algorithms.

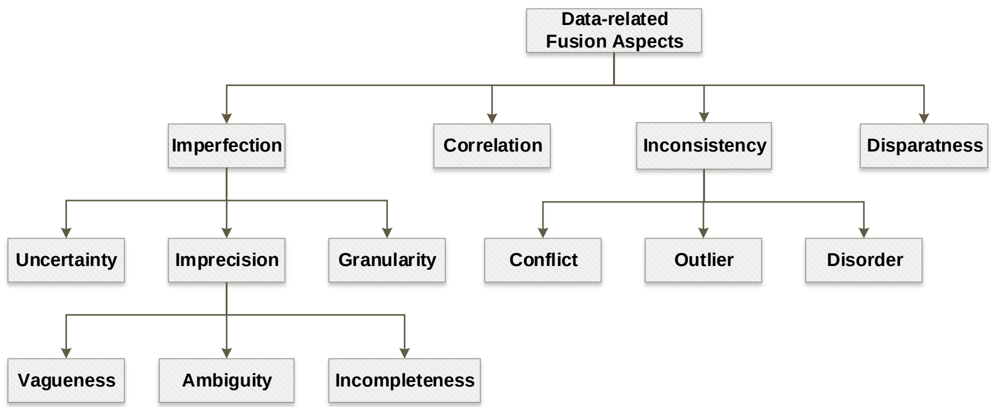


Figure 3.1: Taxonomy of data-related fusion aspects [116]

3.2.1 Low-Level Data Fusion

The basis of VANETs is established on the underlying data that is made available through various sources. VANETs can help the driver to have a safer, more secure and convenient driving experience by exploring these data, and providing useful mobile information, such as safe distance from the leading vehicle, or real-time routing service, while also satisfying the green environment requirements implicitly. However, The main challenge is that such information is hidden among a huge amount of low-level data gathered from various on-board sensors, as well as communication links. This is clearly requires a Low-Level Data Fusion (LLDF) process that provides us with the proper *information* upon necessity.

Various aspects of LLDF, which is also known as Multi-Sensor Data Fusion (MSDF), is thoroughly reviewed by Khaleghi *et al.* [116]. In this paper, the authors propose a data-centric taxonomy of MSDF methodologies, which can be efficiently used to select the appropriate methods in resolving different challenges in MSDF. Khalgehi *et al.* [116] summarize these challenges to: data imperfection, outliers and spurious data, conflicting data, data modality, data correlation, data alignment/registration, data association, processing framework, operational timing, static *vs.* dynamic phenomena, and data dimensionality. These issues are structurally arranged in a taxonomy that is illustrated in Figure 3.1

3.2.2 Localization in VANET

A categorization of different approaches of localization in VANETs is introduced in [41], in which the methods classified are based on: Global Positioning Systems, Map Matching, Dead Reckoning, Cellular Localization, Image/Video Processing, Localization Services, and Relative Distributed Ad-Hoc Localization. Although each of the methods in each of the groups, deals with the problem of localization in VANETs by taking advantages of the basic methods under this category, only a few methods are based on combined approaches. We introduce here some interesting relevant contributions that are either classified under one of the categories mentioned above, or is an integration of some of them.

In [7], the vehicle localization is performed based on the directional information using dual radios. The localization is based on measuring the inter-radio distances between each node (vehicle) to other nodes (vehicles) within their proximity. This directional/dual wireless radio localization (DWRL) scheme does not require the GPS and the algorithm consists of semi-localization and rigid-localization. One of the nodes is designated/located as *sing* node to initiate the localization process. The sink node then selects within its wireless range a node to be semi-localized. Next, in the rigid localization other unknown nodes are located with respect to the known locations of the sink node and semi-localized node.

The idea of using radio-range techniques for measuring the distance (radio-location) between a sender and a receiver is mainly discussed in [45]. In this paper, different radio-ranging distance measurement methods are categorized based on the way the distance is estimated. Accordingly, the distance between the sender and the receiver is estimated based on theoretic modeling of path loss attenuation of the radio-location signal. Moreover, the distance can be estimated from Angle of Arrival (AOA) as well as Time of Arrival (TOA) information as they measure the angle of incoming signal at the receiver and one-way propagation time between transmitter and receiver, respectively. Further the distance can be estimated by measuring the Time Difference Of Arrival (TDOA) between the pair of receivers.

Yao *et al.* in [231] propose a cooperative positioning (CP) method that fuses kinematics information as obtained from GPS or other kinematic sensors, with distance measurements calculated based on radio-ranging techniques such as Time Of Arrival (TOA) and Time Difference Of Arrival (TDOA). Moreover, they improve the accuracy of vehicles positions within each vehicle cluster by employing a routing algorithm presented in [39]. Similarly, Ou [171] presents a localization method for VANETs that relies on Road-Side Units (RSUs). In the proposed scheme, the author assumes that there is a pair of RSUs deployed on either side of the road, communicating with the passing vehicles constantly. Further studies on the effect of RSU deployment, beacon collision and failure are discussed, and enhancements are done to deal with such issues. In this proposed method, each vehicle communicates with each of RSUs through broadcasting beacon messages, and estimates its position by measuring the distance to each RSU using radio-ranging techniques such as TOA or TDOA.

Cooperative vehicle localization using car-to-car communication is addressed in [168] and [149] which are involved with a European project called Cooperative Vehicle Localization (CoVel). In these papers, the authors propose a comprehensive framework that incorporates a variety of localization methods in different classes, as discussed in [41], which are able to tackle different levels of available information. In their framework, the final position estimation is achieved by combining the output of Absolute Positioning, Relative Positioning, and Group Map Matching components. These components also use various sources of data/information obtained from a data access system called EDAS (EGNOS Data Access System), odometer, GNSS (Global Navigation Satellite System) receiver, car-to-car communication, and a digital map.

The idea of using distance measurements for improving the accuracy of location estimation based on GPS is introduced in [6]. In their proposed framework (VLOCI), the authors assume that all vehicles are equipped with GPS, and therefore have an estimation of their current location. Besides, the vehicles are deemed to move on a single lane, and do not communicate with other vehicles on different lanes. Continuously, and after measuring the distance to their neighbors, each vehicle collects estimated location of its neighbors (found by GPS) and computes its own location in their coordinate frames, using the measured distance. Finally, the improved estimated location is obtained by calculating a weighted sum over the obtained estimated locations, where the weights are proportional to the distance of neighbors. In [6], the two techniques used to measure the distance are based on time-of-arrival (TOA) and the received signal strength. The accuracy of these techniques can be modeled such that when the distance to be measured increases, the accuracy of the measurements taken also decreases. Therefore, measuring the distance to vehicles within close range is more accurate than measuring the distance to those further away. Further the vehicles are assumed to be traveled in one lane and in the same direction. In order to estimate the position, each vehicle receives messages from its neighbors containing their estimated positions. Also, each vehicle can measure the distance between itself and its neighbors. To tackle the variance in erroneous distance measurements, multiple measurements can be taken for which the average can be used as the final distance measurement.

Our contribution is relevant to [6], and improves the proposed framework in [6] by eliminating the restriction of all vehicles having GPS, and employing data fusion techniques such as EKF to estimate the location of vehicles by predicting it using predefined dynamic motion model, and refining it using sensor model and the information received from other vehicles. Besides, we use both TOA and AOA for measuring the location of the neighboring vehicles in the polar coordinate frame, which further helps to communicate with the vehicles on other lanes. Finally, we improve the location accuracy by computing a weighted sum over all of the estimated locations for each vehicle, where the weights are proportional to the belief of each vehicle regarding its current position, which can be easily estimated using the above mentioned data fusion filtering method.

3.2.3 State Estimation

In this section, we briefly describe the methods we have adopted for our new localization approach in VANETs.

Bayesian Filtering

The most commonly used algorithm for the state estimation is the Bayesian filtering approach [215], which calculates the probabilities of multiple beliefs to allow a robot (which is vehicle here) to infer its position and orientation. Bayesian filtering provides a general framework to estimate the state of the system (the current location of all vehicles in our case) based on probabilistic theory [181]. The general formulation of Bayesian filtering is as follows:

$$\overline{bel}(\mathbf{x_t}) = \int p(\mathbf{x_t}|\mathbf{u_t}, \mathbf{x_{t-1}}) bel(\mathbf{x_{t-1}}) dx_{t-1} \tag{3.1}$$

$$bel(\mathbf{x_t}) = \frac{p(z_t|x_t, z_{1:t-1}, u_{1:t})\overline{bel}(\mathbf{x_t})}{p(z_t|z_{1:t-1}, u_{1:t})} \tag{3.2}$$

where $\mathbf{x_t}$, $\mathbf{u_t}$, and $\mathbf{z_t}$ are the robot's state vector, input control vector, and sensor measurement at time t, respectively. Moreover, $p(\mathbf{x_t}|\mathbf{u_t}, \mathbf{x_{t-1}})$ and $p(\mathbf{z_t}|\mathbf{x_t})$ are "state transition" and "measurement" probability density functions.

Using Markov assumption and the fact that the denominator does not depend on x (as it will be the same for all values of x), we can express Equation 3.2 as follows:

$$bel(\mathbf{x_t}) = \eta p(\mathbf{z_t}|\mathbf{x_t})\overline{bel}(\mathbf{x_t}) \tag{3.3}$$

where η is a normalizing factor. Therefore, the belief of a vehicle about its current location as denoted by $bel(\mathbf{x_t})$ in Equation 3.3, is normalized between 0 and 1 with constant η.

Extended Kalman Filter

Extended Kalman filter is an extended version of Kalman filter, which is a recursive state estimator that constitutes the earliest tractable implementations of the Bayesian Filter. If the initial uncertainty is assumed to be Gaussian and the observation model and system dynamics are linear functions of the state, then Kalman filters are the optimal estimators. However, since most of the systems are not strictly linear we need to apply EKF, which linearize the system using first-order Taylor series expansions. Although the main advantage of Kalman filter is its computational efficiency, it can represent only unimodal distributions which limits them to local localization. Moreover, Kalman filters cannot satisfactorily solve ambiguities such as symmetries, and therefore, for instance, it is unable to track an object (e.g. robot/vehicle) in case of failure. Another shortcoming of Kalman filters is that, they cannot incorporate negative information. Finally, Kalman filters provide no sound solution to the data association problem and false data associations often lead to disastrous failures [62].

Unscented Transform

Unscented Transform (UT) is incorporated, for further refinement of the EKF estimation. In fact, EKF produces inaccurate estimations when the predict and update phases of the state estimation include highly non-linear functions. This is where using UT helps to overcome such deficiency. The UT is based on choosing a set of S consists of sigma points X_i and weights W_i so that the resulting estimate on the state $\mathbf{x}_t$ and covariance P_t become:

$$\hat{\mathbf{x}}_t = \sum_{i=0}^{2n} W_i^{mean} X_i \tag{3.4}$$

$$\hat{P}_t = \sum_{i=0}^{2n} W_i^{cov} [X_i - \hat{\mathbf{x}}_t][X_i - \hat{\mathbf{x}}_t]^T + Q_t \tag{3.5}$$

where n, T and $Q_\mathbf{t}$ represent the number of states, transpose and process noise, respectively and the weights of W_i^{mean} and W_i^{cov} are calculated accordingly:

$$W_0^{mean} = \frac{\lambda}{n+\lambda} \tag{3.6}$$

$$W_0^{cov} = \frac{\lambda}{n+\lambda} + (1 - \alpha^2 + \beta) \tag{3.7}$$

$$W_i^{mean} = W_i^{cov} = \frac{1}{2(n+\lambda)}, \ \ i = 1, 2, \cdots, 2n \tag{3.8}$$

where the constant α which determines the spread of the sigma points, is usually set to a small positive value (e.g. $10^{-4} \leq \alpha \leq 1$) with a default value of 10^{-3} and the parameter β

is used to incorporate the prior knowledge of the distribution with a default value of 2 [49]. The scaling parameter λ is given by $\lambda = \alpha^2(n + \kappa) - n$ with $\kappa = 3 - n$ [108]. Note that generation of sigma points $X_i(0 \leq i \leq 2n)$ require cholesky factorization of the covariance matrix P_t and can be calculated by:

$$\sigma_t \leftarrow 2n \text{ columns from } \pm c\sqrt{P_t + Q_t}$$

$$X_t(0) = \mathbf{x}_t$$

$$X_t(i) = \sigma_t(i) + \mathbf{x}_t$$

where Q_t matrix is related to the process noise covariance, the coefficient $c = \sqrt{n + \lambda}$ and σ_t represents $2n$ eigenvectors generated by Cholesky decomposition of the positive and negative square roots of the covariance matrix [88].

3.3 Cooperative Localization Using V2V Communication and Data Fusion

Our general VANET framework for information gathering and dissemination consists of Context-Aware Information Processing (CAIP) unit, Low-Level Data Fusion (LLDF) unit and Cognitive Gateway [80]. These units are deployed hierarchically, and interact with each other to provide necessary information for a service (*e.g.*, congestion detection). In this context, a service with an abstract definition, such as congestion detection, first enters into the framework through CAIP unit. Then, after determining different information attributes necessary to define a service (parameters), their values are computed using various sources of data, such as physical sensors in LLDF unit. Among various attributes, one of the very important attributes in VANETs is the location of the vehicles which is often measured using typical GPS sensors. Considering the fact that the required accuracy of location is service-based [41], it is necessary (in critical services such as safe driving) to get more accurate location estimation. We tackle this problem by incorporating data fusion methods along with V2V communication. In the following, we explain the LLDF unit followed by our new localization approach.

3.3.1 Localization Improvement Using V2V Communication

To represent the vehicle motion model, we have used the well-known *Ackermann* steering model [5, 91] coupled with a simple vehicle driving model that uses gas and brake pedal, and the steering wheel sensors to find the wheels angle and vehicle's velocity. In this model, the current normalized amount of gas and brake are calculated as

$$C(t) = \frac{\alpha_C(t)}{\alpha_{Cmax}} \qquad D(t) = \frac{\alpha_D(t)}{\alpha_{Dmax}} \tag{3.9}$$

where α_C and α_D are the current gas and brake pedal angles, respectively, α_{Cmax} and α_{Dmax} are their maximums. Considering that the wheel's radius is R_w and its maximum velocity is V_{max}, current velocity of vehicle is attained using the expressions below.

$$V(t) = \frac{(C(t) - D(t)) \times V_{max}}{R_w} \tag{3.10}$$

Therefore, the current input of the system can be defined by a vector $\mathbf{u}_t = [C(t)\ D(t)\ \phi(t)]$, where $C(t)$ and $D(t)$ are current gas and brake amount respectively, and $\phi(t)$ is the current steering wheel angle. Consequently, nonlinear state transition of the vehicle becomes

$$\mathbf{x}_t = f(\mathbf{u}_t, \mathbf{x}_{t-1}) = \begin{bmatrix} V(t)\cos(\beta(t-1))\Delta t + x(t-1) \\ V(t)\sin(\beta(t-1))\Delta t + y(t-1) \\ \frac{V(t)}{l}\tan(\phi(t))\Delta t + \beta(t-1) \end{bmatrix} \tag{3.11}$$

where $\mathbf{x}_t = [x(t)\ y(t)\ \beta(t)]'$ is the state of the vehicle at time t, Δt is the discretization size, and $\beta(t)$ is the vehicle's yaw (heading) angle with respect to the x-axis. With this dynamic motion model, we can then implement an EKF as follows:

$$\begin{aligned} \mathbf{x}_t &= f(\mathbf{u}_t, \mathbf{x}_{t-1}) + \varepsilon_t \\ \mathbf{z}_t &= h(\mathbf{x}_t) + \delta_t \end{aligned} \tag{3.12}$$

in which ε_t and δ_t are process and measurement Gaussian errors, respectively, and $h(\mathbf{x}_t)$ is the sensor model, and its output is the sensor measurement on state $\mathbf{x}_t$. Therefore, $\mathbf{z}_t$ is the sensor measurement that is perturbed with Gaussian error. Finally, the beliefs of each vehicle about their current state $\mathbf{x}_t$, which is their current location, is calculated using Equation 3.1. By calculating the corresponding beliefs, the vehicles estimate the location of their one-hop neighbors using TOA and AOA measurements and share this measurements with them. One-hop neighbors of a vehicle, are those neighboring vehicles which are accessible using one of the communication protocols such as DSRC (Dedicated Short Range Communications) [161]. Note that DSRC is a medium range wireless communication system currently developed and designed for automotive applications and the spectrum that can be used, in the range of 5GHz, is allocated by local government telecommunications authorities. The spectrum is partitioned into a number of 10 MHz channels following the

802.11p specifications. Also, its physical characteristics makes it suitable for the vehicle on road-side applications, securing reliable communications within the reasonable distances. The range and the data rate of this fully distributed standard protocol are $300m$ and 10-50 Mb/sec, respectively [201]. The detailed discussion about the implementation of the DSRC protocol for the V2V communication can be found in [186]. In Algorithm 1, $\mathbf{Q}_t$ is the

Algorithm 1 Extended Kalman Filter (EKF) Algorithm

```
procedure EKF(x_{t-1}, P_{t-1}, u_t, z_t)
    x̄_t ← f(u_t, x_{t-1})
    P̄_t ← G_t P_{t-1} G'_t + Q_t
    K_t ← P̄_t H'_t (H_t P̄_t H'_t + R_t)^{-1}
    x_t ← x̄_t + K_t (z_t − h(x̄_t))
    P_t ← (I − K_t H_t) P̄_t;

    return x_t, P_t
```

control error matrix, $\mathbf{R}_t$ denotes the measurement error matrix, and $\mathbf{K}_t$, $\mathbf{G}_t$, and $\mathbf{I}$ are the Kalman gain, Jacobian, and Identity matrices, respectively.

Let a vehicle and its neighbors set at time t are represented by i^{th} vehicle (v_i) and $\mathbf{N}_t(v_i)$. Furthermore, the distance and the angle between the two vehicles at time t using TOA and AOA are calculated as:

$$\begin{aligned} \hat{d}_t^{(ij)} &= d_t^{(ij)} + \mathcal{N}(0, g^2) \\ \hat{\gamma}_t^{(ij)} &= \gamma_t^{(ij)} + \mathcal{N}(0, \omega^2) \end{aligned} \tag{3.13}$$

where $d_t^{(ij)}$ and $\gamma_t^{(ij)}$ are error-free distance and angle between v_i and v_j at time t, which are then perturbed with Gaussian error with standard deviation g and ω, respectively. Let $v_j \in \mathbf{N}_t^{(i)}$, then its estimation about the location of v_i is expressed as:

$$\begin{aligned} x_t^{(ij)} &= \hat{d}_t^{(ij)} \cos(\hat{\gamma}_t^{(ij)}) \\ y_t^{(ij)} &= \hat{d}_t^{(ij)} \sin(\hat{\gamma}_t^{(ij)}) \end{aligned} \tag{3.14}$$

where $x_t^{(ij)}$ and $y_t^{(ij)}$ are the estimation of v_i's location in terms of the coordinate frame of v_j. Consequently, v_i improves its belief ρ about its location by calculating a weighted sum over estimated locations by its neighbors.

$$\mathbf{x}_t^{(i)} = \eta(\mathbf{x}_t^{(i)} \times \rho_t^{(i)}(\mathbf{x}_t^{(i)}) + \textstyle\sum_{j \in \mathbf{N}_t^{(i)}} \mathbf{x}_t^{(ij)} \rho_t^{(j)}(\mathbf{x}_t^{(j)})) \tag{3.15}$$

where $\mathbf{x}_t^{(i)}$ is the location of v_i at time t, and $\rho_t^{(i)}(\mathbf{x}_t^{(i)})$ is its own belief about its location, which is calculated using Equation 3.16, for μ and σ^2 as the outputs of the EKF model.

$$\rho_t^{(i)}(\mathbf{x}_t^{(i)}) = \mathcal{N}(\mathbf{x}_t^{(i)}; \mu, \sigma^2) \tag{3.16}$$

Furthermore, $\mathbf{x}_t^{(ij)}$ is the estimated location of v_i in the coordinate frame of v_j, and $\rho_t^{(j)}(\mathbf{x}_t^{(j)})$ is v_j's belief about its own location. Here, η is the normalization factor and is equal to:

$$\eta = \frac{1}{\rho_t^{(i)} + \sum_{j \in \mathbf{N}_t^{(i)}} \rho_t^{(j)}} \tag{3.17}$$

Note that the covariance of $[x_t^{(ij)} \quad y_t^{(ij)}]'$ is [183]:

$$\mathbf{P}_t{}^{(ij)} = \begin{bmatrix} P_{xx}^{(ij)} & P_{xy}^{(ij)} \\ P_{yx}^{(ij)} & P_{yy}^{(ij)} \end{bmatrix} \tag{3.18}$$

where

$$P_{xx}^{(ij)} = \hat{r}_t^{(ij)} g^2 \cos^2(\hat{\gamma}_t^{(ij)}) + \omega^2 \sin^2(\hat{\gamma}_t^{(ij)}) \tag{3.19a}$$

$$P_{yy}^{(ij)} = \hat{r}_t^{(ij)} g^2 \sin^2(\hat{\gamma}_t^{(ij)}) + \omega^2 \cos^2(\hat{\gamma}_t^{(ij)}) \tag{3.19b}$$

$$P_{xy}^{(ij)} = P_{yx}^{(ij)} = (g^2 - \hat{r}_t^{(ij)} \omega^2) \sin(\hat{\gamma}_t^{(ij)}) \cos(\hat{\gamma}_t^{(ij)}) \tag{3.19c}$$

in which $\hat{r}_t$ denotes $\hat{d}_t^2$.

3.4 Traffic Entity Assessment

The Low-Level Data Fusion (LLDF) framework, is located in the middle part of our proposed VANETs framework in [80]. This unit collaborates with the Context-Aware Information Processing (CAIP) unit, which is responsible for context-aware information dissemination after receiving list of the information attributes (parameters) to send out the estimated values. So, LLDF interacts with the low-level/physical sensors and the data pool from which it can select the relevant information, fuse them, and find the values for the required attributes by the CAIP unit. We use the proposed LLDF framework to implement a Traffic Entity Assessment (TEA) unit in our general Attention Assist Framework (see Chapter 6). As shown in Figure 3.2, the inputs of the TEA unit are selected according to the submitted service by the CAIP unit. In fact, the attributes of the information produced by the CAIP unit are the entities that the TEA unit needs to determine. Knowing

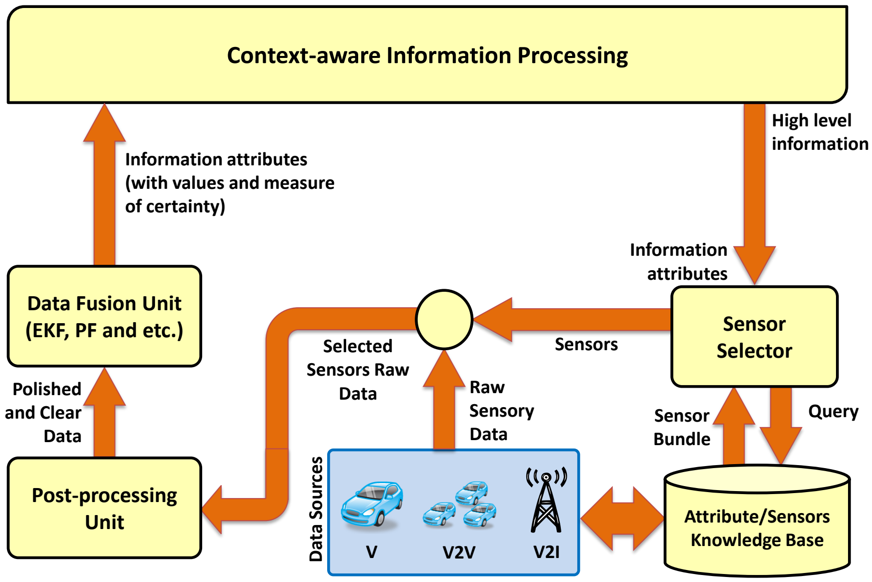


Figure 3.2: The block diagram of our proposed Traffic Entity Assessment unit incorporating Low-Level Data Fusion (LLDF) framework

the attributes, a specific data source (general definition of a sensor) or a bundle of data sources is selected from a set of data sources containing sensors on the ego vehicle, sensors on neighboring vehicles (V2V), and information provided by RSUs or infrastructures. Then the selected data is post-processed, and fed into the data fusion unit that employs various LLDF methods such as Extended Kalman Filter (EKF), to fuse the data and estimate a value for the given attributes. Finally, the completed information is sent back to the CAIP unit for dissemination.

3.5 Cooperative Localization Experiments

We realize the framework proposed in the last section by tackling the localization problem which is an important issue in many VANETs services such as safe driving and congestion detection. For the data source, we use the GPS data of each vehicle and collaborate with other vehicles to find the estimation of the ego vehicle's location using TOA and AOA radio-ranging techniques. EKF is used to predict the next location of the vehicle using its dynamic motion model, and update this prediction using GPS sensor measurements. The estimation of the ego vehicle location is further improved by taking a weighted average over the estimations of all the neighboring vehicles.

3.5.1 Simulations Setup

In our simulations, we define two test cases (Test Case 1 and Test Case 2), for which the results of the proposed approach is evaluated with and without V2V communication. In both cases, we evaluate the efficiency of the method by assuming that all vehicles are equipped with erroneous GPS sensors that may provide inaccurate location information due to blockage problem [41]. The GPS accuracies of all 5 vehicles considered here are varied at different time slots as illustrated in Table 3.1, where each column represents a Time Window (TW) for which the GPS standard deviation is set to a specific value. For example, the standard deviation (STD) of GPS for vehicle 1 is 17 from time slot 1 to 5, 18.9 from time slot 6 to 13, *etc.* as shown in Figure 3.3. In Table 3.1, although the length

Table 3.1: Time-varying GPS STD of all vehicles

V.I.	TW1	TW2	TW3	TW4	TW5	TW6
v_1	[1:5],17.0	[6:13],18.9	[14:19],12.9	[20:24],13.6	[25:28],9.3	[29:40],7.4
v_2	[1:9],7.1	[10:13],8.9	[14:20],7.6	[21:24],6.3	[25:29],6.2	[30:40],5.4
v_3	[1:10],4.1	[11:13],4.3	[14:19],5.7	[20:23],6.3	[24:34],5.2	[35:40],4.4
v_4	[1:9],3.0	[10:16],2.3	[17:20],1.7	[21:23],1.3	[24:33],2.2	[34:40],3.4
v_5	[1:8],1.1	[9:13],2.3	[14:19],3.7	[20:23],2.3	[24:31],1.2	[32:40],5.4

of the TW is different for each vehicle, the total number of time windows are the same which is equal to 6. Note that the motion model error of each vehicle and its initial location are set randomly at the beginning of iteration, where the *iteration* refers to simulation. Also, in the following simulation analysis, the motion model parameters of $C(t)$, $D(t)$ and $\phi(t)$ are varied in the same way for all vehicles in the scenario assuming that all gas, brake and wheel angles have the same value for all vehicles time slot by time slot for sake of simplicity. Moreover, all vehicles are assumed to have communication links between them and could calculate TOA and AOA accordingly. The neighborhood adjacency matrix at time t, $\mathbf{N}_t$, as defined in Equation 3.20, is considered to be static,

$$\mathbf{N}_t = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix} \qquad t = 1, \cdots, T \tag{3.20}$$

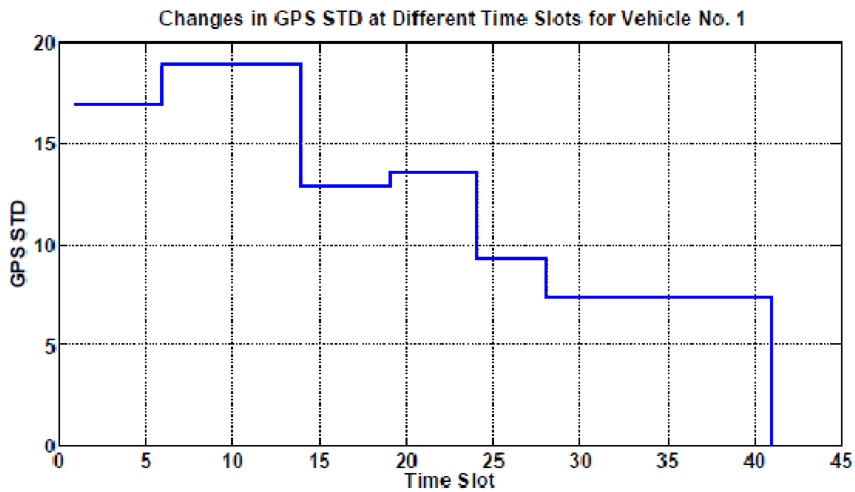


Figure 3.3: Time-varying GPS STD of vehicle 1

where the rows and columns of $\mathbf{N}_t$ matrix are indexed from 1 to 5 representing vehicle's number as 1 to 5, and T refers to total number of time slots. In practice, the duration of each time slot t can be few seconds before the vehicles can change their speeds and directions, whereas the time to compute TOA, DOA and EKF can be less than a second which results in real-time implications. The configuration of the vehicles at their initial states is displayed in Figure 3.4, where x and y-axes are in meters and there are 5 different lanes with lane-width of around $5m$ $(16ft)$.

Simulation results show how different approaches, namely, GPS, EKF, V2V without UT, and V2V with UT perform, given the test cases and scenarios introduced above. In addition, we have run the same test cases on the VLOCI model proposed in [6] to show the advantage of our proposed method among similar works in the literature. VLOCI assumes that all vehicles are in the same lane, thus, have equal y coordinates on an $x-y$ coordinate system. To model this framework, we simply remove the error-free angle measurement $\gamma_t^{(ij)}$ from Equation 3.13, and leave it with just the uncertainty about AOA. Therefore, Equation 3.13 becomes:

$$\begin{aligned} \hat{d}_t^{(ij)} &= d_t^{(ij)} + \mathcal{N}(0, g^2) \\ \hat{\gamma}_t^{(ij)} &= \mathcal{N}(0, \omega^2) \end{aligned} \tag{3.21}$$

where $\mathcal{N}(0, \omega^2)$ is normal distribution with a mean of 0 and standard deviation of ω radians.

The results of the following test cases are evaluated by comparing them with the ground truth and calculating the corresponding Mean Squared Error (MSE). The traversed path of the ego vehicle for each test case is also displayed.

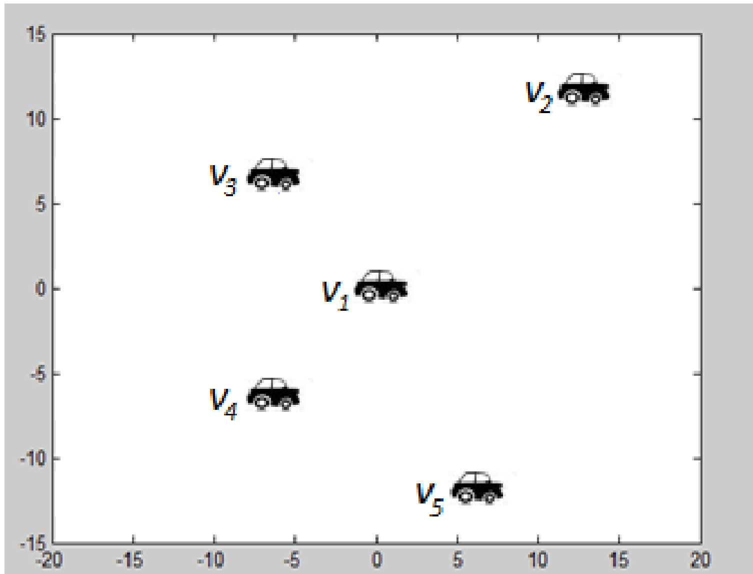


Figure 3.4: Initial configuration of the vehicles

3.5.2 Experimental Results: Straight Path Test Case

Here, we consider the vehicles are moving along a straight-line path with equal velocity. As mentioned earlier, we further consider that the GPS sensor of each vehicle provides data with varying accuracies at different time slots due to the blockage (see Figure 3.3), and the vehicles can communicate with their neighboring vehicles. For our data fusion method, we have used EKF to estimate the state of each vehicle, and run the simulation for 200 times. Motion model error of each vehicle and its initial location are set randomly at the beginning of each iteration, and the observation error is obtained at the specific time slots based on given GPS accuracy. From Figs. 3.3 and 3.5, it is obvious that using sensor data fusion and state estimation by EKF, the corresponding belief of each vehicle varies with the GPS accuracy. Figure 3.5 shows belief about the location of vehicle 1 with and without UT. After all the above experimental setup, we have used our proposed method based on V2V communication to improve the localization. The corresponding result of the estimated traversed path for the ego vehicle is presented in Figure 3.6, which shows the improvement of the location estimation using V2V communication (X and Y are expressed in meter(m) where each X-unit and Y-unit are $15m$ and $1m$, respectively). It is noted that we have the same X- and Y-axis representations for Figs. 3.7 and 3.8 as well.

Table 3.2 shows the results of mean squared error (MSE) calculated over 200 iterations.

For each iteration, the MSE is calculated over a certain period of time t_b and t_e, to eliminate the effect of random setting for initial locations. In our simulation, we set $t_b = 2$ and $t_e = T$, where 2 refers the second time slot and T is the last time slot. Based on the results in Table 3.2, the estimation of traversed path for each vehicle is improved using V2V communication (without and with UT) (in compare to the trivial EKF and GPS sensor only localization). We have also included the results after running an implementation of VLOCI [6] on the same scenario. Noticeably, since VLOCI does not consider lateral displacement between vehicles, its uncertainty about a neighboring vehicle on a direction with a particular angle increases which consequently, leads to very high MSE.

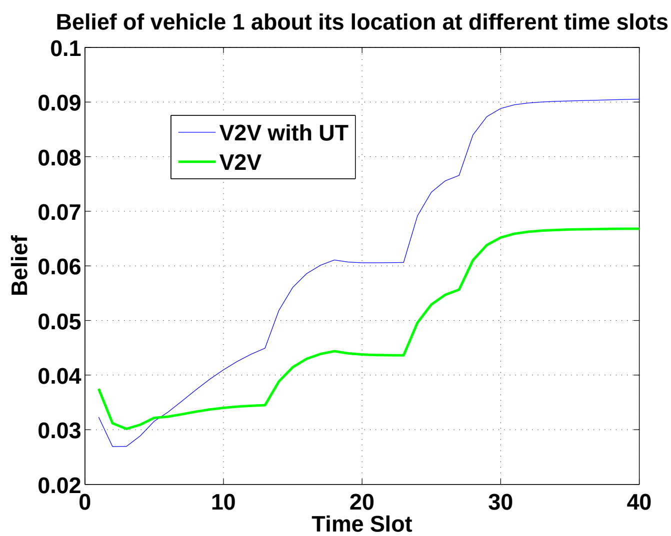


Figure 3.5: Belief of vehicle 1 at different time slots.

Table 3.2: MSE of different methods in Test Case 1

V.I.	GPS	EKF	VLOCI [6]	V2V (w UT)	V2V (w/o UT)
v_1	24.16	22.66	124.5	5.58	5.33
v_2	18.54	17.01	71.76	7.15	6.78
v_3	15.09	8.14	113.5	6.35	5.34
v_4	10.71	11.86	127.3	7.15	5.94
v_5	11.49	8.46	121.6	7.18	5.98

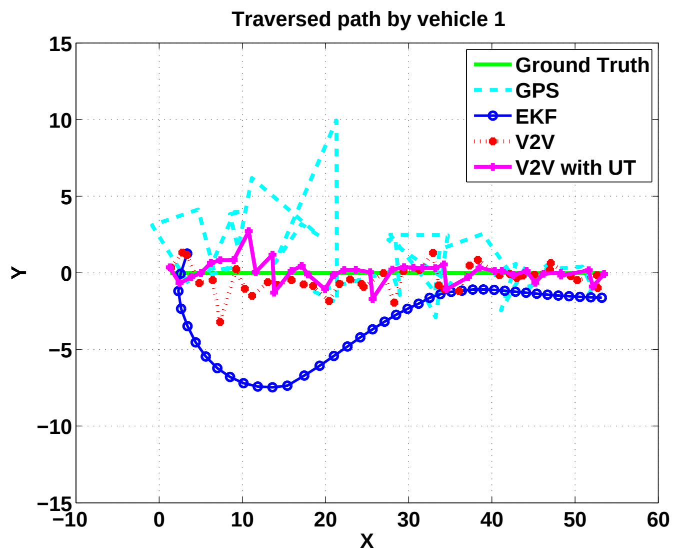


Figure 3.6: Estimation of traversed path by vehicle 1 using different methods.

3.5.3 Experimental Results: Curved Path Test Case

In this test case, the ego vehicle travels in a curved path. The corresponding estimation results using only GPS and using our EKF filter-based fusion technique as used to integrate GPS information with other sensors as well as combining the belief of other neighboring vehicles, are presented. There are two scenarios, i.e. *Scenario I and Scenario II*, for this set of experiments. In *Scenario I*, we consider that GPS is continuously turned on. In *Scenario II*, we consider that GPS fails for a certain period of time (e.g. between time slots 21 to 30 in our simulations) to show the effect of this failure and the result of its improvement using the belief integration of other vehicles. All the experiments are conducted for 200 iterations and the MSE values are calculated by averaging over all runs.

Discussions of the results for Scenario I

Figure 3.7 shows the path traveled by vehicle 1 in Scenario I. The path estimated by V2V (shown in red) is consistently much closer to the ground truth (shown in green) as displayed in Figure 3.7. This can be further illustrated in terms of calculated Mean Squared Error (MSE). As it can be seen in Table 3.3, the MSE obtained using V2V communication is much smaller than that of EKF and GPS alone.

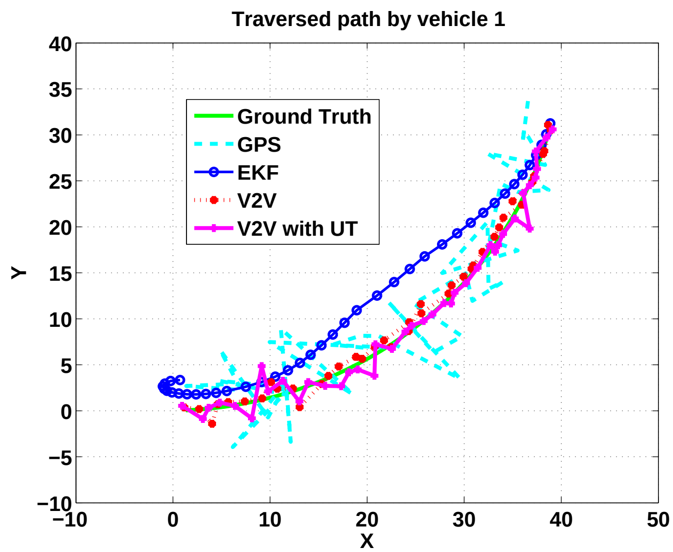


Figure 3.7: Traversed path by vehicle 1, and its estimations using different methods (with GPS is continually turned on)

Table 3.3: MSE of different methods in Test Case 2 (Scenario I)

V.I.	GPS	EKF	VLOCI [6]	V2V (w UT)	V2V (w/o UT)
v_1	23.96	22.23	73.69	5.64	5.56
v_2	18.51	17.02	61.52	7.96	6.99
v_3	14.97	8.09	88.99	6.42	5.47
v_4	10.64	11.65	90.24	7.31	5.66
v_5	11.64	9.27	75.9	7.00	5.85

Discussion of the results for Scenario II

In the second set of experiments, GPS data is lost for a certain period of time and the estimation is not accurate for that period (i.e. between the time slots 21 and 30 in our simulation). As shown in Figure 3.8, the corresponding path cannot be estimated accurately using only the GPS (shown in light blue); however, the path estimated using V2V and EKF are much closer to the ground truth. In Figure 3.8, the EKF results without using V2V communication is also depicted (shown in blue). The MSE of each method calculated for each vehicle is listed in Table 3.4. From the experimental results performed for both scenarios in test case 2, it can be inferred that the MSE achieved using V2V and EKF

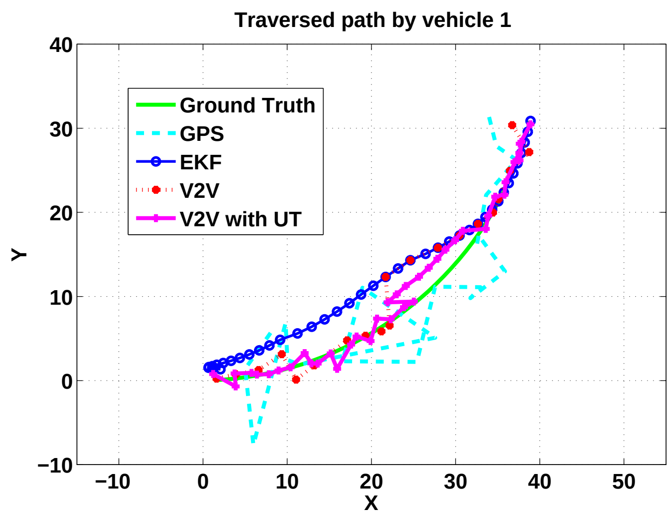


Figure 3.8: Traversed path by vehicle 1, and its estimations using different methods (with GPS is turned off during the time slots 21 to 30)

Table 3.4: MSE of different methods in Test Case 2 (Scenario II)

V.I.	GPS	EKF	VLOCI [6]	V2V (w UT)	V2V (w/o UT)
v_1	24.22	22.64	72.53	8.53	7.63
v_2	18.28	16.24	60.36	7.21	7.04
v_3	15.20	7.96	93.12	6.02	4.60
v_4	10.81	11.36	84.64	7.02	5.63
v_5	11.53	8.44	77.74	6.72	5.80

is much lower than using the other methods as shown in Tables 3.3 and 3.4. Moreover, similar to test case 1, VLOCI still performs weakly due to poor direction detection upon receiving a V2V signal. The proposed method thus provides an accurate estimation of the current state over time for the entire path traveled (see Figs. 3.7 and 3.8).

3.5.4 Performance Evaluation

In this section, we evaluate the performance of the proposed framework from various aspects, namely, number of vehicles, Gaussian *vs.* Non-Gaussian error, and impact of Sequential Decentralized EKF (SDEKF).

Various Number of Vehicles

We have shown here the performance of the proposed method by varying the number of vehicles, or traffic density. As shown in Table 3.5, the performance improves with the number of vehicles increases in normal situation when there is no outage of GPS (Test Case 1 and Scenario I) showing the significance of the proposed method. However, as expected the performance drops slightly with increasing number of vehicles (higher traffic) due to temporary GPS blockage in urban scenario (Scenario II) indicating the robustness of the presented method in urban environments. Here, the neighborhood adjacency matrices $\mathbf{N}_t$

Table 3.5: MSE of the proposed method (with UT) for Vehicle 1 with varying number of vehicles

Num of Vehicles	Test Case 1	Test Case 2 (Scenario I)	Test Case 2 (Scenario II)
5	5.33	5.56	7.63
7	4.57	4.44	8.29
9	3.81	3.82	8.49

for number of vehicles of 7 and 9, are defined in Equations 3.22 and 3.23.

$$\mathbf{N}_t = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \quad t = 1, \cdots, T \tag{3.22}$$

$$\mathbf{N}_t = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix} \quad t = 1, \cdots, T \tag{3.23}$$

Also, the lists of time-varying GPS STD for all 7 vehicles and 9 vehicles are depicted in Tables 3.6 and 3.7.

Table 3.6: Time-varying GPS STD of all 7 vehicles

V.I.	TW1	TW2	TW3	TW4	TW5	TW6
v_1	[1:5],17.0	[6:13],18.9	[14:19],12.9	[20:24],13.6	[25:28],9.3	[29:40],7.4
v_2	[1:9],7.1	[10:13],8.9	[14:20],7.6	[21:24],6.3	[25:29],6.2	[30:40],5.4
v_3	[1:10],4.1	[11:13],4.3	[14:19],5.7	[20:23],6.3	[24:34],5.2	[35:40],4.4
v_4	[1:9],3.0	[10:16],2.3	[17:20],1.7	[21:23],1.3	[24:33],2.2	[34:40],3.4
v_5	[1:8],1.1	[9:13],2.3	[14:19],3.7	[20:23],2.3	[24:31],1.2	[32:40],5.4
v_6	[1:5],2.3	[6:12],1.1	[13:18],3.7	[19:22],2.3	[23:31],1.2	[32:40],5.4
v_7	[1:9],5.4	[10:15],3.7	[16:23],1.1	[24:28],2.3	[29:32],2.3	[33:40],1.2

Table 3.7: Time-varying GPS STD of all 9 vehicles

V.I.	TW1	TW2	TW3	TW4	TW5	TW6
v_1	[1:5],17.0	[6:13],18.9	[14:19],12.9	[20:24],13.6	[25:28],9.3	[29:40],7.4
v_2	[1:9],7.1	[10:13],8.9	[14:20],7.6	[21:24],6.3	[25:29],6.2	[30:40],5.4
v_3	[1:10],4.1	[11:13],4.3	[14:19],5.7	[20:23],6.3	[24:34],5.2	[35:40],4.4
v_4	[1:9],3.0	[10:16],2.3	[17:20],1.7	[21:23],1.3	[24:33],2.2	[34:40],3.4
v_5	[1:8],1.1	[9:13],2.3	[14:19],3.7	[20:23],2.3	[24:31],1.2	[32:40],5.4
v_6	[1:5],2.3	[6:13],1.1	[14:19],3.7	[20:23],2.3	[24:31],1.2	[32:40],5.4
v_7	[1:9],5.4	[10:15],3.7	[16:23],1.1	[24:27],2.3	[28:31],2.3	[32:40],1.2
v_8	[1:5],2.3	[6:13],1.2	[14:22],5.4	[23:28],3.7	[29:36],1.1	[37:40],2.3
v_9	[1:4],2.3	[5:10],3.7	[11:19],5.4	[20:27],1.1	[28:31],2.3	[32:40],1.2

Correlated Gaussian and Non-Gaussian Error

In the above, we have considered uncorrelated Gaussian error. Here we show how the performance varies with respect to correlated Gaussian error, which is generated by the measurement error covariance matrix $\mathbf{R}$ defined as:

$$\mathbf{R} = \begin{bmatrix} \sigma_x & \rho\sigma_x \\ \rho\sigma_y & \sigma_y \end{bmatrix} \tag{3.24}$$

where ρ is the correlation factor varied as $\rho = 0, .1, \cdots, .9$ and σ_x, σ_y are the error variances along x- and y-coordinates, respectively.

For illustration, Table 3.8 lists the average localization accuracy (MSE) of vehicle 1 with respect to ρ by using GPS alone as well as EKF and V2V based methods. The illustrative results are presented for test case 2 and scenario I when the number of vehicles is five and

UT is considered. As we can see in Table 3.8, the results achieved by the proposed V2V based method seem less sensitive to the correlated error due to less variation of the MSE against ρ compared to the results obtained by GPS and EKF. In Table 3.9, the results of

Table 3.8: MSE of the proposed method (with UT) for Vehicle 1 for varying correlation factor ρ

ρ	GPS	EKF	V2V
0.1	24.45	23.08	5.52
0.3	25.51	25.17	5.61
0.5	26.99	25.96	5.72
0.7	28.70	26.47	5.67
0.9	29.78	26.40	5.90

performances for the above situation with non-Gaussian error are illustrated based on the following measurement error covariance [154]:

$$\mathbf{R} = \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix} \tag{3.25}$$

where $R_{11} = (1/\Omega)[(\nu/\varrho(2/\nu, 2/\nu)) - 1]\sigma_x^2$, $R_{22} = (1/\Omega)[(\nu/\varrho(2/\nu, 2/\nu)) - 1]\sigma_y^2$, $R_{12} = (1/\Omega)[(\nu/(2\varrho(2/\nu, 2/\nu)))(1 + \rho) - 1]\sigma_x^2$, $R_{21} = (1/\Omega)[(\nu/(2\varrho(2/\nu, 2/\nu)))(1 + \rho) - 1]\sigma_y^2$ with $\Omega = 50$, $\nu = 0.1$, and $\varrho(m, \ell) = \Gamma(m)\Gamma(\ell)/\Gamma(m + \ell)$ [154]. As we can see in Table 3.9,

Table 3.9: MSE of the proposed method (with UT) for Vehicle 1 for non-Gaussian error

ν	GPS	EKF	V2V
0.1	481.6	136.8	11.20
0.3	502.8	126.9	10.81
0.5	516.8	145.7	12.07
0.7	521.9	135.8	11.89
0.9	536.0	187.7	16.95

the degradation of performance for non-Gaussian error from that for Gaussian error, is minimum for the proposed method.

Sequential Decentralized EKF

Here, we have shown the improvement of our vehicular localization performance by considering Sequential Decentralized EKF (SDEKF) [224], which improves the location estimation of the i^{th} vehicle based on sequential filtering by its neighboring vehicles. Suppose a vehicle and the set of its neighboring vehicles at time t are represented by v_i and $\mathbf{N}_t(v_i)$ where the neighboring vehicles $v_j \in \mathbf{N}_t(v_i)$ and $\mathcal{S}$ is the size of $\mathbf{N}_t(v_i)$. Let the initial location estimation of the vehicle v_i at time t is given by

$$\hat{x}_i(t|t) = \hat{x}_i(t|t-1) + K_i(t)[z_i(t) - h_i(t)\hat{x}_i(t|t-1)] \tag{3.26}$$

where, Kalman gain

$$K_i(t) = P(t|t-1)h_i(t)[h_i(t)P(t|t-1)h_i'(t) + R_i(t)]^{-1}$$

, and error covariance

$$P_i(t|t) = [I - K_i(t)h_i(t)]P(t|t-1)$$

. Then the subsequent j^{th} estimates by the corresponding neighboring vehicles $v_j \in N_t(v_i)$ are as follows.

$$\hat{x}_j(t|t) = \begin{cases} \hat{x}_i(t|t) + K_j(t)[z_i(t) - h_i(t)\hat{x}_i(t|t)], \ j = 1 \\ \hat{x}_{j-1}(t|t) + K_j(t)[z_i(t) - h_j(t)\hat{x}_{j-1}(t|t)], \\ \hfill j \in [2, \mathcal{S}] \end{cases} \tag{3.27}$$

where

$$K_j(t) = P_{j-1}(t|t)h_j(t)[h_j(t)P_{j-1}(t|t)h_j^{'}(t) + R_j(t)]^{-1}$$

and

$$P_j(t|t) = [I - K_j(t)h_j(t)]P_{j-1}(t|t)$$

Finally, the improved location estimate and the error covariance of vehicle v_i at time t become:

$$\begin{aligned} \bar{\hat{x}}_i(t|t) &= \hat{x}_{\mathcal{S}}(t|t) \\ \bar{P}_i(t|t) &= P_{\mathcal{S}}(t|t) \end{aligned} \tag{3.28}$$

The simulation results for Test Case 2 (Scenario I) with Gaussian measurement error and the same experimental setup are presented in Table 3.10. Similarly, the simulation performance for more neighboring vehicles are presented in Table 3.11. Comparing Tables 3.10 and 3.11, it can be found that the larger the number of neighboring vehicles the better the performance of the SEKF scheme. As we see from the results in Tables 3.10-3.12, improved performance has been achieved by SEKF for Gaussian error (for both uncorrelated and correlated error cases). It can be mentioned that the results (not shown here) in case of non-Gaussian error, the SEKF provides much better performance over the EKF, however lower than the V2V based method. Moreover, the performance of the SEKF scheme is evaluated for varying the number of vehicles and shown improved results in all test cases (compare Tables 3.5 and 3.13).

Table 3.10: MSE of the SEKF scheme in Test Case 2 (Scenario I)

V.I.	GPS	SEKF (w UT)	SEKF (w/o UT)
v_1	23.86	6.53	4.93
v_2	18.53	7.65	5.52
v_3	15.14	1.47	1.05
v_4	10.78	2.29	1.79
v_5	11.51	1.27	0.83

Table 3.11: MSE of the SEKF scheme in Test Case 2 (Scenario I)

V.I.	GPS	SEKF (w UT)	SEKF (w/o UT)
v_1	24.36	3.70	3.06
v_2	18.45	4.52	3.53
v_3	15.17	0.75	0.56
v_4	10.79	1.43	0.95
v_5	11.60	0.55	0.30
v_6	11.53	0.41	0.36
v_7	11.15	2.86	2.02
v_8	11.40	1.07	0.77
v_9	11.49	1.86	1.49

Table 3.12: MSE of the SEKF filtering scheme (with UT) for Vehicle 1 for varying ρ in Test Case 2 (Scenario I) with correlated Gaussian error

ρ	GPS	SEKF
0.1	24.19	4.97
0.3	25.13	4.91
0.5	26.65	4.58
0.7	28.31	4.32
0.9	30.37	4.57

Computational Complexity

The computational requirement of the EKF is dominated by the need to store and update the filtering-error covariance matrix. At each time step t, the computational complexity of the EKF is $O(n^2)$, where n (n=3) represents the size of the state [40]. On the other hand, TOA-ranging calculation is based on MLE (Maximum Likelihood Estimation) algo-

Table 3.13: MSE of the SEKF filtering scheme (with UT) for Vehicle 1 with varying number of vehicles

Num of Vehicles	Test Case 1	Test Case 2 (Scenario I)	Test Case 2 (Scenario II)
5	4.80	4.90	5.49
7	3.43	3.47	3.75
9	2.63	2.72	3.23

rithm [192], where the linear complexity of the MLE algorithm is $O(L)$ with L represents the size of the signal which varies as the total number of nodes. The order of the complexity for the conventional MUSIC (Multiple Signal Classification) algorithm used for AOA estimation is $O(N^3)$ where N is the length of the received signal (which is N=2) [158].

The computational time (in elapsed CPU seconds) for the proposed scheme including the execution time of EKF, TOA, DOA are obtained using the PC with Pentium B940 Processor (2.0 GHz) and MATLAB implementation, which are listed as follows: EKF(0.138 sec), TOA (0.029 sec), AOA (0.020 sec). Moreover, the time to change the speed and the direction of a vehicle is compatible with the EKF computation ($<$ 1 sec). Moreover, the computational costs (i.e. CPU time) are 1.310 sec, 1.388 sec, 1.513 sec when the number of vehicles are 5, 7, and 9, respectively. The computational time of the proposed approach in highway scenario (e.g. Test Case 1) is 1.185 sec, where urban scenario (e.g. Test case 2), it is 1.388 sec. Also, the computational time of the proposed method due to UT can be increased by 1%.

3.6 Summary

This chapter presented a LLDF framework for the Internet of Cars, called Traffic Entity Assessment unit, along with a new cooperative approach dealing with the localization problem in VANETs is proposed. In our approach, we combined data fusion and radio-ranging distance measurement techniques along with V2V communication in order to improve the location information of the vehicles. We further extend our ideas of cooperative approach by considering sequential EKF filtering within the neighboring vehicles to further improve the performance. We evaluated the methods by comparing the estimated locations of the vehicles with their ground truth, and demonstrated that using V2V communication for measuring the distance and sharing belief about the estimation of the current location, the neighboring vehicles can cooperatively improve the knowledge of the current location. Moreover, we have tested the robustness of the methods by taking out the GPS sensor for a period of time to show how the ego vehicle can maintain its belief about its current location by communicating with other vehicles. The performances of the methods are also evaluated by using different types of noise as well as varying the number of vehicles. Comparing to another cooperative method of localization based on distance measurements for improving the GPS information, such as reported in [6], the proposed method eliminates the restriction of all vehicles having GPS, and employs data fusion techniques such as EKF to estimate the location of the vehicles by predicting it using predefined dynamic motion model, and refining it using GPS sensor and the information received from other vehicles.

Chapter 4

Situation Assessment Using a Fuzzy Extension to MEBN

This chapter presents a novel comprehensive Fuzzy extension to Multi-Entity Bayesian Networks (MEBN) model, which is deemed a well-studied and theoretically rich language that expressively handles semantics analysis, and effectively model uncertainty management. MEBN lack the capability of modeling the inherent conceptual and structural ambiguity that is delivered with the knowledge gained through human language. In fact, Fuzzy-MEBN is a new version of MEBN that is based on First-order Fuzzy Logic (FOFL), and Fuzzy Bayesian Networks (FBN), and aims to overcome MEBN issues using Fuzzy theory.

The applicability of the proposed model in the Internet of Cars domain is examined in two aspects. Firstly, a traffic situation assessment unit is implemented, in which the entities, situations, and their relationships in specific contexts are modeled using Fuzzy-MEBN fragments. Furthermore, Fuzzy-MEBN inference is used to assess the situations of interest by estimating their states. To demonstrate the capabilities of the proposed framework, a collision warning system simulator has been developed, which evaluates the likelihood of a vehicle being in a near-collision situation using a wide variety of local and global information sources available in the IoC environment. If the threat of being in a near-collision situation is determined to be high, then the driver is warned accordingly.

As the second aspect of our evaluation, a Soft-Hard Data Fusion (SHDF) model in the IoC is proposed that is capable of combining the data generated from human-based sources with those generated by physical sensors. In this model, the unstructured soft data is presented by undergoing a novel soft data matching process, through which the data is semantically analyzed, and accurately structured in a fuzzy random variable. Moreover, the clique tree inference algorithm for Bayesian Networks is modified to handle fuzzy evidence in Fuzzy-MEBN.

Our experimental results for two distinct single-vehicle and multi-vehicles categories of driving scenarios, as well as a novel hybrid Fuzzy-MEBN inference, show the capability of the proposed framework to efficiently achieve situation and threat assessment on the road, while handling both soft and hard data. Besides, the results demonstrate that Fuzzy MEBN is able to efficiently deal with ambiguous semantic and uncertain causal relationships between the knowledge entities

4.1 Introduction

Situation Awareness (SAW) is the main result of High-Level Information Fusion (HLIF), which is followed by knowledge insight extraction. This is made more achievable nowadays as connectivity and mobility have been improved. However, storing, processing, and handling the vast amounts of data and information that come from various sources with different levels of abstraction have become challenging issues. Moreover, these data and information can be any kind of hard data that are generated from physical sensors, or different types of unstructured soft data, which are embedded in a higher level (with respect to hard data), and are generally generated by humans. The human-generated data are usually based on the inherently ambiguous and unconstrained natural language. In other words, the human-based data is deemed *qualitative*, and open to interpretation. This is opposite to hard data which is *quantitative* with fixed interpretation.

Three main sources of human-generated information are: the reporter, the communication links, and the Internet [92]. The reporter is usually deemed a person who contributes to an information fusion model by providing his/her observations about a specific fact. For example, in the IoC context, such a person can be the driver, or a passenger of a certain vehicle, who is reporting his/her observations about a certain traffic status. The generated data are commonly captured through a Human Computer Interaction (HCI) unit first, and then following some pre-processing stages, such as Natural Language Processing (NLP) are made ready for the fusion task. Moreover. the text-based soft data on the Internet is in a different form than those provided by a human source. Two main categories of Internet data are raw text, and semantically interpreted text with the help of meta-data. The later category can be perfectly related to the research on semantic web [28]. Knowledge representation in semantic web is done by using a meta-data-enabled framework such as Resource Description Framework (RDF) [118] that makes it machine-readable.

Clearly, such a diversity increases the need for attaining reasonable automated knowledge discovery methodologies that are able to conveniently handle most of these issues. Multi-Entity Bayesian Networks (MEBN) model, introduced by Laskey [125] is a well-defined and theoretically rich language for HLIF that tackles uncertainty management and semantics analysis simultaneously. A MEBN model is a combination of First-order Logic (FOL) and Bayesian Networks (BN), and is considered as a powerful tool for modeling knowledge for situation assessment. However, despite being a strong bridge connecting structured knowledge (that is often expressed by domain experts) to computational models, MEBN lack the capability of modeling some imperfect aspects of data such as ambiguity, which is an inherent characteristic of human language, and the observations gained from the vague environment. For instance, when referring to an entity in an environment, various sources may use different identifiers that although all can be semantically positioned in one category, they may not be completely the same as identifiers used for defining the semantic relationships. This is basically referred to as semantic similarity in the literature [8], as widely happens specially when dealing with soft data.

To overcome these issues, a novel Fuzzy extension to MEBN is proposed, which enables them to efficiently representing ambiguous semantics relations, and to smoothly recognize soft data. Accordingly, we first redefine the semantics specifications of conventional MEBN by incorporating notions of First-order Fuzzy Logic that is mainly inspired by works of [167]. As a result, contextual constraints of MEBN are generalized in a way to represent the ambiguity that is usually delivered with the imperfect semantic information. Furthermore, a new way of representing Fuzzy Bayesian Networks (FBN) is also presented, and the well-known Junction Tree (JT) inference algorithm in regular BN is updated to include fuzzy states with a certain likelihood.

4.2 Background and Related Work

In this section, an overview of the literature work related to HLIF and SHDF is presented, and theoretical background of MEBN is thoroughly introduced. It begins with exploring the role of HLIF methodologies and situation assessment in realizing safe and efficient transportation systems, and proceeds with highlighting the prominent research work in Soft-Hard Data Fusion (SHDF) area. Moreover, the subjects of ontology-based information fusion and relevant semantic technologies are briefly discussed.

4.2.1 High-Level Information Fusion

Major topics of current research in HLIF are presented by Blasch *et al.* in [34]. In their survey paper, the authors extract the top ten trends of HLIF from the conference papers and panel discussions published within years 2000 to 2011, and categorize them into five main groups of Data and knowledge representation, Situation, threat, and impact assessment, Systems design, Evaluation, and Information management. Furthermore, Uncertainty Analysis and Semantics and Ontologies trends are featured as the most important areas of study that come before other crucial trends such as Reference Model Definition, Social Behavioral Model, and Resource Planning. Therefore, the literature review in this section is narrowed to the analysis of some well-defined HLIF frameworks, and to the substantial research work addressing uncertainty management, and semantics and ontology representation in HLIF.

General HLIF Frameworks

A comprehensive schema with seven building blocks for designing an HLIF system is introduced by D.A. Lambert in [124]. In this schema, the first building block discloses a three level assessment paradigm, and presents a de-constructed JDL model [226] in which the 4th level of its revised version [208] is implicitly embedded as an assessment unit into levels 1 to 3. The resulting model consists of three major levels aiming to render object, recognize relationships, and assess the impact of those relationships, respectively. In the second building block, the fundamental steps in adding the machine-readability capability to situation and impact assessment is discussed and major steps through which syntactic tokens acquire meanings using a sequential schema, are presented. Moreover, the third building block aims to represent the human mental status in machines by defining beliefs, expectations, and anticipations, which are generally called *awareness*. The State Transition Data Fusion (STDF) model, is the content of the fourth building block that goes through the DF processes from the lower level of object assessment to higher levels of situation and impact assessment. Fifth building block demonstrates the applicability of social behavior models, wherein the distributed methods such as Distributed Data Fusion (DDF) techniques, and ubiquitous fusion are among the main topics. In DDF, each level of JDL model will perform its own task in a distributed approach, and then the local information is merged to form global information about the current state. Furthermore, ubiquitous fusion allows the objective of a particular agent to enter a society (*i.e.*, VANET environment), for further examination upon acquiring the acceptance of all other agents, or at least a significant portion of them. Therefore, the agents participate in a society, and increase the robustness of their DF paradigm. Finally, sixth and seventh building blocks evaluate different methods in demonstrating the results to a user-level endpoint, and in studying the human condition in different aspects of this demonstration, respectively.

In order to have a generic HLIF model, active role of human, and a bi-directional interaction between human and technology should be taken into account. Nilsson *et al.* in [166] propose this idea by first arguing the limitations of traditional fusion models, specifically JDL model [208] and OODA loop [135], in incorporating human decision making models, and using humans inherent cognitive capability to enrich the fusion processes. Therefore, Nilsson *et al.* propose their human-technology driven environment in which users are the active parts of the fusion process and while receive useful knowledge from the provided technology through artefacts, help to improve the awareness of the current situation by utilizing their decision making capability, and removing technologically related flaws such as untrustworthy results. The first steps towards a human-technology interactive fusion environment is taken by employing the distributed cognition concept that is originally introduced by Hutchins in [104]. Distributed cognition studies the understanding of the dynamic flow of information (*i.e.*, a process) through a systematic organization of components (*i.e.*, either humans or technological artefacts). Technically speaking, these processes are deemed distributed in three ways: *across* the components, *between* internal and external aspects of components, and *over* time.

Semantics and Ontology Representation

Based on the concept of Situation Theory [21, 20, 19], Kokar et al. [119] define a situation awareness framework based on an ontology described using the OWL, which they refer to as the Situation Theory Ontology (STO). The STO creates different ontology-based concepts of the situation theory by defining OWL classes and connections for objects, types, and their relationships. Furthermore, a situation can be easily represented using a set of classes related to that situation, with appropriate relationship definitions among them. Lastly, new situations are inferred by creating a knowledge base containing horn clauses. Although it is a well-organized framework for situation semantics representation, STO lacks uncertainty management, which is deemed an important aspect of a HLIF model.

Mapping ontologies is another important application of using ontologies in HLIF systems which is also discussed by Wache *et al.* in [220] in two cases. In the first case, an ontology can be mapped to an information source, with the major steps of Structure Resemblance, Term Definition, Structure Enrichment, and Meta Annotation. In the second case, two different ontologies can be mapped to each other, *i.e.*, inter-ontology mapping, through employing pre-defined mappings, lexical and semantic relations correspondence, and top-level grounding. At the end, Wache *et al.* propose three steps for creating an ontology and having an engineering vision on its development process. The first step identifies the underlying scope being processed. Moreover, building the ontology by capturing and coding it, and integrating different ontologies, are the contents of the second step, and finally, the third step evaluates the constructed ontology accordingly.

Abstract semantics and ontology representation is discussed by Little and Rogova in [137] who address the problem of defining formal structures for different entities, their attributes and properties, and relationships between objects, *etc.*. This is done by using hybrid approaches and formal ontologies. In their proposed method, Little and Rogova aim to make the in-between connections between a higher-level ontology with more abstract structure, and a pre-determined domain-specific ontology.

Heintz and Dragisic in [96] tackle semantics representation by using the idea of annotating sensors based on their semantic structure, and reasoning by semantic information integration. Therefore, a source is generating relevant data for a service if their pre-defined ontologies match. Furthermore, they propose an application independent framework which can be customized to find a set of information sources satisfying the given demand for a specific service.

In fact, lacking uncertainty management capability is a common drawback in most of the frameworks that only care for representing the ontological and semantic relationships between the entities existing in a specific environment. Therefore, it is necessary to develop situation awareness frameworks which are also capable of modeling uncertainty.

Uncertainty Management

The fundamentals of a dependable and generic HLIF method for handling uncertainty is proposed by Karlsson in [112]. In his technical report, Karlsson categorizes the methods dealing with uncertainty, so-called Uncertainty Management Methods (UMMs), into three groups of Bayesian, Dempster-Shafer, and Imprecise probability approaches. This technical report raises some interesting open questions such as the possibility of fusing temporal attributes, *i.e.*, information in present, past and future, as well as the definition of evaluation metrics for measuring the performance of different HLIF systems.

Costa *et al.* in [55] propose the Uncertainty Representation and Reasoning Evaluation Framework (URREF) to improve the system-level metrics such as timeliness, accuracy, and confidence. In other words, their main goal is to study the effect of uncertainty on IF systems. Therefore, they present an abstract model in which different uncertainty handling tools such as probabilistic methods, Dempster-Shafer theory, and Fuzzy Sets, can be used in a plug-and-play fashion. Furthermore, they define an ontology for the proposed framework to make sure that all the evaluations are semantically sound. Different types of entities, and relationships between different objects in the domain of uncertainty handling are determined through this ontology.

An Object-oriented Probabilistic Relational Model (OPRM) is introduced by Howard and Stumptner in [101]. This model, which is a new language for First-order Probabilistic Logic (FOPL), aims to handle situation assessment by formalizing object and relationship recognition, IF at different abstract levels, and handling uncertainty and temporal nature. The main structure of OPRM consists of a set of classes featured with a set of descriptive and reference attributes. OPRM is also equipped with a probabilistic component which defines probability distributions on attributes to model uncertainty on them. One of the most strong capabilities of OPRM, is its uncertainty handling power which is imposed on existence, attribute, and structural uncertainties.

Uncertainty handling is tackled in a situation awareness framework for transportation, proposed by Röeckl in [185], which uses dynamic probabilistic causal decision networks to help making decisions with maximum utility. Both forward and backward propagation are studied to demonstrate the capability of the framework in making decisions based on the observed evidence, and in optimizing evidence selection for a given decision. However, this method cannot be used in a wide range of applications due to the lack of semantics representation capability. Besides, when the complexity of the environment increases, it is unlikely to demonstrate efficient inference time, which is inevitably inherited from BN.

4.2.2 Soft-Hard Data Fusion

SHDF is a new research trend in the area of sensor/data fusion that has attracted attention of researchers in the field [116]. Since its early ages, there has been many attempts to construct a general framework based on a well-defined mathematical foundation, and to establish test-cases and evaluation metrics [138]. It was highlighted by Llinas and Nagi [138] that the first major steps towards making such a comprehensive framework include source categorization, soft and hard data alignment and association, state estimation, and test-case development.

A SHDF model based on Dempster-Shafer (DS) theory is presented in [178]. In this model, called DS-TEC, DS theory is employed to develop an evidence updating method, and to handle both probabilistic and possibilistic aspects of SHDF. In a case study presented in the paper, soft evidence is considered the relative databases that contains informative sentences. These sentences are first analyzed through Natural Language Processing

(NLP) techniques, and are then extracted into logical machine-readable form.

Using ontologies for knowledge representation is another important method for modelling a system capable of fusing both soft and hard data. In fact, ontologies help to give structure to soft data, and construct the machine-readable knowledge. As an example, Gómez-Romero *et al.* [89] propose an context-aware information fusion system based on ontologies. Their model consists of two processing levels. In the first level, they use logical reasoning for object recognition (level 1 data fusion). Moreover, they perform situation and threat assessment based on Belief-Argumentation System (BAS), and Transferable Belief Model (TBM).

The theoretically rich foundation of Random Set (RS) theory is another research direction aimed at reaching to reach a well-defined SHDF. In fact, RS is capable of handling both uncertain and ambiguous data, and is deemed a powerful tool for SHDF. For instance, Khaleghi and Karray [115] propose an SHDF framework based on Random RS and a domain-specific ontology for a target tracking application. Moreover, they propose a model that measures the trustworthiness of human agents, and helps to avoid the misleading and adversary observations.

As it was mentioned before, soft data is inherently unstructured and vague. Therefore, importing such data to a pre-deployed information fusion model will cause the model to return imprecise assessments. One common method that aims at associating two different entities, and at determining how much they are semantically related, is the semantic similarity measurement. AlemZadeh [8] discusses various semantic similarity metrics in his Ph.D. dissertation, and utilizes them to analyze Wikipedia's linked data graph. Analogous techniques can also be seen in research work of [142, 164].

What makes our approach different from those in the SHDF literature is that here, we take advantage of the ontological structure of MEBN along with the semantic analysis of soft data to efficiently associate it with the most relevant entity. In most of the other approaches, such as [198], it is naively assumed that a set of pre-defined words are chosen by a reporter to produce soft data.

4.2.3 Multi-Entity Bayesian Networks

Multi-Entity Bayesian Networks aim to improve the conventional Bayesian Networks (BN) by incorporating means of introducing semantic relations among entities, and also further expanding the classic first-order logic using the uncertainty handling power of the probability theory [125]. The foundations of MEBN are the MEBN Theories (MTheories), which are powerful tools for modelling domain-specific knowledge for situation assessment.

The MEBN theories are composed of several fragments defined on a set of related random variables representing a certain entity, called the MEBN Fragments (MFrags). This constitutes the main characteristic of the MEBN language which is its modularity, *i.e.*, MFrags can be readily added/removed to/from the modeled system, without losing the structural consistency. The global probability distributions are defined over a set of small groups of hypotheses with specific local probability distributions. In a broader view, the joint probability distribution over truth-values of sets of semantically connected entities is the output of an MTheory. The following provides more backgrounds on MEBN, as discussed in [125].

MEBN Fragments

An MFrag $\mathcal{F}$ is defined by a tuple $\mathcal{F} = (\mathcal{C}, \mathcal{I}, \mathcal{R}, \mathcal{G}, \mathcal{D})$, in which it hosts three different types of nodes, namely context nodes $\mathcal{C}$ for semantics representation, input nodes $\mathcal{I}$ for MFrags inter-connection, and resident nodes $\mathcal{R}$ as random variables. The context nodes represent the semantic relations among the entities constituting a fragment of the domain knowledge, and specialize the general definition of the related FOL sentences in an MFrag. Furthermore, the input nodes act as ports acquiring external links from the entities lying in other MFrags, and finally, the resident nodes represent the random variables in BN, and are conditioned on the values provided by the context, and the input nodes, or other resident nodes. In an MFrag $\mathcal{F}$, $\mathcal{G}$ represents an MFrag graph, and $\mathcal{D}$ is a set of local distributions. Moreover, the sets $\mathcal{C}$, $\mathcal{R}$, and $\mathcal{I}$ are pairwise disjoint, and $\mathcal{G}$ is a Directed Acyclic Graph (DAG), whose nodes belong to $\mathcal{I} \cup \mathcal{R}$, with root nodes being members of the set $\mathcal{I}$ only.

MEBN Theories

A collection of MFrags which satisfy the consistency constraints that lead to the existence of a unique joint probability distribution over all the random variables in the collection is called the MTheory. The MTheory consists of two types of MFrags, namely, the built-in MFrags that represent basic logical content, and the domain-specific MFrags that are determined by an external knowledge. Observations are included in MTheories in forms of the findings that encompass certain facts about the current state of the environment. Therefore, any query on an MTheory is processed by first including the observed evidence (findings) into the MFrags to represent task-specific information, and then performing the Bayesian inference to compute the response, and to refine the local distributions. It is proved by Laskey in [125] that for an MTheory $\mathcal{T} = \{\mathcal{F}_1, \mathcal{F}_2, ..., \mathcal{F}_n\}$, a joint unique probability distribution $\mathcal{P}_{\mathcal{T}}{}^{gen}$ exists on the set of random variable instances of its MFrags that are consistent with the local distribution assigned by the MFrags in $\mathcal{T}$.

MEBN Inference

The MEBN utilizes Situation-Specific Bayesian Networks (SSBN) as its inference engine. The algorithm is based on works of [143] in which a minimal SSBN is approximated using the combination of particular types of random variables extracted from a given query. A detailed discussion of this algorithm is provided in [125], and is not included here for sake of brevity. Once the minimal SSBN is constructed, a standard Bayesian network inference algorithm can be deployed to compute the marginal distribution of the target random variables (entities of interest) given the evidence data.

Hybrid MEBN Inference

The result of the MEBN inference is a minimal SSBN that is formed based on the given query, current evidences, and the semantic relations present in the knowledge base. Among all these factors that are directly bonded to the structure of the SSBN, only the current evidences may vary over time. Based on this observation, we propose a hybrid version of MEBN inference algorithm that recreates the SSBN structure only if the there is a lateral shift in situation evolution (see our definition of situation evolution in [87]). As a result, in hybrid version of MEBN inference, SSBN will be created whenever a new evidence has arrived or removed, to help with the reduction of the required time for MEBN inference.

4.3 Fuzzy Multi-Entity Bayesian Networks

In this section, the MEBN are improved by replacing the FOL representation with First-order Fuzzy Logic (FOFL) [167] when defining contextual and semantic constraints. Therefore, the original MEBN language is updated, and accordingly, some of its definitions are refined.

4.3.1 Fuzzy Entities and Random Variables

A specific domain in MEBN language is modeled using a predefined set of attributed entities, and determining semantic and causal relationships between them. These entities are identified by constants that are included in an infinite collection of domain-specific constants (with meanings fixed by the language), and are referred to by variables that are included in an infinite collection of variable symbols. Besides, features of entities and the relationships between them are modeled using random variables that are drawn from an infinite collection of both logical and domain specific random variables. Following shows how the entities and random variables are refined in Fuzzy MEBN language.

- *Ordinary variable symbols:* The ordinary[1] variables are deemed containers that refer to non-specific entities. Ordinary variables names are alphanumeric strings that begin with a lower case letter, *e.g.*, $veh13$, env.
- *Phenomenal constant symbols:* Constants are represented by fuzzy sets with just a single member, *i.e.*, fuzzy singletons. Constant names may contain both letters and number, but must start with an uppercase letter, and should be followed by a real-valued membership degree subscript within range $[0, 1]$ *e.g.*, $Vehicle_{0.85}$, $Environment_{1.0}$.
- *Unique identifier symbols:* The entities are assigned a unique identifier symbol that are annotated with a fuzzy membership degree, and are arranged in one of the groups below:
  - *Truth value symbols and undefined symbol:* Truth values can either be a real number within range $[0, 1]$, or a member of a finite chain of truth values $L =< l_1, l_2, ..., l_n >$ predefined by the language.
  - *Entity identifier symbols:* Shown by $\mathcal{E}$, the set of entity identifier symbols are used by an interpretation of the theory to label the specific entities. Entity identifier symbols can be either numbers of alphanumeric symbols starting with an exclamation point and are subscripted with a real-valued membership degree ranging from 0 to 1, *e.g.*, $!V428_{0.75}$.
- *Logical connectives (random variables):* All the logical connective symbols, $\neg$,$\vee$,$\wedge$,$\Rightarrow$, $\Leftrightarrow$, and $=$ are deemed reserved logical random variables whose fuzzy interpretations are predefined by the language. Therefore, expressions such as $(\psi \vee \phi)$ will be interpreted by the fuzzy interpretation $\mathcal{D}$ as: $\mathcal{D}(\psi \vee \phi) =_I \mathcal{D}(\psi) \vee \mathcal{D}(\phi)$, in which $=_I$ is read *"is interpreted as"*, and operators such as $\vee$ or $\wedge$ can be substituted with the corresponding fuzzy logic s-norm or t-norm operators, respectively. Finally, logical

[1] As mentioned by Laskey in [125], the adjective "ordinary" is used to differentiate between ordinary variables and random variables.

connectives look more like random variables with truth-valued outputs, if written in prefix notation. For example, $\Rightarrow (\psi, \phi)$ is an implication random variable with two random variables ψ and ϕ.

- *Quantifiers:* Universal $\forall$ and existential $\exists$ quantifiers are interpreted by the fuzzy interpretation $\mathcal{D}$ predefined by the language as $\mathcal{D}(\forall_x \varphi) = \inf_\Delta \mathcal{D}(\varphi_x(\Delta))$ and $\mathcal{D}(\exists_x \varphi) = \sup_\Delta \mathcal{D}(\varphi_x(\Delta))$, respectively, wherein $\Delta = \langle \varepsilon_1^{(\alpha_1)}, \varepsilon_2^{(\alpha_2)}, \cdots, \varepsilon_n^{(\alpha_n)} \rangle$ is a vector of unique entity identifier symbols (*s.t.* $\varepsilon_i^{(\alpha_i)} \in \mathcal{E}$) with a length equal to the number of arguments that the logical (or domain specific) random variable (see below) φ takes, and x is an exemplar symbol.

- *Findings:* observed evidence is called finding in MEBN, and are stored in the set Ω. Logical findings are assigned a truth value within the range $[0, 1]$ or from the finite chain of truth values $L = \langle l_1, l_2, ..., l_n \rangle$.

- *Domain-specific random variable symbols:* random variable names in Fuzzy MEBN are alphanumeric strings beginning with a capital letter. Each random variable is assigned a positive integer that corresponds to the number of argument it takes. Moreover, random variables can have a set of finite or infinite possible values. Accordingly, possible values of *logical* random variables can be either within the continuous range of $[0, 1]$, or the finite chain of truth values $L = \langle l_1, l_2, ..., l_n \rangle$ predefined by the language. Furthermore, possible values of *phenomenal* random variables are defined as a subset of $\mathcal{E} \cup \{\perp\}$. In addition, the degree of membership of phenomenal random variables are predefined by their fuzzy interpretation, so that a phenomenal random variable R maps a vector of unique entity identifier symbols $\Delta = \langle \varepsilon_1^{(\alpha_1)}, \varepsilon_2^{(\alpha_2)}, \cdots, \varepsilon_n^{(\alpha_n)} \rangle$, called *input arguments*, to another vector of unique identifier symbols $\Gamma = \langle \gamma_1^{(\beta 1)}, \gamma_2^{(\beta 2)}, \cdots, \gamma_m^{(\beta_m)} \rangle$, called *fuzzy state* or *fuzzy value assignment*, with a certain degree. [2] In other words, $R : \Delta \rightarrow_\mu \Gamma$, in which the value of μ for various arrangements of arguments and possible values are predefined in the language by the fuzzy interpretation of R. This can also be represented using fuzzy relations [114] in which the truth values of a relation of set of inputs are resulted. Therefore, using fuzzy relations: $R : \langle \Delta, \Gamma \rangle \rightarrow \mu \in \{l_1, l_2, ..., l_n\}$, in which $\langle \Delta, \Gamma \rangle$ is the concatenation of two vectors Δ and Γ. Finally, it is notable to mention that logical and phenomenal random variables in Fuzzy MEBN are analogous to fuzzy predicate and functions, respectively.

[2] For simplicity, vector representations such as $\Delta = \langle \varepsilon_1^{(\alpha_1)}, \varepsilon_2^{(\alpha_2)}, \cdots, \varepsilon_n^{(\alpha_n)} \rangle$ will be denoted by $\Delta = \langle \varepsilon_i^{(\alpha_i)} | \; i = 1 \cdots n \rangle$ from now on, and throughout the paper.

Random variable terms in Fuzzy MEBN are created exactly the same way as those of regular MEBN. In general, ordinary random variables (say u and v) are deemed atomic random variable terms that may be used as input arguments for both logical and phenomenal random variables (*e.g.*, $AreInCommunication(u, v)$ and $Driver(v)$ respectively) to make more complex random variable terms. Furthermore, random variable terms can be logically related to each other using logical connectives random variables. The resulting interpretation of a complex random variable term is determined by applying individual interpretations, which are predefined by the language on all symbols of the language, and merging them using the logical connectives interpretations.

4.3.2 Fuzzy MEBN Fragments (MFrags)

The building blocks of a MEBN Theory (MTheory) are MEBN Fragments (MFrags) that semantically and causally represent a specific notion of the knowledge.

Definition 1. *A Fuzzy MFrag (Fuzzy-MFrag) is defined as* $\mathcal{F} = (\mathcal{C}, \mathcal{I}, \mathcal{R}, \mathcal{G}, \mathcal{D}, \mathcal{S})$ *which hosts three different types of nodes, namely context nodes* $\mathcal{C}$*, input nodes* $\mathcal{I}$ *and resident nodes* $\mathcal{R}$*. Context nodes represent the semantic structures of knowledge by using First-order Fuzzy Logic sentences. Moreover, input nodes act as bridges to resident nodes in other Fuzzy-MFrags, and faciliate feeding any relevant information to the current Fuzzy-MFrag. Finally, resident nodes are random variables that are conditioned on the values of the context and input nodes. Additionally, in an Fuzzy-MFrag* $\mathcal{F}$*,* $\mathcal{G}$ *represents a Fuzzy-MFrag graph, set,* $\mathcal{D}$ *contains local distributions per each resident node, and* $\mathcal{S}$ *encompasses a set of fuzzy if-then rules to be used by the Fuzzy Inference System (FIS). It should be noted that the sets* $\mathcal{C}$*,* $\mathcal{R}$*, and* $\mathcal{I}$ *are pairwise disjoint, and* $\mathcal{G}$ *is a Directed Acyclic Graph (DAG) whose nodes belong to* $\mathcal{I} \cup \mathcal{R}$*, and the root nodes are members of* $\mathcal{I}$ *only. Finally, context value assignment terms in* $\mathcal{C}$ *are used for enforcing constraints under which the local distributions apply.*

In Fuzzy-MFrags, contextual constraints will be assigned a truth value that implies how much a constraint is satisfied. The *consistency constraint degree* of Fuzzy-MFrags are then determined by referring to the fuzzy interpretations of the terms defined in the Fuzzy-MFrag and built-in Fuzzy-MFrags, and calculating the degree of satisfiability of the constraints as a whole. In addition to the consistency constraints, the local probability distribution in Fuzzy-MFrags are defined as a conditional probability distribution of the resident nodes given input/parent, and context nodes. Calculating this conditional probability is easy when the ordinary random variables used in the parents of the resident random variable, and the resident random variable itself are exactly the same. The problem arises when there exist ordinary variables in the parents that do not exist in the child. Such problems are usually tackled by applying aggregation functions and combining rules [163]. Laskey in [125] uses the notion of influence counts to combine the influence of multiple parents. Here, the same approach is adopted, with some refinements for the new Fuzzy MEBN.

Definition 2. *Let us assume that $\mathcal{F}$ is an Fuzzy-MFrag within which there exists a resident random variable $\psi(\Theta)$ parametrized by a vector of ordinary variables $\Theta = \langle \theta_i |\ i = 1 \dots n \rangle$:*

1. $\mathcal{B} = \{(\theta_1, {\varepsilon_1}^{(\alpha_1)}), (\theta_2, {\varepsilon_2}^{(\alpha_1)}), \dots, (\theta_1, {\varepsilon_n}^{(\alpha_n)})\}$ *is a binding set of ordered pairs wherein θ_is are ordinary variables, and ${\varepsilon_i}^{(\alpha_i)}$s demonstrate unique entity identifier symbols that are represented by fuzzy singletons whose membership degrees are shown by α_is. Additionally, $\Delta = \langle \varepsilon_i^{(\alpha_i)} |\ i = 1 \cdots n \rangle$ is a vector of size n with elements arranged in the same order of θ_is in $\psi(\Theta)$.*

2. *With $\mathcal{B}$ being a binding set, and $\psi(\Delta)$ as the instance of ψ after substituting respective ${\varepsilon_i}^{(\alpha_i)}$ for each θ_i, the value assignment $\{(\Gamma = \phi(\Delta)\}$ is achieved that is called a potential influencing fuzzy configuration for $\psi(\Delta)$ in which $\phi(\Delta)$ is either an instance of one of its parents, or a context random variable residing in its Fuzzy-MFrag. Accordingly, Γ is a truth value (membership degree) for context random variables, and denotes a possible fuzzy state of $\phi(\Delta)$ for parent random variables. The fuzzy states are gained using the local distribution $\mathcal{D}$ and the fuzzy rule-sets $\mathcal{S}$ defined in the Fuzzy-MFrag $\mathcal{F}$.*

3. *With $\mathcal{B}$ as the binding set, and upon substituting each unique entity identifier ${\varepsilon_i}^{(\alpha_i)}$ with ordinary random variables θ_i, context constraints, which are reflected by context random variables, are satisfied to some degree based on their predefined fuzzy interpretation. Thus, the truth value of context random variable ϕ_j is calculated using Eq. 4.1:*
$$L^*_{\mathcal{B}} = \sup_{\Delta_{\mathcal{B}}}(\phi_j(\Delta_{\mathcal{B}})) \tag{4.1}$$
where the size of $\Delta_{\mathcal{B}}$ is equal to the number of inputs that ϕ_j takes, and its elements are borrowed from the binding set $\mathcal{B}$. $L^*_{\mathcal{B}}$ is also considered to be a set that contains all the equal supremum values. [3]. Accordingly, an influencing fuzzy configuration is

[3] Supremum is used here for the sake of generality. In real experiments with finite sets, it can be substituted with maximum.

a potential fuzzy configuration whose unique identifier assignments are found using Eq. 4.2.

$$\Delta^*_{\mathcal{B}} = \arg\sup_{\Delta_{\mathcal{B}}}(\phi_j(\Delta_{\mathcal{B}})) \quad (4.2)$$

in which $\Delta^*_{\mathcal{B}}$ is a set of all the potential fuzzy configurations that yield the supremum value in Eq. 4.1. Using Eq. 4.2, equivalent influential fuzzy configurations are those in which $\phi(\Delta^i_{\mathcal{B}}) = \phi(\Delta^j_{\mathcal{B}})$ for $\Delta^i_{\mathcal{B}}, \Delta^j_{\mathcal{B}} \in \Delta^*_{\mathcal{B}}$ and $i \neq j$, and equivalence classes are distinct fuzzy configurations of parents of $\psi(\theta)$.

4. *Assuming that $\mathcal{E} = \{{\varepsilon_1}^{(\alpha_1)}, {\varepsilon_2}^{(\alpha_1)}, ..., {\varepsilon_n}^{(\alpha_n)}\}$ is a set of unique identifier symbols, a partial fuzzy world $\mathcal{W}_f$ for $\psi(\Theta)$ is constructed by instantiating its parents as well as context random variables with each member of $\mathcal{E}$. Moreover, a partial fuzzy world state $\mathcal{S}_{\mathcal{W}_f}$ will be the fuzzy value assignments of the generated partial world.*

5. *Finally, the influence counts $|\mathcal{S}_{\mathcal{W}_\psi}|$ for $\psi(\Delta)$ is defined as the number of influencing fuzzy configurations that $\mathcal{S}_{\mathcal{W}_f}$ has for each equivalence class.*

It is obvious that finding influence counts in Fuzzy MEBN is exactly the same as in regular MEBN when all the context random variables are assigned the same truth value in different potential configurations. Otherwise, the number of cases in which the consistency constrained are satisfied are reduced by applying Eq. 4.1 and Eq. 4.2.

Upon having their consistency constrained analyzed, and after determining the configuration parent nodes, Fuzzy-MFrags will have the probability distribution of their resident nodes calculated as a conditional probability on the possible values of the resident node given the values of its parents (input nodes or findings), and context nodes. Next definitions show how regular Bayesian Networks are replaced with Fuzzy Bayesian Networks.

Definition 3. *Let us assume that $\mathcal{E}$ is the set of unique entity identifiers, and in an Fuzzy-MFrag $\mathcal{F}$, N_ψ is the set of all possible values of an instance of the resident node $\psi(\Theta)$ residing in $\mathcal{F}$ (showed by $\psi(\Delta)$). Then:*

1. *The fuzzy state of $\psi(\Delta)$ is defined as the vector $\Gamma_\psi = \langle {\gamma_j}^{(\beta_j)} |\ j = 1 \ldots |N_\psi| \rangle$ in which ${\gamma_j}^{(\beta_j)} \in N_\psi$ for $\gamma_j \in \mathcal{E}$ and β_j as the degree of being in the individual state γ_j.*

2. *The local probability distribution $\pi_\psi(\Gamma_\psi | \mathcal{S}_{\mathcal{W}})$ is a conditional probability density function that shows the likelihood of resident random variable ψ being in fuzzy state Γ_ψ given partial fuzzy world state $\mathcal{S}_{\mathcal{W}_f}$ that contains the fuzzy states of both parent and context nodes.*

3. Since $\pi_\psi(\Gamma_\psi|\mathcal{S}_\mathcal{W})$ is a probability density function, then $\eta\sum_k \pi_\psi(\Gamma_{k,\psi}|\mathcal{S}_\mathcal{W}) = 1$ wherein $\Gamma_{k,\psi}$s are various fuzzy states that can be generated by exchanging the membership degree β_j of each individual state γ_j, and η is the normalization factor.

Regular Bayesian Networks (BN), are Directed Acyclic Graphs (DAG) in which nodes represent random variables with finite states, and edges specify conditional dependencies between random variables. For each random variable R, a Conditional Probability Table (CPT) is defined that determines the likelihood of R being in one of its states conditioned on the configuration of its parents.

Inference on BN is performed by first adding the observations to the network and then finding the likelihood of the desired random variable by running one of the common inference algorithms in BN such as Variable Elimination, or Belief Propagation [121].

4.3.3 Fuzzy-MEBN Inference

In our proposed Fuzzy Bayesian Network (FBN) [81], *Fuzzy* Random Variables (FRVs) are defined to be composed of both uncertainty and ambiguity factors, which are respectively represented by a discrete Probability Density Function (PDF), and the fuzzy sets defined on their universe of discourse. This model is able to cover both discrete and continuous PDFs.

The fuzzy state of an FRV, called A, is represented by

$$\Gamma_A(x) = \langle a_1^{\mu_1(x)}, a_2^{\mu_2(x)}, \cdots, a_N^{\mu_N(x)} \rangle \tag{4.3}$$

in which N is the number of fuzzy sets defined on the universe of discourse of A, γ_is are the linguistic terms, and $\mu_i(x)$s are their corresponding membership functions, s.t. $\sum_{i=1}^N \mu_i(x) = 1$. Moreover, the *a priori* probability distribution of an FRV is represented through applying Ordered Weights Aggregation (OWA) operator [229] on the fuzzy sets defined on the universe of discourse of A. Therefore, the probability of being at the fuzzy state $\Gamma_A(x)$ will be

$$P(\Gamma_A(x)) = \tilde{P}_A(x) = \frac{1}{Z}\sum_{i=1}^{N} w_i \mu_{\sigma(i)}(x) \tag{4.4}$$

wherein $\sigma : \mathbb{N} \to \mathbb{N}$ is a permutation function, $1/Z$ is the normalization factor, and w_is are the OWA weights that are calculated based on the PDF of the corresponding random variable. Thus, for discrete random variables $w_i = p_i$, where p_i is simply the probability of being at ith state. If the random variable is continuous, then

$$w_i = \frac{1}{Z'} \int \mu_i(x).P(x)\mathrm{d}x \tag{4.5}$$

for $Z' = \sum_{j=1}^{N} w_j$ as the normalization factor.

The rest of the approach is based on the configuration of specific random variable and its parents. Accordingly, if an FRV, called A, has only one parent, say $\pi(A)$, then the fuzzy joint probability distribution is calculated as $\tilde{P}_{(A,\pi(A))}(x, y) = \tilde{P}_A(x)\tilde{P}_{(A|\pi(A))}(x|y)$, wherein

$$P(\Gamma_{(A|\pi(A))}(x|y)) = \tilde{P}_{(A|\pi(A))}(x|y) = \sum_{i=1}^{N}\sum_{j=1}^{M} \mu_i(x|b_j)\mu_j(y)P_{(A|\pi(A))}(a_i|b_j) \tag{4.6}$$

In Equation 4.6, $\mu_j(y)$s are the fuzzy sets defined on the universe of discourse of $\pi(A)$, and $\mu_i(x|b_j)$s are the resulting membership degrees after setting the antecedent of *if-then* rules in A's Fuzzy-Rule Set (FRS) to $y = b_j$, and performing the fuzzy the inference. Moreover, $P_{(A|\pi(A))}(a_i, b_j)$ is the Conditional Probability Table (CPT) entry when $A = a_i$ and $\pi(A) = b_j$.

Finally, if the FRV A has more than one parent, shown by $\pi_k(A)$, then the fuzzy joint probability distribution $\tilde{P}_{(A,\pi_1(A),\cdots,\pi_K(A))}(x, y_1, \cdots, y_K)$ can be simply calculated using a combination of the chain rule [121], Equation 4.6, and Equation 4.9. For example, if A has three parents $\pi_1(A)$ to $\pi_3(A)$, then applying chain rule on the fuzzy joint probability distribution yields: [4]

$$\tilde{P}(x, y_1, y_2, y_3) = \tilde{P}(x|y_1, y_2, y_3)\tilde{P}(y_1|y_2, y_3)\tilde{P}(y_2|y_3)\tilde{P}(y_3) \tag{4.7}$$

Equation 4.7 can be further simplified knowing the conditional dependency between parents. For instance if all the parents are mutually conditionally independent, then Equation 4.7 can be rewritten as:

$$\tilde{P}(x, y_1, y_2, y_3) = \tilde{P}(x|y_1, y_2, y_3)\tilde{P}(y_1)\tilde{P}(y_2)\tilde{P}(y_3) \tag{4.8}$$

In general, the fuzzy conditional probability distribution $\tilde{P}_{(A|\pi_1(A),\cdots,\pi_K(A))}(x|y_1, \cdots, y_K)$ is

[4]Subscripts are dropped for sake of notational simplicity.

calculated as:

$$
\begin{aligned}
P(\Gamma_{(A|\pi_1(A),\cdots,\pi_K(A))}(x|y_1,\cdots,y_K)) \\
&= \tilde{P}_{(A|\pi_1(A),\cdots,\pi_K(A))}(x|y_1,\cdots,y_K) \\
&= \sum_{i=1}^{N}\sum_{j=1}^{R}\widehat{\mu}_i(x|\bar{r}_j)P_{(A|\pi_1(A),\cdots,\pi_K(A))}(a_i|\bar{r}_j) \quad (4.9)
\end{aligned}
$$

in which $\bar{r}_j$ is chosen from the cross product of possible values of parents: $\bar{r}_j \in \{\Gamma_{\pi_1(A)} \times \Gamma_{\pi_2(A)} \times \cdots \times \Gamma_{\pi_K(A)}\}$, R is the total number of combinations that the parents can make: $R = |\Gamma_{\pi_1(A)}||\Gamma_{\pi_2(A)}|\cdots|\Gamma_{\pi_K(A)}|$, and $\widehat{\mu}_i(x|\bar{r}_j)$s are the resulting membership degrees after setting the antecedents of *if-then* rules in A's Fuzzy-Rule Set (FRS) to $\langle y_1,\cdots,y_K\rangle = \bar{r}_j$, and performing the fuzzy the inference. Finally, $P_{(A|\pi_1(A),\cdots,\pi_K(A))}(a_i|\bar{r}_j)$ is the Conditional Probability Table (CPT) entry when $A = a_i$ and $\langle\pi_1(A)\cdots\pi_K(A)\rangle = \bar{r}_j$.

Evidence in FBNs are in the form of fuzzy observations that may include inherent ambiguity [5]. However, an FBN behaves similar to a regular BN, when dealing with non-ambiguous evidence. Ambiguous evidences in SHDF originate from either hard or soft data. In the case of hard data, the fuzzification process is applied on the evidence data first, and then after having its vagueness factor [84, 81] determined, an ambiguous hard evidence enters the FBN. On the other side, two main sources of soft data are human and the internet (also called H-Space and I-Space by [92]), which are inherently ambiguous. In the next sub-section, we propose a novel algorithm for importing soft evidence to an FBN through measuring semantic similarity between the given evidence and the context of network. As a result of this procedure, both hard and soft evidences are in the form of fuzzy states shown in Equation 4.3 that can be consequently used in an inference algorithm.

Inference is the probabilistic answer to the query $P_{Q|E}(\cdot|\bar{e})$, in which Q is the set of query nodes, and $E = \bar{e}$ is the set of evidence nodes with the corresponding assignments. Variable Elimination and Clique Tree [121] (*a.k.a.*, Junction Tree) are two most common inference algorithms in regular BNs. Here, we modify the Junction Tree (JT) algorithm to tackle inference in FBNs. For detailed information about the JT algorithm, interested reader is referred to [121].

In Fuzzy-MEBN, the JT algorithm is applied on the resulting Situation-Specific Fuzzy Bayesian Network (SSFBN) that is created once all the consistency constraints of an FMTheory, reflected by the context nodes, are satisfied. It is notable that solving satisfiability in Fuzzy-MEBN is different from regular MEBN as mentioned in [81]. Nevertheless,

[5] The terms *ambiguity* and *vagueness* may be used interchangeably throughout the paper.

the required steps to construct a situation-specific belief network are defined by Mahoney and Laskey in [143], which are used in Fuzzy-MEBN as well. The output of this algorithm in Fuzzy-MEBN is an FBN that is contextually related to a specific situation.

Based on the same idea proposed by Koller and Friedman [121], the factors $\phi_i \in \Phi$ for a regular BN without evidence are the Conditional Probability Densities (CPDs), whose production is the joint probability distribution of the whole network. Therefore, in our proposed FBN, the *fuzzy* factors $\tilde{\phi}_i$s are set to the Conditional Fuzzy Probability Densities (CFPDs) that are constructed using Equation 4.4, 4.6, and 4.9. In case of an FBN with fuzzy evidences $\tilde{e}$, the CFPDs are first recalculated given $\tilde{e}$, and then they are assigned to the corresponding factors. The fuzzy joint probability distribution of the network will be:

$$\tilde{P}_\Phi(X) = \prod_{\phi_i \in \Phi} \tilde{\phi}_i \tag{4.10}$$

The JT algorithm starts with *moralizing* the FBN (since it is a directed graph) by simply connecting parents that have a common child, and then undirecting all the edges in the network. Once the network is moralized, *maximal* cliques are determined, and the clique potentials are initialized as:

$$\tilde{\psi}_c^{[0]}(X_c) = \prod_{i:x_i \in X_c} \tilde{\phi}_i \tag{4.11}$$

wherein X_c is the set of all FRVs in clique c. After this step, the graph goes through a *message passing* process that propagates the probabilities across the network, wherein each clique sends a message to its neighboring cliques that are computed as:

$$\tilde{\delta}_{i,j}^{[t]}(S_{i,j}) = \sum_{C_i - S_{i,j}} \tilde{\psi}_i^{[t-1]} \prod_{k \in N_i - j} \tilde{\delta}_{k,i}^{[t-1]} \tag{4.12}$$

in which $\tilde{\delta}_{i,j}^{[t]}(S_{i,j})$ is the message clique i sends to clique j at step t, $S_{i,j} = C_i \cup C_j$ is called the separator set, and N_i is the neighbors set of clique i. This process is iteratively performed until convergence occurs: $\tilde{\delta}_{i,j}^{[t]}(S_{i,j}) = \tilde{\delta}_{i,j}^{[t-1]}(S_{i,j})$. At the end, assuming that for each clique C_i, and separator set $S_{i,j}$, fuzzy beliefs are shown by $\tilde{B}_i(C_i) = \tilde{\psi}_i^{[t]} \prod_{k \in N_i} \tilde{\delta}_{k,i}$, and $\tilde{M}_{i,j}(S_{i,j}) = \tilde{\delta}_{j,i}\tilde{\delta}_{i,j}$, respectively, the fuzzy joint probability distribution can be found using Equation 4.13 [121].

$$\tilde{P}_\Phi(X) = \frac{\prod_i \tilde{B}_i(C_i)}{\prod_{i,j} \tilde{M}_{i,j}(S_{i,j})} = \prod_i \tilde{\psi}_i^{[t]} \tag{4.13}$$

Among all the factors that are directly bonded to the structure of the SSFBN, such as the given query, current evidences, and the semantic relations, only the current evidences may vary over time. Based on this observation, we propose a hybrid version of Fuzzy-MEBN inference algorithm that recreates the SSFBN structure only if the added evidence causes any changes to the structure of the previously built SSFBN. As a result, in hybrid version of Fuzzy-MEBN inference, SSFBN will be created whenever a new evidence has arrived or removed, to help with the reduction of the required time for Fuzzy-MEBN inference.

4.4 Traffic Situation Assessment

This section introduces our proposed Traffic Situation Assessment (TSA) model, which will complete the Attention Assist Framework (see Chapter 6). The overall block diagram is demonstrated in Figure 4.1. TSA operates on the data and information that come from various sources with different levels of abstraction. Accordingly, LLDF is performed at the TEA that outputs entities and/or information attributes that are previously determined by the context (see Figure 3.2 in Chapter 3 for implementation details of the TEA). Upon being settled in the same level of abstraction, various information streams enter a fuzzification process through which continuous variables are discretized first, and along with the ambiguous discrete variables, are assigned an ambiguity factor. Also, the soft data is processed in the SDEA unit and based on semantics similarity measurements, are assigned a vagueness factor. All these sets of variables are then fed to the Fuzzy-MEBN model for further higher-level fusion, wherein domain expert knowledge is used to define semantic and causal structures, and to construct an FRS for each node. At the end, a Situation-Specific Fuzzy Bayesian Network (SFSBN) [143] is created on the FBN that is resulted from the fusion of the information uncertainty and ambiguity. The rest of this section describes different parts of this abstract model in more details.

4.4.1 Data and Information Sources

There are diverse sources of data and information sources in the IoC environment, which can be ubiquitously utilized by interested entities. Accordingly, some sets of the lower-level data sources need to get fused first to create higher-level information (see [86]). For example, in VANET context, the lower-level data that mainly come from physical sensors on a vehicle (*i.e.*, GPS, range finders, and speedometer), or its surrounding vehicles (through V2V communication) can be fused to create higher-level information attributes/entities (*i.e.*, distance). Discretization of such data using fuzzy logic is causes ambiguity. For

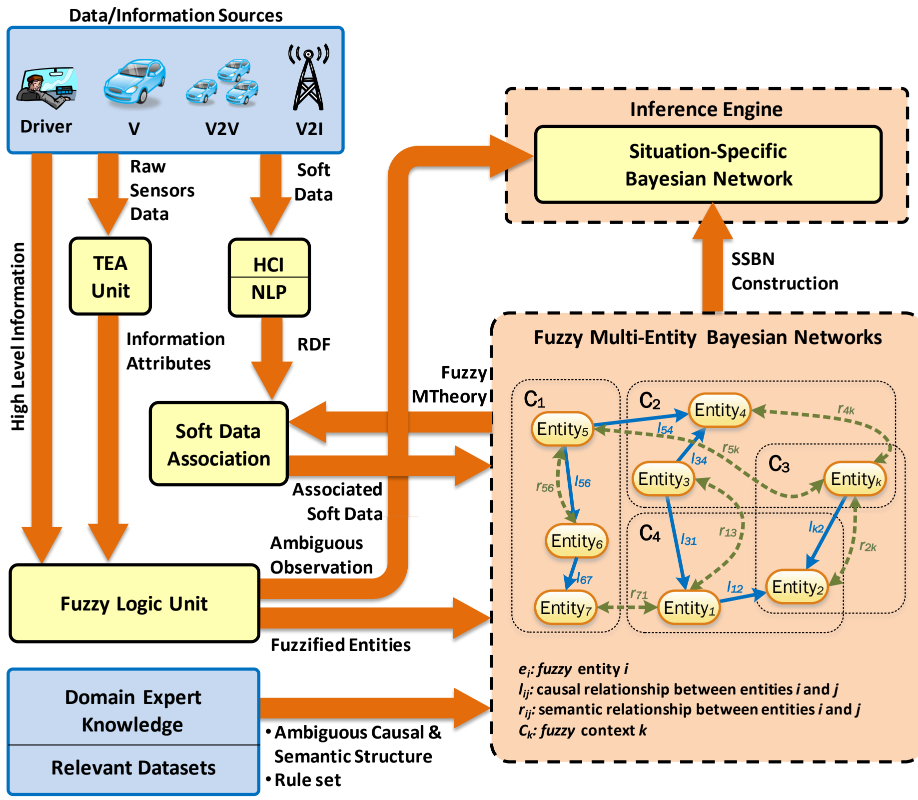


Figure 4.1: The Block diagram of the overall paradigm of the proposed TSA unit

instance, when discretizing the speed of a vehicle, a $70km/h$ speed might be ambiguously interpreted both normal and fast.

Examples of the soft data can be the information generated by the driver or any of the on-board passengers that can be categorized as the H-Space [92]. Moreover, there also exist flows of information in higher-level that can directly import a deployed HLIF framework. For instance, the information made by the available infrastructure (*i.e.*, cloudy weather or busy road) through V2V, V2R, or V2I communication is of this type. Both the higher-level information and the soft data contain inherent ambiguity as they are usually made by human entities through ambiguous linguistic terms. For instance, when reporting the weather condition, a driver might say: *"the weather is cloudy"* whereas it can be both partly cloudy, and cloudy. Consecutively, ambiguity can be easily propagated throughout the fusion process when the constituting entities of a certain situation are formed with inherently ambiguous attributes. To tackle such imperfectness, a fuzzy-based technique is proposed whose implementation details are discussed as follows.

4.4.2 Fuzzy Logic Unit

Fuzzy Logic Unit is represented with the tuple $F\langle\Gamma, \Delta, S\rangle$ in which Γ and Δ represent the sets of continuous and discrete variables respectively, and the set S contains the specifications of fuzzy rule sets pre-defined by a domain expert. Both continuous and discrete inputs are fuzzified and annotated with membership degrees with respect to their corresponding fuzzy set to deal with the underlying ambiguity. Therefore, the resulting nominal state of a specific variable $C \in \Gamma$ or $C \in \Delta$ will be denoted by a vector $C' = [c_1^{\mu_1}, c_2^{\mu_2}, ..., c_N^{\mu_N}]$, called *Fuzzy State*, in which $c_i^{\mu_i}$s are corresponding fuzzy set labels (*'components'* as they are named Fogelberg *et al.* in [73]), annotated with their membership degrees μ_i such that $\sum_i \mu_i = 1$.

Fuzzification of a continuous variable, is straight forward and follows exactly the same conventional common procedure presented by L.A. Zadeh in [235], that maps a continuous value to fuzzy set labels with their relative membership degrees. For instance, using Gaussian fuzzy sets of *Slow*, *Normal*, *Fast*, and *VeryFast* defined on the universe of discourse of vehicle speed, a speed entity with the value of $S = s_1 km/h$ will be fuzzified and lies in the fuzzy state $S' = [Slow^{\mu_1}, Normal^{\mu_2}, Fast^{\mu_3}, VeryFast^{\mu_4}]$, wherein the membership degrees are normalized. Furthermore, and for discrete variables, linguistic modifiers that increase/decrease the observation ambiguity are used to dynamically increase/decrease the fuzziness of the sets defined on a particular discrete universe of discourse. In the proposed model, the fuzziness is altered using an ambiguity factor imposed by function $f(\mu_A, \alpha) = (\mu_A)^\alpha$, for μ_A as the membership function defined over set A, and α as the ambiguity factor. Function f is the general form of the conventional *dilation* and *contradiction* concepts in fuzzy set theory [114]. For instance, if in regular case, the state of weather is rather confidently determined by the fuzzy state vector $W' = [Sunny^{\mu_1}, Cloudy^{\mu_2}, Rainy^{\mu_3}, Snowy^{\mu_4}]$, then after applying an ambiguous interpretation on the observation, using function f, the resulting fuzzy state will become $W'' = [Sunny^{\mu'_1}, Cloudy^{\mu'_2}, Rainy^{\mu'_3}, Snowy^{\mu'_4}]$, in which membership portions are more smoothly distributed around the crisp value of the observed state. Numerical representation of ambiguous linguistic terms are assumed to be specified by a domain expert who also helps in structuring the causal and semantic relations among the entities in the current context, as well as defining a FRS (as a complementary feature to causal relationships) between them. Such domain-specific consultation, along with the resulting fuzzy states determined by the Fuzzy Logic Unit, are then fed to a HLIF framework, which is the Fuzzy-MEBN.

4.4.3 Soft-Hard Data Fusion Unit

As it was previously mentioned, the ambiguity of soft data comes from the inherent vagueness of its sources. Commonly, the first step towards reducing such ambiguity is to pre-process the soft data is and reformat them to the machine-readable *subject-predicate-object* RDF triplets [118]. Furthermore, each soft evidence needs to be associated to its corresponding FRV in the underlying Fuzzy-MTheory. We tackle this soft data matching problem, by utilizing the semantic similarity metrics, and the capabilities of Fuzzy-MEBN to find the best match for the given subject, predicate, and the object.

In the SDEA unit, subjects, predicates, and objects in RDF triplets are respectively associated to the fuzzy entities, resident nodes, and resident node states in Fuzzy-MEBN. This is done by first measuring the semantic similarity between the given subject sub_i and each of the defined entities in the FMTheory, and then, assigning the entity with maximum similarity sub_i's class. We call this process *entity matching* that further helps to narrow down the search to only those Fuzzy-MFrags whose consistency constraints are satisfied given the entity class of sub_i. After pin pointing these Fuzzy-MFrags, the semantic similarity metric is once again used to find the resident nodes that are semantically closest to the given pred_i. This process is called *context matching*. Finally, the given obj_i is treated as a fuzzy state whose degrees of membership are calculated by measuring the semantic similarity between obj_i and all the states defined in the resident node. This process, called *states matching*, generates an entity, called soft entity, that can be directly exported to a situation assessment model (*i.e.*, TSA unit in our case). Algorithm 2 summarizes our novel soft data matching problem in Fuzzy-MEBN. In this algorithm, N is the input Fuzzy-MTheory, and (sub_i,pred_i,obj_i) is the extracted RDF triplet from an input soft data. Furthermore, entity, context, and states matching processes are executed in lines 3 to 6, 8 to 20, and 21 to 23, respectively. In particular, this algorithm returns an entity (or a set of entities with different ambiguities) that can be used as input evidences to the underlying Fuzzy-MEBN. In fact, the convergence of the algorithm is dependent on the context matching process (lines 8 to 20), wherein a set of potential Fuzzy-MFrags are selected based on the truth values of the context nodes residing in them. If no Fuzzy-MFrags meet the threshold, then it means that the input soft data is not contextually related to the current application domain, and it should discarded. In such case, F' in Algorithm 2 will remain null, and consequently, $R' = \varnothing$ will be returned as the output.

Algorithm 2 Soft-Data Matching Algorithm

function SOFT-DATA-ASSOC(N,sub_i,pred_i,obj_i)
    $D_E \leftarrow D_R \leftarrow \varnothing$
    for all e in $N.E$ **do**
        $D_E \leftarrow D_E \cup \text{dist}_{TT}(e, \text{sub}_i)$
    $\text{sub}_i.\alpha \leftarrow \text{argmax}(D_E)$
    $F' \leftarrow \varnothing$
    for all f in $N.F$ **do**
        for all c in $f.C$ **do**
            if $\mu(c|\text{sub}_i) \geq \sigma$ **then**
                $F' \leftarrow F' \cup f$
    for all f in F' **do**
        for all r in $f.R$ **do**
            $D_R \leftarrow D_R \cup \text{dist}_{ET}(r, \text{pred}_i)$
    $R' \leftarrow \text{argmax}(D_R \geq \epsilon)$
    for all r in R' **do**
        $r.\Gamma \leftarrow \text{dist}_{EE}(r.S, \text{obj}_i)$
    return R'

4.5 Situation and Threat Assessment Experiments

In order to evaluate the applicability of the proposed TSA framework for a real world safety application, we have conducted experiments on a simulated Collision Warning (Avoidance) System (CWS). The CWS operates on a constant basis using the various sources of data and information as inputs and aims at assessing accident situations involving the vehicle running the framework. Accordingly, the situation of interest is considered to be the near-collision situation, which is defined as the likelihood of a set of contextually related entities in a specific environment. At the end, the performance of both regular and hybrid Fuzzy-MEBN inference algorithms along with a BN-based implementation of the CWS are assessed for different number of neighboring vehicles.

4.5.1 Expert Knowledge Extraction

Involving entities and their structures in a near-collision situation can be achieved with the help of a domain expert's analysis and interpretation. Based on a recent study by [47], such knowledge extraction and modelling procedure can be accomplished using a set of iterations through which the extracted knowledge can be formed into semantic and causal structures/rules. Inspired by the work of [117] and their analysis results for near-collision incidents with respect to driver inattentiveness, a list of four major hypotheses have been compiled along with their corresponding specifications that may be involved in attributing a vehicle to a near-collision situation. As shown in Table 4.1, several entities are identified that are involved in evaluating the near-collision situation related hypotheses and their specifications; namely, driving behavior, driver attention, and environment and surrounding conditions.

Table 4.1: The questions and their specifications that model a vehicle in a near-collision situation

1. Is the vehicle moving on an erratic path? • Does the vehicle have an inattentive driver? • Is the environment unsuitable for the vehicle? • Does the driver have a bad driving history? • Is the vehicle in danger by its surrounding vehicles?
2. Is the vehicle moving with an unsafe speed? • Is the driver distracted or drowsy? • Does the driver have a bad driving history? • Is the environment tempting the driver to drive fast? • Is the current speed safe when considering that of surrounding vehicles?
3. Is the vehicle moving with an irregular driving pattern? • Is the driver an amateur one? • Is the driver distracted by environmental factors?
4. Is the surrounding environment making danger for the vehicle? • Are there any surrounding vehicles being driven dangerously? • Are the drivers of surrounding vehicles attentive?

4.5.2 Modeling

In order to model the role of inattention in a near-collision situation, a semantic relations diagram is illustrated in Figure 2.2, wherein all the entities corresponding to the inattention factors, and the main entities involved in a near-collision situation along with their attributes and semantic relationships are presented. In Figure 2.2, four major entities (shown in green squares) give rise to the second level of entities, after taking the situation-specific knowledge (Table 4.1) into account. The second level of entities (shown in orange squares with curved corners) include driving behavior, driver distraction, drowsiness and history, environment, and VANET. These entities have semantic relations within themselves and the first-level entities. Furthermore, the information provided by the expert knowledge can help with distinguishing the attributes of each entity as well. These attributes (see [117]) are shown by blue ovals in Figure 2.2.

To continue with the knowledge extraction process, it is required to aim for the situation-specific knowledge that incorporates all the entities and shows their causal relations in modelling the near-collision situation. The required knowledge for this part can be either gained through consulting a domain expert, or analyzing the available datasets for that specific situation. For our case study, we rely on the insights provided in the technical report of [117], and incorporating all the entities introduced so far, since there is limited access to datasets that contain information regarding factors related to road incidents. Besides, we employ our latest analysis on structures of situations, and situation evolution, in order to organize entities efficiently and relative to their contextual nature [87].

The causal relationships involved in a near-collision situation are modelled according to a large set of rules. A small subset of them are presented in Table 4.2. Consequently, one can determine the causal relationship between different entities and propose the corresponding structure to address uncertainty by using the rules presented in this table. The resultant structure is shown in Figure 4.2.

4.5.3 Implementation

The last step in modeling the near-collision situation is defining the MEBN Fragments (MFrags) through which the corresponding fragments of the desired situations are modeled. For this part of our experimental setup, an open-source Java platform software, called UnBBayes [46] is used to model and simulate the situations of interest. In this environment, our MEBN Theory (MTheory) represents a collision threat and its constituting MFrags encapsulate the component situations which include the entities that share common semantic and causal relations. Therefore, from our structural situation analysis standpoint [87], the super situation in CWS will be Collision Threat Level (CTL) that reflects the current level of collision threat imposed on the vehicle v. CTL is composed of four component situations, namely, *VehicleMovementSituation(v)*, *EnvironmentSituation(e)*, *DriverSituation(d)*, and *VANETThreatLevel(n,v)*, whose contexts perfectly match with those of factors presented

Table 4.2: A Sample of the rules related to the near-collision situation

1. If VEH has aggressive VMS, then CTL is certainly high for VEH.
2. If SDL shows high, then VEH probably has aggressive VMS.
3. If SPD is normal and ROT is highway, then SDL is usually low.
4. If SPD is high and ROT is not highway, then SDL is definitely high.
5. If SPD is high and ROC is icy, then SDL is often high.
6. If SPD is high and DIS is close, then DDL is definitely high.
7. If DDL is high, then VMS is aggressive.
8. If DRS is inattentive, then CTL is likely high for VEH.
9. If DRD is high, then DRS is likely to be inattentive.
10. If DRV eye closure is long, then DRD is usually high.
11. If DRV uses hand-held device, then DSD is certainly high.
12. If DRS and ENV are inconsistent, then VEH has high CTL.
13. If DRF is many, then DEL is somehow low.
14. If DEL is low, then DRS is usually inconsistent.
15. If DRS is inconsistent and SDL is low, then VEH has high VMS.
16. If WEA is snowy and ROC is icy, then ENV is almost inconsistent.
17. If WEA is foggy, then ENV is definitely inconsistent.
18. If ROC is wet and WEA is rainy, then ENV is less consistent.
19. If V2M is aggressive, then VTL is definitely high.
20. If VTL is high, then CTL is usually high.

in Figure 2.2. Figure 4.3 shows how these component situations are semantically related (green pentagons), and construct the body of *CollisionThreatLevel(v)* super situation. In Figure 4.3, *VehicleMovementSituation*(v_1,*t*) encapsulate the entities/events that are related to the movement of the vehicle v_1 at time t. These entities/events are *ManeuveringLevel*(v_1, *t*), *SpeedDangerLevel*(v_1), and *DistanceDangerLevel*(v_1) that model the maneuvering behavior of vehicle v_1 at time t, its speeding pattern, and the distance to its surroundings. Moreover, the entities/events related to *DriverSituation(d)* are *DriverDistractionLevel(d)*, *DriverDrowsiness(d)*, and *DriverExperienceLevel(d)*, which are estimated by data fusion of the sensors installed inside the vehicle, and extracting driver's background information through V2I communication, respectively. In addition, environmental situation is also modelled through *EnvironmentSituation(e)* component situation whose entities/events are mainly assessed by communicating with the infrastructures that provide the information regarding current environmental conditions. Finally, the current threat imposed by the surrounding vehicles in VANET is assessed in *VANETThreatLevel(n,v)* component situa-

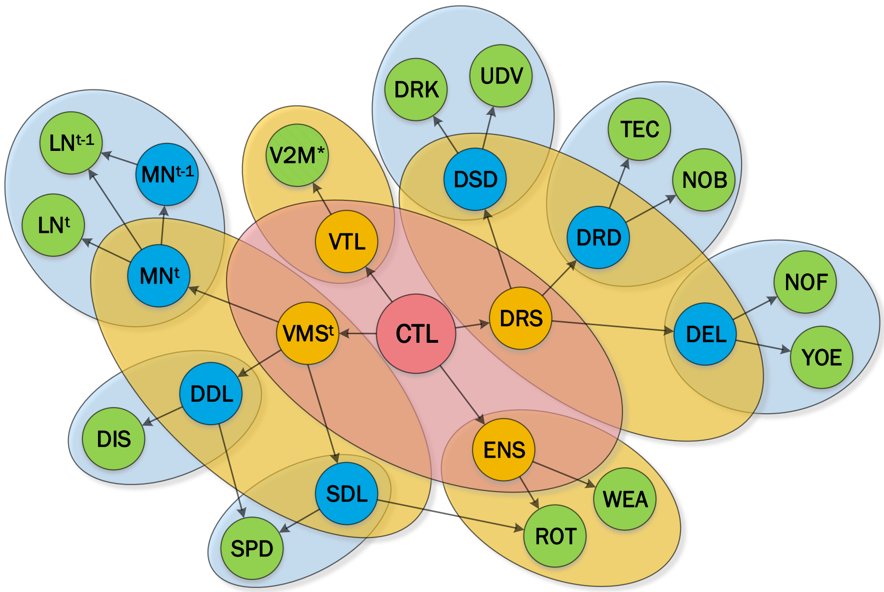


Figure 4.2: Involved entities in a near-collision situation and their causal relations. Super situation is shown in the middle of the network, and all the other component situations are arranged around it within orange ovals. The nodes structured in blue ovals are also high level entities that construct the component situations. See the list of abbreviations for more information.

tion whose entities/events are inbound data coming through V2V communication. In our case, this data is the current *VehicleMovementSituation*(v_2) of all the surrounding vehicles v_2, which are registered in the same VANET as the vehicle v_1. Note the context nodes $n = RegisteredVANET(v_2)$ and $\neg(v_1 = v_2)$ in the *VANETThreatLevel(n,v)* component situation.

4.5.4 Simulations Setup

The CWS is an implementation of the proposed TSA that is constructed using the Fuzzy-MEBN Theory introduced before. The evidences are imported to CWS through either physical sensors, and V2V or V2I communications. Moreover, the driver is able to communicate with the TSA through an HCI unit that is equipped with a microphone. The voice signal is processed by the Sphinx speech recognition engine [130], and then, the RDF triplets are extracted from the recognized voice using Natural Language Processing (NLP)

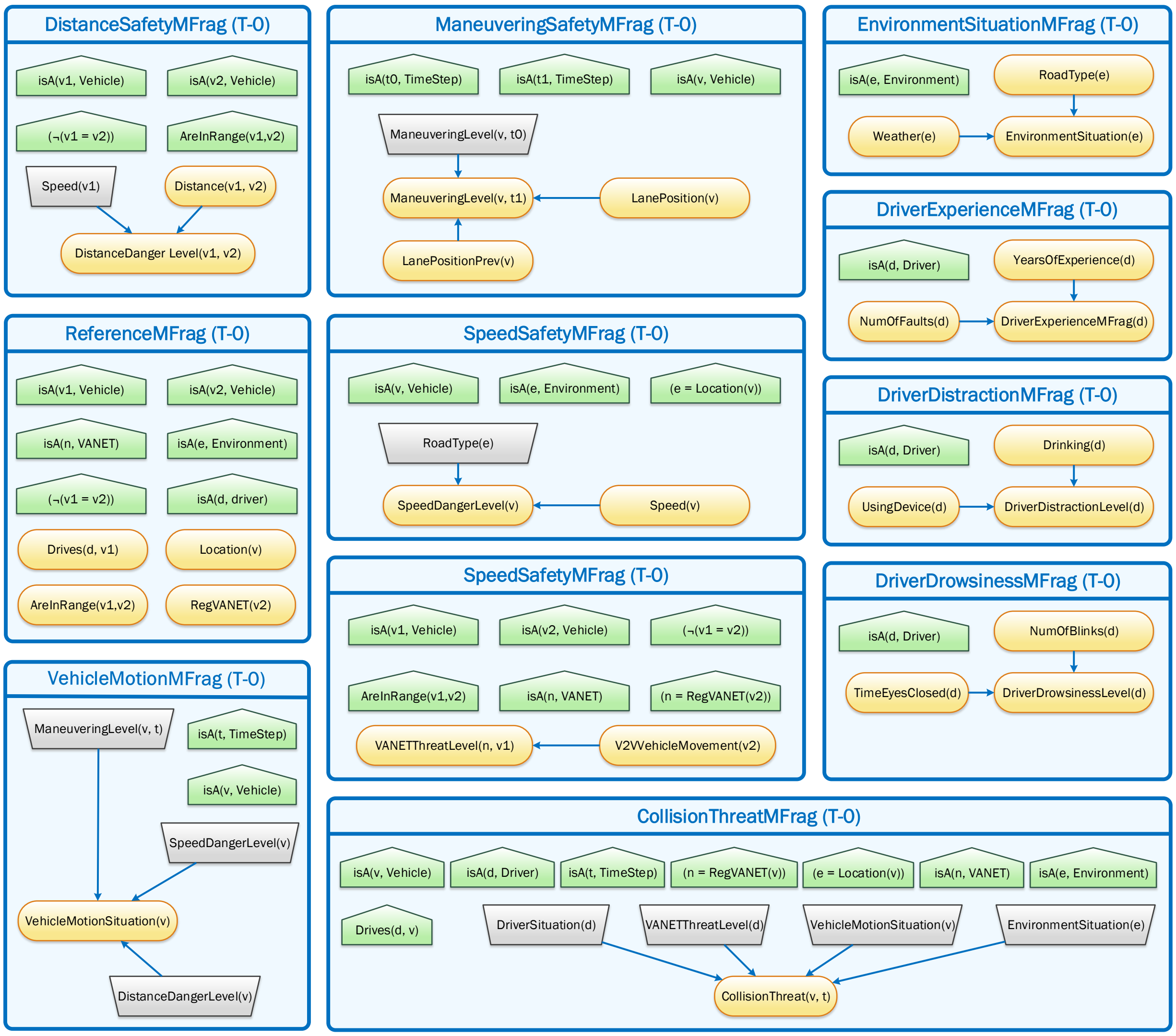


Figure 4.3: Fuzzy-MTheory that models the collision threat for vehicle v at time t. Green pentagons are context nodes, gray trapezoids are input nodes and yellow ovals are resident nodes.

techniques. Finally, the subject, predicate, and object, along with the designed Fuzzy-MTheory for road safety in VANET are set as the inputs of Algorithm 2. Simultaneously, CWS will be constantly running as long as the vehicle is turned on, and updates the status of its situation at pre-defined intervals. In other words, the Fuzzy-MEBN inference is performed on the CWS Fuzzy-MTheory based on a given query, and calculates the marginal probability distribution of the resident nodes of interest. In our simulations, the query will be CollisionThreatLevel($!VEHEGO_{1.0}, !T1_{1.0}$) that shows the collision threat level imposed on the ego vehicle, denoted by the fuzzy unique entity identifier $!VEHEGO_{1.0}$, at current time step $!T1_{1.0}$.

To show the functionality and adaptability of the CWS, we have separated our simulations to two main categories of single-vehicle and multi-vehicle. For the single-vehicle case, five scenarios are tested on a vehicle that is moving on a straight path. For the multi-vehicle case, a scenario is tested on an environment involving the ego vehicle and a number of neighboring vehicles all running the CWS continuously. In each scenario, all the situations of interest are assessed, and are used accordingly as the inputs for the corresponding super-situation.

Lastly, a fully customized version of an open-source driving simulator, OpenDS [146], is used as our simulations platform. We improved OpenDS by implementing some sensors such as range finders, lane camera, GPS, emitter and receiver, map and so on. The data of all these sensors go through a low-level data fusion process (see Figure 3.2) that outputs the required entities. In particular, we have chosen low-pass filter for range finders and Hough Transform [105] for lane detection.

4.5.5 Soft Entity Matching Results

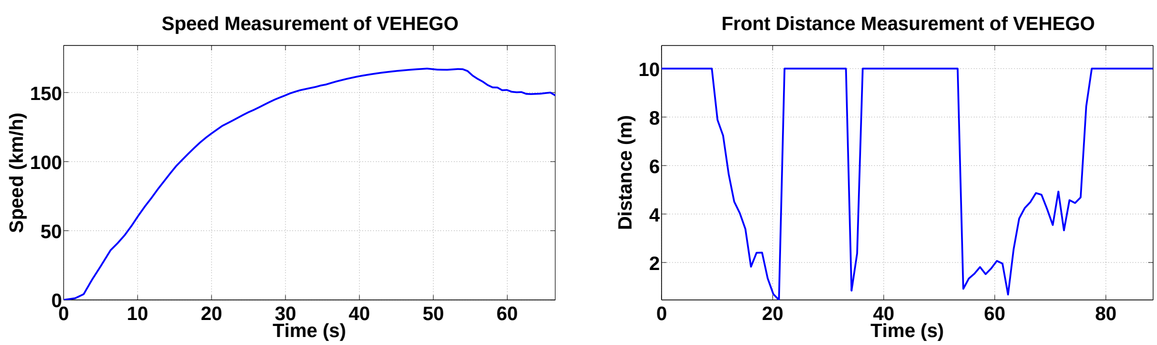
The accuracy of our proposed soft data matching algorithm is tested by providing it with different soft data reporting the condition of either a traffic vehicle, a traffic vehicle driver, or an environment. The resulting structured observation is then compared with the ground truth and the accuracy is found by simply calculating the ratio of correct matching. A subset of RDF triplets extracted from the user's voice signal, and their associated fuzzy states are shown in Tables 4.3 and 4.4. The results of soft data matching shows that for each three main factor in road accidents (see Figure 2.2) more than 72% of the times, the soft data was correctly associated to its relative entity. It is notable that this result is dependant on the words used by the driver to report a the condition of a traffic vehicle, a driver, or an environment. These data are used along with the hard data to export a better understanding for the current situation. Sample data of the speedometer and the front distance sensor are depicted in Fig.4.4a and Fig.4.4b, respectively. Here, the situation of interest is CollisionThreat, which is composed of the following component situations: VehicleMotionSituation, DriverSituation, EnvironmentSituation($!VEHEGO_{1.0}$), and VANETThreatLevel, wherein the later is assessed through the soft data provided by the driver (see Table 4.5). The estimated states of all situations are then fused to measure

Table 4.3: A subset of soft data sentences and their RDF triplets

#	**Soft Data**	**Subject**	**Predicate**	**Object**
1	VEH02 is going fast	VEH02	Go	Fast
2	Truck is moving slowly	Truck	Move	Slow
3	Bus's speed is high	Bus	Speed	High
4	Driver blinks rapidly	Driver	Blink	Rapid
5	Driver is distracted	Driver	Distraction	Yes
6	VEH02 advances slowly	VEH02	Advance	Slow
7	Weather is partly cloudy	Waterloo	Weather	Partly Cloudy

Table 4.4: A subset of soft data sentences and their RDF triplets

#	**Fuzzy State**	**Vagueness Factor**
1	$\mathrm{SPD} = \langle \mathrm{SLW}^{0.0}, \mathrm{REG}^{0.0}, \mathrm{FST}^{1.0} \rangle$	0.0
2	$\mathrm{SPD} = \langle \mathrm{SLW}^{1.0}, \mathrm{REG}^{0.0}, \mathrm{FST}^{0.0} \rangle$	0.0
3	$\mathrm{SPD} = \langle \mathrm{SLW}^{0.0}, \mathrm{REG}^{0.29}, \mathrm{FST}^{0.71} \rangle$	0.66
4	$\mathrm{NOB} = \langle \mathrm{FEW}^{0.25}, \mathrm{MNY}^{0.75} \rangle$	0.75
5	$\mathrm{DSD} = \langle \mathrm{LOW}^{0.0}, \mathrm{MED}^{0.25}, \mathrm{HIG}^{0.75} \rangle$	0.66
6	$\mathrm{SPD} = \langle \mathrm{SLW}^{1.0}, \mathrm{REG}^{0.0}, \mathrm{FST}^{0.0} \rangle$	0.0
7	$\mathrm{WEA} = \langle \mathrm{SUN}^{0.35}, \mathrm{CLD}^{0.65}, \mathrm{RIN}^{0.0}, \mathrm{SNW}^{0.0} \rangle$	0.4



(a) Speed measurement

(b) Front distance measurement

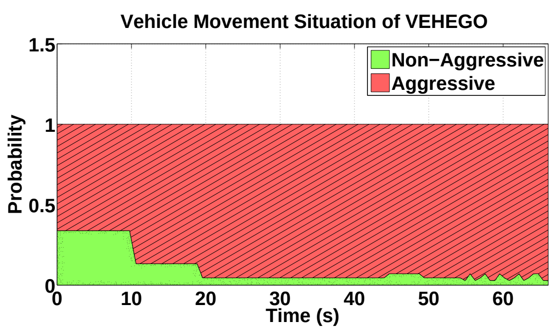
Figure 4.4: Physical sensor measurements of $!VEHEGO_{1.0}$

the current threat level. Figures 4.5a, 4.5b, and 4.6 demonstrate the state estimation of VehicleMotionSituation, VANETThreatLevel, and CollisionThreat, respectively. In these figures, Low, Medium and High states are shown with solid green, single-lined blue, and

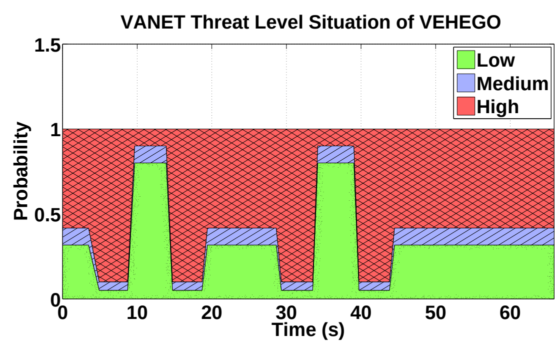
Table 4.5: The soft data used for SHDF

Soft Data	**Time Range (Time Slots)**
VEH02 goes fast	[5, 10]
Truck moves slowly	[10, 15]
VEH02's speed is high	[15, 20]
Truck goes fast	[30, 35]
Bus goes slowly	[35, 40]
VEH02 advances fast	[40, 45]

cross-lined red hatching, respectively. Besides, Non-aggressive and Aggressive states are also respectively shown with solid green and cross-lined red hatching. As demonstrated



(a)



(b)

Figure 4.5: The state estimations of (a) VehicleMotionSituation($!VEHEGO_{1.0}, !T1_{1.0}$), and (b) VANETThreatLevel component situations

in Figure 4.5b, the associated soft data that is reported by the driver influences the state estimation of VANETThreatLevel situation correctly, *i.e.*, its state is more towards high in time slots within which the reported soft data reports high traffic vehicle speed. This is further fused with the situations assessed through the physical sensors (Figure 4.5a, and presented in the threat level in Figure 4.6.

4.5.6 Traffic Situation Assessment Results

In this part, first we assume $!VEHEGO_{1.0}$ is not in communication with other vehicles in the VANET environment. Our goal in this part is to evaluate the performance of the CWS

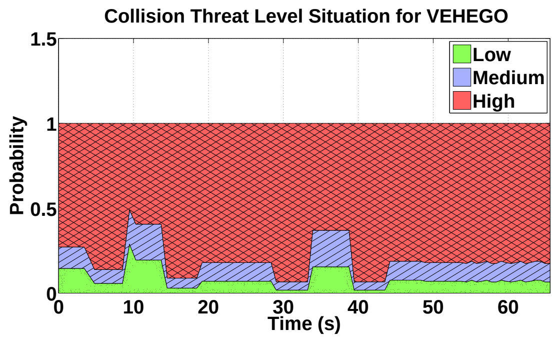
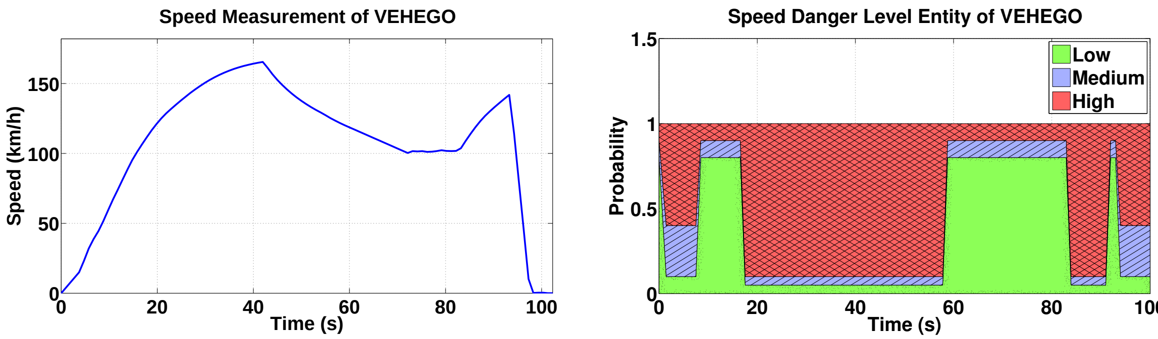


Figure 4.6: The overall CollisionThreat assessment

when operated in a local-only fashion. Four main entities, namely, SpeedSafetyLevel, ManeuveringLevel, and DistanceDangerLevel along with VehicleMotionSituation component situations are assessed in separate scenarios. Besides, DriverDistraction and DriverDrowsiness entities and the way they contribute to DriverSituation component situation are also analyzed.

Scenario 1: Speeding

In this scenario, $!VEHEGO_{1.0}$ moves on a straight path on a highway with varying speed for 100 seconds. The CWS samples from road type just once (see Figure 4.3) and at the beginning of the simulation, and samples from the speedometer data every 1 second. The measured speed along with its impact on SpeedSafetyLevel are depicted in Figures 4.7a and 4.7b. The varying speed of $!VEHEGO_{1.0}$ during 100 seconds of simulation, seen in Figure 4.7a, causes the uncertainty changes in the states of SpeedSafetyLevel entity, as illustrated in Figure 4.7b. In the first 20 seconds, $!VEHEGO_{1.0}$ increases its speed from 0km/h to almost 50km/h. Consequently, SpeedSafetyLevel outputs with high certainty that it is *not* in Low state (*i.e.*, the solid green shaded area that shows the probability of being in Low state is small in the first 20 seconds of the simulation). This is due to the fact that very low speed is dangerous in highways. As the speed of $!VEHEGO_{1.0}$ exceeds 120km/h SpeedSafetyLevel entity recognizes dangerous speed, and therefore, gives high certainty to High state (*i.e.*, larger portion of state estimation is dedicated to High state that is shaded with cross-lined red color).

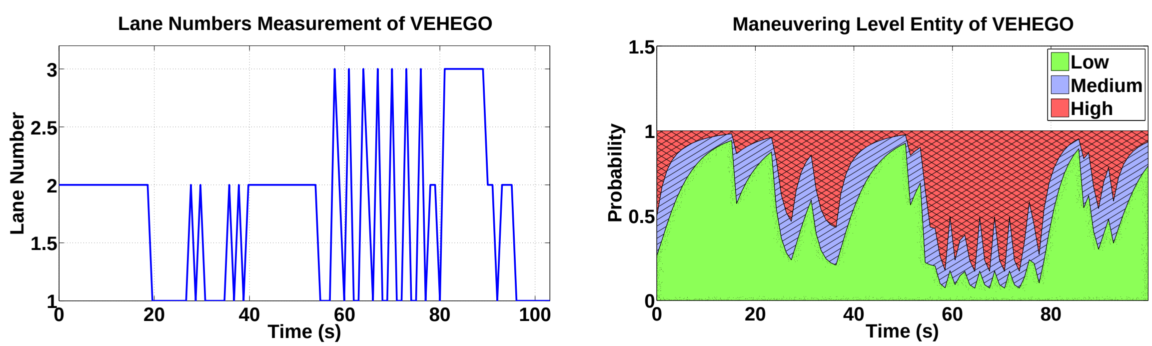


(a) (b)

Figure 4.7: (a) Output of the Speed entity in scenario 1, and (b) Measurement of the SpeedSafetyLevel entity in scenario 1. Low, Medium and High states are shown with solid green, single-lined blue, and cross-lined red hatching, respectively.

Scenario 2: Maneuvering

Maneuvering driving behavior is studied in this scenario in which $!VEHEGO_{1.0}$ changes lanes irregularly. $!VEHEGO_{1.0}$ uses lane camera and Hough transform to detect lane-based location, and utilizes the Markov model presented in Maneuvering Fuzzy-MFrag (see Figure 4.3) to assess the maneuvering event. Figure 4.8a and 4.8b show the lane detection and maneuvering level outputs, respectively. Current lane on which the vehicle is moving is



(a) (b)

Figure 4.8: (a) Output of the LanePosition entity, and (b) Measurement of the ManeuveringLevel entity in scenario 2. Low, Medium and High states are shown with solid green, single-lined blue, and cross-lined red hatching, respectively.

shown in Figure 4.8a. As it is clearly illustrated, $!VEHEGO_{1.0}$ goes on the second lane for the first 20 seconds of the simulation, and hence, its maneuvering level converges to Low state gradually (see Figure 4.8b). From second 20 to 40, $!VEHEGO_{1.0}$ changes between lanes 1 and 2 for 10 times, and consequently causes the altering manuvering level during that period. More adverse lane changing behavior is observed within time slots 60 to 80, which is correctly assessed by ManeuveringLevel event and is depicted in Figure 4.8b.

Scenario 3: Keeping Distance

In this scenario, $!VEHEGO_{1.0}$ goes on a straight line with varying speed and sometimes gets close to the moving traffic. Front, rear, left and right distances are measured by range finders, and the speed of the vehicle is sampled by the CWS every 500 milliseconds. The measured distances on all four directions, speed of the vehicle, and DistanceDangerLevel entity are presented in Figures 4.9a to 4.9f. Figure 4.9f shows how DistanceDangerLevel entity is estimated given the distance sensors and the speedometer data. All the periods during which the DistanceDangerLevel entity is in Low state with high certainty (*i.e.*, bigger solid green shaded area), $!VEHEGO_{1.0}$ is in safe distance with its surrounding. Conversely, whenever the vehicle's distance to the moving traffic gets to a level that is unsafe regarding its speed, the certainty of being in High state in DistanceDangerLevel entity grows accordingly. For instance, during time slots 78 to 98, $!VEHEGO_{1.0}$ is running with an average speed of 125km/h, while keeping a short distance (around 2 meters) to the leading traffic. Such an unsafe speed and distance is correctly assessed by DistanceDangerLevel entity during the same period.

Scenario 4: Vehicle Movement Situation

It is presented in Figure 4.3 that vehicle movement situation at a specific point of time is dependent on its speed, maneuvering, and distance danger levels. Here in this scenario, $!VEHEGO_{1.0}$ is moving freely on a highway with light traffic, while CWS continuously assesses DistanceDangerLevel, ManeuveringLevel, and DistanceDangerLevel entities along with VehicleMotionSituation component situation. The output of entity and situation assessments are shown in Figure 4.10d. The CWS determines with high certainty that $!VEHEGO_{1.0}$ has aggressive movement pattern whenever it adopts either a dangerous speed, an irregular maneuvering pattern, an unsafe distancem, or a combination of them. For instance, $!VEHEGO_{1.0}$'s movement is 90% Aggressive (10% Non-aggressive) at time slots 58, 60, and 70 since SpeedDangerLevel, ManeuveringLevel, and DistanceDangerLevel entities are all assessed to be in High state with high certainty.

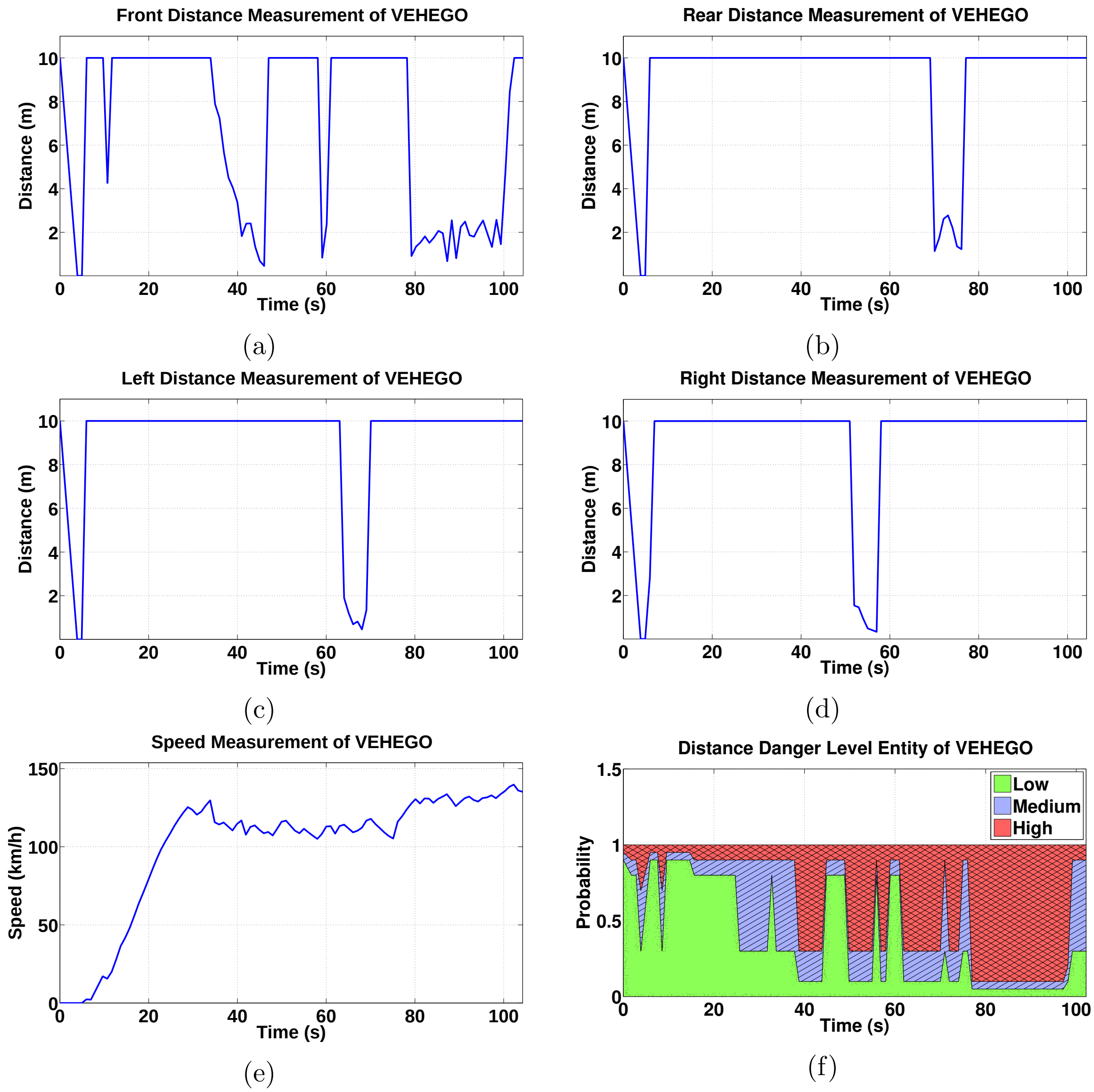


Figure 4.9: Output of the (a) front (b) rear (c) left, and (d) right Distance entity. (e) output of the Speed entity, and (f) Measurement of the DistanceDangerLevel entity of $!VEHEGO_{1.0}$ in scenario 3. Low, Medium and High states are shown with solid green, single-lined blue, and cross-lined red hatching, respectively.

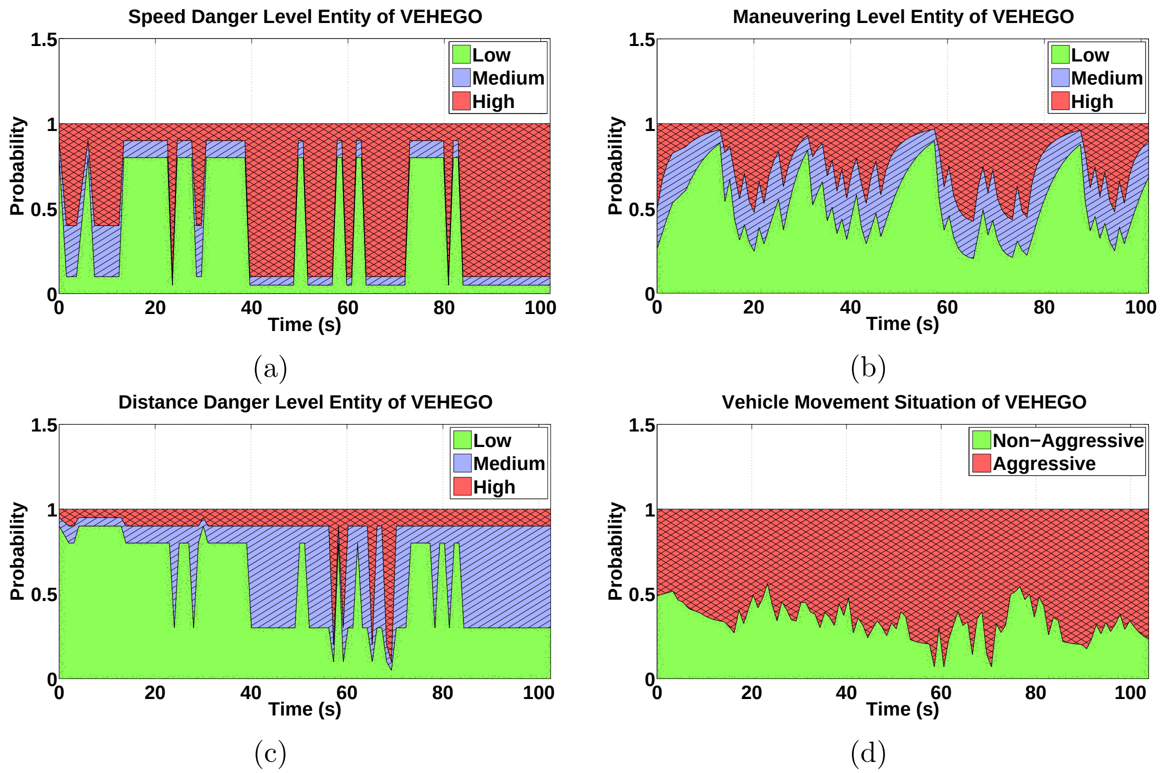


Figure 4.10: Measurement of the (a) SpeedDangerLevel, (b) ManeuveringLevel, and (c) DistanceDangerLevel entities. (d) Assessment of the VehicleMotionSituation component situation in scenario 4. Non-aggressive and Aggressive states are shown with solid green and cross-lined red hatching, respectively.

Scenario 5: Driver Situation

For assessing the driver situation, we assumed that the driver's distraction and drowsiness changes during the simulation based on some random actions such as having a drink or more frequent blinking. However, driver distraction and drowsiness can be detected using the sensors installed inside the vehicle, as it is also shown in [179]. Figures 4.11c, 4.12c, and 4.13 respectively show the results of drowsiness, distraction, and situation assessment of driver $!DRVVEH02_{1.0}$ who drives $!VEH02_{1.0}$. As it is seen in Figures 4.11a to 4.11c, different number of blinks per time unit, and the period during which eye closure is detected have high impact on drowsiness detection. For instance, during time slot 80 to 92, the driver is blinking about 5 times per time unit and keeps his/her eyes closed for 7 seconds. This causes the CWS to determine with high certainty that the driver drowsiness

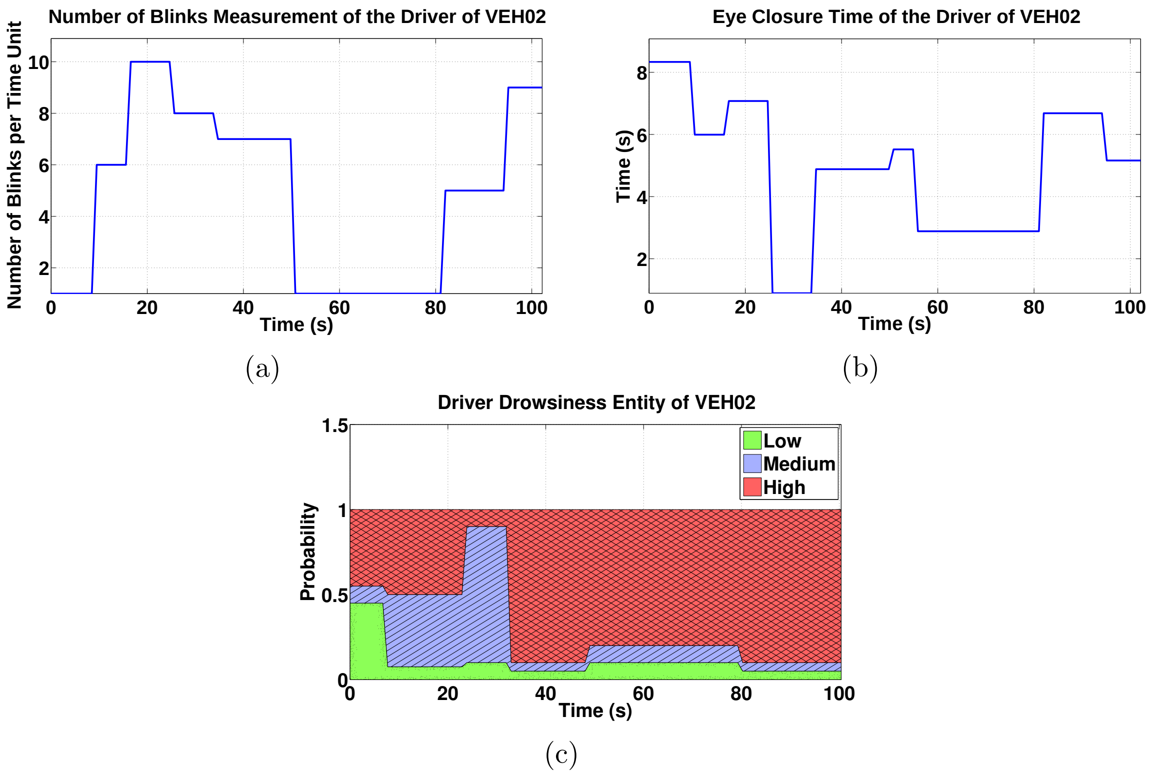
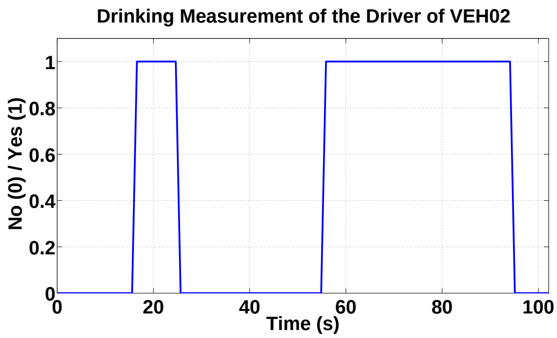
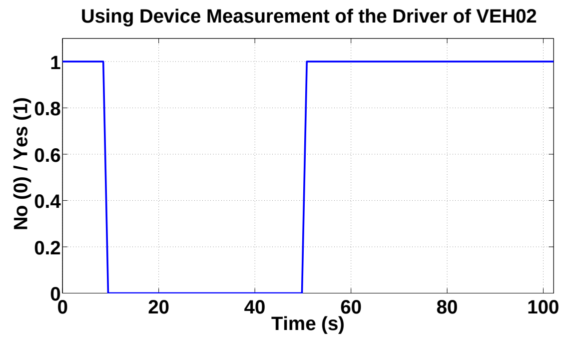


Figure 4.11: Output of the (a) NumOfBlinks and (b) TimeEyesClosed entities. (e) Measurement of the DriverDrowsinessLevel entity in scenario 5. Low, Medium and High states are shown with solid green, single-lined blue, and cross-lined red hatching, respectively.

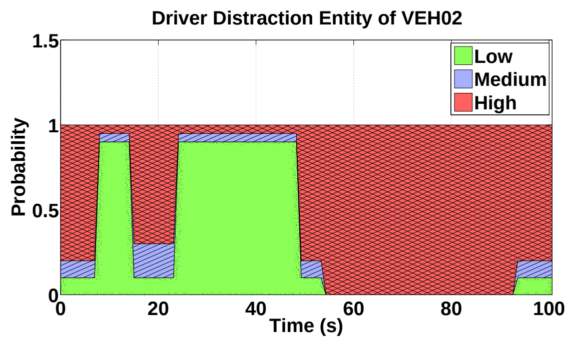
is in High state. Moreover, driver distraction is assessed based on knowing whether the driver is drinking or using any devices (such as cell phone, GPS, *etc.*). The result of the assessment, presented in Figure 4.12c, shows high inattentiveness whenever the driver is getting either distracted, drowsy, or both (as seen within seconds 55 to 95). Finally, the attentiveness of the driver is represented in DriverSituation component situation that is influenced by its parents, namely, DriverDrowsinessLevel and DriverDistractionLevel entities (see Figure 4.13). When the number of vehicles increases, it is assumed that they are operated in VANET environment, and therefore, are able to transmit data through V2V communication. We take advantage of this capability and use V2V communication to transmit the current movement behavior of the vehicle to its surrounding. In other words, each vehicle broadcasts its VehicleMotionSituation at specific intervals. The vehicles residing in the coverage area of the transmitter, receive the data, and provide it to



(a) Output of the entity in scenario 5



(b) Output of the entity in scenario 5



(c)

Figure 4.12: Output of the (a) Drinking and (b) UsingDevice entities. (c) Measurement of the DriverDistractionLevel entity in scenario 5. Low, Medium and High states are shown with solid green, single-lined blue, and cross-lined red hatching, respectively.

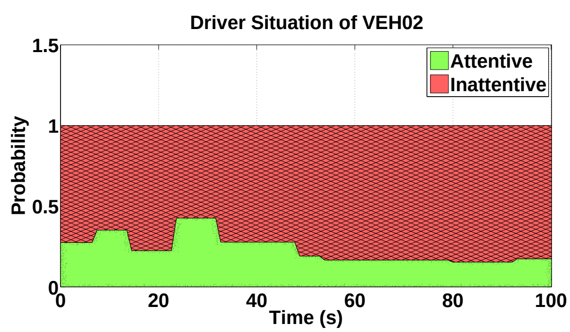


Figure 4.13: The DriverSituation component situation in scenario 5 whose parents are DriverDrowsinessLevel and DriverDistractionLevel entities. Attentive and Inattentive states are shown with solid green and cross-lined red hatching, respectively.

their own CWS as new evidence. This strategy makes the drivers aware of the threat that might be imposed within their own surroundings. Additionally, this information can be used to improve the collision threat assessment.

Scenario 6: VANET Threat Assessment

All the traffic vehicles and the ego vehicle, shown by ordinary variable v, perform hybrid version of the MEBN inference algorithm on the query VANETThreatLevel. Therefore, depending on whether a new evidence is available or not, the CWS of each vehicle recreates the SSBN, and then outputs the threat level of VANET n by adding the evidences and calculating the marginal of the resident node of interest. Figures 4.14a to 4.14d demonstrate the state estimation of the VehicleMotionSituation situation for 4 different traffic vehicles. In these figures, Low, Medium and High states are shown with solid green, single-lined blue, and cross-lined red hatching, respectively. Figure 4.14f summarizes the state estimation of VANETThreatLevel component situation. The default probability distribution of this situation is $\langle 0.3, 0.6, 0.1 \rangle$ for Low, Medium, and High, respectively, which is clearly seen in the periods during which $!VEHEGO_{1.0}$ is not in communication with any traffic vehicle (64 to 79 and 107 to 120 time slots). However, aggressive moving situation of V_5 within 36 to 45, and 50 to 55 time slots has correctly influenced the state estimation of VANETThreatLevel during the same periods. Similar cases can be observed for V_3 at 93th time slot. Moreover, VANETThreatLevel is in High state with high certainty within time slots 3 to 26, since there are more than two vehicles in the same VANET which are demonstrating aggressive movement (see Figure 4.14e).

Scenario 7: Collision Threat Assessment

Collision threat assessment is the main purpose of the CWS. Four different component situations, one of which modelling a key factor in car incidents (*i.e.*, driver, vehicle, environment, and the surrounding traffic) constitute the body of Collision threat super situation (see Figure 4.3). To demonstrate the efficiency of the CWS in assessing the collision threat, a 120 seconds driving scenario is set up that includes a highway with the ego vehicle $!VEHEGO_{1.0}$, driven by a user, and 8 traffic vehicles V_2 to V_9 moving with random behavior. All the vehicles continuously assess all the relevant situations at 500 milliseconds time steps, and also transmit and receive VehicleMotionSituation messages through V2V communication whenever residing in the coverage zone. The same scenario is run for 20 iterations and the state estimation of the CollisionThreat is stored at each time slot. Table 4.6 shows the average of state estimation for time slots at which a collision

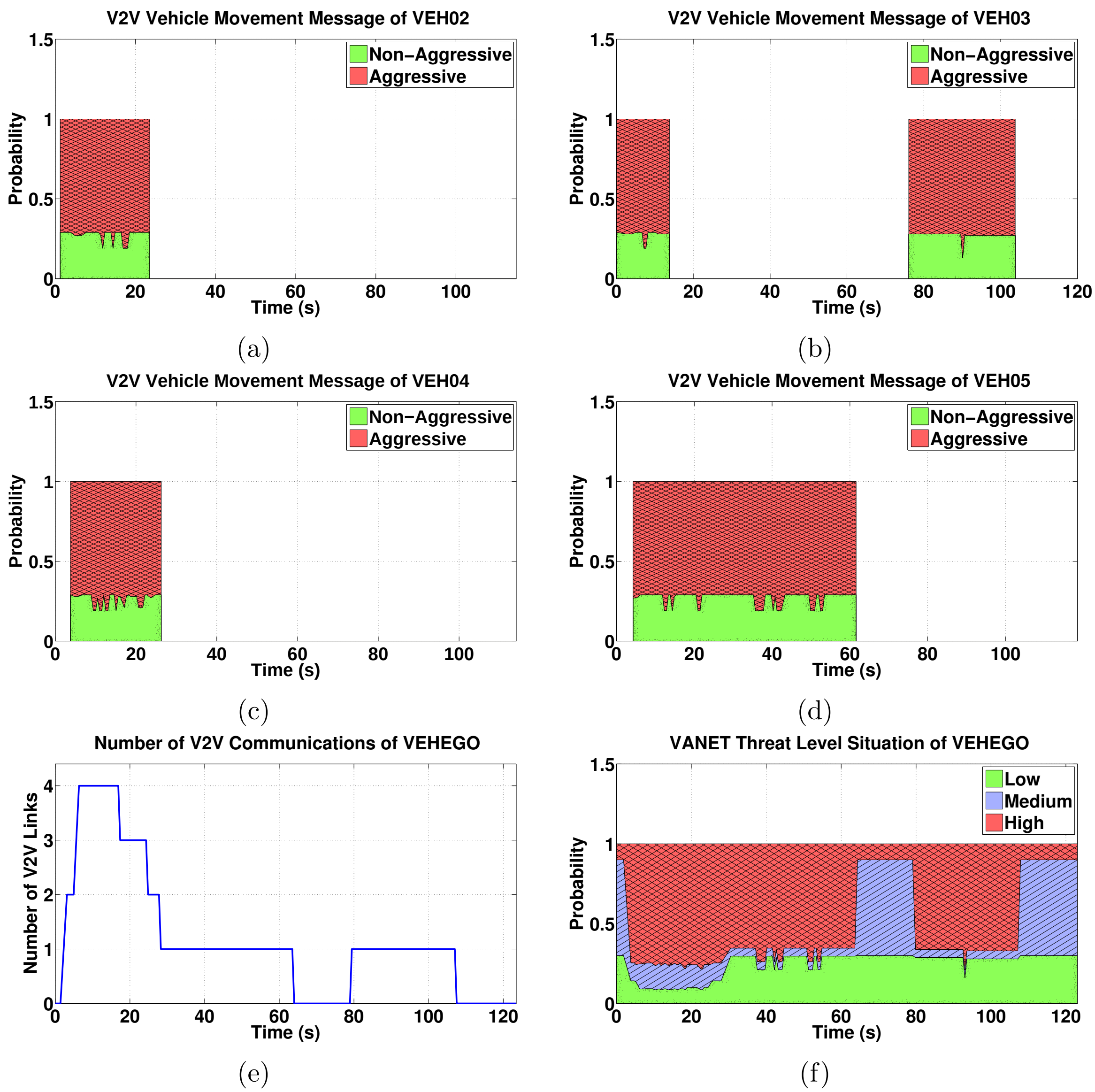


Figure 4.14: The VehicleMotionSituation messages of the traffic vehicles (a) $!VEH02_{1.0}$, (b) $!VEH03_{1.0}$, (c) $!VEH04_{1.0}$, and (d) $!VEH05_{1.0}$. (e) Number of traffic vehicles in communication with $!VEHEGO_{1.0}$. (f) VANETThreatLevel component situation. Non-aggressive and Aggressive states are shown with solid green and cross-lined red hatching, respectively.

has occurred. If at i-th iteration, $C^+ = \{s_{i,j} | s_{i,j} \to \text{"}Collision\text{"}\}$ shows the set of collision threat states $s_{i,j}$ at which a collision has occurred, then the average of their certainty will be:

$$\overline{s_i} = \sum_j p(s_{i,j}) \tag{4.14}$$

Furthermore, let $C^- = \{s_{i,j} | s_{i,j} \geq \overline{s_i} \wedge s_{i,j} \to \neg\text{"}Collision\text{"}\}$ be the set of non-collision threat states whose estimations are above the calculated average $\overline{s_i}$. Hence, the false alarm rate for window size of 0 time slots is defined as:

$$f = |C^-|/|C^+| \tag{4.15}$$

Lastly, to investigate the collision threat state estimation just before the collision occurrence, the window size is increased up to 5 time slots to include the time slots prior to the one at which a collision is occurred.

As shown in Table 4.6, in 80% of the simulations, the average of collision threat state in the case of accident shows with at least 80% of certainty that CollisionThreat is in High state. The performance of CWS is further evaluated by measuring the false alarms during the 120 seconds simulation run-time for each iteration. The results for different window sizes show that zero-sized window performs the best as the average of false alarms over all the iteration is 15% for that. However, this metric is increased to 15.3%, 16.45%, 18.43%, 19.59%, and 21.95% for window sizes of 1 to 5 time slots, respectively. This means that if the time step is set to 500 milliseconds no previous time slots is needed for efficiently assessing the collision threat.

The performance of our proposed model is compared against a similar approach [185] in which the authors employ regular Bayesian Networks (BN) with simple structures to tackle situation assessment in Intelligent Transportation Systems. However, BN-based method is limited, and not empirical in our case, since the overall structure of BN grows due to the nature of VANET, and this yields to an exponential growth in the required inference time. Here, we compare our method, using both regular and hybrid inference algorithm, with that of [185], by increasing the number of traffic vehicles in VANET and measuring the inference time accordingly. The result is demonstrated in Figure 4.15. [6]

Figure 4.15 clearly shows that both regular and hybrid MEBN inference algorithms outperform the BN one. The main reason is related to the size of network as the number

[6] All the simulations were run on a platform with a Core i7 3.4 GHz processing power and 16GBs of RAM.

Table 4.6: Average of state estimation, in different scenarios, when a collision occurs, and the corresponding false alarms rates for different window sizes. The average of all scenarios is also shown in the last row.

	CTL State Estimation			**False Alarms per Window Size**					
Scenario	**Low**	**Medium**	**High**	**W:0**	**W:1**	**W:2**	**W:3**	**W:4**	**W:5**
1	0.08	0.1	0.82	0.15	0.2	0.26	0.38	0.4	0.41
2	0.07	0.09	0.84	0.18	0.18	0.18	0.2	0.23	0.3
3	0.09	0.1	0.8	0.27	0.27	0.21	0.27	0.25	0.27
4	0.07	0.08	0.85	0.06	0.06	0.06	0.06	0.06	0.06
5	0.09	0.1	0.81	0.22	0.22	0.24	0.25	0.25	0.26
6	0.08	0.1	0.82	0.17	0.17	0.19	0.31	0.31	0.32
7	0.06	0.08	0.86	0.03	0.11	0.19	0.22	0.23	0.23
8	0.11	0.1	0.79	0.31	0.26	0.3	0.31	0.35	0.35
9	0.09	0.1	0.81	0.16	0.15	0.15	0.15	0.16	0.17
10	0.07	0.08	0.85	0.13	0.14	0.13	0.13	0.13	0.13
11	0.09	0.1	0.81	0.11	0.1	0.12	0.11	0.13	0.13
12	0.11	0.1	0.79	0.14	0.14	0.13	0.12	0.13	0.13
13	0.08	0.09	0.83	0.15	0.16	0.23	0.27	0.28	0.3
14	0.08	0.09	0.83	0.13	0.13	0.13	0.12	0.12	0.13
15	0.07	0.09	0.85	0.06	0.06	0.06	0.11	0.12	0.12
16	0.09	0.1	0.81	0.08	0.08	0.08	0.08	0.09	0.1
17	0.08	0.1	0.82	0.09	0.09	0.09	0.09	0.14	0.15
18	0.09	0.1	0.81	0.18	0.18	0.18	0.2	0.22	0.45
19	0.1	0.11	0.79	0.16	0.16	0.16	0.17	0.18	0.19
20	0.1	0.11	0.79	0.2	0.2	0.18	0.13	0.13	0.2
Average	0.08	0.1	0.82	0.15	0.15	0.16	0.18	0.2	0.22

of vehicles in VANET increases. Since there is no extra information about the semantic relations between the entities constituting the body of the network in BN, all of the nodes should be added to the network at its initialization step. This results in a BN wherein the number of traffic vehicles becomes very large, and consequently yields inefficient inference time. Such behavior is not observed in the regular MEBN, as it recreates the structure of network at each time step with the help of semantic relations' extra information. Therefore, the nodes representing the movement of traffic vehicles are restricted only to those residing in V2V communication coverage zone of the ego vehicle. Inference time is even better in the hybrid MEBN inference algorithm, because instead of recreating the network at each

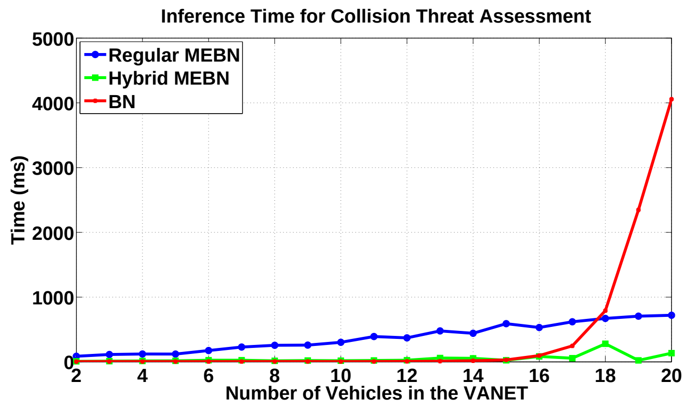


Figure 4.15: The inference time of collision threat assessment for regular and hybrid MEBN, and regular BN.

time slot, it utilizes the same structure for the same set of evidences, and only recreates the network upon addition of a new evidence, or removal of an old one.

Another important performance difference between MEBN and BN is their network construction time. In MEBN, random variables are contextually grouped together using the FOL ordinary variables. Therefore, the network is re-constructed (random variables are instantiated) at run-time, and based on the available entities. However, the topology of the network is constant in BN as the number of random variables and their types are fixed throughout the whole process. Although this may be deemed a drawback for MEBN, we observed the network construction time to be as small as 200 milliseconds in the worst case scenario, which can be simply ignored.

4.6 Summary

In this chapter, the theoretical foundation of Fuzzy Multi-Entity Bayesian Networks (MEBN) was comprehensively discussed. Fuzzy-MEBN handles the semantics analysis by making use of First-order Fuzzy Logic (FOFL), and manages uncertainty by employing Fuzzy Bayesian Networks (FBN) as its causal reasoning core. In other words, Fuzzy-MEBN adds the imprecise knowledge representation and reasoning capability to the conventional MEBN by incorporating Fuzzy logic into both its semantics and causal sides.

The second novelty of this chapter was an algorithm for soft data matching using Fuzzy-MEBN and semantic analysis concepts. To provide Fuzzy-MEBN with soft data handling capability, we first defined the basis of its underlying Fuzzy Bayesian Networks, and accordingly, modified the Junction Tree inference algorithm. Moreover, the soft data was analyzed through topic-to-topic, entity-to-topic, and entity-to-entity semantic analysis algorithms to find its corresponding entity class, context, and state, respectively.

To evaluate the applicability of our model, we proposed a Traffic Situation Assessment (TSA) framework for the Internet of Cars, which applies high-level information fusion techniques to perform situation and threat assessment by utilizing Fuzzy-MEBN. In the modelling procedure, the inattention-related entities along with their causal and semantic relationships were identified first, and then were modelled in specific contexts using the proposed framework. In order to show the capabilities of the framework, we also implemented a collision warning system based on the TSA to measure the likelihood of a vehicle being in a near-collision situation while using a wide range of information sources made available through VANET platform.

Two distinct groups of driving scenarios were designed and tested on the proposed system, and our simulation results helped to demonstrate the capability of TSA in achieving situation assessment on the road. Results showed that proper implementation of Fuzzy MEBN enables imprecise knowledge representation and reasoning, which can be used to tackle many real world applications such as collision warning in Vehicular Ad-hoc Networks area. Besides, the proposed model showed that it is able to tackle the inherent ambiguity in the natural language of soft data by efficiently performing soft data matching, and casting a positive impact in more accurate situation/threat assessment.

The future of this research work will be the incorporation of imprecise context-based ontologies, and analysis of OSINT data. Furthermore, automatic learning of the Fuzzy-MTheory structure and the formation of Fuzzy-MFrags can be seen as future activities. Besides, considering the construction of more complex situations in which the likelihood of both entities and their relationships are taken into account, and addressing the idea situation distance definition, as introduced in [22], can also be considered as the seed for pertinent research work.

Chapter 5

Game Theoretic Impact Assessment and Decision Making

The best protection against road accidents is to prevent them. However, preventing crashes is a challenging issue as the traffic density is increasing spectacularly. A remarkable number of crashes could be prevented if the driver is warned at least one-half second prior to an accident. Therefore, future active safety systems are urged to assess the danger involved in some situations and to take the convenient maneuvers, appropriately.

This chapter tackles the Impact Assessment (IA) and Decision Making (DM) problems in the Internet of Cars context. A novel game-theoretic version of Fuzzy Multi-Entity Bayesian Network, called Active Fuzzy-MEBN (for short ATFY-MEBN), is proposed. Our approach inherits most of the advantages of game theory in enumerating different possibilities of action profiles, and finding an equilibrium.

To demonstrate the capabilities of ATFY-MEBN, a Traffic Impact Assessment (TIA) and Decision Making (DM) model is implemented. In fact, TIA accept the the situation assessment results, generated by the Traffic Situation Assessment (TSA) framework, as its inputs, and adaptively aims to find the best action by hypothesizing the possible future situations and choosing the one with maximum profit. Our experiments on a collision warning system exhibit promising results in IA and DM in the IoC.

5.1 Introduction

Driving is a complex behavior that is composed of various fast-evolving situations. These situations are generally bound to a set of actions that directly impact their future state. Therefore, it is a difficult and crucial task to predict the future of the situations, asses their impacts, and take the optimal action based on that. Therefore, Impact Assessment (IA) and Decision Making (DM) in the context of connected cars are challenging problems that needs proper solutions.

As we already mentioned in Chapter 2, prediction and risk analysis are the main steps of IA. An IA model obtains the previously assessed situations and their threats (the situations capabilities and intents) to calculate the likelihood of future hypothesized situations. Therefore, any approach capable of generating a set of hypothesized situations, given the current situations and their threats, can be potentially used as an IA model. Among several approaches proposed to tackle IA in the context of connected cars, Probabilistic Graphical Models (PGMs) are the most promising methods (see Chapter 2).

Along with IA, Decision Making (DM) is also an important part of the projection phase of a SAW model [32, 35, 33]. An efficient DM model should evaluate the hypothesized situations by measuring the benefit of the actions that can be taken on them. Generally, the evaluation metric is defined either based on a domain expert interpretation, or statistically compiled relevant datasets. Furthermore, involvement of the other intelligent decision makers, *i.e.*, agents, is another important factor that can influence the resulting actions of a DM system. Indeed, the actions of the other agents can be either in favor, or against our utility value. Such agents are commonly referred to as *cooperative* and *adversarial* agents.

Game Theory is internally a decision making scheme, which is provided with a set of players, states, transition function, and a payoff function. A conventional game model contains a decision space that is formed from the actions of individual players. Therefore, for a specific situation wherein two or more active entities are involved, the decision space of the game model is all of combinations of their actions. Furthermore, the best action to be taken would be the combination that satisfies both the consensus and the individual utilities which is measured by a payoff function [110, 212].

According to our review in Chapter 2, Fuzzy Multi-Entity Bayesian Network (MEBN) have the maximum number of SAW-based features. However, it cannot handle major prediction aspects. In this chapter, we propose a new tier for Fuzzy-MEBN that is laid to monitor the underlying assessed situations of interest, evaluate their impact, and finally, suggest optimal actions. The new Fuzzy-MEBN model introduces the concepts of player entity classes and situation specific games, as well as some new game-theoretic components such as game and action nodes. Therefore, we name it Active Fuzzy-MEBN, or for short ATFY-MEBN.

5.2 Background and Related Work

This section introduces relevant research work on IA, and incorporation of game theory to solve IA problems. Moreover, the required theoretical background of game theory on different forms of games, and game solution techniques are briefly explained.

5.2.1 Related Work on Game Theoretic Impact Assessment

Higher layers of JDL model call for situation prediction. Since situations, in the context of VANETs, are developed by actions performed by drivers, vehicles, pedestrians, *etc.*, game-theoretic techniques can provide an improved degree of prediction. Therefore, many efforts in this direction have been proposed in the literature.

The authors in [50] introduce a game-theoretic information fusion approach to tackle threat and impact assessment in military domain applications. The core of the proposed framework is based on Markov (stochastic) game theory. Furthermore, Hierarchical Entity Aggregation (HEA), and Hierarchical Task Networks (HTN) are also used in different levels of the framework. Crucial segments of the Markov game, namely, Players, State Space, Decision, Transition Rule, and Payoff Functions, along with different Strategies (Pure Nash, Mixed Nash, and Correlated Equilibria) are studied.

Tang *et al.* in [212] propose a model for threat and situation assessment in cyber insider scenarios. They use dynamic Bayesian Networks (BN) as their information fusion module, and then couple it with a game module that uses Quantal Response Equilibrium (QRE) as its strategy method. The main intuition behind choosing QRE is its capability in modeling rationality of players. Their proposed algorithm performs efficiently in terms of convergence and precision. However, it suffers from additional computational cost.

A Threat Assessment (TA) model is proposed by Aoude *et al.* that is based on a combination of game theory and Rapidly-exploring Random Trees (RRT). The authors tackle the problem of collision avoidance at intersections containing a host vehicle, and an erratic traffic vehicle, which is similar to an adversary unit in a game model. The algorithm is split into two modes: exploration and pursuit. In exploration mode, regular RRT is employed and the state-space is explored to find reachable states for the erratic car. Moreover, in the pursuit mode, the sampling of RRT is biased towards where the host vehicle is going. Threat assessment is then calculated based on the Time To Collision (TTC) [113] metric for each expanded node in RRT.

A prediction and planning framework for collision avoidance is introduced in a technical report by Broadhurst *et al.* [43]. The authors determine the main entities involved in prediction, and the most important elements of planning. Furthermore, the main aspects of a game such as states, actions, strategies and payoff functions are formulated. In the experiments, sequential game playing is employed and it is assumed that the states are updated in turn-based manner.

An information fusion game component based on dynamic Bayesian Networks and Bayesian Game Theory is proposed by [44]. Goals, utilities, and decisions are represented by a graphical model that is further used by a game component. Moreover, the agents are deemed to interact with partial knowledge about the other agents' strategies. Henceforth, a Bayesian game is used to model high-level agent interaction to improve situation awareness.

Although game theory is widely used for the analysis of impact and threat assessment, few studies focus on road safety. Pioneering works discuss strategy interactions between one errant driver and one victim car. Unlike earlier studies, our proposed technique for IA will profit from different types of communication in VANETs, and adopt an adaptive process asses a situation and its impact, where there might be multiple road users.

5.2.2 Game Theory: Theoretical Background

A conventional game $\mathcal{G}$ in general is shown by a 5-tuple $\mathcal{G} = \langle \mathcal{P}, \mathcal{S}, \mathcal{D}, \mathcal{R}, f \rangle$, wherein $\mathcal{P} = \{1, 2, \cdots, N\}$ is the set of players, $\mathcal{S}$ is the set of states, $\mathcal{D} = \mathcal{D}_1 \times \mathcal{D}_2 \times \cdots \times \mathcal{D}_N$ is called the decision space and is created from the actions of each individual player $\mathcal{D}_i = \{a_1, a_2, \cdots, a_m\}$, $\mathcal{R} : \mathcal{S} \times \mathcal{D} \to \Delta(\mathcal{S})$ is the transition function that calculates the likelihood of residing in each state after taking decision instance $d_i \in \mathcal{D}$ when observing world state $s_i \in \mathcal{S}$, and finally $\{ : \mathcal{S} \times \mathcal{D} \to \mathbb{R}^N$ is the payoff function that awards/punishes the players upon perceiving world state $s_i \in \mathcal{S}$ and taking decision instance $d_i \in \mathcal{D}$.

A strategy s_p for a player $p \in P$ is defined as a probability distribution π over the actions a_i that p takes, when dealing with different instances of games. Therefore, $\sum_i \pi(a_i) = 1$. A strategy s_p is called pure strategy if $\pi(a_k) = 1$ and $\forall_{j \neq k} \pi(a_j) = 0$. Otherwise, s_p is called a mixed strategy.

Two main forms of games are: 1. strategic games, and 2. extensive games with perfect/imperfect information[160].

Strategic (Normal Form) Games

In strategic games, it is assumed that the players know which actions their opponents can take and what the outcomes will be, but they are not informed about which action their opponents are willing to take. Alternatively, one ca assumed that actions are taken simultaneously. A strategic game $\mathcal{G}_N$ is a 3-tuple in abstract form, wherein $\mathcal{G}_N = \langle \mathcal{P}, \mathcal{D}, f \rangle$. The definition of the consisting components are the same as above. Normal Form Games (NFGs) are the most well-known strategic type games. It is important to know that in NFGs, a players utility is not only dependant on his own utility, but also on the strategies played by its opponents. A player p_i is rational if and only if it tries to maximize the expected value of its payoffs. NFGs are usually represented by a hyper-matrix.

Extensive Form Games

Knowing the complete information about the game is the main difference between the strategic games and Extensive Form Games (EFGs). In EFGs, each player knows when to play (and when other players play), what actions are available to them (and what actions are available to their opponents), and where their actions lead them to (and where those of the other players do). An extensive form game $\mathcal{G}_E$ is a 5-tuple such that $\mathcal{G}_E = \langle \mathcal{P}, mathcalT, \mathcal{S}, \mathcal{R}, f \rangle$, wherein $\mathcal{P}$,$\mathcal{S}$, $\mathcal{R}$, and f are defined same as above, and $\mathcal{T}$ is the tree representation of $\mathcal{G}_E$. Having complete or partial information regarding the transition rules in $\mathcal{R}$, divides the EFGs into two sub-groups of EFGs with perfect, and imperfect information, respectively.

Solving Games

Common game solution algorithms are Dominant Strategy Equilibrium (DSE), and Nash Equilibrium (NE). An equilibrium, which is also known as a strategy profile, is a combination of each player's strategies, from which they will not deviate. Such strategies are called dominant (best response) strategies, and are defined as follows.

A strategy s_p^* *strictly* dominates s_p if and only if $f_p(s_p^*, s_{-p}) > f_p(s_p, s_{-p})$, wherein $s_{-p} \in S_{-p}$ are the strategies played by all the players except for p, and f_p is the payoff function of p. s_p^* *weakly* dominates s_p if and only if $f_p(s_p^*, s_{-p}) \geq f_p(s_p, s_{-p})$.

Similarly, for any player p, a strategy s_p^{BR} is a *best response* to s_{-p} if and only if $f_p(s_p^{BR}, s_{-p}) > f_p(s_p, s_{-p}), \forall s_p \in S_p$. Note that this is different from the dominant strategy definition, in which s_{-p} is not fixed.

Finally, a strategy profile $(s_1^{NE}, s_2^{NE}, \cdots, s_P^{NE})$ is a *Nash equilibrium* if and only if s_i^{NE} is a best response to s_{-i}^{NE}.

5.3 Active Fuzzy Multi-Entity Bayesian Networks

ATFY-MEBN is a novel 2-tier model that is designed on top of Fuzzy-MEBN to enable IA and DM. While ATFY-MEBN model shares most of the Fuzzy-MEBN language and components, it introduces some modifications in the language, and a number of new game-theoretic components. Besides, two separate algorithms for generating Situation-Specific Normal Form Games, and Situation-Specific Active Fuzzy Bayesian Networks are presented.

5.3.1 ATFY-MEBN Language Specifications

The language of ATFY-MEBN is very similar to what we introduced in Chapter 4, and is mainly based on the original MEBN language proposed by Laskey [125]. The following explains just the differences between ATFY-MEBN and Fuzzy-MEBN.

- *Phenomenal constant symbols:* Constants are represented by fuzzy sets with just a single member, *i.e.*, fuzzy singletons. Constant names may contain both letters and number, but must start with an uppercase letter, and should be followed by a real-valued membership degree subscript within range $[0, 1]$ *e.g.*, $\text{Vehicle}_{0.85}$, $\text{Environment}_{1.0}$. The constant symbol $\text{Action}_{1.0}$ is pre-defined and fixed by the language to specify actions.
- *Unique identifier symbols:* The entities are assigned a unique identifier symbol that are annotated with a fuzzy membership degree, and are arranged in one of the groups below:
    - *Action identifier symbols:* Action identifier symbols are shown by the set Λ, and are used to label the domain-specific actions taken by the entities defined in the. Action identifier symbols are alphanumeric symbols starting with an at-sign, *e.g.*, $@A1$, $@A2$.
- *Action binders:* The action binders $\mathcal{B}$ are simply mapping functions that bind phenomenal constant symbols to a subset of action identifier symbols. If such a binding exists, then the corresponding specific phenomenal constant symbol is called a player.

- *Domain-specific game variable symbols:* Analogous to random variable names, game variable symbols are also alphanumeric strings beginning with a capital letter character. It is necessary for the arguments in game variables to have an action binder, *i.e.*, the arguments should be players. Moreover, the number of arguments for each game variable is fixed to two, showing the fact that at least two players are required for the game to be played. In this setting, the first variable v_i is automatically associated with the main player, and the second variable v_{-i}, which is actually a vector of variables, is assigned to the other players. All the possible values of game variables are found by finding the cross product of the actions in each binding. In other words, $G : \{v_i, v_{-i}\} \to \prod_i \mathcal{B}(v_i)$, wherein G is the game variable, and $\mathcal{B}(v_i)$ is the set of actions bound to v_i.

- *Domain-specific action variable symbols:* Action variable names are also alphanumeric strings with a capitalized first letter. These variables have only one argument that needs to have an action binder. The possible values of action variables are defined as a subset of actions bound to its underlying argument, *i.e.*, $A : \{v_i\} \to \Lambda' \subseteq \Lambda$.

5.3.2 ATFY-MEBN Components and Structure

In the following, we propose our novel 2-tier architecture for ATFY-MEBN that ensures separating game components from those of conventional Fuzzy MEBN. In this architecture, the ATFY-MFrags that contain at least one game component belongs to tier 1, and the rest lay in tier 0. In other words, ATFY-MFrags determine to which tier different components belong. For all the definitions below, assume that T_0 and T_1 refer to tiers 0 and 1, respectively.

Definition 4. *The fuzzy context nodes $C_i(\Omega) \in \mathcal{C}$ are the graphical representations of FOFL expressions that include the ordinary variables in the set Ω. The output of $C_i(\Omega)$ ranges from 0 to 1, and reflects the truth value of its underlying FOFL expression. Context nodes are usable on both T_0 and T_1.*

Definition 5. *The input nodes $I_i(\Phi) \in \mathcal{I}$ are defined as placeholders for the fuzzy resident nodes $R(\Phi)$, wherein Φ is the set of the ordinary variables in the original fuzzy resident node. Input nodes can be created on both T_0 and T_1. An input node is an inter-tier portal if it is created on T_1, and its fuzzy resident node is on T_0.*

Definition 6. *The output nodes $O_i(\psi) \in \mathcal{O}$ are defined as placeholders for the action nodes $A(\psi)$, wherein ψ is the ordinary variables residing in the original action node. Output nodes* *can be created on both T_0 and T_1. However, they play the role of a portal between two tiers, if they are in an ATFY-MFrag on T_0 (they point to an action node, which is on T_1).*

Definition 7. *Let us assume that $\Psi = \{\psi_j, \psi_{-j}\}$ is the set of ordinary variables, wherein ψ_j is the main player and ψ_{-j} are all its opponents. It is necessary that all the ordinary variables in Ψ have non-empty action binders. Moreover, let $\mathcal{I}_i$ and $\mathcal{C}_i$ be subsets of input nodes and fuzzy context nodes, respectively. Therefore, the game nodes $M_i(\Psi|\mathcal{I}_i, \mathcal{C}_i) \in \mathcal{M}$ are defined as the graphical representations of the games whose players existence is conditioned on $\mathcal{C}_i$. The payoff function of M_i is conditioned on the subset of input nodes $\mathcal{I}_i$, and is defined through aggregating the results of the input nodes in $\mathcal{I}_i$. All the game nodes are on T_1.*

Definition 8. *An action node is the graphical representation of a the random variable $A_i(\psi|\mathcal{R}_i, \mathcal{I}_i, \mathcal{M}_i)$ that is the probability distribution over the actions that ψ can take, given a subset of fuzzy resident nodes $\mathcal{R}_i$, and a subset of input nodes $\mathcal{I}_i$. Moreover, the game nodes playing the actions of A_i are included in $\mathcal{M}_i$. All the action nodes are on T_1.*

Definition 9. *Let us assume that $\alpha \in \Lambda$ is an action identifier. Then, an influence set $L(\alpha)$, is the set of fuzzy resident nodes whose states are affected if action α is taken.*

Definition 10. *A fuzzy resident node is a graphical representation of the random variable $R_i(\Theta|\mathcal{R}_i, \mathcal{I}_i, \mathcal{C}_i, \mathcal{O}_i)$ that is expressed through a set of ordinary variables Θ, and is conditioned on the values of a subset of other fuzzy resident nodes $\mathcal{R}_i \subset \mathcal{R}$ where $R_i \notin \mathcal{R}_i$, a subset of input nodes $\mathcal{I}_i \subset \mathcal{I}$, and a subset of fuzzy context nodes $\mathcal{C}_i \subset \mathcal{C}$. The output of R_i is a probability distribution over its states. It is notable that $0 \leq |\mathcal{I}_i| \leq |\mathcal{I}|$ and $0 \leq |\mathcal{R}_i| \leq |\mathcal{R}| - 1$. Besides, for any $0 \leq i, j \leq |\mathcal{R}| - 1$, where $i \neq j$, $\mathcal{R}_i$ and $\mathcal{R}_j$ are not necessarily disjoint, i.e., fuzzy resident nodes can have multiple children. Furthermore, $|\Theta| \leq |\mathcal{C}_i| \leq |\mathcal{C}| + |\Theta|$, showing that $\mathcal{C}_i$ needs to contain the at least $|\Theta|$ number of fuzzy context nodes that demonstrate the entity assignments of ordinary variables in Θ. Depending on their underlying ATFY-MFrag, fuzzy resident nodes can be on both T_0 and T_1.*

The MFrags in ATFY-MEBN are called ATFY-MFrags and are defined as below.

Definition 11. *An ATFY-MFrag $\mathcal{F} = (\mathcal{R}, \mathcal{C}, \mathcal{I}, \mathcal{A}, \mathcal{M}, \mathcal{G}_{0,1}, \mathcal{D}_{0,1}, \mathcal{S})$ is a 2-tier 8-tuple wherein $\mathcal{R}$, $\mathcal{C}$, and $\mathcal{I}$, are respectively the conventional sets of resident, context, and input nodes that reside on tier 0. Furthermore, $\mathcal{A}$ and $\mathcal{M}$ are action nodes and game nodes sets, which are new in ATFY-MFrag and are laid on tier 1. Moreover, $\mathcal{G}_{0,1}$ and $\mathcal{D}_{0,1}$ are respectively the graph representation of $\mathcal{F}$ and the probability distributions defined for $\mathcal{R}$ and $\mathcal{A}$.* [1] *The fuzzy rules are also defined in the fuzzy rule-set $\mathcal{S}$.*

[1]The sub-scripts in $\mathcal{G}_{0,1}$ and $\mathcal{D}_{0,1}$ show the tier index

A number of important characteristics of $\mathcal{F}$ are as follows:

- $\mathcal{C}$, $\mathcal{R}$, and $\mathcal{I}$ are pairwise disjoint sets
- $\mathcal{G}_{0,1} = \langle V_0 \cup V_1, E_0 \cup E_1 \cup E_c \rangle$ wherein $V_0 = \mathcal{I} \cup \mathcal{R}$ and $V_1 = \mathcal{A} \cup \mathcal{M}$ are the sets of graph vertices on each tier with corresponding edges specified in E_0 and E_1, respectively. Moreover E_c contains the cross edges that originate from one tier and rest on the other.
- Context value assignment terms in $\mathcal{C}$ are used for enforcing constraints under which the local distributions apply

5.3.3 Situation-Specific Active Fuzzy Bayesian Networks

The Situation Specific Fuzzy Bayesian Networks (SSFBNs) which are enhanced with action and game nodes are called Situation Specific Active Fuzzy Bayesian Networks (Active SSFBNs). An Active SSFBN is constructed when finding the probability of the query Q happening given a set of evidences E, *i.e.*, $p(Q|E)$. Therefore, the whole process starts by submitting the query Q that is an instantiated resident node, say $R_1(\Theta_1)$, with a set of bound ordinary variables (see [125] for the definition of binding sets). In addition, all the present observations, say $E = /e_1, e_2, \cdots, e_M/$, are also added to the network. Furthermore, the parents of these nodes are recursively analyzed and then imported to the network. In fact, parents are created following exactly the same instructions in constructing an SSBN [143]. As long as the parent nodes expanded are resident nodes, they are laid on T_0. However, as it is also mentioned in the definition of resident nodes in ATFY-MEBN, there might be cases wherein an output node, say O_p, is the parent of a resident node, *i.e.*, R_q. Such configurations happens when R_q is in the influence set of at least on of the actions defined in the action node host of O_p. This results to an inter-portal link between R_q (residing on T_0) and the action node A_p (laid on T_1), to which O_p points. Subsequently, the algorithm continues by creating the parents of the currently unexpanded nodes including A_p. When expanding the network from A_p, all of its parents that are either resident or input nodes are created as before, but, those which are game nodes are treated differently.

To create the parent game node M_p, all of the context nodes on which M_p is conditioned should be satisfied initially. The satisfiability check is done given the bindings of the ordinary variables present in the ATFY-MFrag in which M_p resides. This normally should results in instantiating the game with its main player ψ_i and all the other opponents ψ_{-i} using the outputs of context nodes. However, if the total number of players is less than 2, then the game cannot be created. Game nodes are also conditioned on a given subset of input node parents that serve as defining the payoff function. Similarly, the resident nodes which are pointed by these input nodes are expanded following the same instructions as before. The result of this stage is a situation specific game whose players along with their actions are determined based on the current situation.

The whole process produces a 2-tier Active SSFBN that is capable of assessing the impact of situations by playing games. The Active SSFBN construction algorithm is presented in Algorithm 3. This algorithm is very similar to SSFBN construction algorithm except for lines 12 and 7 where a node and its parents are created. Conclusively, the

Algorithm 3 Situation-Specific Active Bayesian Network Construction Algorithm

1: **procedure** GENERATE-SSATFYBN($Q(\Theta)$,E)
2: $\quad V \leftarrow Q \cup E$
3: $\quad V_c \leftarrow \varnothing$
4: $\quad G \leftarrow \varnothing$
5: $\quad$ **while** $V \neq \varnothing$ **do**
6: $\quad\quad v \leftarrow V - \{v\}$
7: $\quad\quad \Pi \leftarrow$ CREATE-PARENTS(v)
8: $\quad\quad$ **for all** $\pi \in \Pi$ **do**
9: $\quad\quad\quad$ **if** $pi \notin V_c$ **then**
10: $\quad\quad\quad\quad V \leftarrow V \cup \pi$
11: $\quad\quad V_c \leftarrow V_c \cup v$
12: $\quad\quad G \leftarrow$ CREATE-NODE(v)
13: $\quad$ **return** G

output of Algorithm 3 is a 2-tier graph that contains the conventional situation specific fuzzy Bayesian network on tier 0, and a set of game components on tier 1. These outputs are used to predict the future states of the situation modelled on tier 0.

In the next section, we explain how actual games are instantiated from the game nodes, and how they are used to assess the impact of the current situation.

5.3.4 Situation-Specific Normal Form Games

The games are played in a distributed manner and as a separate step after all the nodes are created, and the network is initialized. The result of the game, which is a strategy, will then be utilized to predict the impact of the current situation. Depending on what type of game being created, different approaches may be chosen. Here, we propose a novel distributed approach to generate a Situation Specific Normal Form Game.

In a nutshell, we first fix the main player entity, and then determine all the other opponents by exploring the knowledge base and checking satisfiability of the constraints imposed by the context nodes. The next steps would be generating all the possible action profiles, and calculating the payoffs of each player. The later needs to be done in a distributed manner, since the main player entity has partial knowledge about its opponents situations. Upon setting all the payoffs for all the action profiles, a game solution algorithm such as Nash equilibrium is performed to extract the best strategies. It is notable that all the players are assumed to be rational agents, which means that they take the actions that ultimately maximize their payoff. The body of SSNFG generation algorithm is presented in Algorithm 4.

Algorithm 4 Situation-Specific Normal Form Game (SSNFG) Generation Algorithm

1: **procedure** GENERATE-SSNFG($M_i(\psi_{-i}|\psi_i = \epsilon_i, \mathcal{I}_i, \mathcal{C}_i)$,$A''$,$f$)
2: $\quad B' \leftarrow \varnothing$
3: $\quad B' \leftarrow B' \cup (\psi_i, \epsilon_i)$
4: $\quad B' \leftarrow B' \cup$ EXTRACT-VALID-PLAYERS(ψ_{-i},$\mathcal{C}_i$))
5: $\quad A' \leftarrow$ GENERATE-ACTION-PROFILES(B',A'')
6: $\quad \Sigma \leftarrow$ CALCULATE-PAYOFFS(B',ψ_{-i},A',$\mathcal{I}_i$,f)
7: $\quad (A'_k, L_{A'_k}) \leftarrow$CALCULATE-EQUILIBRIUM($\Sigma$)
8: **return** $(A'_k, L_{A'_k})$

In Algorithm 4, it is assumed that the main player is already known. The unique entity identifier symbol assigned to ψ_i usually comes from a child action node that is an ancestor of a query resident node. Figure 5.1 shows an example. Another input of the algorithm is the actions binding set A'', or (partial) action profile, that is a subset of all player-action pairs. In other words, A'' can be defined as below:

$$A'' \subseteq \{(\psi, \alpha)|\psi \in P \text{ and } \alpha \in A(\psi)\} \tag{5.1}$$

wherein P is the set of all players participating in the current game, and $A(\psi)$ is the set of all possible actions that the player ψ can take. If A'' contains the player-action pairs of all the players, then it is an action profile. Otherwise, we refer to A'' as partial action profile.

Having the main player entity fixed, other valid opponents are extracted at line 4, and are stored in the players binding set B'. Afterwards and at line 5 of Algorithm 4, all the possible action profiles are generated based on the players binding set B', and the partial

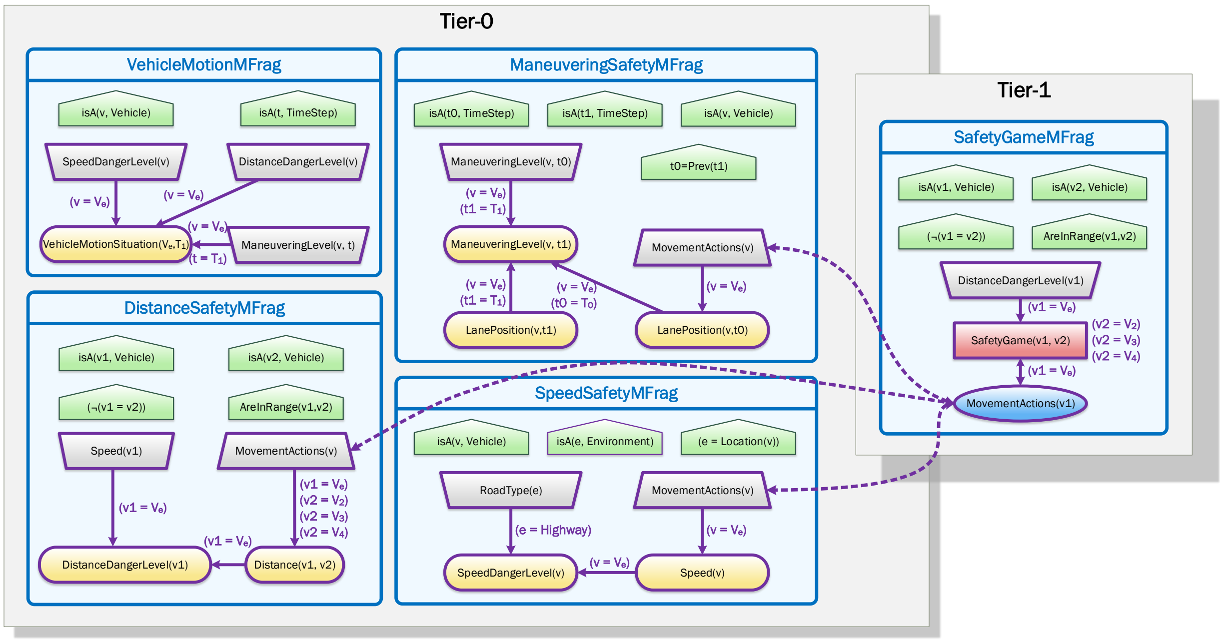


Figure 5.1: The instantiation of ordinary variables based on the given query and the knowledge base

action profile A''. Furthermore, the game table is constructed at line 6, and finally the optimal strategy is calculated by finding the Nash equilibrium at line 7. The following describes the last four steps in mode details.

Player Entity Extraction

The opponent players entities ψ_{-i} of an SSNFG node are determined using a series of steps that are presented in Algorithm 5. The entities associated to these players are the unique entity identifier symbols that belong to a player class (line 4), and all their consistency constraints defined by the context nodes are satisfied (lines 5 to 8). It is notable that the binding set B' is generated using the set of observed unique entity identifiers as well as those stored in the knowledge base (see line 2).

Action Profile Generation

All the possible action profiles are generated based on the given player binding set B', created on the previous step, and an actions binding set A'', *a.k.a.* partial action profile

Algorithm 5 Opponent player extraction for the current context

procedure EXTRACT-VALID-PLAYERS(ψ_{-i},$\mathcal{C}_i$))
    $B' \leftarrow$ EXTRACT-BINDINGS(ψ_{-i})
    for all $(\psi^j_{-i}, \epsilon_j) \in B'$ **do**
        if IS-PLAYER(ϵ_j) **then**
            for all $c \in \mathcal{C}_i$ **do**
                if NOT-SATISFIED($c(\Omega|\epsilon_j)$) **then**
                    $B' \leftarrow B' - \{(\psi^j_{-i}, \epsilon_j)\}$
                    go to 3
    return B'

(see Equation 5.1) that determines which players know which actions they should take. For sake of simplicity, let us assume that all the players belong to the same type, meaning that all the available actions to them are the same. The actions set of the players are determined by the corresponding action nodes $A_i(\psi_i)$ and $A_{-i}(\psi_{-i})$ linked to the game node M_i.

Depending on the size of A'', two cases might happen. If $A'' = \varnothing$, then the *maximum* total number of action profiles is:

$$|A(\psi_i)| \prod_{j \in \psi_{-i}} |A(\psi_j)| \tag{5.2}$$

wherein $A(\psi_j)$s are the actions set of the players ψ_j that are in the opponents set ψ_{-i}. However, if the actions binding set $A'' \neq \varnothing$ (A'' contains the pre-fixed actions that some of the players take), then the total number of action profiles will be:

$$H^1_{A''}(\psi_i) \prod_{j \in \psi_{-i}} H^1_{A''}(\psi_j) \tag{5.3}$$

wherein H is an indicator function, and is defined as:

$$H^1_{A''}(\psi_j) = \begin{cases} 1 & \text{if } \exists_{(x,y) \in A''} \;\; s.t. \;\; x = \psi_j \\ |A(\psi_j)| & \text{o.w.} \end{cases} \tag{5.4}$$

Briefly speaking, fixing the actions of a subset of players by the partial action profile A'' helps to reduce size of the action profiles set that consequently leads to the reduction in the size of the game table.

Theorem 1. *If $A'' = \{(\psi_1, \alpha_1), (\psi_2, \alpha_2), \cdots, (\psi_K, \alpha_K)\}$, then the total number of action profiles for a game with N players, where $K \leq N$, is reduced by a factor of at least 2^K, assuming that each player can take the minimum of two actions.*

Proof. The proof is very straight-forward. Let us assume that $A'' = \varnothing$. Therefore, assuming that each player can take at least two actions, we will have: $\forall_\psi\ 2 \leq |A(\psi)|$. Plugging-in this equation into Equation (5.2) yields:

$$2^N \leq |A(\psi_i)| \prod_{j \in \psi_{-i}} |A(\psi_j)| \tag{5.5}$$

in which N is the total number of players. Furthermore, for an arbitrary $A'' = \{(\psi_1, \alpha_1), (\psi_2, \alpha_2), \cdots, (\psi_K, \alpha_K)\}$, using Equation 5.3 results in:

$$\prod_{j=1}^{N} H^1_{A''}(\psi_j) = \underbrace{1 \times 1 \times \cdots \times 1}_{K} \times \overbrace{\prod_{\psi_j \notin A''(\cdot,\alpha)} |A(\psi_j)|}^{N\text{-}K \text{ products}} \tag{5.6}$$

wherein $\psi_j \notin A''(\cdot, \alpha)$ determines the players not included in the partial action profile A''. Since each player takes minimum of two actions, Equation 5.6 is bounded below by 2^{N-K}, and can be rewritten as:

$$\frac{2^N}{2^K} \leq \prod_{j=1}^{N} H^1_{A''}(\psi_j) \tag{5.7}$$

Dividing Equation 5.5 by 5.7 yields:

$$2^K \leq \frac{|A(\psi_i)| \prod_{j \in \psi_{-i}} |A(\psi_j)|}{\prod_{j=1}^{N} H^1_{A''}(\psi_j)} \tag{5.8}$$

showing that the total number of action profiles is reduced by a factor of at least 2^K. □

Payoff Calculation

The entries of a game table are created for each action profile, based on the input nodes $\mathcal{I}_i$ and an arbitrary aggregation function f. Therefore, the size of the game table Σ is proportional to the number of players and action profiles, which is precisely $|\Sigma| = |A'| \times |B'|$, wherein A' is the set of all action profiles, and B' is the players binding set. As it is also shown in Algorithm 6, each cell of Σ is associated with an action profile

Algorithm 6 Payoff calculation for each action profile

1: **procedure** CALCULATE-PAYOFFS(B',A',$\mathcal{I}_i$,f)
2: $\quad\Sigma[\,] \leftarrow \varnothing$
3: $\quad$**for all** $A'_j \in$ ACTION-PROFILES(A') **do**
4: $\qquad\Sigma[A'_j] \leftarrow f($PREDICT-SITUATION$(\mathcal{I}_i, A'_j))$
5: $\quad$**return** Σ

$A'_j = \langle \alpha_{j1}, \alpha_{j2} \cdots \alpha_{j|B'|} \rangle$, and is set with a vector containing the payoff of each player (see line 4). This is where we introduce two different types of situation prediction algorithm, namely, Type-0 (centralized approach) and Type-1 (distributed approach).

In the Type-0 situation prediction algorithm, it is naively assumed that the main player knows how the payoff of its opponents change when the actions in an arbitrary action profile A'_j are taken. This can be pictured as a centralized approach in which the main player has complete knowledge about its opponents dynamic model, state-space, worlds, *etc.*, and therefore, can estimate their future states. Accordingly, the payoff values are calculated by measuring the impact of the situations in $\mathcal{I}_i$ upon virtually taking the actions in A'_j. Algorithm 7 shows the situation prediction instructions for Type-0 in more details.

Algorithm 7 Centralized situation prediction based on the given action profile A'_j

1: **procedure** PREDICT-SITUATION-TYPE-0($\mathcal{I}_i$,A'_j)
2: $\quad L_{A'_j} \leftarrow \varnothing$
3: $\quad$**for all** $\alpha_{ji} \in A'_j$ **do**
4: $\qquad L_{A'_j} \leftarrow$UPDATE-FINDINGS($L(a_j)$,$\alpha_{ji}$)
5: $\quad p[\,] \leftarrow \varnothing$
6: $\quad$**for all** $\psi_j \in A'_j(\cdot, \alpha)$ **do**
7: $\qquad p[\psi_j] \leftarrow p_{\psi_j}(\mathcal{I}_i | L_{A'_j})$
8: $\quad$**return** p

In Algorithm 7, the states of the entities in the influence set L of each action α_{ji} are estimated first, by calling the Update-Findings function at line 4. These entities, L_α, are then used as new evidences when assessing (predicting) the new states of the situations of interest given in $\mathcal{I}_i$ for each player ψ_j. This method performs well in the centralize architectures, where all the information regarding the participating agents is accessible. However, this is definitely a naive assumption in multi-agent systems. Therefore, we propose Type-1 situation prediction algorithm that is based on a distributed architecture and the message exchanging schemes.

In the distributed version of the situation prediction algorithm, it is assumed that the agents have limited local information, but are able to communicate through message passing protocols. Therefore, the main player does not need to predict the impact of the situations of interest for each opponent anymore. Instead, it communicates with each opponent by sending a message that contains an action profile, and receiving their corresponding payoff. VANETs are a relevant context of distributed approach. vehicles are equipped with communications technologies that allow them to exchange messages with each other . VANETs has the flexibility in seamlessly supporting both single-hop and multi-hop communications.

Once an opponent ψ_j receives an action profile A'_j from the main player ψ_i, it will use A'_j as an actions binding set when generating action profiles for its own game M_j. In other words, if P_j denotes the set of players included in the game that ψ_j plays, then two cases may happen:

1. $P_j - \{\psi_i, \psi_j\} = \varnothing$, which means that ψ_j's only opponent is ψ_i who is the one sending the partial action profile. In this case, the size of action profiles set for ψ_j's game will be reduced to 1 (see Theorem 1), and ψ_j can simply return its payoff without playing any game.

2. $P_j - \{\psi_i, \psi_j\} \neq \varnothing$, is the case in which ψ_j has some other opponents along with ψ_i. Depending on the number of opponents that ψ_i and ψ_j inclusively share, say K opponents, the size of the action profiles set will be reduced by a factor of at least 2^K (see Theorem 1). However, ψ_j still needs to play its own game M_j by calling Algorithm 4 wherein A'' is equal to the partial action profile sent by ψ_i. Finally, the calculated payoff after solving the game will be returned to the sending player.

The above steps can be repeated up to a specific number of hops or until a threshold is met (exceeding the active area of the game). The Type-1 situation prediction algorithm is presented in Algorithm 8. It is very similar to Algorithm 7, except for line 8 at which the payoff of the opponent ψ_j upon taking actions in A'_j is found by sending messages using the Ask-Payoff function. After finding the payoffs of each player for every action profile, the game is ready to be solved.

Algorithm 8 Distributed situation prediction based on the given action profile A'_j

procedure PREDICT-SITUATION-TYPE-1($\mathcal{I}_i$,A'_j)
    $L_{A'_j} \leftarrow \varnothing$
    for all $\alpha_{ji} \in A'_j$ **do**
        $L_{A'_j} \leftarrow$UPDATE-FINDINGS($L(a_j)$,α_{ji})
    $p[\,] \leftarrow \varnothing$
    $p[\psi_i] \leftarrow p_{\psi_i}(\mathcal{I}_i|L_{A'_j})$
    for all $\psi_j \in A'_j(\cdot, \alpha) \wedge \psi_j \neq \psi_i$ **do**
        $p[\psi_j] \leftarrow$ ASK-PAYOFF(ψ_i, A'_j)
    return p

Equilibrium Calculation

An SSNFG is ready be solved when the game table is generated, and all of the players payoffs for each action profile are calculated. Different game solution algorithms such as Nash equilibrium can be used to output the optimal strategy for the main player as well as the estimated payoff. A common solution to calculate the best optimal strategy is linear programming that is widely used for normal form games equilibrium calculation. As also depicted in Algorithm 4, this is the last stage of the main SSNFG generation algorithm whose results can be used to decide on the next action to be taken.

Performance Analysis

The complexity of the SSNFG generation algorithm highly depends on the number of players. Accordingly, its performance can be studied in three different phases: 1. players detection, 2. game table generation, and 3. equilibrium calculation.

Theorem 2. *The complexity of the player entity extraction algorithm (see Algorithm 5) is linear in the number of players, i.e., $O(\varphi|B'|)$, wherein B' is the set of player bindings, and $\varphi > 0$ is a constant.*

Proof. The algorithm can be analyzed in two different parts. The first one is the Extract-Bindings method where we literally examine every observed entity identifier symbol $\epsilon_j \in \mathcal{E}'$, which is valid in the current ATFY-MFrag, and assign them to opponents ordinary variables set ψ_{-i}. Therefore, this method performs in $O(|\mathcal{E}'|)$, with $\mathcal{E}'$ as the set of valid entity identifier symbols in the current ATFY-MFrag.

The second part involves deleting the player bindings whose consistency constraints are not satisfied by the subset of context nodes $\mathcal{C}_i$. Therefore, its time complexity will be:

$$O(|B'| \times |\mathcal{C}_i| \times (|\mathcal{E}'|^{|\Omega|})) \tag{5.9}$$

in which the terms $|B'|$ and $|\mathcal{C}_i|$ are caused by the outer and inner loops, respectively. Moreover, $|\mathcal{E}'|^{|\Omega|}$ is the maximum number of combinations that entity identifier symbols $\mathcal{E}'$ can be matched with the variables Ω of the context nodes. If we assume that Ω^* is the largest set of context node variables, then:

$$O(|B'| \times |\mathcal{C}_i| \times (|\mathcal{E}'|^{|\Omega|})) \preceq O(|B'| \times |\mathcal{C}_i| \times (|\mathcal{E}'|^{|\Omega^*|})) \tag{5.10}$$

Merging Equation 5.10 with $O(|\mathcal{E}'|)$, which is the performance of the first part, yields:

$$O(|\mathcal{E}'|) + O(|B'| \times |\mathcal{C}_i| \times (|\mathcal{E}'|^{|\Omega^*|})) \tag{5.11}$$

It is obvious that the first term is dominated by the second term. Besides, $|\mathcal{C}_i| \times (|\mathcal{E}'|^{|\Omega^*|})$ can be treated as a constant, say φ, because the number of context nodes and the number of entity identifier symbols assigned to the current ATFY-MFrag are fixed. Therefore, Equation 5.11 reduces to $O(\varphi|B'|)$, which is linear in the number of player bindings B'. $\square$

In the worst case scenario, the action profile generation algorithm performs in $O(|A^*|^{|B'|})$, where $|A^*|$ is the largest action set among the action sets of all players, and B' is the players binding set. As it is thoroughly discussed previously, if the partial action profile A'' is non-empty, then the total number of possible action profiles is reduced. Therefore, for $A'' \neq \varnothing$, the action profile generation algorithm performs in $O(|A^*|^{|B'|-|A''|})$. See Theorem 1 for more details.

The bottleneck of the payoff calculation algorithm is revoking the inference algorithm that performs in $O(e^w)$ for w as the width of the largest clique in the junction tree algorithm. In the centralized case, this should be done for all the players individually that makes the total complexity worse, and increases it to $O(e^w|B'|)$. However, the complexity remains at $O(e^w)$ for Type-1 payoff calculation algorithm. The inference algorithm should be called for every cell of the game table that is at most $|A'| = |A^*|^{|B'|}$. Altogether, the performance of Type-0 and Type-1 payoff calculation algorithms are $O(e^w|B'||A^*|^{|B'|}$ and $O(e^w|A^*|^{|B'|}$, respectively.

Finally, the last stage of generating a situation-specific normal form game is actually solving that game. It is shown in [57] that computing a (mixed) Nash equilibrium is of class PPAD-complete (Polynomial Parity Arguments on Directed graphs complete).

To summarize, the performance of Algorithm 4 heavily relies on the payoff calculation algorithm as it dominates the time complexities of other subroutines. In the next section, we statistically analyze the performance of our SSNFG-generation algorithm and discuss its tractability, Specifically in the context of VANETs.

5.4 Traffic Impact Assessment And Decision Making

In this section, we propose a novel Traffic Impact Assessment (TIA) and Decision Making (DM) unit that makes use of ATFY-MEBN to perform impact assessment and decision making in Vehicular Ad-hoc Networks (VANETs). This model comes after the Traffic Situation Assessment (TSA) unit introduced in the Chapter 4, and aims to complete the Attention Assist Framework (see Chapter 6).

Tier-0 of ATFY-MEBN is sufficient to implement such a situation assessment paradigm. Accordingly, an individual fuzzy resident node is assigned to each traffic entity. Depending on the type of the data/information to be captured, the state of the resident nodes are estimated from the relevant input *i.e.*, raw sensors data, human operator soft data [85], and so on. Finally, the situations of interest are assessed by running an inference query on the constructed SSFBN.

Tier-1 is linked to the TIA and DM units to assess the impact of the resulted situations of interest, and to consequently make the best decision based on that, the TIA and DM units implement and use the game components residing on tier-1 of the constructed domain-specific ATFY-MEBN. The games created at this stage are *non-cooperative* games, in which the players intend to optimize their own payoff function, without considering those of their opponents. In other words, the payoff function of each player is only composed of the situation states that directly impact the safety of that specific player.

The input situations of interest to TIA are used as the current state of the game nodes whose main player is the vehicle running the TIA instance, and the contextually relevant opponents are determined using VANETs communication links. The TIA and DM units operate interactively to generate different possibilities of the players actions, *i.e.*, action profiles, and measure their impact on the current situation. After the game nodes are played, and the best strategy for the main player is found, the respective action is either directly taken, or is advised to the driver. Figure 5.2 shows how tier-0 of ATFY-MEBN is implemented by the TIA and the DM units. In the following, different essential aspects of a game model are explained thoroughly, and are defined as generic as possible, so that they can be used to implement various services applicable in the VANETs area such as safety and convenience services.

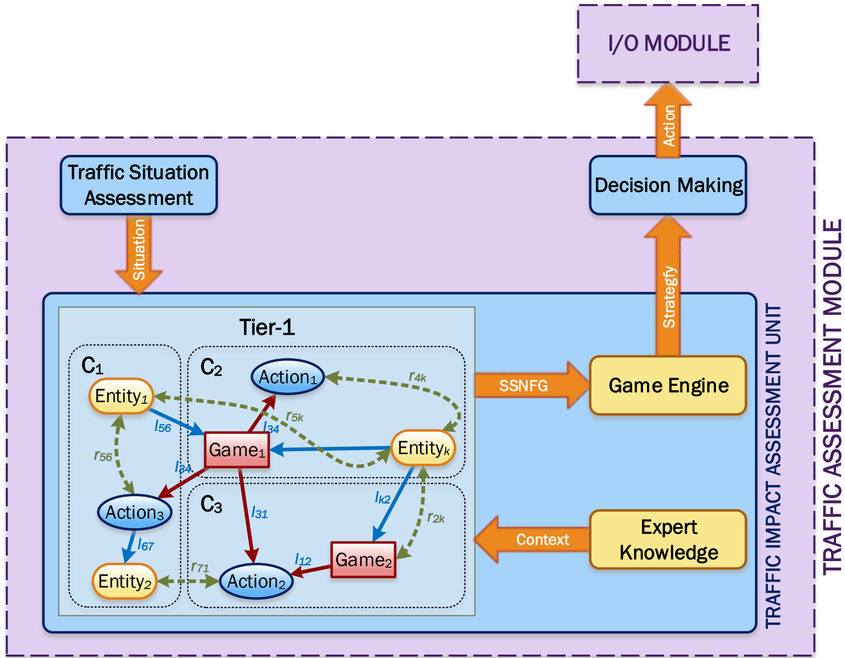


Figure 5.2: The details of the TIA and the DM units in AAF

5.4.1 Players

The players of the games generated in VANETs mainly depend on the scenarios/applications that they are involved, and also on the problems they intend to solve. For example, in the intersection safety applications, the ego vehicle along with other traffic vehicles, and the pedestrians can all be assumed to be the players of the safety games. In such scenarios, the players (no matter of which type they are) intend to avoid vehicle-to-vehicle and vehicle-to-pedestrian collisions. As another example, vehicles are the only players in highway scenarios which similarly try to minimize their risk of accident by tracking some crucial factors including safe speed and distance. Here, we assume that the players are only the vehicles. However, drivers can be easily taken into account without loss of generality.

For each vehicle $v_e(t)$, there are $v_i(t) \in N_e(t)$ game opponents ($e \neq i$), which together create a game with $|N_e(t)| + 1$ players. We denote the main vehicle player (ego vehicle) with v_e, and all the other vehicle players (opponents) with v_i. The neighboring vehicles set of v_e is also shown as $N_e(t)$. Since, the neighboring vehicles of v_e may change over time, N_e is annotated with the time variable t. Using the notations introduced in the section 5.3.4,

$\psi_i = v_e$ is the main player, and ψ_{-i} is the set of opponents (all the players except for ψ_i).

5.4.2 States Space

As we mentioned before, the output of the TSA unit is a set of situations of interest along with their estimated states. Depending on which situations need to have games played for, one or more situations constitute the state of each player in our game setting. Therefore the state space can be defined as $\widehat{S} = \{S_1(\Theta_1), S_2(\Theta_2), \cdots, S_K(\Theta_K)\}$ wherein S_is are the situations and Θ_is are their ordinary variables that are bound with unique identifier symbols.

5.4.3 Decisions (Actions)

All the possible actions that each player entity is able to take is related to the class of that player. As it is thoroughly investigated by Klauer [117], and also based on our slightly different compilation on major factors involved in road incidents, the main entity classes in the VANETs domain include: Vehicle, Driver, Environment, and VANET. Correspondingly, the Vehicle and Driver classes are among those factors which can actively alter the states of the situations by taking actions. Therefore, the actions sets in the AAF are categorized into two abstract groups of *Vehicle Actions*, and *Driver Actions*.

The Vehicle Actions group includes all the *operating-level actions* that a vehicle takes. For instance, steering left, steering right, accelerating, decelerating, braking, etc., are deemed the operating-level actions of a vehicle. An ordered subset of these actions leads to a higher-level action that help the vehicle to reach a goal. Higher-level actions are situation-specific, and may differ in different scenarios. High-level actions can also be organized based on the *environment* in which they are taken. For example, the high-level actions that a vehicle takes in a highway, differ from those it takes in urban areas, or in a parking lot.

5.4.4 Transition Rule

Since we model the states of the players based on the situations of interest set, the transition rules will be simply the changes in their states estimations that are cause by taking actions. This results in situation evolution on lateral dimension, temporal dimension, or both [87]. Therefore, the state space will be a network of situations that are connected with lateral or temporal links. Two situations S_1 and S_2 are temporally connected if taking the action a_1 at S_1 leads to S_2, where S_2 has the same topology as S_1. Furthermore, if taking the action a_1 causes S_1 to change topologically and create S_2, then the two situations are connected with a lateral link. Figure 5.3 depicts a sample network of situations connected via transition rule links.

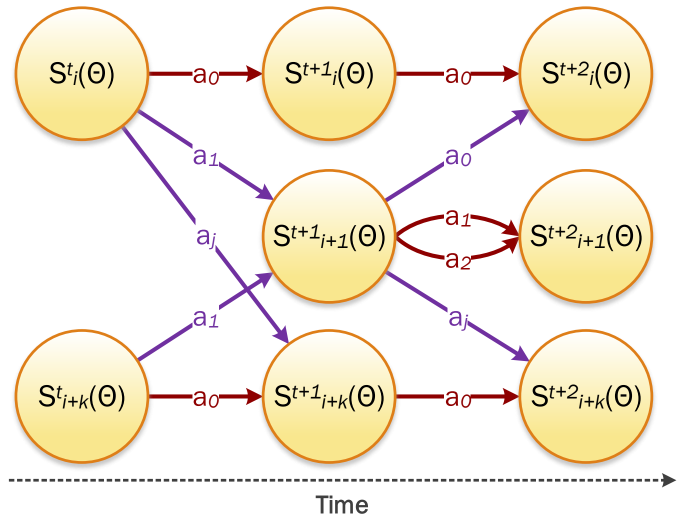


Figure 5.3: A sample network of interconnected situations

5.4.5 Payoff Function

A payoff function is a tool through which a player determines whether an action is beneficial to take or not. In normal form games, payoff functions should be defined in a way to represent the outcome of the strategy of the opponents as well. In other words, the payoff functions need to be designed in a way to express how the actions of the opponents influence the payoff of the main player.

We define a payoff function g as a mapping from a situation, with an estimated state, to a real number that shows how much the current situation is desired. Therefore, g can be defined as: $g : S_i(\Theta_i) \rightarrow \mathbb{R}$, in which $S_i(\Theta_i)$ is an arbitrary state-space situation of interest with Θ_i as its ordinary variables vector, and $\mathbb{R}$ is the set of real numbers.

In ATFY-MEBN, a situation S_i is accessed through its representing fuzzy resident node R_i, which has a set of possible values defined using fuzzy sets [84, 81]. When a situation is assessed, its corresponding random variable node s in the generated probabilistic network is assigned an uncertain state such that $\sum_{v \in V(S)} p(s = v) = 1$, wherein $V(s)$ is the set of possible values of s. We use the estimated state probabilities as the weights vector, and calculate the weighted sum of the fuzzy sets defined on the possible values, such that:

$$\sum_{v \in V(s)} \mu_v(x) p(s = v) \tag{5.12}$$

in which $\mu_v(x)$ is the fuzzy membership function defined for the possible value v. Finally, the payoff function g is defined as the center of gravity of the calculated weighted sum. Therefore:

$$g = \frac{1}{A} \int_x x \sum_{v \in V(s)} \mu_v(x) p(s = v) dx \tag{5.13}$$

wherein A is the total area, and x ranges over the universe of discourse.

5.4.6 Equilibrium Calculation

As we mentioned before, the best strategy is commonly found using linear programming techniques. Here we use the same technique in [177], to find the Nash equilibrium for n-player games. In this algorithm, all the possible pairs of supports (pure strategies played with non-zero probabilities) are enumerated, and for each pair a feasibility program is solved to verify the existence of a Nash equilibrium. For more details, the reader is referred to [177, 152, 59].

5.5 Impact Assessment and Decision Making Experiments

In this section, we propose a framework that makes use of ATFY-MEBN to perform impact assessment and decision making in the IoC. Most parts of the underlying situation assessment task is similar to what we have previously presented in Chapter 4. In fact, major situations and their building blocks still reside on tier-0, and can be used directly for situation assessment.

What is extra in our new setting is the involvement of relevant game components that are used to predict the future situations. Accordingly, we use our novel ATFY-MEBN model and its underlying components to re-design the Collision Warning System (CWS) introduced in Chapter 4, and to enable impact assessment and decision making.

The following explains the main components of the new CWS model. First of all, game ingredients such as Players, State Space, Decisions, Transition Rules, and Payoff Functions are specified. Moreover, respective ATFY-MEBN game components, *i.e.*, actions, action nodes, game nodes, *etc.*, are defined and used to build the second tier of the new 2-tier ATFY-MTheory of the CWS.

The performance of the newly introduced game components are evaluated in various VANETs configurations for both centralized and distributed games. Moreover, two different scenarios of driving in a highway, and at an intersection are outlined to demonstrate the IA and DM capabilities of CWS in situations of different types.

5.5.1 Game Components

A number of components need to be defined for every game to be complete and solvable. These components along with their specifications are introduced below.

Players

The players classes vary in the two separate scenarios we will investigate. In the intersection safety scenario, the ego vehicle along with other traffic vehicles, and the pedestrians can all be assumed to be the game players. In such scenarios, the players (no matter of which type they are) intend to avoid vehicle-to-vehicle and vehicle-to-pedestrian collisions. Moreover, vehicles are the only type of players participating in the highway safety scenario. To make our simulations simpler, we assume that the only type of players in our scenarios are vehicles. As we mentioned before, the main player is shown with v_e, and its opponents v_i, where $e \neq i$, at time t are in the neighboring set $N_e(t)$.

States Space

The states are selected based on the underlying ATFY-MTheory constructed for our domain. The main situation that can be uses to measure the threat of the current driving situation is CollisionThreatLevel(v_e, t), which reflect the degree of how much a vehicle is close to an accident. Another possible situations of interest can be ManeuveringLevel(v_e, t), DistanceDangerLevel(v_e, t), and so on.

Table 5.1: List of operating-level actions of the Vehicle and Driver classes

Vehicle
Accelerate, Decelerate, Steer Right, Steer Left, Shift Gear Up, Shift Gear Down, Stop, Toggle Reverse Gear
Driver
Look Forward, Look Back, Shoulder Check Right, Shoulder Check Left, Turn on/off Lights, Turn on/off Wiper, Signal Right, Signal Left, Take the Cell Phone, Put Down the Cell Phone, Keep Driving, Take a Rest,

Decisions (Actions)

The list of the operating-level actions that vehicles usually take are presented in Table 5.1. Moreover, the high-level actions, based on *the environment*, of a vehicle in a highway are depicted in Figure 5.4, which also illustrates how operating-level actions are put together to create high-level ones. For instance, a vehicle should *accelerate* (upto a safe speed) and *steer left* at the same time to merge onto the passing vehicles in the highway. Moreover, acceleration, and steering wheel requires changing throttle pedal angle (and possibly gears), and the wheels angle, respectively. The actions presented for each class in Table 5.1 are in direct relation with how the situation of the entities defined in each class evolves as those action are taken. In other words, most of these actions (in both the Vehicle and the Driver classes) are literally taken by the driver, but those that may cause alteration in driver's attentiveness are categorized in a separate group. For instance, the operating-level driver actions can be easily related to driver's distraction or drowsiness. Since the actions in the new CWS model are grouped based on the environment factor, the environment determines which actions are available to a specific vehicle. In ATFY-MEBN language, the actions are collected in relevant action nodes that are further connected to corresponding game nodes. See Figure 5.1 for an example.

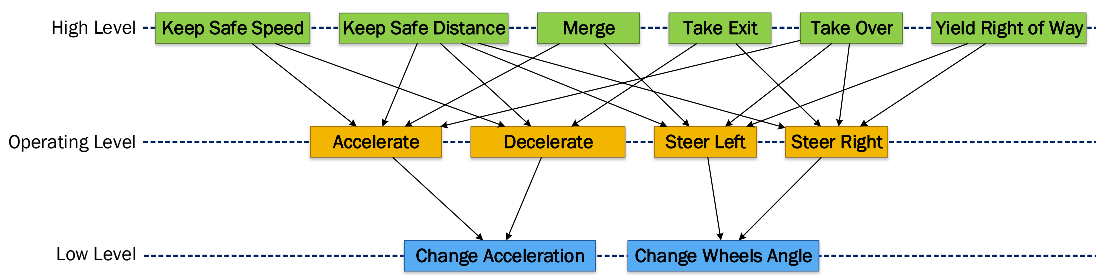


Figure 5.4: The high-level, operating level, and low-level actions taken by the drivers on a highway

Transition Rule

The situations of interest set in our case study contains only the CollisionThreatLevel(v_e) situation that is composed of four main sub-level situations on Vehicle, Driver, Environment, and VANET, and therefore, reflects a comprehensive look of all the crucial situations. A fragment of state space along with the transition links is illustrated in Figure 5.5. In

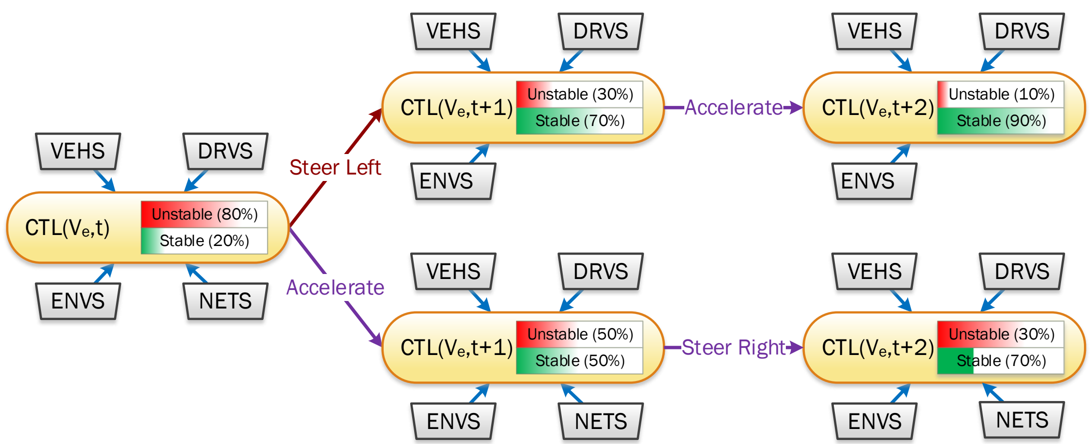


Figure 5.5: The state space of the CollisionThreatLevel(v_e) situation

the simple example shown in Figure 5.5, the vehicle V_e is estimated to be in unstable situation with 80% certainty. The movement situation of the vehicle changes at the next time step upon taking the "steer left", or the "accelerate" actions. Accordingly, if the "steer left" action is taken (considering the actions that the neighboring vehicles will take), and assuming that this action gets the maximum payoff to V_e, then the collision threat level is estimated to be at *Stable* state with 70% of certainty. If we assume that V_e is a rational agent and is gonna take the "accelerate" action, then future situations can be predicted based on the same strategy.

Payoff Function

The payoff function is declared based on its definition in AAF and the assigned state space. Therefore, the possible values of the CollisionThreatLevel(v_e) situation along with their fuzzy sets are used to calculate the payoff. As seen in Figure 5.6, the fuzzy sets are defined using triangular membership functions on a universe of discourse in the range of [0..100]. Moreover, the likelihood of each state is estimated, and consequently, the fuzzy

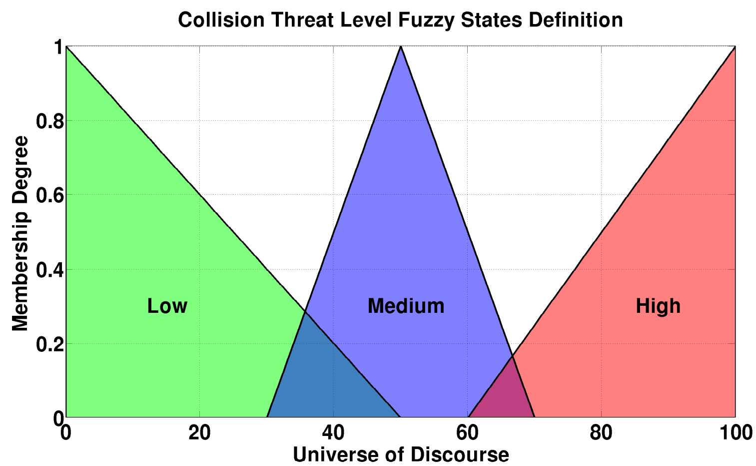


Figure 5.6: The fuzzy sets definitions of the possible values of CollisionThreatLevel(v_e) situation

state is calculated using Equation 5.12. The center of gravity of the resulting function, which is presented in Figure 5.7, is used to find the payoff value. Centroid and Bisector methods for calculating the center of gravity are employed in this paper.

5.5.2 Modeling

The completed version of CWS contains additional entities that make it ready for impact assessment and decision making. These entities and the way they are structured can be achieved by exploiting a domain expert's knowledge. We follow the same process used

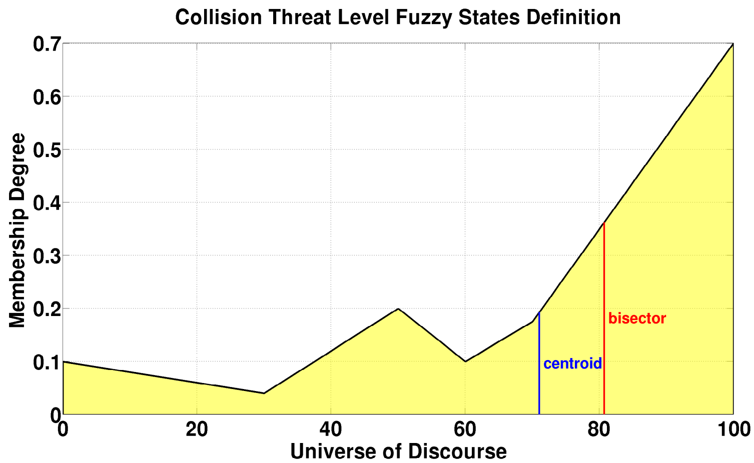


Figure 5.7: The aggregation of the fuzzy sets definitions of the possible values of CollisionThreatLevel(v_e) situation

in [47] to schematically extract the knowledge and form it into semantic and causal structures/rules. The questionnaire is similar to what is presented in 4.1. This results in a semantics network, and a causality structure that are previously presented in Figures 2.2 and 4.2, respectively.

Finally, we utilize ATFY-MEBN to implement the semantics and causal relationships, and to build the game components. Figure 5.8 presents the ATFY-MFrags involved in the new ATFY-MTheory of CWS. As shown in Figure 5.8, the SafetyMFrag contains the game components, and therefore, resides on tier 1. The inter-tier links are provided by the output nodes in SpeedSafetyMFrag, DistanceSafetyMFrag, and ManeuveringSafetyMFrag, whose action node is MovementActions(v_1), and resides in the SafetyMFrag.

5.5.3 Simulations Setup

Two scenarios are defined to present the capabilities of our proposed framework in different applications. The first scenario represent the ego vehicle $!VEHEGO_{1.0}$ in a highway with various number of traffic vehicles $!VEH0X_{1.0}$s with random movement patterns, and the second one models an intersection at which two (or more) vehicles arrive. In both scenarios, it is assumed that all of the vehicles are geared with the CWS that is able to assess the current collision threat, *i.e.*, CollisionThreat($!VEHEGO_{1.0}$) , as well as to send/receive messages to/from the neighboring vehicles through communication links. It is also notable that for sake of simplicity, we temporarily assumed that the vagueness of each entity

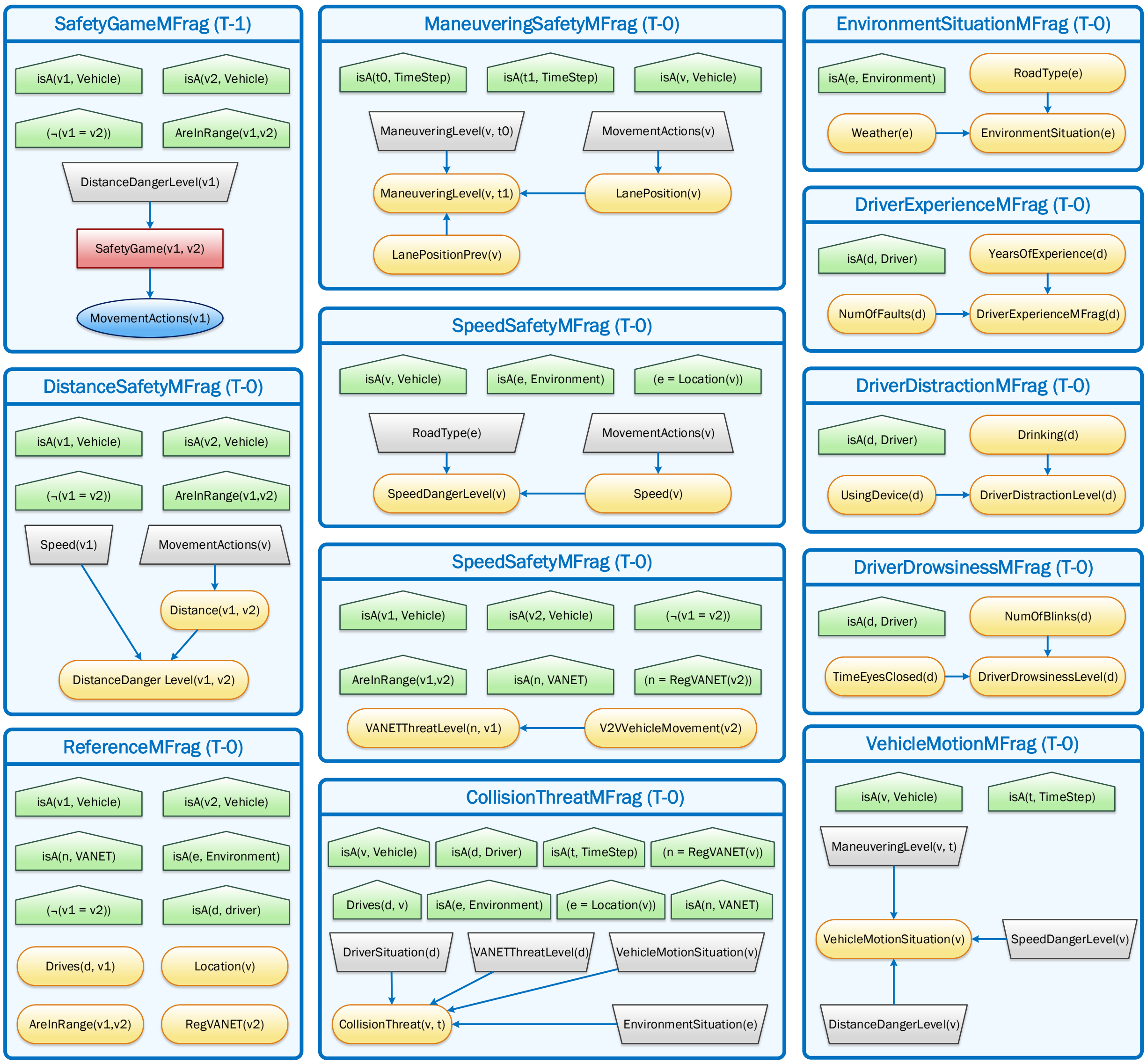


Figure 5.8: The ATFY-MTheory designed for the new CWS

identifier symbol is 0.0, and therefore, all of them are sub-scripted with the membership degree of 1.0. The details of the scenarios are presented next.

Scenario 1: Highway

In the highway scenario, we assume that $!\mathit{VEHEGO}_{1.0}$ is in a highway and is occasionally surrounded by the traffic vehicles $!\mathit{VEH0X}_{1.0}$s along the path. The observations are added to the knowledge base according to a prefixed configuration, and in different intervals. For instance, some of them such as driver's years of experience (or number of faults) are read just once, converted into relative entities, and are permanently kept in the knowledge base. Additionally, other observations are added in entities form through sensor reading and LLDF. The *fixed* details of the scenario are presented in Table 5.2. In the information provided below, the vehicle and its driver, and the environment are observed to be $!\mathit{VEHEGO}_{1.0}$, $!\mathit{DRVEGO}_{1.0}$, and $!\mathit{H401}_{1.0}$, respectively. In other words, *i.e.*, $(v = !\mathit{VEHEGO}_{1.0})$, $(d = !\mathit{DRVEGO}_{1.0})$, and $(e = !\mathit{H401}_{1.0})$ are already in the knowledge base. Other observations along with their assessed entities are presented in the next section, where the simulations are actually run, and the sensors readings are recorded. All

Table 5.2: The details of the highway scenario

Entity	**Fuzzy State**
$\mathrm{ROT}(e = !\mathit{H401}_{1.0})$	$\langle !\mathit{Street}^{0.1}_{1.0}, !\mathit{Highway}^{0.9}_{1.0}\rangle$
$\mathrm{WEA}(e = !\mathit{H401}_{1.0})$	$\langle !\mathit{Sunny}^{0.7}_{1.0}, !\mathit{Cloudy}^{0.3}_{1.0}, !\mathit{Rainy}^{0.0}_{1.0}, !\mathit{Snowy}^{0.0}_{1.0}\rangle$
$\mathrm{YOE}(d = !\mathit{DRVEGO}_{1.0})$	$\langle !\mathit{Few}^{0.2}_{1.0}, !\mathit{Many}^{0.8}_{1.0}\rangle$
$\mathrm{NOF}(d = !\mathit{DRVEGO}_{1.0})$	$\langle !\mathit{Few}^{0.9}_{1.0}, !\mathit{Many}^{0.1}_{1.0}\rangle$

the vehicles in this scenario have an actions set that are presented in Table 5.3. As it is

Table 5.3: List of the highway actions and their influence set

Action	**Influence Set**
$@\mathit{Stay}$	$\varnothing$
$@\mathit{Accelerate}$	$\{\mathrm{SPD}(v), \mathrm{DIS}(v, v_i), \mathrm{AIR}(v, v_i)\}$
$@\mathit{Decelerate}$	$\{\mathrm{SPD}(v), \mathrm{DIS}(v, v_i), \mathrm{AIR}(v, v_i)\}$
$@\mathit{SteerRight}$	$\{\mathrm{DIS}(v, v_i), \mathrm{AIR}(v, v_i), \mathrm{LN}(v, t)\}$
$@\mathit{SteerLeft}$	$\{\mathrm{DIS}(v, v_i), \mathrm{AIR}(v, v_i), \mathrm{LN}(v, t)\}$

shown in Table 5.3, each action a_i is assigned to an influence set that determines which entities need to be refined when a_i is taken.

Scenario 2: Intersection

In this scenario, it is assumed that $!VEHEGO_{1.0}$ along with the traffic vehicles, $!VEH0X_{1.0}$s, approach an intersection. The new CWS running on each vehicle performs in a way to avoid any accident, or minimize its cost. Similar to the first scenario, the following observations: ($v = !VEHEGO_{1.0}$), ($d = !DRVEGO_{1.0}$), and ($e = !H401_{1.0}$) are already in the knowledge base, and the rest are measured in real-time. The scenario specifications are presented in Table 5.4. The action set of the vehicles and their influence set are also introduced in Table 5.5.

Table 5.4: The details of the intersection scenario

Entity	**Fuzzy State**
ROT($e = !KingSt_{1.0}$)	$\langle !Street^{0.9}_{1.0}, !Highway^{0.1}_{1.0} \rangle$
WEA($e = !KingSt_{1.0}$)	$\langle !Sunny^{0.1}_{1.0}, !Cloudy^{0.1}_{1.0}, !Rainy^{0.8}_{1.0}, !Snowy^{0.0}_{1.0} \rangle$
YOE($d = !DRVEGO_{1.0}$)	$\langle !Few^{0.2}_{1.0}, !Many^{0.8}_{1.0} \rangle$
NOF($d = !DRVEGO_{1.0}$)	$\langle !Few^{0.9}_{1.0}, !Many^{0.1}_{1.0} \rangle$

Table 5.5: List of the highway actions and their influence set

Action	**Influence Set**
$@Stay$	$\varnothing$
$@Stop$	$\{\text{SPD}(v), \text{DIS}(v, v_i), \text{AIR}(v, v_i)\}$
$@Accelerate$	$\{\text{SPD}(v), \text{DIS}(v, v_i), \text{AIR}(v, v_i)\}$
$@Decelerate$	$\{\text{SPD}(v), \text{DIS}(v, v_i), \text{AIR}(v, v_i)\}$
$@SteerRight$	$\{\text{DIS}(v, v_i), \text{AIR}(v, v_i), \text{LN}(v, t)\}$
$@SteerLeft$	$\{\text{DIS}(v, v_i), \text{AIR}(v, v_i), \text{LN}(v, t)\}$

5.5.4 Situation-Specific Normal Form Games Results

In this section, the performance of the Situation-Specific Normal Form Games (SSNFG) generation algorithm is studied from different aspects, when deployed in an Internet of Cars environment. In fact, we mainly emphasize on the time performance of both game generation and solution algorithms utilized in centralized and distributed configurations (see section 5.3.4).[2]

Centralized Configuration

In the centralized configuration, it is assumed that the players participating in the situation specific games, generated by the ego vehicle, are its K nearest neighbors (opponents), whose consistency constraints are satisfied. We assume that the number of opponents is bounded by 8 in the safety scenarios of the IoC. This is a sound assumption, as there can be at most 8 other close threatening vehicles surrounding the ego vehicle. Figure 5.9 illustrates the arrangement of vehicle players in the centralized configuration. The time

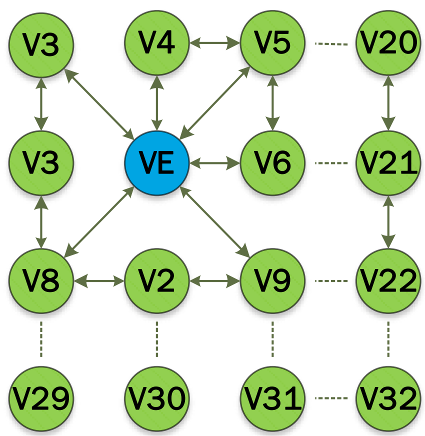


Figure 5.9: The centralized configuration of the vehicles in safety scenarios of the Internet of Cars

performance of the safety game node (see Figure 5.8) is measured when deployed in the centralized configuration with various number of opponents. Besides, we also study the effect of splitting the single game node, linked to an action node with many number of actions, into multiple game nodes connected to the corresponding actions nodes with fewer number of actions. As it is shown in Figure 5.10, the time performance of the game generation algorithm increases exponentially (log-linearly) as the number of players increases. However,

[2]All the simulations were run on a Microsoft Windows PC with a Core i7 3.4 GHz processing power and 16 Gigabytes of memory.

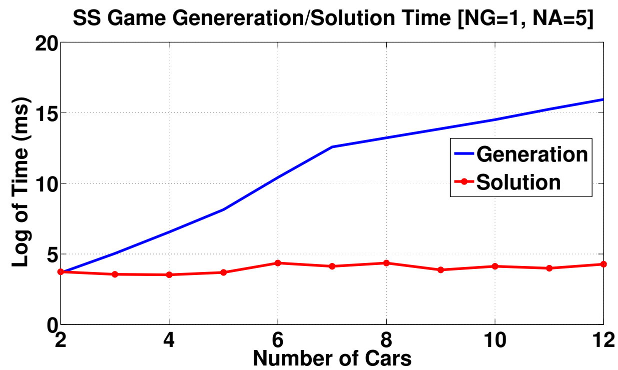


Figure 5.10: The time performance of a single game node linked to an action node with 5 actions.

the game solution algorithm performs almost constantly, when called with various number of players. Figure 5.10 shows that for 8 number of opponent vehicles (9 players in total) the required time to generate a game is almost 268 seconds, which is way over the proper time period required to plan safe driving. Based on these results, such a configuration is only practical in the configurations where the total number of players is not more than 5, since the generation time will be less than or equal to 1 second in such cases.

To increase the applicability of the game generation algorithm for the configurations with more than 5 players, we split the SafetGame(v_1, v_2) to two separate nodes of SafeDistanceGame(v_1, v_2) and SafeSpeedGame(v_1, v_2), which are connected to DistanceActions(v_1) and SpeedActions(v_1) action nodes, respectively. In this new setting, the DistanceActions(v_1) action node consists of @*SteerRight*, @*SteerLeft*, and @*Stay* actions, and the SpeedActions(v_1) action node contains @*Accelerate*, @*Decelerate*, and @*Stay* actions. Figure 5.11 shows the time performance of the generation and solution algorithms for 2 game nodes each connected to action nodes with 3 actions. Similar to the previous case, the time increases exponentially (log-linearly) as the number of players increases. However, generating the 9-players game needs around 60 seconds this time, and the algorithm is practically useful when the number of players are less than or equal to 6.

To further optimize the time performance of our proposed algorithm in the IoC environment, we drop the @*Stay* from both the DistanceActions(v_1) and SpeedActions(v_1) action nodes in the multiple game nodes case. Instead, we only play the game if only the payoff of the resulting optimal strategy is better than not taking any actions. As a result, we will have two game nodes linked to two action nodes each consists of only 2 actions. The time performance of this setting is demonstrated in Figure 5.12. As it is shown in

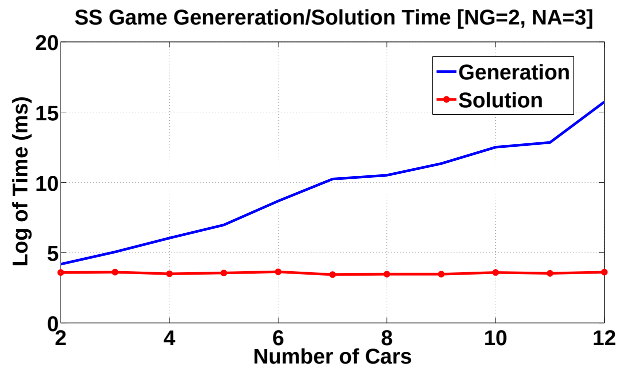


Figure 5.11: The time performance of the generation and solution algorithms for 2 game nodes each connected to action nodes with 3 actions.

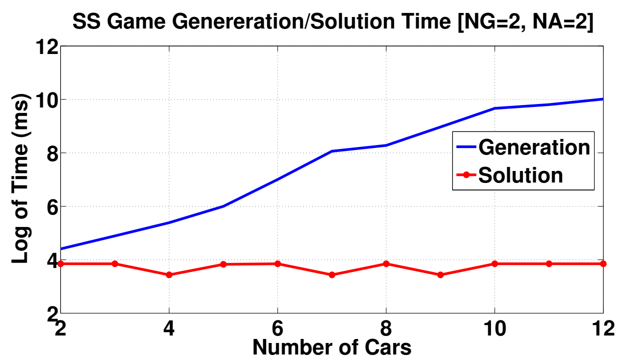


Figure 5.12: The performance of the generation and solution algorithms for 2 game nodes each connected to action nodes with 2 actions.

Figure 5.12, the algorithm requires around 8 seconds to generate the 9-players game, and it will need only the maximum of around 3 seconds for less number of players. Figure 5.13 puts together the time performance of all the mentioned cases for sake of comparison.

Distributed Configuration

The opponent players in the distributed configuration are able to locally play their own games and provide the main players with their optimal strategy in a recursive manner. In the IoC environment, the ego vehicle contacts its K nearest neighbors, and asks for their payoffs for each of the action profiles in the generated game. Similarly, each of the neighbors start a new game with its own set of K nearest neighbors, and so on. When

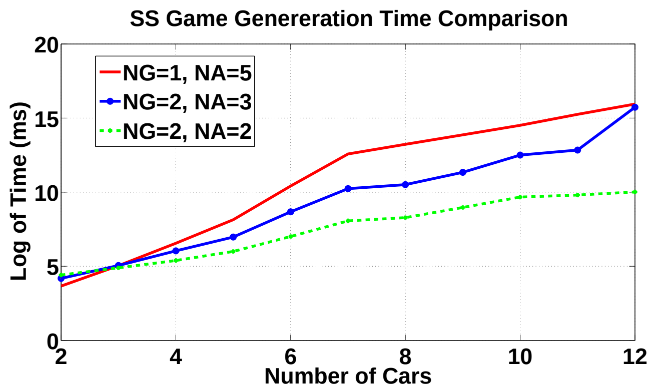


Figure 5.13: The time performance of the generation and solution algorithms for various number of game nodes and actions.

a certain threshold is met, such as the maximum number of hops, or elapsed time, the resulting optimal strategies are calculated by solving the games, and furthermore, are back propagated to the ego vehicle. We have analyzed the time performance of the game generation and solution algorithms by running it in 50 different driving scenarios, with 1 to 25 vehicles randomly positioned and linked to one another, and averaging over the needed time to generate and solve the game.

In the first set of experiments, we bound the maximum number of neighbors of each vehicle to 4, and define a hopping threshold of 1. This means that all the games generated in this setting have no more than 5 players, and the games are played with only the ego vehicle, and its level 1 neighbors (number of hops is 1). Figure 5.14 shows the time performance of the game generation algorithm in this case. Figure 5.14 demonstrates that the game generation algorithm performs well for all number of cars. The maximum time record is 66.1 milliseconds for a scenario with 17 randomly positioned and linked vehicles. Moreover, the game solution algorithm performs constantly, and requires an average of around 27.1 milliseconds to solve the generated games.

We increase the hopping threshold up to 4 to study the effect local game generation/solution, and distributed impact assessment. Figures 5.15 to 5.17 show the time performance of the game generation/solution algorithms for hopping thresholds of 2, 3, and 4. As it is demonstrated in these figures, the maximum needed time to generate the complete game on the ego vehicle is around 403.4, 601.8, and 3600 milliseconds for hopping threshold of 2, 3, and 4, respectively. In all of these cases, the required game generation time increases exponentially up to a certain point, but it is then kept constant. This is due to the bounded maximum number of neighbors of each vehicle, which prevents exponential

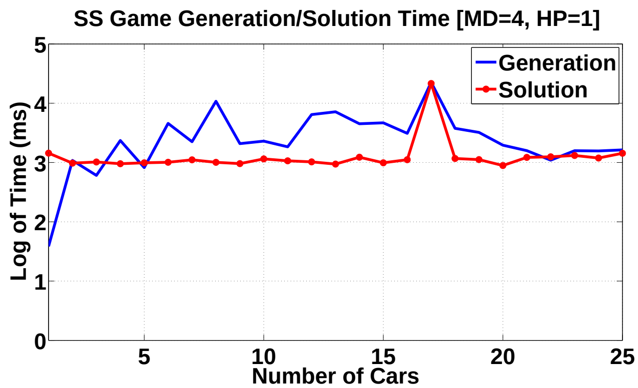


Figure 5.14: The time performance of the game generation algorithm in the distributed configuration. Maximum number of neighbors of each vehicle is 4, and the hopping threshold is 1.

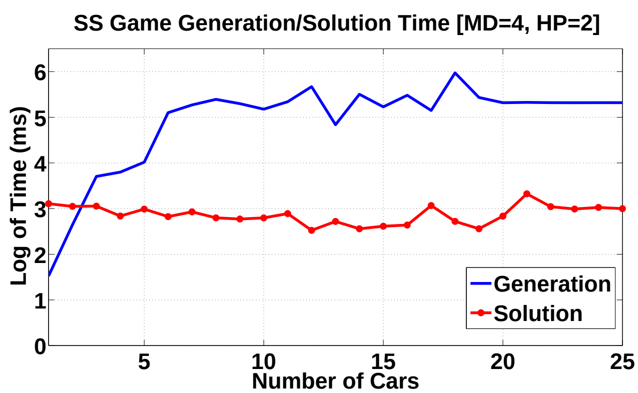


Figure 5.15: The time performance of the game generation algorithm in the distributed configuration. Maximum number of neighbors of each vehicle is 4, and the hopping threshold is 2.

growth in the size of communication links between the vehicles.

In the second set of experiments, we bound the maximum number of neighbors of each vehicle to 8, and study the time performance of the game generation and solution algorithms for the hopping thresholds of 2 and 4. As it is demonstrated in Figures 5.18 and 5.19, the required time to generate a game increases exponentially (log-linearly) as the number of present vehicles in the region grows. This growth is up to 12 and 19 vehicles for when the hopping threshold is equal to 2 and 4, respectively. Moreover, the maximum time record for 2 and 4 number of hops are around 60 and 162 seconds, respectively. Besides,

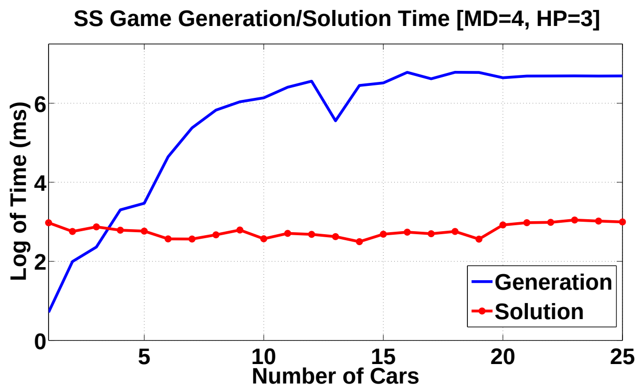


Figure 5.16: The time performance of the game generation algorithm in the distributed configuration. Maximum number of neighbors of each vehicle is 4, and the hopping threshold is 3.

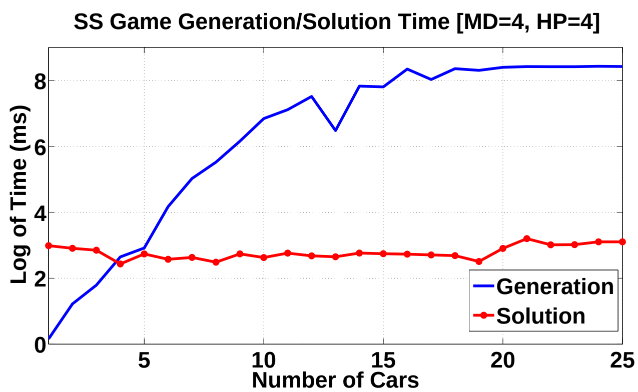


Figure 5.17: The time performance of the game generation algorithm in the distributed configuration. Maximum number of neighbors of each vehicle is 4, and the hopping threshold is 4.

for the maximum of 8 neighbors, the game generation algorithm is practical only when the number of vehicles is less than or equal to 10. However, this is not really an impractical assumption as in many driving scenarios the number of vehicles in a specific region are not more than 10.

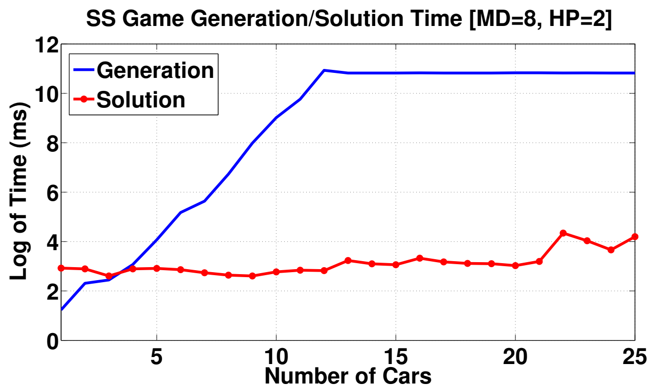


Figure 5.18: The time performance of the game generation algorithm in the distributed configuration. Maximum number of neighbors of each vehicle is 8, and the hopping threshold is 2.

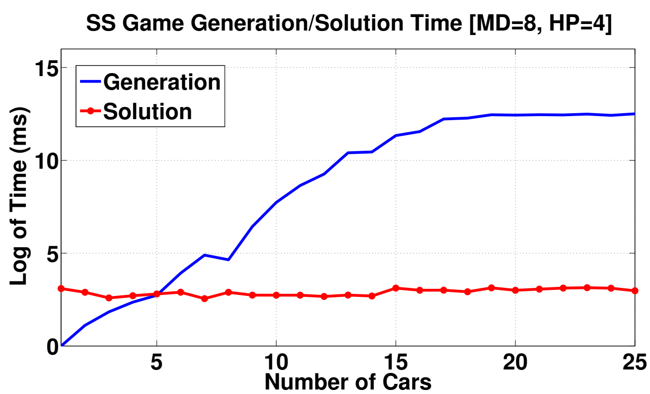


Figure 5.19: The time performance of the game generation algorithm in the distributed configuration. Maximum number of neighbors of each vehicle is 8, and the hopping threshold is 4.

5.5.5 Impact Assessment Results

The applicability of the proposed model and algorithms are evaluated in two real world driving scenarios of highway and intersection. In both scenarios, Situation-Specific games are generated and solved at each 500 milliseconds and according to the current set of neighbors. Moreover, the available actions are either @*SteerRight* and @*SteerLeft* for the SafeDistanceGame(v_1, v_2) game, and @*Accelerate* and @*Decelerate* for the SafeSpeedGame(v_1, v_2) game. The resulting optimal strategy is processed and the appropriate action is automatically applied on the vehicle.

Scenario 1: Highway

In the highway scenario, it is assumed that the ego vehicle is driven in a highway with randomly positioned and controlled normal traffic, which are able to communicate with each other through V2V communication. The games are generated for the maximum of 5 players (maximum 4 neighbors), and the hopping threshold is set to 2. We repeat the scenario for 33 iterations and calculate the ratio of collisions occurred with and without the game nodes deployed. See Figure 5.20 for more details. As it is depicted in Figure 5.20,

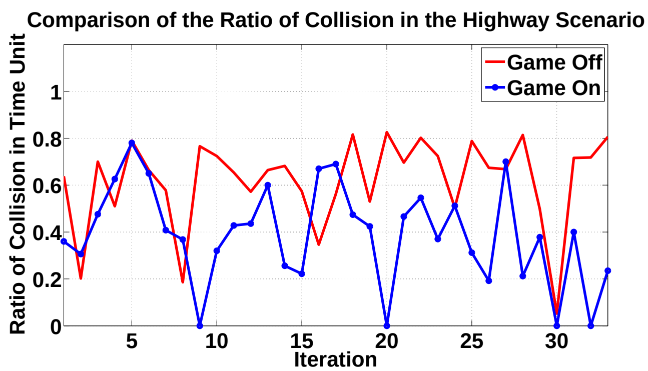


Figure 5.20: The ratio of collisions for turned on and off impact assessment unit

the ratio of collisions is decreased when the impact assessment is done and its resulting actions are performed by the vehicle. The averages of the ratio of collisions for turned off and on impact assessment unit are 61.95% and 38.84%, respectively. Another interesting aspect about this experiment is that in about 12% of iterations the ratio of collision is zero (iterations 9, 20, 30, and 32), when the impact assessment unit is turned on. This is a significant improvement when compared to 0% collision rate with the turned off impact assessment unit. Finally, non-zero collision rates, when games are played, are attributed to the situation-invariant method that actions are taken. In other words, the optimal actions proposed by the impact assessment unit are high-level (see Figure 5.4), and lack the detailed information about how it should be taken. For example, when the optimal action is to steer left, it does not deliver how much the vehicle needs to go to its left. This may result in translating to an unforeseen situation that leads to a collision. However, this can be improved by providing lower-level information about the actions as well, and let the driver know how an action need to be take.

The VehicleMotionSituation($!VEHEGO_{1.0}, !T1_{1.0}$) for both cases of turned on and off impact assessment unit shows that the game components are successful in keeping the vehicle motion situation in stable state. Figures 5.21 and 5.22 demonstrate the state estimation of the VehicleMotionSituation($!VEHEGO_{1.0}, !T1_{1.0}$) for these two cases at the iteration with highest collision ratio. Clearly, with the impact assessment unit turned on, the vehicle motion situation of $!VEHEGO_{1.0}$ is more stable during the simulation. Finally,

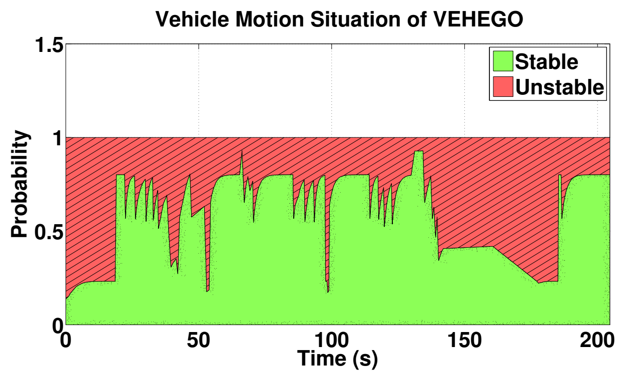


Figure 5.21: The VehicleMotionSituation($!VEHEGO_{1.0}, !T1_{1.0}$) for the highest ratio iteration when the impact assessment unit is turned on.

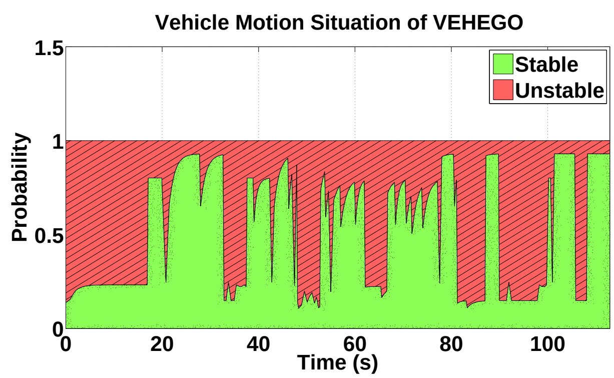


Figure 5.22: The VehicleMotionSituation($!VEHEGO_{1.0}, !T1_{1.0}$) for the highest ratio iteration when the impact assessment unit is turned off.

the evolution of the calculated payoffs for the lowest and highest collision ratios when the impact assessment unit is turned on are shown in Figures 5.23 and 5.24. It is obvious that in both best and worst cases, the impact assessment unit is able to keep the payoff of the vehicle at high values in longer periods of time.

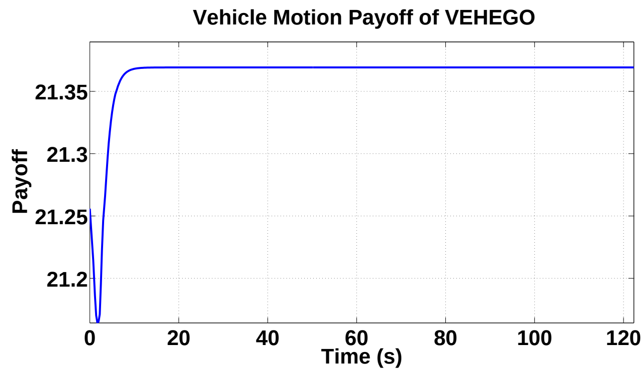


Figure 5.23: The payoff of $!VEHEGO_{1.0}$ for the lowest collision ratio with the impact assessment unit is turned on.

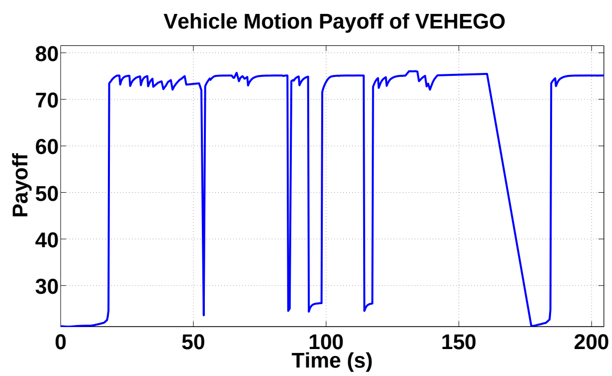


Figure 5.24: The payoff of $!VEHEGO_{1.0}$ for the highest collision ratio with the impact assessment unit is turned on.

Scenario 2: Intersection

The intersection scenario is studied in 7 separate sub-scenarios created with 2 to 8 cars. In each sub-scenario, the vehicles are initialized with on random roads and with random speeds, and have their Collision Warning System (CWS) deployed and running. We study the effect of impact assessment and decision making by turning on and off the impact assessment units, running the simulation for 22 iterations, and finally, counting the number of times the vehicles collide at the intersection. Table 5.6 briefly presents the ratio of collisions in each sub-scenario. As it is shown in Table 5.6 shows, the impact assessment unit remarkably decrease the collision ratio by projecting the status of the current driving situation into the future, assessing its impact, and taking an appropriate action accordingly.

Table 5.6: The ratio of collision in the 2 to 8 cars intersection scenarios with the impact assessment unit on and off

	Game Off	**Game On**
2-Cars	85.2%	11.4%
3-Cars	91.7%	29.8%
4-Cars	83.1%	23.5%
5-Cars	93.6%	26.5%
6-Cars	94.5%	29.6%
7-Cars	88%	33.2%
8-Cars	89.8%	49.1%

Besides, the applicability and the performance of generating situation specific games in clear when facing various situations with different number of players and topologies at an intersection.

5.6 Summary

Highways are now high-risk areas, where drivers select high speed to reach their destination. Drivers must monitor the traffic constantly and carefully for sudden events, such as errant vehicles changing lanes, slowing vehicles, entering and exiting high speed roads, *etc.* Assessing such hazardous situations, evaluating their impacts, and taking proper actions accordingly, is a challenging task. Therefore, road safety is tightly related to the drivers, whose improved awareness and efficient interaction to their environment lead to prevent accidents.

In this chapter, we proposed a game-theoretic approach based on Fuzzy Multi-Entity Bayesian Networks (Fuzzy-MEBN) to simulate the strategic interactions between different entities involved in the IoC environment. The novelty of our approach is explicit in extending the classical Fuzzy-MEBN by introducing active entities and arranging them in a new 2-tier architecture. Moreover, new game components such as game nodes and action nodes are added to MEBN to prepare this model for impact assessment and decision making. Games are generated based on a specific situation whose contexts are determined through context nodes in the original MEBN. Accordingly, two novel algorithms for constructing situation-specific normal form games and 2-tier situation-specific active fuzzy bayesian networks are presented. Furthermore, the game and action nodes participate in hypothesizing a set of future situations that are created upon taking actions.

Another novelty of ATFY-MEBN is its decentralized technique in calculating the payoffs of opponents, which can be utilized in the environments where communication links between different player entities are available. Subsequently, the players of game nodes can overcome their lack of knowledge about their opponents' states by cooperatively calculate their payoffs through inter-player communication.

We tested ATFY-MEBN by first examining the performance of its game components when used in different cases. Besides, an impact assessment and decision making unit based on ATFY-MEBN was created that aimed to demonstrate the applicability of the proposed model in real world problems. Our results showed that ATFY-MEBN can be conveniently used in various applications composed of different types and numbers of players and actions. This is due to the fact that the computational complexity of the new algorithms defined for ATFY-MEBN does not act as a bottleneck through the inference process. Moreover, correctly assessing the impact of hazardous situations and advising proper optimal actions based on that, is another improvement that is clearly shown in the results.

Future trends include the involvement of another types of games that are optimized to excel in a particular aspect, such as space complexity that is tackled by Action Graph Games [29, 107]. Moreover, parametrized decisions that allow the actions to be taken differently, and specific to different situations, can be another important topic of future research.

Chapter 6

Attention Assist Framework: A Comprehensive Tool for High-Level Information Fusion in VANETs

The High-Level Information Fusion research has recently attracted considerable interest in the data fusion community as reflected by the review articles published in the data fusion community within the last few years [34, 31, 216]. In particular, Blasch et al. [34] identify the top ten trends of research in HLIF among which the issues of uncertainty analysis and semantics/ontologies are described as the most important areas of study that have not received enough attention in the past.

Motivated by this background, we introduce here a high-level information fusion framework to achieve enhanced safety in VANETs, titled Attention Assist Framework (AAF), which aims specifically at improving the road safety by enhancing the driver's attentiveness. This framework is consisted of three major units: a Low-Level Data Fusion model for Traffic Entity Assessment (TEA), and two High-Level Information Fusion models for Traffic Situation Assessment (TSA), and Traffic Impact Assessment (TIA) and Decision Making (DM). The AAF enables autonomous assessment of the current driving situation, prediction of potential impacts, and assist with making decisions or taking actions accordingly.

The AAF is composed of three major data/information fusion units, namely, the Traffic Entity Assessment (TEA) unit, the Traffic Situation Assessment (TSA) unit, and the Traffic Impact Assessment (TIA) and Decision Making (DM) units, whose details and theoretical foundations are specifically presented in the previous chapters.

6.1 Introduction

As discussed previously, safety is one of the main applications of Situation Awareness (SAW) in the Internet of Cars (IoC). Frameworks that aim to bring the machine-interpreted *e*-safety to the passengers are called the Advanced Driver Assistance Systems (ADAS) in the literature. ADAS are designed to actively/passively automate vehicles' functionality for safer journeys. The main purpose of ADAS is to alert drivers from potential threats, or mitigate collisions and fully control the vehicle. Based on their level of interference, ADAS are categorized into active and passive systems. Active ADAS act preemptively to avoid an accident by taking control of the car [71], and passive ADAS refer to the safety embedded technologies in the car that target occupant protection and injuries reduction during a crash.

Dangerous driving situations in the Internet of Cars are limited to a number of cases where an incident is threatening the vehicle and its driver. Such situations are analyzed by sensing the surrounding relevant entities, and then assessing whether a particular situation is being observed or not. For instance, Isermann *et al.* [106] define a set of driving situations such as *blocked lanes* and *overtaking*, and attribute them as unsafe ones. Moreover, they create the relevant entities using the information produced from a lower level data fusion system, and use situation assessment to estimate the states of the situations of interest. A framework in the ADAS family needs to identify major hazardous driving situations need to perform efficiently. Statistics show that road departure and collision with oncoming vehicles are the top two causes of fatal accidents [106], which are mainly due to improper SAW.

6.2 Background and Related Work

In integrated ADAS, the cooperation between the driver, the vehicles, and infrastructure aims to mitigate accidents, and maintain a full awareness of dangerous situations [71, 202]. In driver centric techniques, ADAS ensure situational awareness and include the driver in the decision process. Moreover, vehicle centric techniques use more sensors to provide decision support, and finally, network centric techniques employs wireless communications (V2X) to share useful information and provide the driver with larger telematic vision. In the following, the main safety applications of the IoC are introduced first, and then prominent research work in designing ADAS are highlighted.

6.2.1 Safety Applications in VANETs

The main safety-related situations in the IoC are: lane changing, frontal collision, turning, the post-accident, and the case of emergency vehicles.

Lane Changing Situation

Lane changing situation is the aggregation of a set of entities, along with their relationships that is estimated as lane changing intention. In this situation, the target car keeps receiving periodic updates of the positions and the speed of the surrounding vehicles, through V2V communication. Upon detecting the lane changing intention, the presence or absence of a sufficient gap between vehicles in adjacent lanes are assessed to ensure a safe lane changing. Figure 6.1 depicts a sample scenario.

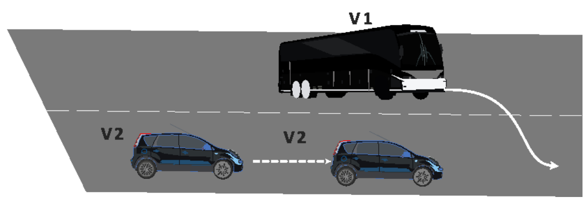


Figure 6.1: A sample situation of the Lane changing situation

Frontal Collision Situation

Similar to the previous situation, a frontal collision situation is a set of relative entities and their relationships that demonstrate a scenario where cars, travelling on opposite directions, may collide. This use case is linked to a situation where a vehicle tries to overcome another vehicle, which consequently causes a potential risk of collision for the vehicle that is getting closer from the opposite direction (see Figure 6.2).

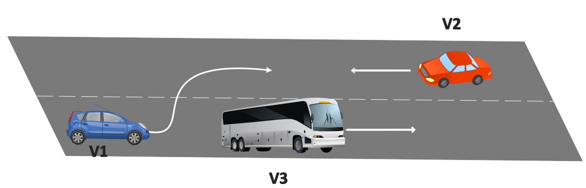


Figure 6.2: A sample scenario of the frontal collision situation

Turning Situation

Turning situations can also be handled by having RSUs to inform the approaching car about the next turn curvature sharpness and the proper speed to switch to, so the driver can make anticipated actions. Figure 6.3 illustrates such a scenario in which all vehicles, along with the RSUs, broadcast messages about the road curvature. As a result, another car approaching a sharp curve may continue its travel safely without any map or other navigational support system on board.

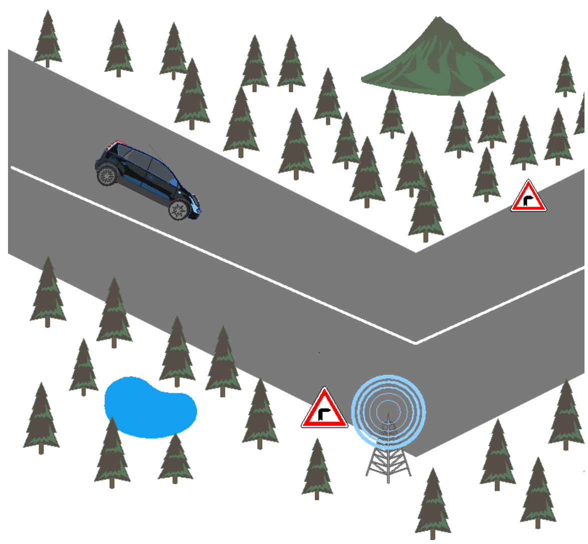


Figure 6.3: A sample scenario of the turning situation

Post Accident Situation

Post-accident situations are important safety situations that connected cars concept may assist in dealing with. In such cases, the cars approaching an accident are notified about an accident in their neighboring zone. This is actually more applicable in low visibility conditions as it can drastically reduce the danger of serial accidents. Figure 6.4 depicts a sample scenario in which the damaged car transmits its position, identity, and status to the closest RSU using V2I communications. The RSU will issue a warning to vehicles approaching the scene of the accident until the accident scene is cleared. Moreover, warning messages can be transmitted directly to approaching vehicles using V2V communication.

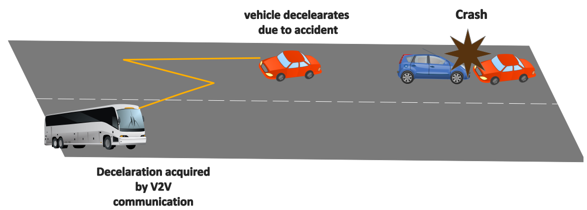


Figure 6.4: Accident ahead scenario

Emergency Vehicle Situation

Lastly, the situations including an emergency vehicle are deemed very important in the IoC context. In such situations, an emergency vehicle is allowed to ask other cars for a reserved corridor relief (see Figure 6.5). Potentially, it can also ask RSUs located at traffic lights to facilitate its mobility. The broadcast message by the emergency vehicle includes information about its position, route, speed, and destination. The embedded application in the surrounding cars use this information to alert the drivers and accordingly give way to the emergency vehicle.

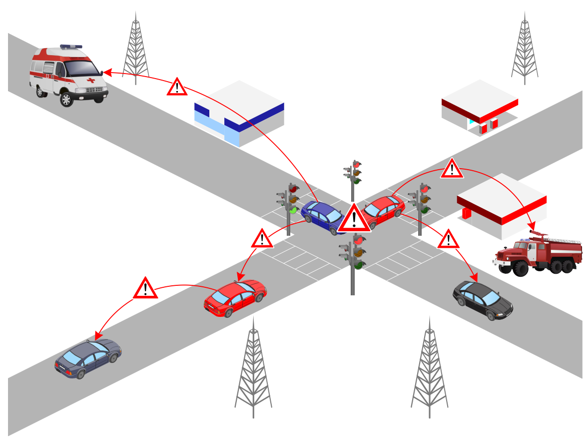


Figure 6.5: A sample scenario of a situation that includes an emergency vehicle

6.2.2 Active Driver Assistance Systems

The European HAVEit project [98] aims the development of ADAS designed for automated vehicles. HAVEit is composed of three main layers: driver interface, command layer, and execution layer, which are in touch with the driver, computing processes, and the physical sensors/actuators, respectively. The different layers of this project are depicted in Figure 6.6. Holzmann *et al.* [99] introduces SPARC model that is a data fusion-based

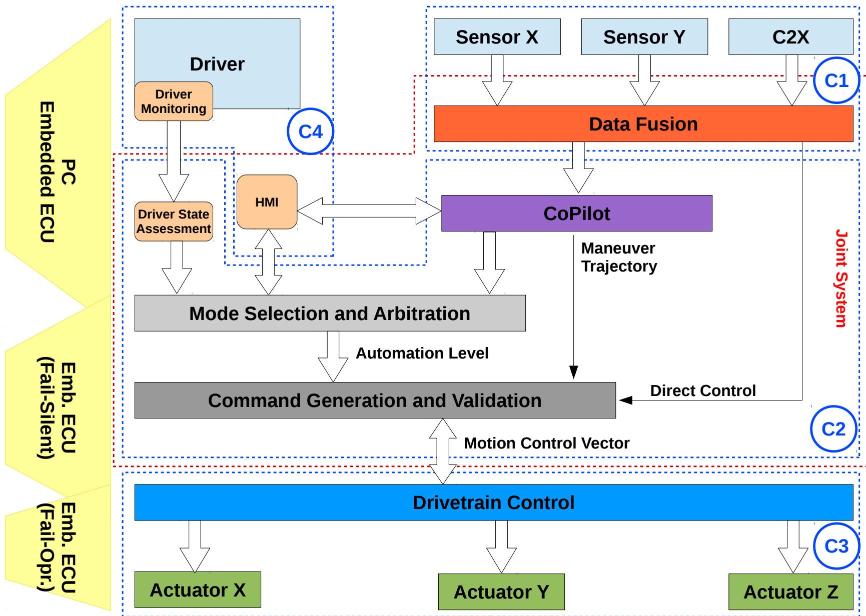


Figure 6.6: The four HAVEit Layers [98]

approach for driver assistance. SPARC architecture is a layered design that starts from environment sensing and goes up to driving command.

PRORETA 1 and 2 are two versions of a collision avoidance system developed by Isermann *et at.* [106]. PRORETA 1 is basically used in highway in which sudden appearance of stationary obstacles on the road, *i.e.* end of a traffic jam, or cutting-in vehicles, are deemed common causes of accident. This system is able to handle such situations by either evading the stationary obstacle, or decelerating enough before colliding with it. An overview of PRORETA1 is highlighted in Figure 6.7. PRORETA 2 is designed to tackle unsafe driving situations; particularly, overtaking in rural areas. The modeled system assumes that the oncoming traffic is viewable by the sensors, which can be a potential source of inaccurate assessment. This is the case in which capabilities of the Internet of Cars may come to play and tackle such problems. Overtaking is detected using the Enhanced Time To Collision (ETTC) measure fused with the accelerator pedal position in a fuzzy-logic system.

Situation-Aware Driver Assistance System (SADAS), developed by Röeckl *et al.* [185], is composed of four main components: 1. Utility-based Knowledge Exchange, 2. Knowledge

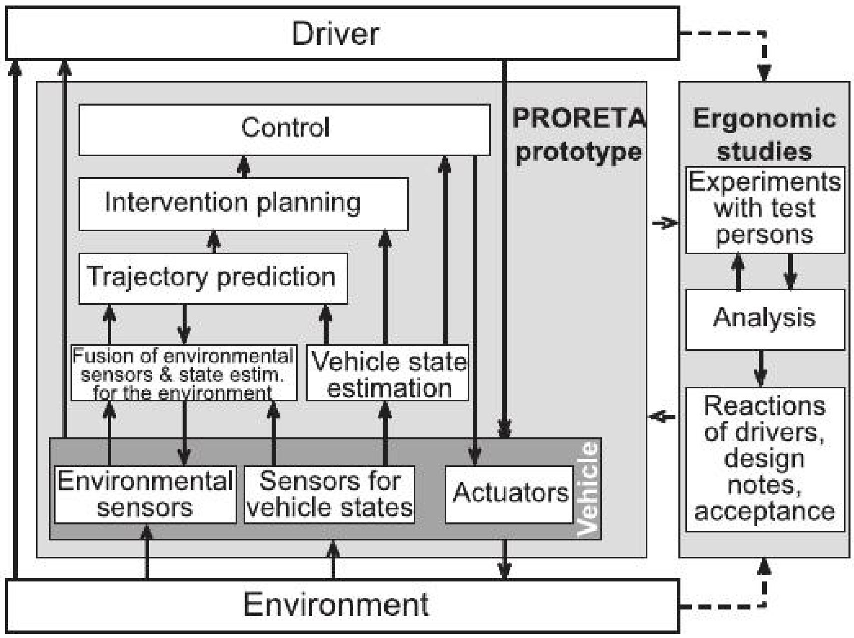


Figure 6.7: Collision-avoidance system overview for the development of PRORETA 1 (The figure is taken from [106] with permission.)

Broker, 3. Reasoner, 4. Human-Machine and Machine-Machine Interface as illustrated in Figure 6.8. Utility-based Knowledge Exchange component is designed for V2V communication. Besides, Knowledge Broker resembles a data set that stores all the relevant knowledge for later use. Reasoner is the main component of the architecture that deals with different aspects of situational information such as uncertainty or semantics. In SADAS, reasoning is used for learning, hazard detection, prediction, assessment of partner's knowledge, and consistency check. Finally, HMI/MMI component warns the driver about any dangerous situation by accepting inputs from the Reasoner component.

Schubert *et al.* in [197] propose an automatic lane-change maneuvers for intelligent transportation. They deploy some sensors including two radars and two cameras (front and rear) and a CAN for vehicle motion data. Signal processing methods are used for vehicle and lane detection and the authors use Unscented Kalman Filter (UKF) for low level data fusion and estimating the trajectory of the ego and surrounding vehicles. Furthermore, Bayesian Networks (BN) are used for situation assessment and relationships recognition. State estimation between different nodes in BN is constructed based on some evaluation criteria, namely, Deceleration to Safety Time (DST), and likelihood functions defined based

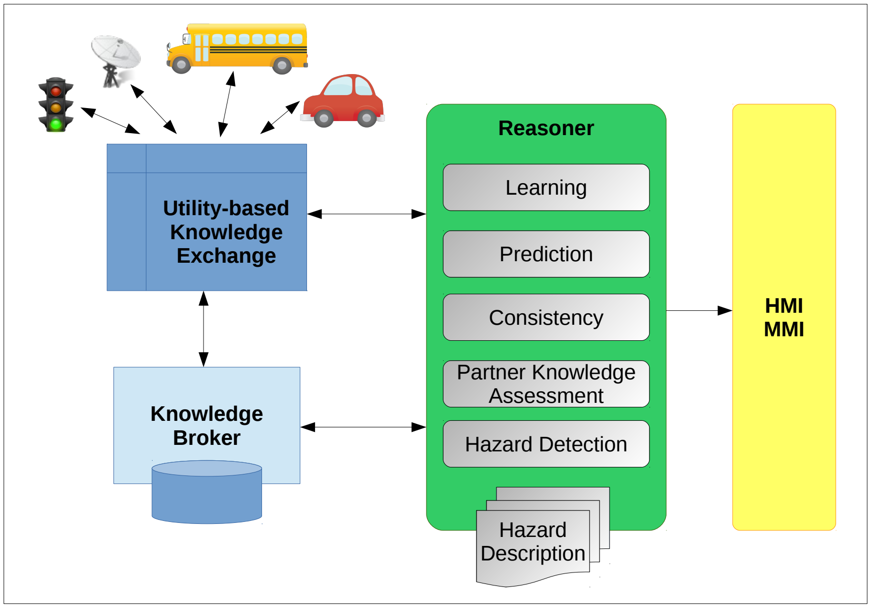


Figure 6.8: SADAS core components [185]

on that.

In [195], authors propose a Cooperative ADAS that tackles blind spot assistance. It uses information received by V2X communication to enhance the perception in case of limited perception range. Attention monitoring systems [133, 213, 27, 227] are also among methods that mainly employ SAW to assess driver situation accordingly.

To evaluate SAW in a connected cars context, two basic estimates for assessing safety impact, namely, analytical approach, expert knowledge approaches [136], can be used. As another evaluation paradigm, Ledoux and Archer [128] introduce the SINDI project that utilizes a bottom-up approach to measure the changes in driving patterns based on a human-behavior model.

To include V2X communication, [196] introduces a language for representing relevant parameters in V2V and V2I information transmission. The language is equipped with grammars, runtime processes, and a build system that make it useful for defining relationships between entities and rule creation.

In [48], authors investigate how V2X communication could be used for sharing decision making between a group of cooperative cars such as a platoon. The cooperative driving mainly deals with collision avoidance and optimal path planning scenarios.

6.3 Attention Assist Framework

The AAF encompasses the major aspects of a transportation system and models them using the HLIF techniques [82]. Moreover, it takes advantage of the wide range of different information sources provided by the VANET platform, and tackles major issues regarding the HLIF systems, namely, semantic relationships, uncertainty management, and ambiguity handling. A block diagram of the proposed framework is depicted in Figure 6.9. As

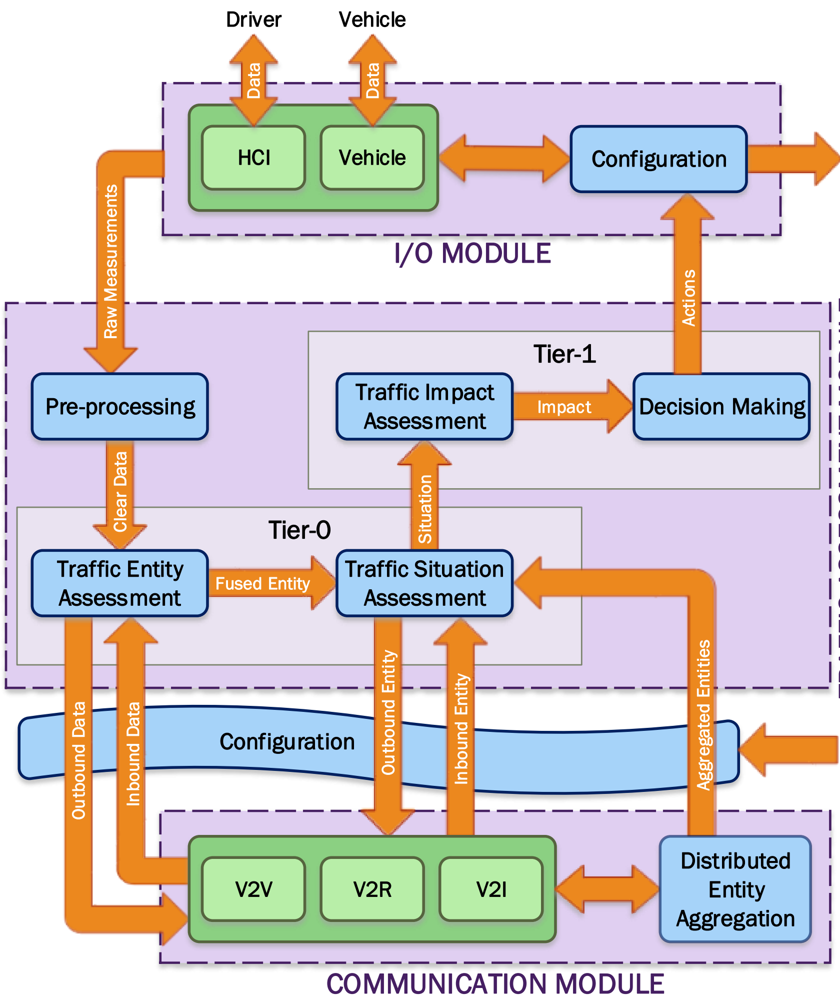


Figure 6.9: The Block Diagram of the Attention Assist Framework (AAF)

shown in Figure 6.9, three basic modules, namely I/O Module (IOM), Traffic Assessment Module (TAM), and Communication Module (CM), are in collaboration with each other to implement the whole process. The TAM is the main module, which is being fed by the local and global data originated from the IOM and the CM, respectively. The local data, such as that of the vehicle itself or the one generated by its driver, are supplied by the IOM, whereas the global data/information is obtained through V2V, V2R and V2I communication. Consequently, various local and global entities are refined and used as the inputs of the Traffic Situation Assessment (TSA) unit. Upon assessing a situation, the outcome is provided to the Traffic Impact Assessment (TIA) unit, and after having its impact assessed, the situation/impact pair is sent to the Decision Making (DM) unit, which determines the final action(s) to be taken. These actions are then submitted to the IOM to be applied to the HCI (targeted at the driver), or to be exerted onto the vehicle. In the following, the entire process is described by detailing the involved units.

The traffic-related entities created in the TEA unit through Low-Level Data Fusion (LLDF), along with the entities received through communication links are used for assessing various situations of interest and their threats. These situations mainly include those that represent the status of the vehicle, its driver, and their surrounding environment.

6.3.1 I/O Module (IOM)

The IOM is responsible for acquiring local data. The local sources of data are mainly the vehicle and its driver. Moreover, in order to obtain the vehicle and/or driver specific preferences, a Configuration Unit (CU) is also envisioned, which is in direct contact with an interface, and keeps track of both the vehicle and its driver behaviors and stores user-specific and vehicle-dependent preferences in order to enable feeding the system with the relevant data.

6.3.2 Traffic Assessment Module (TAM)

As mentioned earlier, the inputs to this module could be either local and/or global and supplied through the IOM and CM, respectively. The raw measurement input data from local sources is first processed in the Pre-processing (PP) unit, and then the clean (*e.g.*, denoised) data goes to the Traffic Entity Assessment (TEA) unit. The inbound global data, gained from the CM through V2V and V2I communication, is also fed to the TEA unit. Inversely, the TEA unit sends its outcome to the CM, where it is broadcast to the other vehicles (in the VANET), as well as the other relevant parts of the road infrastructure.

Traffic Situation Assessment Unit

The TSA unit plays a major role in the TAM. The TSA unit accepts inputs as refined entities from the TEA unit, and inbound situation entities along with aggregated entities from the CM, evaluates the current traffic situation, and finally sends it out to the TIA unit to perform the impact assessment phase.

The TSA unit consists of a modelling phase in which different entities are organized using the domain-specific knowledge. With an expert's help, the situation-related entities are grouped based on their domain of origin, *i.e.*, vehicle, environment, driver, or VANETs (see Figure 2.2), as well as their relationship types, *i.e.*, causal or semantic relations.

Tier-0 of the proposed ATFY-MEBN is the underlying modelling approach (see Chapter 4) for the TSA unit. From ATFY-MEBN standpoint, the entities contributing to a particular domain-specific context, are included within a specific ATFY-MFrag. These entities can act as either the fuzzy resident nodes or the input nodes of an ATFY-MFrag. The resident nodes can be fed along with the evidence gathered from the environment, which in the TSA unit structure are either the local refined entities, global inbound entities, or aggregated entities. The semantic relations between entities in the same Fuzzy-MFrag, or those in two separate ones, are also modelled through defining context nodes. Finally, the ATFY-MEBN inference constructs a minimal SSFBN and then a standard Bayesian inference algorithm is applied to compute the marginal distribution for the entities of interest given the evidence. The outcome of this process, which shows the likelihood of the situation, is then provided to the TIA unit. Upon measuring the impact of the assessed situation in the TIA unit, the best action can be chosen by the Decision Making (DM) unit, and finally presented to the IOM to be displayed on the driver's HCI (e.g. visual notifications), or exerted through the vehicle's physical interface (e.g. reducing speed by applying brakes).

Traffic Impact Assessment Unit

The TIA unit is located next on the traffic assessment path as it accepts the assessed situations from the TSA unit. A well-designed TIA unit should be able to generate a set of hypothesized situations, and send them to the DM unit. One applicable approach can be the idea of situation evolution towards temporal or lateral dimensions (see Figure 6.10 and [87] for more information). Accordingly, an internal method needs to be defined to make the situation evolution happen. Tier-1 of the ATFY-MEBN presented in Chapter 5 is responsible of modeling the basis of the TIA unit. The core of the TIA unit is implemented through an arrangement of action nodes and game nodes, and a pre-defined set of actions,

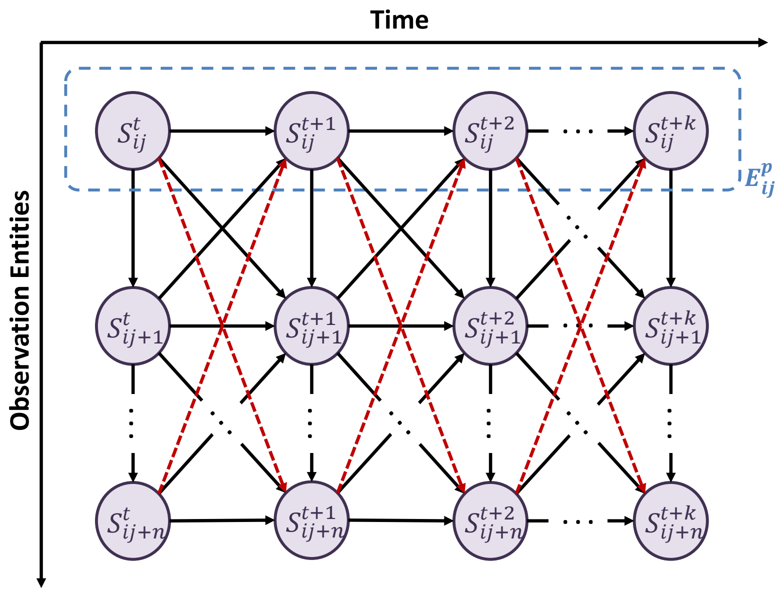


Figure 6.10: Super-situation structure and its composing component situations

whose collaborations results in the generation of a set of hypothesized situations that are evolved towards both lateral and temporal dimensions. These situations are subsequently fed to the DM unit.

Decision Making Unit

The DM unit analyzes the set of situations hypothesized by the TIA unit, and chooses the one that returns the maximum payoff (according to a utility function). Subsequently, a proper decision needs to be made upon determining the most profitable future situation.

It is very important for a DM method to also consider other active agents, if any, as their favorable (or opposing) actions may result in a whole different approach for making the best action. For example, if a vehicle changes its lane to left, without considering an approaching neighboring vehicle on its left (which is probably trying to take it over), it will possibly be in a more dangerous situation that what it is currently in. Therefore, it is critical for a DM method to consider other active entities, in such situations. The DM unit implemented in Chapter 5 is fundamentally based on the concepts of game theory that inherently handle multi-players decision making.

6.3.3 Communication Module (CM)

The Communication Module (CM) is composed of a set of components that enables the whole framework to be in contact with other information sources in the environment. Since the deployment platform is a VANET, two ports for communication with other vehicles (V2V) and infrastructure (V2I) are designed to collect inbound data and situation entities from these sources. Besides, the Distributed Entity Aggregation (DEA) unit relies on distributed signal processing methodologies, e.g. consensus protocols [60], to compute an aggregation of the available global data/information if desired, and send the results to the TSA unit for traffic assessment. Lastly, the configuration data provided by the CU is deployed as a filtering mechanism to screen both the incoming and outgoing data/information.

6.4 Discussion

An Active Driver Assistance System needs to be verified from different aspects. We mainly evaluated the major units of AAF in a safety application in the IoC by having a Collision Warning System (CWS) instantiated from its abstract model. The CWS can be utilized in various driving scenarios through which its different capabilities can be measured.

Moreover, The HLIF-related evaluation criteria groups introduced by Costa *et al.* in [55] can be interpreted by using the results presented in the previous sections. The first group of this categorization, called the Input criteria, contains the measures that influence the way observations are provided to the deployed system. The criteria in this group are: *Relevance*, *Weight of Evidence*, and *Credibility*. Representation criteria is the second group that encompass the measures which target the *Knowledge* and *Evidence Handling* capability of an HLIF system. Finally, the third group, named Reasoning criteria, contains the subclasses of *Correctness*, *Consistency*, *Scalability*, and *Computational Cost*.

The proposed AAF is specially deployed in VANET-related tasks, which by nature, benefit from a broad range of data and information sources. The semantically *relevant* data and information sources are grouped into MEBN Fragments to construct a particular entity. Furthermore, a subset of them are put together to model a specific situation. Such capability of AAF assures that only the *relevant* inputs enter the system and other (irrelevant) ones are kept away from the reasoning system. Besides, it is also clear in the assessments that these evidences *correctly* influence the certainty of a situation. As an important criteria of the second group, knowledge and evidence handling of AAF is also perfectly managed by expressiveness, and uncertainty management capabilities of

MEBN. Finally, the results presented in Table 4.6 show that AAF performs *efficiently* in assessing the collision threat when run in different scenarios. Additionally, AAF is able to *consistently* work in different scenarios with varying number of available evidence and still assess the desired situation. Lastly, the hybrid version of MEBN inference algorithm helps improve the *scalability* of the framework in such a way that it bounds the exponential order of complexity of the inference algorithm, and handles large VANETs in an efficient way. In other words, assuming that the inference time is $O(c^n)$ for common algorithms such as variable elimination and belief propagation, where n is the total number of nodes throughout the network, and c is a constant, hybrid MEBN inference bounds this time complexity by assuring that n does not go over a certain threshold, which is the total number of neighboring vehicles in our case.

6.5 Summary

This chapter proposed a comprehensive framework, called Attention Assist Framework (AAF) that applies high-level information fusion techniques on VANETs platform to perform situation, threat, and impact assessment, as well as decision making. The proposed framework takes advantage of Active Fuzzy Multi-Entity Bayesian Networks (ATFY-MEBN), which is introduced in detail in the following chapters.

In the modelling procedure, the inattention-related entities along with their causal and semantic relationships were identified first, and then were modelled in specific contexts using the proposed ATFY-MEBN-based framework.

In order to show the capabilities of the framework, we also implemented a collision warning system based on the AAF to measure the likelihood of a vehicle being in a near-collision situation while using a wide range of information sources made available through VANET platform. The information sources supplied to the system come from the vehicle itself (local information), and also the communication infrastructure of VANET (global information). This framework was capable of assessing a collision threat, and furthermore, alarming the driver if the likelihood of being in a near-collision situation was measured to be high. Accordingly, it could be used to improve the attentiveness of the driver, and consequently enable avoiding potentially fatal accidents. Two distinct groups of driving scenarios were designed and tested on the proposed system, and our simulation results helped to demonstrate the capability of AAF in achieving situation assessment on the road.

At the end, specifying the details of the traffic impact assessment, and decision making units, can be considered as two major potential future works. Furthermore, automatic learning of the MTheory structure and the formation of MFrags can also be seen as future activities. Besides, considering the construction of more complex situations in which the likelihood of both entities and their relationships are taken into account, and addressing the idea situation distance definition, as introduced in [22], can also be considered as the seed for pertinent research work.

Chapter 7

Concluding Remarks and Future Directions

The Internet of Cars is born where the Internet of Things and Vehicular Ad-hoc Networks (VANETs) meet. This results in connectivity and mobility features, which are utilized by various sources to generate data/information with different levels of abstraction, and furthermore, transmit them to the interested entities. As a result, significant amount of research work in this field has been focused on specific areas where data/information plays and important role, *i.e.*, safety, routing, broadcasting, Quality of Service (QoS), and security. Among these research areas, road safety issues are deemed one of the most challenging problems of the Internet of Cars, which is often related to lack of situational awareness, which has been identified as one of the main reasons that lead a driving scenario to an accident.

In this thesis, we tackled situation awareness in the Internet of Cars by proposing a comprehensive active driver assistance system, called Attention Assist Framework, which was able to fully utilize connectivity feature of VANETs through vehicle-to-vehicle, vehicle-to-RSU, and vehicle-to-infrastructure communication links. The introduced framework was composed of four main levels that complete a full information fusion procedure by performing low-level data fusion, situation assessment, impact assessment, and decision making. Each of these parts were modeled with novel methods that solve important challenges in the corresponding information fusion step. The performance of these methods, along with their applicability, were evaluated through running them on various specifically-designed driving scenarios. Our results showed that while each level of the proposed Attention Assist Framework performs well to achieve its goal, it could successfully alleviate a major information fusion-related issue.

The rest of this chapter highlights the contributions made in this thesis, discusses the conclusive remarks, and determines the future directions.

7.1 Contributions Highlights

The main contribution of this thesis was the introduction of the Attention Assist Framework (AAF), which was a novel generic data/information fusion model to achieve enhanced safety in the IoC aiming at improving the road safety by enhancing the drivers' attentiveness. The proposed model could handle various types of low-level data that were available in the VANETs environment (*i.e.*, the data generated by the physical sensors, or those received through different means of communication). Besides, four main levels were implemented in AAF that complete a full information fusion procedure that ultimately resulted in proper situation awareness for the entities of interest. In summary, these levels along with the corresponding contribution at each level were:

- **Entity Assessment:** low-level data fusion framework and cooperative localization using vehicle-to-vehicle communication and data fusion to integrate the available data, and to cooperatively improve the accuracy of the localization information of the vehicles. The model was further improved by estimating the vehicle location using Unscented Transform (UT) along with Sequential Decentralized Extended Kalman (SDEK) filtering.

- **Situation and Threat Assessment (SA/TA):** high-level information fusion using a novel fuzzy extension to multi-entity Bayesian networks that model some imperfect aspects of data such as ambiguity that is an inherent characteristic of human language, and the observations gained from the environment. It was showed that Fuzzy-MEBN incorporated First-order Fuzzy Logic and Fuzzy Bayesian networks to handle ambiguity.

- **Impact Assessment (IA):** situation evolution towards lateral and temporal dimensions, and structural situation analysis. Hierarchical structure of situations helped to manage how entities form component situations, and how they cooperated to make the situations of interest. Besides, situation evolution caused by temporal and lateral alterations were studied and the idea of event was presented.

- **Decision Making (DM):** game-theoretic impact assessment and decision making through situation-specific games generation and solution using the novel active fuzzy multi-entity Bayesian networks. Accordingly, various combinations of actions were generated, future situations were hypothesized, and the best action that gives the maximum payoff was reported. We called this version of Fuzzy-MEBN, AcTive FuzzY-MEBN (ATFY-MEBN).

7.2 Conclusion

In the Low-Level Data Fusion (LLDF) part of the AAF, data fusion and radio-ranging distance measurement techniques were combined with V2V communication in order to improve the location information of the vehicles. This idea was further extended by adding sequential EKF filtering within the neighboring vehicles to improve the localization performance. The methods were then evaluated by comparing vehicle's estimated locations with their ground truth. The results demonstrated that using V2V communication for measuring the distance and sharing belief about the estimation of the current location, the neighboring vehicles could cooperatively improve the knowledge of the current location.

In chapter 2, we conducted a comprehensive study on different methods used for achieving complete situation awareness in the Internet of Cars through passing the three main steps of perception, comprehension, and projection. As it is discussed in chapter 2, the Multi-Entity Bayesian Networks (MEBN) model is among the most efficient approaches to perform situation awareness in the Internet of Cars (see Table 2.1). This is the main reason we opted MEBN as the core of our information fusion framework. However, as we also discussed in chapters 4 and 5, MEBN lack the capability of handling the ambiguity inherent in human language, as well as predicting the future of the situations of interest. We overcame these issues by importing fuzzy logic and game theory into MEBN and introduce ATFY-MEBN that was capable of accepting a wider range of data/information, including soft data, as well as, predicting the future situations, and proposing a proper action through generating and solving situation-specific games.

Referring to the discussion made in chapter 2, the ATFY-MEBN can now be placed below Fuzzy-MEBN in Table 2.1, and while inheriting all the features of Fuzzy-MEBN, have its Game Theory (GT) one check-marked. This is a considerably major step towards having a comprehensive model for high-level information fusion, as the new model is now able to perform prediction.

7.3 Future Work

The future of Situation Awareness (SAW) in the Internet of Cars can be simply attributed to the future of SAW in general. Here, we outline major future trends in SAW, which have recently attracted researchers.

7.3.1 Cloud-Enabled SAW

In cloud-enabled SAW, or Vehicular Cloud, the data/information resources of a certain vehicle with those of other cars can be pooled and drivers can process data on demand at anytime from anywhere [129]. This new paradigm was first introduced by M. Gerla [76]. The cloud is constructed by collaborations among cloud, cars, and RSUs which enables the fusion and sharing of databases. When the driver is not behind the wheels, the processing will not be in the car but in the cloud and between neighboring smart cars to offer traffic map, appropriate paths without obstacles. The main purpose is to provide drivers and passengers in roads with advanced vehicular services that individual cars cannot offer. Nevertheless, vehicular cloud poses many issues to SAW, which impact the following axes [232]: context discovery, acquisition and dissemination, situation analysis and recognition, and situation-triggered response.

7.3.2 Cognitive SAW

According to M. R. Endsley [68], common setting of a SAW model mainly concentrates on the physical and perceptual attributes of its human part, and does not take his cognitive abilities into account. In Cognitive SAW (CSAW), a system is designed in a way to take advantage of human's information processing power. Thus, CSAW handles important cognitive aspects such as perception and attention, working memory, mental models, scripts, and schema, and finally goals. In fact, a CSAW imitates how cognition is performed in humans' mental model, and aims to facilitate it. Crucially, cognitive SAW is important when the complexity of the situations increases in a way that is difficult for humans to become aware of it appropriately.

7.3.3 Distributed SAW

In a nutshell, Distributed SAW (DSAW) is the third perspective in SAW proposed by [189], and is defined as a set of interacting entities wherein each entity has its own situation awareness, which may be different from (or even in conflict with) that of others [206]. Therefore, DSAW is a dynamic and collaborative process that leads to shared awareness. In a well-designed DSAW system, the situational awareness is defined as the combination of individual entities' SAW. Besides, entities abilities to perform their assigned tasks should not be dependant on others' SAW. Moreover, sharing SAW needs to be performed intelligently and upon request, to avoid the propagation of misleading and confusing SAW. One interesting architecture for a DSAW can be a set of agents that each are assigned to a specific SAW component introduced earlier in this paper.

References

[1] http://www.asirt.org/.

[2] More than 50 Billion Connected Devices, White paper. Technical report, Ericsson, 2011.

[3] The Internet of Things: How the Next Evolution of the Internet is Changing Everything. Technical report, Cisco, 2011.

[4] The Smart/Connected City and Its Implications for Connected Transportation. Technical report, U.S. Department of Transportation John A Volpe National Transportation Systems Center, October 2014.

[5] Juergen Ackermann. Robust control prevents car skidding. *Control Systems, IEEE*, 17(3):23–31, 1997.

[6] Farhan Ahammed, Javid Taheri, AlbertY. Zomaya, and Max Ott. Vloci: Using distance measurements to improve the accuracy of location coordinates in gps-equipped vanets. In Patrick Sénac, Max Ott, and Aruna Seneviratne, editors, *Mobile and Ubiquitous Systems: Computing, Networking, and Services*, volume 73 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 149–161. Springer Berlin Heidelberg, 2012.

[7] Hüseyin Akcan and Cem Evrendilek. Gps-free directional localization via dual wireless radios. *Computer Communications*, 35(9):1151 – 1163, 2012. Special Issue: Wireless Sensor and Robot Networks: Algorithms and Experiments.

[8] Milad AlemZadeh. *Semantic Analysis of Wikipedia's Linked Data Graph for Entity Detection and Topic Identification Applications.* Thesis, University of Waterloo Department of Electrical and Computer Engineering, 2012.

[9] Milad AlemZadeh, Richard Khoury, and Fakhri Karray. Query Classification using Wikipedia's Category Graph. *Journal of Emerging Technologies in Web Intelligence*, 4(3):207–220, 2012.

[10] Pongtep Angkititrakul, Ryuta Terashima, and Toshihiro Wakita. On the use of stochastic driver behavior model in lane departure warning. *Intelligent Transportation Systems, IEEE Transactions on*, 12(1):174–183, 2011.

[11] G. S. Aoude, B. D. Luders, and J. P. How. Sampling-Based Threat Assessment Algorithms for Intersection Collisions Involving Errant Drivers. In *Proceedings of the IFAC Symposium on Intelligent Autonomous Vehicles*, pages 581–586, Lecce, Italy, September 2010.

[12] Georges S Aoude and Jonathan P How. Using Support Vector Machines and Bayesian Filtering for Classifying Agent Intentions at Road Intersections. Technical report, 2009.

[13] Georges S Aoude, Brandon D Luders, Kenneth KH Lee, Daniel S Levine, and Jonathan P How. Threat Assessment Design for Driver Assistance System at Intersections. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pages 1855–1862. IEEE, 2010.

[14] G.S. Aoude, V.R. Desaraju, L.H. Stephens, and J.P. How. Driver Behavior Classification at Intersections and Validation on Large Naturalistic Data Set. *Intelligent Transportation Systems, IEEE Transactions on*, 13(2):724–736, June 2012.

[15] Ribal F. Atallah, Maurice J. Khabbaz, and Chadi M. Assi. Vehicular networking: A survey on spectrum access technologies and persisting challenges. *Vehicular Communications*, 2(3):125 – 149, 2015.

[16] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The Internet of Things: A survey. *Computer Networks*, 54(15):2787–2805, 2010.

[17] Franz Baader and Werner Nutt. Basic Description Logics. In *Description logic handbook*, pages 43–95, 2003.

[18] Javier Barrachina, Piedad Garrido, Manuel Fogue, Francisco J Martinez, Juan-Carlos Cano, Carlos T Calafate, and Pietro Manzoni. VEACON: A Vehicular Accident Ontology Designed to Improve Safety on the Roads. *Journal of Network and Computer Applications*, 35(6):1891–1900, 2012.

[19] J. Barwise. *The Situation in Logic*. CSLI Lecture Notes. Cambridge University Press, 1988.

[20] Jon Barwise. Scenes and Other Situations. *The Journal of Philosophy*, 78(7):pp. 369–397, 1981.

[21] Jon Barwise and John Perry. Situations and Attitudes. *The Journal of Philosophy*, 78(11):pp. 668–691, 1981.

[22] Norbert Baumgartner, Wolfgang Gottesheim, Stefan Mitsch, Werner Retschitzegger, and Wieland Schwinger. BeAware!—Situation awareness, the ontology-driven way. *Data and Knowledge Engineering*, 69(11):1181–1193, 2010. Special issue on contribution of ontologies in designing advanced information systems.

[23] Norbert Baumgartner and Werner Retschitzegger. A Survey of Upper Ontologies for Situation Awareness. In *In Proceedings of the 4th IASTED International Conference on Knowledge Sharing and Collaborative Engineering*, pages 1–9, 2006.

[24] Norbert Baumgartner, Werner Retschitzegger, and Wieland Schwinger. Application Scenarios of Ontology-Driven Situation Awareness SystemsExemplified for the Road Traffic Management Domain. In *Proceedings of the 2008 Conference on Formal Ontologies Meet Industry*, pages 77–87, Amsterdam, The Netherlands, The Netherlands, 2008. IOS Press.

[25] Falko Bause and Pieter S Kritzinger. *Stochastic Petri Nets*. Springer, 1996.

[26] A. Benavoli, B. Ristic, Alfonso Farina, M. Oxenham, and L. Chisci. An Approach to Threat Assessment Based on Evidential Networks. In *Information Fusion, 2007 10th International Conference on*, pages 1–8, July 2007.

[27] L.M. Bergasa, D. Almeria, J. Almazan, J.J. Yebes, and R. Arroyo. DriveSafe: An app for alerting inattentive drivers and scoring driving behaviors. In *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pages 240–245, June 2014.

[28] Tim Berners-Lee, James Hendler, Ora Lassila, et al. The semantic web. *Scientific american*, 284(5):28–37, 2001.

[29] Navin AR Bhat and Kevin Leyton-Brown. Computing nash equilibria of action-graph games. In *Proceedings of the 20th conference on Uncertainty in Artificial Intelligence*, pages 35–42. AUAI Press, 2004.

[30] Rahul Biswas, Sebastian Thrun, and Kikuo Fujimura. Recognizing Activities with Multiple Cues. In *Proceedings of the 2Nd Conference on Human Motion: Understanding, Modeling, Capture and Animation*, pages 255–270, Berlin, Heidelberg, 2007. Springer-Verlag.

[31] E. Blasch, J. Llinas, D. Lambert, P. Valin, S. Das, Chee Chong, M. Kokar, and E. Shahbazian. High Level Information Fusion developments, issues, and grand challenges: Fusion 2010 panel discussion. In *Information Fusion (FUSION), 2010 13th Conference on*, pages 1–8, july 2010.

[32] E. Blasch and S. Plano. Proactive decision fusion for site security. In *Information Fusion, 2005 8th International Conference on*, volume 2, pages 8 pp.–, July 2005.

[33] E. Blasch, P. Valin, E. Bosse, M. Nilsson, J. van Laere, and E. Shahbazian. Implication of culture: User roles in information Fusion for enhanced situational understanding. In *Information Fusion, 2009. FUSION '09. 12th International Conference on*, pages 1272–1279, July 2009.

[34] E. Blasch, P. Valin, A.-L. Jousselme, D. Lambert, and E. Bosse. Top ten trends in High-Level Information Fusion. In *Information Fusion (FUSION), 2012 15th International Conference on*, pages 2323–2330, july 2012.

[35] Erik Blasch, Ivan Kadar, John Salerno, Mieczyslaw M. Kokar, Subrata Das, Gerald M. Powell, Daniel D. Corkill, and Enrique H. Ruspini. Issues and Challenges of Knowledge Representation and Reasoning Methods in Situation Assessment (Level 2 Fusion). *Proc. SPIE*, 6235:10–14, 2006.

[36] Erik Blasch, Ivan Kadar, John Salerno, Mieczyslaw M. Kokar, Gerald M. Powell, Daniel D. Corkill, and Enrique H. Ruspini. Issues and Challenges in Situation Assessment (Level 2 Fusion). *Journal of Advances in Information Fusion*, 1(2):122–139, November 2006.

[37] Erik Blasch, Kathryn B Laskey, Gee Wah Ng, Rakesh Nagi, Dafni Stampouli, and Johan Schubert. Issues of Uncertainty Analysis in High-Level Information Fusion. pages 1–12, 2012.

[38] Erik P Blasch and Susan Plano. JDL Level 5 Fusion Model: User Refinement Issues and Applications in Group Tracking. In *AeroSense 2002*, pages 270–279. International Society for Optics and Photonics, 2002.

[39] Jeremy J Blum, Azim Eskandarian, and Lance J Hoffman. Challenges of intervehicle ad hoc networks. *Intelligent Transportation Systems, IEEE Transactions on*, 5(4):347–351, 2004.

[40] Vanderlei Bonato, Eduardo Marques, and George A Constantinides. A floating-point extended kalman filter implementation for autonomous mobile robots. *Journal of Signal Processing Systems*, 56(1):41–50, 2009.

[41] Azzedine Boukerche, Horacio ABF Oliveira, Eduardo F Nakamura, and Antonio AF Loureiro. Vehicular ad hoc networks: A new challenge for localization-based systems. *Computer communications*, 31(12):2838–2849, 2008.

[42] Ronald J Brachman and James G Schmolze. An Overview of the KL-ONE Knowledge Representation System. *Cognitive science*, 9(2):171–216, 1985.

[43] Adrian Broadhurst, Simon Baker, and Takeo Kanade. A Prediction and Planning Framework for Road Safety Analysis, Obstacle Avoidance and Driver Information. Technical report, Carnegie Mellon University - Robotics Institute, 2004.

[44] Joel Brynielsson and Stefan Arnborg. An Information Fusion Game Component. *J. Adv. Inf. Fusion*, 1(2):108–121, 2006.

[45] Lames J Caffery Jr and Gordon L Stüber. Overview of radiolocation in cdma cellular systems. *Communications Magazine, IEEE*, 36(4):38–45, 1998.

[46] R. Carvalho, KB Laskey, P. Costa, M. Ladeira, L. Santos, and S. Matsumoto. UnBBayes: modeling uncertainty for plausible reasoning in the semantic web. *Semantic Web, IN-TECH Publishing, ISBN*, pages 953–978, 2010.

[47] R.N. Carvalho, R. Haberlin, P.C.G. Costa, K.B. Laskey, and K.C. Chang. Modeling a probabilistic ontology for Maritime Domain Awareness. In *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*, pages 1–8, july 2011.

[48] D. Caveney and W.B. Dunbar. Cooperative driving: Beyond V2V as an ADAS sensor. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 529–534, June 2012.

[49] Lubin Chang, Baiqing Hu, An Li, and Fangjun Qin. Transformed unscented kalman filter. *Automatic Control, IEEE Transactions on*, 58(1):252–257, 2013.

[50] Genshe Chen, Dan Shen, Chiman Kwan, Jose B. Cruz, and Martin Kruger. Game Theoretic Approach to Threat Prediction and Situation Awareness. *Journal of Advances in Information Fusion*, 2(1):35–48, June 2007.

[51] Tingting Chen, Liehuang Zhu, Fan Wu, and Sheng Zhong. Stimulating Cooperation in Vehicular Ad Hoc Networks: A Coalitional Game Theoretic Approach. *Vehicular Technology, IEEE Transactions on*, 60(2):566–579, Feb 2011.

[52] Wai Chen, R.K. Guha, T. Kwon, J. Lee, and I.Y. Hsu. A survey and challenges in routing and data dissemination in vehicular ad-hoc networks. In *Vehicular Electronics and Safety, 2008. ICVES 2008. IEEE International Conference on*, pages 328–333, Sept 2008.

[53] Hsu-Yung Cheng, Bor-Shenn Jeng, Pei-Ting Tseng, and K-C Fan. Lane detection with moving vehicles in the traffic scenes. *Intelligent Transportation Systems, IEEE Transactions on*, 7(4):571–582, 2006.

[54] Madhav V Chitturi, Juan C Medina, and Rahim Ray F Benekohal. Effect of shadows and time of day on performance of video detection systems at signalized intersections. *Transportation research part C: emerging technologies*, 18(2):176–186, 2010.

[55] P.C.G. Costa, K.B. Laskey, E. Blasch, and A.-L. Jousselme. Towards unbiased evaluation of uncertainty reasoning: The URREF ontology. In *Information Fusion (FUSION), 2012 15th International Conference on*, pages 2301–2308, july 2012.

[56] B. Dasarathy. Information Fusion – what, where, why, when, and how? *Information Fusion*, 2(2):75–76, June 2001.

[57] Constantinos Daskalakis, Paul W Goldberg, and Christos H Papadimitriou. The complexity of computing a nash equilibrium. *SIAM Journal on Computing*, 39(1):195–259, 2009.

[58] Keith Devlin. *Logic and information*. Cambridge University Press, 1995.

[59] John Dickhaut and Todd Kaplan. A program for finding nash equilibria. *The Mathematica Journal*, 1(4):87–93, 1991.

[60] A.G. Dimakis, S. Kar, J.M.F. Moura, M.G. Rabbat, and A. Scaglione. Gossip Algorithms for Distributed Signal Processing. *Proceedings of the IEEE*, 98(11):1847–1864, nov. 2010.

[61] Thomas A Dingus, SG Klauer, VL Neale, A Petersen, SE Lee, JD Sudweeks, MA Perez, J Hankey, DJ Ramsey, S Gupta, et al. The 100-car naturalistic driving study, Phase II-results of the 100-car field experiment. Technical report, 2006.

[62] MWM Gamini Dissanayake, Paul Newman, Hugh F Durrant-Whyte, Steve Clark, and M Csorba. An experimental and theoretical investigation into simultaneous localisation and map building. In *Experimental Robotics VI*, pages 265–274. Springer, 2000.

[63] Yue Dongli, Wang Suihua, and Zhao Ailing. Traffic Accidents Knowledge Management Based on Ontology. In *Fuzzy Systems and Knowledge Discovery, 2009. FSKD'09. Sixth International Conference on*, volume 7, pages 447–449. IEEE, 2009.

[64] A. Doshi and M.M. Trivedi. On the Roles of Eye Gaze and Head Dynamics in Predicting Driver's Intent to Change Lanes. *Intelligent Transportation Systems, IEEE Transactions on*, 10(3):453–462, Sept 2009.

[65] Amit Dua, Neeraj Kumar, and Seema Bawa. A systematic review on routing protocols for vehicular ad hoc networks. *Vehicular Communications*, 1(1):33 – 52, 2014.

[66] Mica R Endsley. Design and evaluation for situation awareness enhancement. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 32, pages 97–101. SAGE Publications, 1988.

[67] Mica R Endsley. Situation awareness and human error: Designing to support human performance. In *Proceedings of the High Consequence Systems Surety Conference*, pages 2–9. Lawrence Eribaum Associates, 1999.

[68] Mica R Endsley. *Designing for situation awareness: An approach to user-centered design*. CRC Press, 2011.

[69] Mica R Endsley et al. Theoretical Underpinnings of Situation Awareness: A Critical Review. *Situation awareness analysis and measurement*, pages 3–32, 2000.

[70] Mica R Endsley and Daniel J Garland. Pilot situation awareness training in general aviation. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 44, pages 357–360. SAGE Publications, 2000.

[71] Azim Eskandarian. Fundamentals of Driver Assistance. In Azim Eskandarian, editor, *Handbook of Intelligent Vehicles*, pages 491–535. Springer London, 2012.

[72] Nour-Eddin El Faouzi, Henry Leung, and Ajeesh Kurian. Data fusion in intelligent transportation systems: Progress and challenges — a survey. *Information Fusion*, 12(1):4–10, 2011.

[73] Christopher Fogelberg, Vasile Palade, and Phil Assheton. Belief propagation in fuzzy bayesian networks. In *1st International Workshop on Combinations of Intelligent Methods and Applications (CIMA)*, pages 19–24, 2008.

[74] Tarak Gandhi and Mohan M Trivedi. Vehicle surround capture: Survey of techniques and a novel omni-video-based approach for dynamic panoramic surround maps. *Intelligent Transportation Systems, IEEE Transactions on*, 7(3):293–308, 2006.

[75] Leonidas Georgiadis, Michael J Neely, and Leandros Tassiulas. *Resource Allocation and Cross-layer Control in Wireless Networks*. Now Publishers Inc, 2006.

[76] M. Gerla. Vehicular Cloud Computing. In *Ad Hoc Networking Workshop (Med-Hoc-Net), 2012 The 11th Annual Mediterranean*, pages 152–155, June 2012.

[77] Mario Gerla, Eun-Kyu Lee, Giovanni Pau, and Uichin Lee. Internet of Vehicles: From Intelligent Grid to Autonomous Cars and Vehicular Clouds. In *IEEE World Forum on Internet of Things*, March 2014.

[78] Mario Gerla, Chuchu Wu, Giovanni Pau, and Xiaoqing Zhu. Content distribution in {VANETs}. *Vehicular Communications*, 1(1):3 – 12, 2014.

[79] Anouck Renée Girard, J Borges de Sousa, James A Misener, and J Karl Hedrick. A control architecture for integrated cooperative cruise control and collision warning systems. In *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, volume 2, pages 1491–1496. IEEE, 2001.

[80] K. Golestan, A. Jundi, L. Nassar, F. Sattar, F. Karray, M. Kamel, and S. Boumaiza. Vehicular Ad-hoc Networks(VANETs): Capabilities, Challenges in Information Gathering and Data Fusion. In *Autonomous and Intelligent Systems*, Lecture Notes in Computer Science, pages 34–41. Springer Berlin Heidelberg, 2012.

[81] K. Golestan, F. Karray, and M.S. Kamel. Fuzzy Multi Entity Bayesian Networks: A Model for Imprecise Knowledge Representation and Reasoning in High-level Information Fusion. In *Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on*, pages 1678–1685, July 2014.

[82] K. Golestan, B. Khaleghi, F. Karray, and M.S. Kamel. Attention assist: A high-level information fusion framework for situation and threat assessment in vehicular ad hoc networks. *Intelligent Transportation Systems, IEEE Transactions on*, PP(99):1–15, 2015.

[83] K. Golestan, F. Sattar, F. Karray, M. Kamel, and S. Seifzadeh. Localization in vehicular ad hoc networks using data fusion and {V2V} communication. *Computer Communications*, 71:61 – 72, 2015.

[84] Keyvan Golestan, Fakhri Karray, and Mohamed S Kamel. High Level Information Fusion Through a Fuzzy Extension to Multi-Entity Bayesian Networks in Vehicular Ad-hoc Networks. In *Information Fusion (FUSION), 2013 16th International Conference on*, pages 1180–1187. IEEE, 2013.

[85] Keyvan Golestan, Fakhri Karray, and Mohamed S. Kamel. An integrated approach for fuzzy multi-entity bayesian networks and semantic analysis for soft and hard data fusion. In *Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on*, 2015.

[86] Keyvan Golestan, Sepideh Seifzadeh, Mohamed Kamel, Fakhri Karray, and Farook Sattar. Vehicle localization in VANETs Using Data Fusion and V2V Communication. In *Proceedings of the second ACM international symposium on Design and analysis of intelligent vehicular networks and applications*, DIVANet '12, pages 123–130, New York, NY, USA, 2012. ACM.

[87] Keyvan Golestan, Ridha Soua, Fakhri Karray, and Mohamed S Kamel. A Model for Situation and Threat/Impact Assessment in Vehicular Ad-hoc Networks. In *Proceedings of the fourth ACM international symposium on Development and analysis of intelligent vehicular networks and applications*, pages 87–94. ACM, 2014.

[88] G.H. Golub and C.F. Van Loan. *Matrix Computations*. Matrix Computations. Johns Hopkins University Press, 2012.

[89] Juan Gómez-Romero, Miguel A Serrano, Jesús García, José M Molina, and Galina Rogova. Context-based multi-level information fusion for harbor surveillance. *Information Fusion*, 21:173–186, 2015.

[90] A. Gray, M. Ali, Yiqi Gao, J.K. Hedrick, and F. Borrelli. A Unified Approach to Threat Assessment and Control for Automotive Active Safety. *Intelligent Transportation Systems, IEEE Transactions on*, 14(3):1490–1499, Sept 2013.

[91] Joaquín Gutiérrez, Dimitrios Apostolopoulos, and José Luis Gordillo. Numerical comparison of steering geometries for robotic vehicles by modeling positioning error. *Autonomous Robots*, 23(2):147–159, 2007.

[92] D.L. Hall, M. McNeese, J. Llinas, and T. Mullen. A framework for dynamic hard/soft fusion. In *Information Fusion, 2008 11th International Conference on*, pages 1–8, 30 2008-july 3 2008.

[93] Joseph Y. Halpern. An analysis of first-order logics of probability. *Artificial Intelligence*, 46(3):311–350, 1990.

[94] J. Harri, F. Filali, and C. Bonnet. Mobility models for vehicular ad hoc networks: a survey and taxonomy. *Communications Surveys Tutorials, IEEE*, 11(4):19–41, Fourth 2009.

[95] Hamid R Hashemipour, Sumit Roy, and Alan J Laub. Decentralized structures for parallel kalman filtering. *Automatic Control, IEEE Transactions on*, 33(1):88–94, 1988.

[96] F. Heintz and Z. Dragisic. Semantic Information Integration for Stream Reasoning. In *Information Fusion (FUSION), 2012 15th International Conference on*, pages 1454–1461, july 2012.

[97] J. Hillenbrand, A.M. Spieker, and Kristian Kroschel. A Multilevel Collision Mitigation Approach mdash;Its Situation Assessment, Decision Making, and Performance Tradeoffs. *Intelligent Transportation Systems, IEEE Transactions on*, 7(4):528–540, Dec 2006.

[98] Reiner Hoeger, Angelos Amditis, Martin Kunert, Alfred Hoess, Frank Flemish, Hans-Peter Krueger, Arne Bartels, Achim Beutner, and Katia Pagle. Highly automated vehicles for intelligent transport: Have-it approach. In *15th World Congress on Intelligent Transport Systems and ITS America's 2008 Annual Meeting*, 2008.

[99] F. Holzmann, M. BeHino, S. Kolskit, A. Sulzmann, G. Spiegelberg, and R. Siegwart. Robots go automotive - the SPARC approach. In *Intelligent Vehicle, IEEE Symposium*, 2005.

[100] Haijing Hou, Lisheng Jin, Qingning Niu, Yuqin Sun, and Meng Lu. Driver Intention Recognition Method Using Continuous Hidden Markov Model. *International Journal of Computational Intelligence Systems*, 4(3):386–393, 2011.

[101] C. Howard and M. Stumptner. Situation Assessments Using Object Oriented Probabilistic Relational Models. In *Information Fusion, 2005 8th International Conference on*, volume 2, pages 8 pp.–, July 2005.

[102] Pau-Lo Hsu, Hsu-Yuan Cheng, Bol-Yi Tsuei, and Wen-Jing Huang. The adaptive lane-departure warning system. In *SICE 2002. Proceedings of the 41st SICE Annual Conference*, volume 5, pages 2867–2872. IEEE, 2002.

[103] Jihua Huang and Han-Shue Tan. Error analysis and performance evaluation of a future-trajectory-based cooperative collision warning system. *Intelligent Transportation Systems, IEEE Transactions on*, 10(1):175–180, 2009.

[104] E. Hutchins. *Cognition in the Wild*. Bradford Books. MIT Press, 1995.

[105] J. Illingworth and J. Kittler. A survey of the hough transform. *Computer Vision, Graphics, and Image Processing*, 44(1):87 – 116, 1988.

[106] Rolf Isermann, Roman Mannale, and Ken Schmitt. Collision-avoidance systems PRORETA: Situation analysis and intervention control. *Control Engineering Practice*, 20(11):1236–1246, 2012. Special Section: Wiener-Hammerstein System Identification Benchmark.

[107] Albert Xin Jiang, Kevin Leyton-Brown, and Navin AR Bhat. Action-graph games. *Games and Economic Behavior*, 71(1):141–173, 2011.

[108] Simon J Julier, Jeffrey K Uhlmann, and Hugh F Durrant-Whyte. A new approach for filtering nonlinear systems. In *American Control Conference, Proceedings of the 1995*, volume 3, pages 1628–1632. IEEE, 1995.

[109] M.S. Kakkasageri and S.S. Manvi. Information management in vehicular ad hoc networks: A review. *Journal of Network and Computer Applications*, 39:334 – 350, 2014.

[110] Ioanna Kantzavelou and Sokratis Katsikas. A Game-based Intrusion Detection Mechanism to Confront Internal Attackers. *Computers & Security*, 29(8):859–874, 2010.

[111] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, and T. Weil. Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions. *Communications Surveys Tutorials, IEEE*, 13(4):584–616, Fourth 2011.

[112] Alexander Karlsson. Dependable and generic high-level information fusion: methods and algorithms for uncertainty management. Report (other academic), University of Skövde, School of Humanities and Informatics, 2007.

[113] R. Karlsson, J. Jansson, and F. Gustafsson. Model-based Statistical Tracking and Decision Making for Collision Avoidance Application. In *American Control Conference, 2004. Proceedings of the 2004*, volume 4, pages 3435–3440 vol.4, June 2004.

[114] Fakhreddine O Karray and Clarence W De Silva. *Soft Computing and Intelligent Systems Design: Theory, Tools, and Applications*. Addison-Wesley, 2004.

[115] B. Khaleghi and F. Karray. Random set theoretic soft/hard data fusion framework. *Aerospace and Electronic Systems, IEEE Transactions on*, 50(4):3068–3081, October 2014.

[116] Bahador Khaleghi, Alaa Khamis, Fakhreddine O. Karray, and Saiedeh N. Razavi. Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, 14(1):28–44, 2013.

[117] S.G. Klauer, T. A. Dingus, V. L. Neale, J.D. Sudweeks, and D.J. Ramsey. The Impact of Driver Inattention on Near-Crash/Crash Risk: An Analysis Using the 100-Car Naturalistic Driving Study Data. Technical Report DOT HS 810 594, U.S. Department of Transportation, April 2006.

[118] Graham Klyne and Jeremy J Carroll. Resource description framework (rdf): Concepts and abstract syntax. 2006.

[119] Mieczyslaw M. Kokar, Christopher J. Matheus, and Kenneth Baclawski. Ontology-based situation awareness. *Inf. Fusion*, 10(1):83–98, January 2009.

[120] Mieczyslaw M Kokar, Christopher J Matheus, Jerzy A Letkowski, Kenneth P Baclawski, and Paul Kogut. Association in Level 2 fusion. In *Defense and Security*, pages 228–237. International Society for Optics and Photonics, 2004.

[121] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT press, 2009.

[122] Daphne Koller and Avi Pfeffer. Object-oriented Bayesian Networks. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, pages 302–313. Morgan Kaufmann Publishers Inc., 1997.

[123] Toru Kumagai, Yasuo Sakaguchi, Masayuki Okuwa, and Motoyuki Akamatsu. Prediction of Driving Behavior Through Probabilistic Inference. In *Proc. 8th Intl. Conf. Engineering Applications of Neural Networks*, pages 117–123, 2003.

[124] Dale A. Lambert. A blueprint for higher-level fusion systems. *Information Fusion*, 10(1):6–24, 2009.

[125] Kathryn B. Laskey. MEBN: A language for first-order Bayesian knowledge bases. *Artif. Intell.*, 172(2-3):140–178, 2008.

[126] K.B. Laskey, P.C.G. Costa, and T. Janssen. Probabilistic Ontologies for Knowledge Fusion. In *Information Fusion, 2008 11th International Conference on*, pages 1–8, June 2008.

[127] Gal Lavee, Artyom Borzin, Ehud Rivlin, and Michael Rudzsky. Building Petri Nets from Video Event Ontologies. In George Bebis, Richard Boyle, Bahram Parvin, Darko Koracin, Nikos Paragios, Syeda-Mahmood Tanveer, Tao Ju, Zicheng Liu, Sabine Coquillart, Carolina Cruz-Neira, Torsten Müller, and Tom Malzbender, editors, *Advances in Visual Computing*, volume 4841 of *Lecture Notes in Computer Science*, pages 442–451. Springer Berlin Heidelberg, 2007.

[128] Corinne Ledoux and Jeffery Archer. Assessing the safety impact of intelligent transport systems. *Center for Traffic Engineering and Traffic Simulation, Royal Institute of Technology*, 1999.

[129] Euisin Lee, Eun-Kyu Lee, M. Gerla, and S.Y. Oh. Vehicular cloud networking: architecture and design principles. *Communications Magazine, IEEE*, 52(2):148–155, February 2014.

[130] K-F Lee, H-W Hon, and Raj Reddy. An overview of the sphinx speech recognition system. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 38(1):35–45, 1990.

[131] S. Lefevre, J. Ibanez-Guzman, and C. Laugier. Context-based Estimation of Driver Intent at Road Intersections. In *Computational Intelligence in Vehicles and Transportation Systems (CIVTS), 2011 IEEE Symposium on*, pages 67–72, April 2011.

[132] VI Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707, 1966.

[133] Xuanpeng Li, E. Seignez, Wenjie Lu, and P. Loonis. Vehicle safety evaluation based on driver drowsiness and distracted and impaired driving performance using evidence theory. In *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pages 82–88, June 2014.

[134] M. Liebner, M. Baumann, F. Klanner, and C. Stiller. Driver intent inference at urban intersections using the intelligent driver model. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 1162–1167, June 2012.

[135] M.E. Liggins, D.L. Hall, and P.D. James Llinas. *Handbook of Multisensor Data Fusion: Theory and Practice*. The Electrical Engineering and Applied Signal Processing Series. CRC PressINC, 2009.

[136] G. Lind and Kommunikationsforskningsberedningen. *Test-site-oriented Scenario Assessment: Possible Effects of Transport Telematics in the Göteborg Region : TOSCA II Final Report*. KFB-rapport. Swedish Transport & Communications Research Board, 1996.

[137] Eric G. Little and Galina L. Rogova. Designing ontologies for higher level fusion. *Information Fusion*, 10(1):70–82, 2009. ¡ce:title¿Special Issue on High-level Information Fusion and Situation Awareness¡/ce:title¿.

[138] J. Llinas, R. Nagi, D. Hall, and J. Lavery. A multi-disciplinary university research initiative in hard and soft information fusion: Overview, research strategies and initial results. In *Information Fusion (FUSION), 2010 13th Conference on*, pages 1–7, July 2010.

[139] Mark Locher and Paulo Costa. Ontological Considerations for Uncertainty Propagation in High Level Information Fusion. In *Semantic Technology for Intelligence, Defense, and Security*, 2012.

[140] Ning Lu, Nan Cheng, Ning Zhang, Xuemin Shen, and J.W. Mark. Connected Vehicles: Solutions and Challenges. *Internet of Things Journal, IEEE*, 1(4):289–299, Aug 2014.

[141] Fengjun Lv and R. Nevatia. Single View Human Action Recognition using Key Pose Matching and Viterbi Path Searching. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, June 2007.

[142] Alexander Maedche and Steffen Staab. Measuring Similarity Between Ontologies. In *Knowledge engineering and knowledge management: Ontologies and the semantic web*, pages 251–263. Springer, 2002.

[143] Suzanne M. Mahoney and Kathryn Blackmond Laskey. Constructing situation specific belief networks. In *In Proc. 14th Conf. on Uncertainty in Artificial Intelligence*, pages 370–378, 1998.

[144] P. Makris, D.N. Skoutas, and C. Skianis. A survey on context-aware mobile and wireless networking: On networking and computing environments' integration. *Communications Surveys Tutorials, IEEE*, 15(1):362–386, First 2013.

[145] Francisco J. Martinez, Chai Keong Toh, Juan-Carlos Cano, Carlos T. Calafate, and Pietro Manzoni. A survey and comparative study of simulators for vehicular ad hoc networks (vanets). *Wireless Communications and Mobile Computing*, 11(7):813–828, 2011.

[146] Rafael Math, Angela Mahr, Mohammad M Moniri, and Christian Müller. Opends: A new open-source driving simulator for research. *GMM-Fachbericht-AmE 2013*, 2013.

[147] C.J. Matheus, M.M. Kokar, and K. Baclawski. A Core Ontology for Situation Awareness. In *Information Fusion, 2003. Proceedings of the Sixth International Conference of*, volume 1, pages 545–552, July 2003.

[148] Paul Mathias. The intelligent cooperative intersection as part of urban traffic control systems. In *12th World Congress on Intelligent Transport Systems*, pages 2796–2804, 2005.

[149] Norman Mattern, Marcus Obst, Robin Schubert, and Gerd Wanielik. Co-operative vehicle localization algorithm – evaluation of the covel approach. In *Systems, Signals and Devices (SSD), 2012 9th International Multi-Conference on*, pages 1–5. IEEE, 2012.

[150] J.C. McCall, D.P. Wipf, M.M. Trivedi, and B.D. Rao. Lane Change Intent Analysis Using Robust Operators and Sparse Bayesian Learning. *Intelligent Transportation Systems, IEEE Transactions on*, 8(3):431–440, Sept 2007.

[151] M Douglas McIlroy, JM Buxton, Peter Naur, and Brian Randell. Mass-produced software components. In *Proceedings of the 1st International Conference on Software Engineering, Garmisch Pattenkirchen, Germany*, pages 88–98. sn, 1968.

[152] Andrew McLennan and Johannes Berg. Asymptotic expected number of nash equilibria of two-player normal form games. *Games and Economic Behavior*, 51(2):264–295, 2005.

[153] J-i Meguro, Taishi Murata, J-i Takiguchi, Yoshiharu Amano, and Takumi Hashizume. Gps multipath mitigation for urban area using omnidirectional infrared camera. *Intelligent Transportation Systems, IEEE Transactions on*, 10(1):22–30, 2009.

[154] James H Michels. Covariance matrix estimator performance in non-gaussian clutter processes. In *Radar Conference, 1997., IEEE National*, pages 309–313. IEEE, 1997.

[155] Brian Milch, Bhaskara Marthi, Stuart Russell, David Sontag, Daniel L Ong, and Andrey Kolobov. BLOG: Probabilistic Models with Unknown Objects. *Statistical relational learning*, page 373, 2007.

[156] Brian Milch and Stuart Russell. Inductive Logic Programming. chapter First-Order Probabilistic Languages: Into the Unknown, pages 10–24. Springer-Verlag, Berlin, Heidelberg, 2007.

[157] Michiel M Minderhoud and Piet HL Bovy. Extended Time-To-Collision Measures for Road Traffic Safety Assessment. *Accident Analysis & Prevention*, 33(1):89–97, 2001.

[158] Mahmoud Mohanna, Mohamed L Rabeh, Emad M Zieur, and Sherif Hekala. Optimization of music algorithm for angle of arrival estimation in wireless communications. *NRIAG Journal of Astronomy and Geophysics*, 2(1):116–124, 2013.

[159] Sagarika Mohanty and Debasish Jena. Secure data aggregation in vehicular-adhoc networks: A survey. *Procedia Technology*, 6:922 – 929, 2012. 2nd International Conference on Communication, Computing & Security [ICCCS-2012].

[160] Roger B Myerson. *Game Theory*. Harvard university press, 2013.

[161] L. Nassar, A. Jundi, K. Golestan, F. Sattar, F. Karray, M. Kamel, and S. Boumaiza. Vehicular Ad-hoc Networks(VANETs): Capabilities, Challenges in Context-Aware Processing and Communication Gateway. In *Autonomous and Intelligent Systems*, Lecture Notes in Computer Science, pages 42–49. Springer Berlin Heidelberg, 2012.

[162] Lobna Nassar, Fakhri Karray, and Mohamed S Kamel. VANET IR-CAS for Commercial SA: Information Retrieval Context Aware System for VANET Commercial Service Announcement. *International Journal of Intelligent Transportation Systems Research*, 13(1):37–49, 2014.

[163] Sriraam Natarajan, Prasad Tadepalli, ThomasG. Dietterich, and Alan Fern. Learning first-order probabilistic models with combining rules. *Annals of Mathematics and Artificial Intelligence*, 54(1-3):223–256, 2008.

[164] Yair Neuman, Dan Assaf, and Yohai Cohen. Fusing Distributional and Experiential Information for Measuring Semantic Relatedness. *Information Fusion*, 14(3):281–287, 2013.

[165] M. Nilsson and T. Ziemke. Rethinking Level 5: Distributed Cognition and Information Fusion. In *Information Fusion, 2006 9th International Conference on*, pages 1–8, july 2006.

[166] Maria Nilsson, Joeri van Laere, Tarja Susi, and Tom Ziemke. Information fusion in practice: A distributed cognition perspective on the active role of users. *Information Fusion*, 13(1):60–78, 2012.

[167] Vilém Novák. First-Order Fuzzy Logic. *Studia Logica*, 46(1):87–109, 1987.

[168] M. Obst, N. Mattern, R. Schubert, and G. Wanielik. Car-to-car communication for accurate vehicle localization – the covel approach. In *Systems, Signals and Devices (SSD), 2012 9th International Multi-Conference on*, pages 1–6, March 2012.

[169] N. Oliver, E. Horvitz, and A. Garg. Layered representations for human activity recognition. In *Multimodal Interfaces, 2002. Proceedings. Fourth IEEE International Conference on*, pages 3–8, 2002.

[170] N.M. Oliver, B. Rosario, and A.P. Pentland. A Bayesian computer vision system for modeling human interactions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):831–843, Aug 2000.

[171] Chia-Ho Ou. A roadside unit-based localization scheme for vehicular ad hoc networks. *International Journal of Communication Systems*, 27(1):135–150, 2014.

[172] Heping Pan and Lin Liu. Fuzzy Bayesian Metworks—A General Formalism for Representation, Inference and Learning with Hybrid Bayesian Networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 14(07):941–962, 2000.

[173] S. Panichpapiboon and W. Pattara-Atikom. A review of information dissemination protocols for vehicular ad hoc networks. *Communications Surveys Tutorials, IEEE*, 14(3):784–798, Third 2012.

[174] Cheol Young Park, K.B. Laskey, P.C.G. Costa, and S. Matsumoto. Multi-Entity Bayesian Networks learning for hybrid variables in situation awareness. In *Information Fusion (FUSION), 2013 16th International Conference on*, pages 1894–1901, July 2013.

[175] James L Peterson. Petri net theory and the modeling of systems. 1981.

[176] Aris Polychronopoulos, Manolis Tsogas, Angelos J Amditis, and Luisa Andreone. Sensor fusion for predicting vehicles' path for collision avoidance systems. *Intelligent Transportation Systems, IEEE Transactions on*, 8(3):549–562, 2007.

[177] Ryan Porter, Eugene Nudelman, and Yoav Shoham. Simple search methods for finding a nash equilibrium. *Games and Economic Behavior*, 63(2):642–662, 2008.

[178] K. Premaratne, M.N. Murthi, Jinsong Zhang, M. Scheutz, and P.H. Bauer. A dempster-shafer theoretic conditional approach to evidence updating for fusion of hard and soft data. In *Information Fusion, 2009. FUSION '09. 12th International Conference on*, pages 2122–2129, July 2009.

[179] Amira Ragab, Celine Craye, MohamedS. Kamel, and Fakhri Karray. A Visual-Based Driver Distraction Recognition and Detection Using Random Forest. In Aurélio Campilho and Mohamed Kamel, editors, *Image Analysis and Recognition*, Lecture Notes in Computer Science, pages 256–265. Springer International Publishing, 2014.

[180] Mark Rahmes, Kathy Wilder, Kevin Fox, and Rick Pemble. A Game Theory Model for Situation Awareness and Management. In *Consumer Communications and Networking Conference (CCNC), 2013 IEEE*, pages 909–913. IEEE, 2013.

[181] Ioannis Rekleitis. *Cooperative localization and multi-robot exploration.* McGill University, 2003.

[182] Matthew Richardson and Pedro Domingos. Markov Logic Networks. *Machine learning*, 62(1-2):107–136, 2006.

[183] Branko Ristic, Sanjeev Arulampalam, and Neil Gordon. Beyond the kalman filter. *IEEE AEROSPACE AND ELECTRONIC SYSTEMS MAGAZINE*, 19(7):37–38, 2004.

[184] David Antolino Rivas, José M. Barceló-Ordinas, Manel Guerrero Zapata, and Julián D. Morillo-Pozo. Security on vanets: Privacy, misbehaving nodes, false information and secure data aggregation. *Journal of Network and Computer Applications*, 34(6):1942 – 1955, 2011. Control and Optimization over Wireless Networks.

[185] Matthias Röckl, Patrick Robertson, Korbinian Frank, and Thomas Strang. An Architecture for Situation-Aware Driver Assistance Systems. In *VTC Spring*, pages 2555–2559, 2007.

[186] Francisco J. Ros, Juan A. Martinez, and Pedro M. Ruiz. A survey on modeling and simulation of vehicular networks: Communications, mobility, and tools. *Computer Communications*, 43:1 – 15, 2014.

[187] James Rumbaugh, Ivar Jacobson, and Grady Booch. *Unified Modeling Language Reference Manual, The*. Pearson Higher Education, 2004.

[188] S Rasoul Safavian and David Landgrebe. A Survey of Decision Tree Classifier Methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(3):660–674, 1991.

[189] Paul M. Salmon, Neville A. Stanton, and Kristie L. Young. Situation awareness on the road: review, theoretical and methodological issues, and future directions. *Theoretical Issues in Ergonomics Science*, 13(4):472–492, 2011.

[190] Dario D. Salvucci. Inferring Driver Intent: A Case Study in Lane-Change Detection. In *in Proceedings of the Human Factors Ergonomics Society 48th Annual Meeting*, pages 2228–2231, 2004.

[191] Dario D Salvucci. Modeling Driver Behavior in a Cognitive Architecture. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 48(2):362–380, 2006.

[192] Thuraiappah Sathyan, Mark Hedley, and Mahendra Mallick. An analysis of the error characteristics of two time of arrival localization techniques. In *Information Fusion (FUSION), 2010 13th Conference on*, pages 1–7. IEEE, 2010.

[193] F. Sattar, F. Karray, M. Kamel, L. Nassar, and K. Golestan. Recent Advances on Context-Awareness and Data/Information Fusion in ITS. *International Journal of Intelligent Transportation Systems Research*, pages 1–19, 2014.

[194] Nicolas Saunier and Tarek Sayed. Probabilistic Framework for Automated Analysis of Exposure to Road Collisions. *Transportation Research Record: Journal of the Transportation Research Board*, 2083(1):96–104, 2008.

[195] O. Sawade, B. Schaufele, J. Buttgereit, and I. Radusch. A cooperative active blind spot assistant as example for next-gen cooperative driver assistance systems (CoDAS). In *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pages 76–81, June 2014.

[196] J. Schafer and D. Klein. Implementing Situation Awareness for Car-to-X Applications Using Domain Specific Languages. In *Vehicular Technology Conference (VTC Spring), 2013 IEEE 77th*, pages 1–5, June 2013.

[197] R. Schubert, K. Schulze, and G. Wanielik. Situation Assessment for Automatic Lane-Change Maneuvers. *Intelligent Transportation Systems, IEEE Transactions on*, 11(3):607–616, Sept 2010.

[198] S. Seifzadeh, B. Khaleghi, and F. Karray. Soft-data-constrained multi-model particle filter for agile target tracking. In *Information Fusion (FUSION), 2013 16th International Conference on*, pages 564–571, July 2013.

[199] Baraa T. Sharef, Raed A. Alsaqour, and Mahamod Ismail. Vehicular communication ad hoc routing protocols: A survey. *Journal of Network and Computer Applications*, 40:363 – 396, 2014.

[200] SE Shladover. Effects of Traffic Density on Communication Requirements for Cooperative Intersection Collision Avoidance Systems (CICAS), Mar. 2005, Richmond, CA: California PATH. *Partners for Advanced Transit and Highways (PATH)*, 2005.

[201] Mihail L Sichitiu and Maria Kihl. Inter-vehicle communication systems: a survey. *Communications Surveys & Tutorials, IEEE*, 10(2):88–105, 2008.

[202] S. Sivaraman and M.M. Trivedi. Towards cooperative, predictive driver assistance. In *Intelligent Transportation Systems - (ITSC), 2013 16th International IEEE Conference on*, pages 1719–1724, Oct 2013.

[203] Sayanan Sivaraman and Mohan Manubhai Trivedi. A general active-learning framework for on-road vehicle recognition and tracking. *Intelligent Transportation Systems, IEEE Transactions on*, 11(2):267–276, 2010.

[204] Raymond M Smullyan. *First-order logic*, volume 21968. Springer, 1968.

[205] John F. Sowa. *Knowledge representation: logical, philosophical, and computational foundations*. Computer Science Series. Brooks/Cole, 2000.

[206] Neville A Stanton, Rebecca Stewart, Don Harris, Robert J Houghton, Chris Baber, Richard McMaster, Paul Salmon, Geoff Hoyle, Guy Walker, Mark S Young, et al. Distributed situation awareness in dynamic systems: theoretical development and application of an ergonomics methodology. *Ergonomics*, 49(12-13):1288–1311, 2006.

[207] Alan N Steinberg. An Approach to Threat Assessment. In *Harbour Protection Through Data Fusion Technologies*, pages 95–108. Springer, 2009.

[208] Alan N. Steinberg and Christopher L. Bowman. Rethinking the JDL Data Fusion Levels. In *NSSDF Conference Proceedings*, pages 1–6. JHAPL, June 2004.

[209] Erik G Ström, Hannes Hartenstein, Paolo Santi, and Werner Weisbeck. Vehicular communications [scanning the issue]. *Proceeedings of the IEEE*, 99(7):1158–1161, 2011.

[210] Yusuke Takatori and Takaaki Hasegawa. Stand-alone Collision Warning Systems Based on Information Fusion from On-board Sensors: Evaluating Performance Relative to System Penetration Rate. {*IATSS*} *Research*, 30(2):39–47, 2006.

[211] Minoru Tamura, Seiki Takahashi, Shinji Yasuhara, Masachika Kojima, and Kouki Minegishi. Development of intersection safety support systems using vehicle-to-vehicle communication. In *12th World Congress on Intelligent Transport Systems*, pages 3966–3972, 2005.

[212] Ke Tang, Mingyuan Zhao, and Mingtian Zhou. Cyber Insider Threats Situation Awareness Using Game Theory and Information Fusion-based User Behavior Predicting Algorithm. *Journal of Information & Computational Science*, 8(3):529–545, 2011.

[213] A. Tawari, S. Sivaraman, M.M. Trivedi, T. Shannon, and M. Tippelhofer. Looking-in and looking-out vision for Urban Intelligent Assistance: Estimation of driver attentive state and dynamic surround for safe merging and braking. In *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pages 115–120, June 2014.

[214] Andrew Thomas, David J Spiegelhalter, and WR Gilks. BUGS: A Program to Perform Bayesian Inference Using Gibbs Sampling. *Bayesian statistics*, 4(9):837–842, 1992.

[215] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005.

[216] G. Toth, M.M. Kokar, K. Wallenius, K.B. Laskey, M. Sudit, M. Hultner, and O. Kessler. Higher-Level Information Fusion: Challenges to the Academic Community. In *Panel presented at the 11th International Conference on Information Fusion, Cologne, Germany*, 2008.

[217] Son D. Tran and Larry S. Davis. Event modeling and recognition using markov logic networks. In *IN ECCV*, 2008.

[218] Christian Vogler and Dimitris Metaxas. A Framework for Recognizing the Simultaneous Aspects of American Sign Language. *Computer Vision and Image Understanding*, 81(3):358–384, 2001.

[219] W3C. OWL 2 Web Ontology Language Document Overview. http://www.w3.org/TR/2009/REC-owl2-overview-20091027/, 2009.

[220] H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hübner. Ontology-Based Integration of Information - A Survey of Existing Approaches. pages 108–117, 2001.

[221] Guy H Walker, Neville A Stanton, Paul M Salmon, and Daniel P Jenkins. A review of sociotechnical systems theory: a classic concept for new command and control paradigms. *Theoretical Issues in Ergonomics Science*, 9(6):479–499, 2008.

[222] Eric Wan, Ronell Van Der Merwe, et al. The unscented kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pages 153–158. IEEE, 2000.

[223] Xiaoyang Wang and Qiang Ji. Context augmented Dynamic Bayesian Networks for event recognition. *Pattern Recognition Letters*, (0):–, 2013.

[224] Chuanbo Wen, Yunze Cai, Chenglin Wen, and Xiaoming Xu. Optimal sequential kalman filtering with cross-correlated measurement noises. *Aerospace Science and Technology*, 26(1):153–159, 2013.

[225] Md Whaiduzzaman, Mehdi Sookhak, Abdullah Gani, and Rajkumar Buyya. A survey on vehicular cloud computing. *Journal of Network and Computer Applications*, 40:325 – 344, 2014.

[226] F. E. White. A Model for Data Fusion. In *Proceedings of the 1st National Symposium on Sensor Fusion*, volume 2, pages 5–8, 1988.

[227] Bing-Fei Wu, Ying-Han Chen, Chung-Hsuan Yeh, and Yen-Feng Li. Reasoning-Based Framework for Driving Safety Monitoring Using Driving Event Recognition. *Intelligent Transportation Systems, IEEE Transactions on*, 14(3):1231–1241, Sept 2013.

[228] Tao Xiang and Shaogang Gong. Beyond Tracking: Modelling Activity and Understanding Behaviour. *International Journal of Computer Vision*, 67(1):21–51, 2006.

[229] R.R. Yager. On ordered weighted averaging aggregation operators in multi criteria decision making. *Systems, Man and Cybernetics, IEEE Transactions on*, 18(1):183–190, Jan 1988.

[230] J. Yamato, Jun Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden Markov model. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference on*, pages 379–385, Jun 1992.

[231] Jun Yao, Asghar Tabatabaei Balaei, Mahbub Hassan, Nima Alam, and Andrew G Dempster. Improving cooperative positioning for vehicular networks. *Vehicular Technology, IEEE Transactions on*, 60(6):2810–2823, 2011.

[232] Stephen S. Yau and Dazhi Huang. Development of Situation-Aware Applications in Services and Cloud Computing Environments. *Int Journal of Software Informatics*, 7(1):21–39, 2013.

[233] B Yegnanarayana. *Artificial neural networks*. PHI Learning Pvt. Ltd., 2009.

[234] Kristie Young, John D Lee, and Michael A Regan. *Driver distraction: Theory, effects, and mitigation*. CRC Press, 2008.

[235] L.A. Zadeh. Fuzzy sets. *Information Control*, 8:338–353, 1965.

[236] D. Zhang, D. Gatica-Perez, S. Bengio, and I. McCowan. Semi-supervised adapted HMMs for unusual event detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 611–618 vol. 1, June 2005.

[237] Junping Zhang, Ben Tan, Fei Sha, and Li He. Predicting pedestrian counts in crowded scenes with rich and high-dimensional features. *Intelligent Transportation Systems, IEEE Transactions on*, 12(4):1037–1046, 2011.

[238] Junping Zhang, Fei-Yue Wang, Kunfeng Wang, Wei-Hua Lin, Xin Xu, and Cheng Chen. Data-Driven Intelligent Transportation Systems: A Survey. *Intelligent Transportation Systems, IEEE Transactions on*, 12(4):1624–1639, dec. 2011.

[239] Yongmian Zhang and Qiang Ji. Efficient Sensor Selection for Active Information Fusion. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 40(3):719–728, June 2010.