Context Aware Drivers' Behaviour Detection System for VANET

PhD Thesis

Saif Jamal Al-Sultan

This thesis is submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Software Technology Research Laboratory

De Montfort University

Leicester - United Kingdom

September 2013

Dedication

To my beloved father

Dr. Jamal Al-Sultan

Who without him my PhD dream would not be a reality

This thesis is a reminder that all my life through I will be thanking heaven for a special father like you.

To my loving mother

Fawzia Al-Nuaimi

For her special love, ceaseless prayers and encouragement

Thank you for everything you have done for me since the day I was born.

Abstract

Wireless communications and mobile computing have led to the enhancement of, and improvement in, intelligent transportation systems (ITS) that focus on road safety applications. As a promising technology and a core component of ITS, Vehicle Ad hoc Networks (VANET) have emerged as an application of Mobile Ad hoc Networks (MANET), which use Dedicated Short Range Communication (DSRC) to allow vehicles in close proximity to communicate with one another, or to communicate with roadside equipment. These types of communication open up a wide range of potential safety and non-safety applications, with the aim of providing an intelligent driving environment that will offer road users more pleasant journeys.

VANET safety applications are considered to represent a vital step towards improving road safety and enhancing traffic efficiency, as a consequence of their capacity to share information about the road between moving vehicles. This results in decreasing numbers of accidents and increasing the opportunity to save people's lives. Many researchers from different disciplines have focused their research on the development of vehicle safety applications. Designing an accurate and efficient driver behaviour detection system that can detect the abnormal behaviours exhibited by drivers (i.e. drunkenness and fatigue) and alert them may have an impact on the prevention of road accidents. Moreover, using Context-aware systems in vehicles can improve the driving by collecting and analysing contextual information about the driving environment, hence, increasing the awareness of the driver while driving his/her car.

In this thesis, we propose a novel driver behaviour detection system in VANET by utilising a context-aware system approach. The system is comprehensive, non-intrusive and is able to detect four styles of driving behaviour: drunkenness, fatigue, reckless and normal behaviour. The behaviour of the driver in this study is considered to be uncertain context and is defined as a dynamic interaction between the driver, the vehicle and the environment; meaning it is affected by many factors and develops over the time. Therefore, we have introduced a novel Dynamic Bayesian Network (DBN) framework to perform reasoning about uncertainty and to deduce the behaviour of drivers by combining information regarding the above mentioned factors.

A novel On Board Unit (OBU) architecture for detecting the behaviour of the driver has been introduced. The architecture has been built based on the concept of context-awareness; it is divided into three phases that represent the three main subsystems of context-aware system; sensing, reasoning and acting subsystems. The proposed architecture explains how the system components interact in order to detect abnormal behaviour that is being exhibited by driver; this is done to alert the driver and prevent accidents from occurring. The implementation of the proposed system has been carried out using GeNIe version 2.0 software to construct the DBN model. The DBN model has been evaluated using synthetic data in order to demonstrate the detection accuracy of the proposed model under uncertainty, and the importance of including a large amount of contextual information within the detection process.

Declaration

I declare that the work described in this thesis is original work undertaken by me for the degree of Doctor of Philosophy, at the software Technology Research Laboratory (STRL), at De Montfort University, United Kingdom.

No part of the material described in this thesis has been submitted for any award of any other degree or qualification in this or any other university or college of advanced education.

This thesis is written by me and produced using LaTeX.

Saif Jamal Al-Sultan

Publications

1. Saif Al-Sultan, Moath M. Al-Doori, Ali H. Al-Bayatti and Hussein Zedan. A Comprehensive Survey on Vehicular Ad Hoc Network. *Journal of Network and Computer Applications*, In Press,http://dx.doi.org/10.1016/j.jnca.2013.02.036, 2013.

2. Saif Al-Sultan, Ali H. Al-Bayatti and Hussein Zedan. Context Aware Driver Behaviour Detection System in Intelligent Transportation Systems (ITS). *IEEE Transactions on Vehicular Technology*, 2013. (accepted).

Acknowledgments

First and foremost, my deepest thankfulness goes to the most merciful **Allah** for all the things he blessed me with throughout my whole life, without those blessings, I would not even be in this place.

My sincerest gratitude and deepest appreciation goes to **Prof. Hussein Zedan**, the head of the STRL for his guidance throughout my PhD study. The love and care he offers to his students, including me, has affected this work in so many good ways.

I would also like to thank my supervisor **Dr. Ali Al-Bayatti** for his critical comments, professional guidance, perceptive suggestions regarding the research work and constant encouragement. He was the first to help and support me when I started pursuing my PhD.

I would also like to thank my mentor **Prof. Hilal Al-Bayatti** for his encouragement, support and motivation.

Special gratitude and appreciation also goes to **Prof. Waheeb Al-Khaja** for his support, help and advice.

On my family side, I have not enough words to describe my ever first teacher, my father **Jamal**, who has been inspiring me and pushing me forward always. His prayers and blessings are the true reason behind any success I have achieved in my life. I owe it all to you father.

Moreover, I would like to thank my mother **Fawzia**; without her endless support, prayers and guidance, I would not be the man I am today. Thank you Mom for everything you have done for me since the day I was born.

Also, my special thanks and appreciation go to my beloved brother, **Mohanad** who has always been by my side giving me his encouragement and support throughout my life.

I also would like to thank my lovely sister **Shahad** for her continual prayers and encouragement.

Last but not least, I want to thank my friend and office mate **Abdulmalik**, for the many discussions and laughs we have had. I would also like to thank all my colleagues and the staff of the Software Technology Research Laboratory (STRL), Faculty of Technology, De Montfort University for the friendly and convenient working atmosphere I experienced there.

Contents

List of Figures

List of Tables

List of Abbreviations

AI	Artificial intelligence
API	Application Program Interface
ASR	Automatic Speech Recognition
AU	Application Unit
BN	Bayesian Network
BNT	Bayes Net Toolbox
CALM	Continuous Air interface for Long and Medium range
CAS	Context-aware Systems
CASS	Context-awareness Sub-Structure
CCD	Charged Coupled Device
CCH	Control Channel
CoBrA	Context Broker Architecture
CPT	Conditional Probability Table
DAG	Directed Acyclic Graph
DBN	Dynamic Bayesian Network

DFT	Department for Transport
DSRC	Dedicated Short Range Communication
EEBL	Emergency Electronic Break lights
EM	Expectation Maximisation
GMTk	Graphical Model Toolkit
GPRS	General Packet Radio Service
GPS	Global Position System
GSM	Global System for Mobile communications
HMM	Hidden Markov Models
HSDPA	High-Speed Downlink Packet Access
IEEE	Institute of Electrical and Electronics Engineers
ITS	Intelligent Transportation Systems
MAC	Medium Access Control
MANET	Mobile Ad Hoc Network
NS	Navigation System
OBU	On Board Unit
PDA	Personal Digital Assistant
PNL	Probabilistic Network Library
PPG	Photoplethysmograpgy
RCP	Resource Command Processor
RSU	Road Side Unit

SCH	Service Channel
SOCAM	Service-oriented Context-aware Middleware
TMC	Traffic Management Centre
UML	Unified Modelling Language
UMTS	Universal Mobile Telecommunications System
UWB	Ultra Wideband
V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle
VANET	Vehicle Ad Hoc Network
WAVE	Wireless Access in Vehicular Environments
Wi-Fi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
WUSB	Wireless Universal Serial Bus

Chapter 1

Introduction

Objectives:

- Provide an introduction and explain the motivation for this research
- List the research questions
- Present the research methodology
- List the main contributions of the research
- Outline the thesis structure

1.1 Motivations

At the present time cars and other private vehicles are used daily by many people. The biggest problem resulting from the increased use of private transport is the large number of fatalities that occur due to accidents on the roads. Related expense and dangers have been recognised as a serious problem that needs confronting by modern society. According to the UK department for transport (DFT) report for road casualties in Great Britain in the first quarter of 2011, there were 24,770 people killed or seriously injured due to road accidents. This number represents a decrease of 5 per cent compared with the previous 12 months period [14]. Although the number seems to have decreased, it remains high and explains why road safety is such a concern for researchers and governments. There is a pressing need to improve road safety and efficiency in order to prevent road crashes, decrease the number of fatalities and save people's lives.

Wireless communications and mobile computing have led to the enhancement of, and improvement in, intelligent transportation systems (ITS) focused on road safety applications [1, 15]. As a core component of ITS, Vehicle Ad hoc Network (VANET) has emerged as an application of Mobile Ad hoc Networks (MANET), that use Dedicated Short Range Communication (DSRC) to allow nearby vehicles to communicate either with each other or with roadside equipment. These forms of communication offer a wide range of safety applications to improve road safety and improve traffic efficiency in order to save people's lives. VANET safety applications are considered to represent a vital step towards enhancing road safety and towards improving traffic efficiency by preventing accidents from occurring; for example, intersection collision avoidance, warning about violating traffic signal and approaching emergency vehicle warning etc. [16, 17].

Consequently, driver error is considered to be the main factor in most road accidents [18]. According to a report by the UK department for transport [19], in 2011, the number of accidents that occurred due to reckless, drunk and fatigued drivers reached 26496. Therefore, detecting driver's behaviour with the aid of sensing, wireless and technological devices equipped within the vehicle may have a significant impact on reducing the number of accidents and provide a safe driving environment.

As a result, it is anticipated that by designing a more flexible and accurate driver behaviour detection system in VANET by utilising a context-aware system approach to detect different styles of driver's behaviour during driving it will be possible to achieve the task of preventing road accidents and saving people's lives.

1.2 Problem description and driver behaviour detection solution

Current developments in cognitive science, which is a multidisciplinary science that studies the mind and the way it performs its processes like reasoning, categorisation, etc. [20, 21], have shown that the human emotional state (i.e. that caused by drunkenness or fatigue) plays a vital role in human behaviour [22, 23, 24]. Therefore, many researchers have been working in the area of driver monitoring and detection over recent decades, and multiple systems have been proposed to monitor and detect the status of drivers. Some researchers [25, 26, 27, 28] have tried to monitor the behaviour of the vehicle or the driver in isolation, while others have focused on monitoring a combination of the driver and the vehicle or the driver and the environment, so as to detect the status of the driver in an attempt to prevent accidents. Road accidents result from three interconnected elements, which are: the driver, the vehicle and the environment [29].

However, there is still no comprehensive system that can both, effectively monitor different types of driving behaviour (e.g. drunk, fatigue, reckless and normal behaviour) by capturing the driver's and the vehicle's state and environmental changes, and perform effective reasoning regarding uncertain contextual information so as to alert the driver and other vehicles on the road by disseminating a warning message in time to relevant vehicles in the vicinity (including implementing practical corrective action to prevent accidents from happening).

The main goal of this research is to develop a novel approach based on the concept of context-awareness to detect abnormal behaviour of the driver in VANET by taking into account many parameters. In other words, the system can accurately and proactively detect four types of driving behaviour during driving: normal, fatigued, drunken and reckless driving by taking into account context regarding the driver, the vehicle and the environment; it will then alert the driver and other vehicles on the road by operating in vehicle alarms and advising corrective action (in this thesis we will focus on the behaviour detection algorithm, and generating the corrective actions will be left for the future work). This will be carried out by proposing a novel context aware architecture for VANET, based on driver behaviour detection, this architecture explains how this operation is related to the OBU. The functionality of the architecture is divided into three phases: the sensing, reasoning and acting phase; these phases represent the three main subsystems of context-aware system: sensing, reasoning and acting respectively. In the sensing phase, the system will collect information about the driver, the vehicle state and environmental changes. The reasoning phase is responsible for performing reasoning about uncertain contextual information, to deduce the behaviour of the driver.

The behaviour of the driver is considered as representing an uncertain context (high level contextual information) and also develops over the course of driving; therefore, effective reasoning techniques about uncertain contextual information must be performed. We will propose a novel Dynamic Bayesian Network (DBN) model to infer the behaviour of the driver. The proposed model will combine information from different kinds of sensors to capture the static and temporal aspects of behaviour and perform probabilistic inference to deduce the driver's current driving behaviour.

The acting phase is responsible for operating in vehicle alarms and for sending corrective actions to other vehicles, via the wireless technology provided by VANET. In this thesis we focus on a behaviour detection algorithm, while calculating corrective actions for the other vehicles on the road is left for the future work.

1.3 Research Questions

We have analysed the driver behaviour detection systems currently available in the literature and we have outlined the main research question as follows:

How can we detect abnormal behaviour exhibited by a driver in VANET utilising a context-aware system approach?

The main aim of the work in this research is to address the question above efficiently. However, partitioning the research question into sub-questions would make it easier to tackle each one individually. These questions can be summarised as follows:

- **What kind of information is needed to detect different styles of driver behaviour accurately?**

- **How can we design an effective driver behaviour detection system architecture for VANET by utilising a context-aware system approach?**
- **How can we design an efficient driver behaviour detection model that can perform reasoning over time (temporal reasoning) and under uncertainty?**

1.4 Research Methodology

The methodology used in this research is a scientific research method, which explains constructive research. The term 'constructive' denotes the contribution in knowledge developed in terms of new architecture, models, techniques, etc. However, it is hard to carry out scientific research in a definite field without thorough knowledge of the field, and hence, acquiring this knowledge represents a component of such research.

The scientific methodology for this research was carried out through five phases. The first phase concerns the research literature review, and the second phase elaborates the uncertainty reasoning method on which the behaviour detection model is based. The third phase deals with the proposed architecture. The fourth phase illustrates in detail the proposed DBN driver's behaviour detection model. The last phase addresses the process of validating the work.

- **Phase 1:** Research background

  Several resources have been used such as books, articles, digital libraries, etc. in order to conduct a literature review in three stages; covering VANET, context-aware system and an overview of current driver behaviour detection systems. The first stage provides an overview of VANET, VANET architecture, wireless access technology in VANET and VANET applications. In the second stage, an overview of context and context-aware system and several proposed architectures was discussed. The third stage illustrates in detail existing driver behaviour monitoring and detection systems and their main drawbacks, as the aim of this research is to design a novel driver behaviour detection system for VANET by utilising a context-aware systems approach to prevent road accidents and save people's lives.

- **Phase 2:** Reasoning techniques

  Define the normal and abnormal behaviour of drivers and explore available reasoning techniques, which can be used to infer behaviour, and then, illustrate in detail the chosen technique on which our detection model will be based.

- **Phase 3:** Architecture

  Design the OBU architecture in order to capture contextual information about the driver, the vehicle and the environment, and then deduce the behaviour of the driver.

- **Phase 4:** Driver behaviour detection model

  Design the DBN driver behaviour detection model to combine information from different kinds of sensors and then deduce the behaviour of the driver. In particular, choosing network variables and specifying the conditional independence between them, and parameterising the network.

- **Phase 5:** Implementation and Evaluation

  Validate the performance of the proposed model using synthetic data and present experiments to show the system's ability for detection.

1.5 Measure of Success

Success in regards to the reported work in this research will be evaluated as follows:

- The research questions set at section 1.3 have to be answered.
- A study presenting how our proposed architecture can be applied in VANET, in order to detect the abnormal behaviours exhibited by drivers has to be conducted.
- An analysis of why DBN was chosen from among other reasoning techniques, and a determination of the advantages of this technique must be performed.
- A study showing how our proposed driver behaviour detection model is different from others has to be carried out.

1.6 Thesis Contributions

The main contributions of the work reported in this thesis can be illustrated as follows:

- **On Board Unit Architecture for driver behaviour detection:** An On Board Unit architecture for detecting the behaviour of the driver in VANET has been presented. The architecture collects information about the driver (i.e. eyes movements), the vehicle (i.e. vehicle's speed) and the environment (i.e. temperature). The architecture has been designed based on aspects of a context-aware system.
- **DBN driver behaviour detection model:** A novel DBN model for deducing the behaviour of the driver during driving has been introduced, this model takes into account the static and the temporal aspects of the behaviour and is

able to detect four styles of driver behaviour, which are: normal, drunk, reckless and fatigue behaviour. The model combines information from different sources in order to detect the behaviour of the driver accurately. In addition to current sensors readings, the state of the driver in the previous time instant is taken into consideration when inferring the current state in order to model the evolving system (driver's behaviour). Network parameters were chosen manually, following a critical analysis of a large number of reports published by the UK department for transport and other transportation organizations, as well as a set of published papers with similarities to the system.

- The system introduced in this thesis is the first driver behaviour detection system that deduces four styles of driver behaviour (normal, drunk, reckless and fatigue) by taking into account information about the driver, the vehicle and the environment, performing reasoning over time and under uncertainty.

1.7 Thesis Outlines

This section describes the outline of the remaining chapters of this thesis:

Chapter 2: Literature Review

At first, this chapter presents an overview of VANET; explaining its architecture, its communication domains and the types of applications provided by VANET. This chapter then describes the context-aware system by defining context, how to capture context, what a context-aware system is, how to model context and how to perform reasoning regarding uncertain context using diverse methods. After this, it provides justification for the reasoning method used in this thesis. Finally, it presents a critical review for the main work that has been carried out in the field of driver behaviour monitoring and detection to show their weaknesses and how the proposed system will differ from previous work.

Chapter 3: Fundamental Principles of Bayesian Networks

This chapter is divided into three parts. The first part introduces an overview of the behaviour of the driver, how normal and abnormal behaviour are defined in this thesis from the perspective of a context-aware system. It also describes the assumptions made when designing a new driver behaviour detection system. The second part of this chapter presents an overview of the reasoning method (DBN) that has been used to combine information from several sensors and to infer the behaviour of the driver. The last part demonstrates the main existing software that supports the implementation of a DBNs with the main reasons for choosing the GeNIe as an implementation tool in this thesis.

Chapter 4: On Board Unit Architecture Based on Context-aware System

This chapter describes the mechanism for detecting the behaviour of the driver in VANET. Moreover, it presents On Board Unit architecture for driver behaviour detection, which was designed based on the aspects of the context aware system and divided into three phases based on the context-aware system. The three phases are described in detail to show the functions and components of each phase in the architecture and how these components interact with each other to deduce driver's behaviour efficiently.

Chapter 5: A Dynamic Bayesian Network Model for Driver's Behaviour Detection

Using DBN, this chapter introduces the design and development of the new driver behaviour detection model. It includes a detailed description of each step when designing the proposed behaviour detection model starting from choosing network variables ending with inferring the hypothesis node.

Chapter 6: Evaluation and Experiments

This chapter illustrates the validity of the proposed system using synthetic data. It explains how the system is able to detect four styles of driver's behaviour (fatigue, drunk, reckless and normal behaviour) applying all possible combinations of evidence. Furthermore, it presents experiments with different scenarios in order to show the ability of the proposed system to detect the above mentioned styles of behaviour during driving.

Chapter 7: Conclusion and Future Work

This chapter demonstrates a summary of the work presented throughout the thesis. It then draws some conclusions and provides suggestions for future work.

Chapter 2

Literature Review

Objectives:

- Present an overview of VANET
- Present literature on context, context-aware systems, context modelling and reasoning
- Justify the use of DBN
- Investigate existing driver behaviour monitoring and detection systems

2.1 Overview of MANET

Mobile ad hoc networks (MANET) composed of self-organised nodes communicates with each other without the need for a pre-established infrastructure. The main principle of an ad hoc networking is that it relies on multi-hop communication, whereby each node can act as an end user or as a router simultaneously [30]. MANET has become widespread and successful in the marketplace of wireless technology for the future, as is evidenced by the increasing use of Wireless Local Area Networks (WLANs) and Bluetooth technologies.

2.2 VANET Overview

Vehicle ad hoc Networks (VANET) is classified as an application of mobile ad hoc networks (MANET) [31]. It provides a wireless communication between moving vehicles, using a dedicated short range communication (DSRC). DSRC is essentially IEEE 802.11a amended for low overhead operation to 802.11p; the IEEE then standardises the whole communication stack by the 1609 family of standards referring to wireless access in vehicular environments (WAVE). A vehicle can communicate with other vehicles directly forming vehicle to vehicle (V2V) communication or communicate with a fixed equipment next to the road , referred to as Road Side Unit (RSU) forming vehicle to infrastructure (V2I) communication. Each vehicle is equipped with a wireless interface known as an On Board Unit (OBU), to allow it to communicate with other vehicles or with the RSU. These types of communications allow vehicles to share information collected pertaining to safety issues; for the purpose of accident prevention, post-accident investigation or traffic jams. The intention behind distributing and sharing this information is to provide a safety message to warn drivers about expected hazards in order to decrease the number of accidents and save people's lives, or share other traveller related information (non-safety messages) [1, 32, 33, 3].

2.2.1 VANET Architecture

The communication between vehicles, or between a vehicle and an RSU is achieved through a wireless medium called WAVE. This method of communication provides a wide range of information to drivers and travellers and enables safety applications to enhance road safety and provide comfortable driving. The main system components are the Application Unit (AU), OBU and RSU. Typically the RSU hosts an application that provides services and the OBU is a peer device that uses the services provided. The application may reside in the RSU or in the OBU; the device that hosts the application is called the provider and the device using the application is described as the user. Each vehicle is equipped with an OBU and a set of sensors to collect and process the information then send it on as a message to other vehicles or RSUs through the wireless medium; it also carries a single or multiple AU that use the applications provided by the provider using OBU connection capabilities. The RSU can also connect to the Internet or to another server which allows AU's from multiple vehicles to connect to the Internet [34, 35, 1, 17].

2.2.1.1 On board Unit (OBU)

An OBU is a WAVE device usually mounted on-board a vehicle used for exchanging information with RSUs or with other OBUs. It consists of a Resource Command Processor (RCP), and resources include a read/write memory used to store and retrieve information, a user interface, a specialised interface to connect to other OBUs and a network device for short range wireless communication based on IEEE 802.11p radio technology. It may additionally include another network device for non-safety applications based on other radio technologies such as IEEE 802.11a/b/g/n. The

OBU connects to the RSU or to other OBUs through a wireless link based on the IEEE 802.11p radio frequency channel, and is responsible for the communications with other OBUs or with RSUs; it also provides a communication services to the AU and forwards data on behalf of other OBUs on the network. The main functions of the OBU are wireless radio access, ad hoc and geographical routing, network congestion control, reliable message transfer, data security and IP mobility [34, 35, 1].

2.2.1.2 Application unit (AU)

The AU is the device equipped within the vehicle that uses the applications provided by the provider using the communication capabilities of the OBU. The AU can be a dedicated device for safety applications or a normal device such as a Personal Digital Assistant (PDA) to run the Internet, the AU can be connected to the OBU through a wired or wireless connection and may reside with the OBU in a single physical unit; the distinction between the AU and the OBU is logical. The AU communicates with the network solely via the OBU which takes responsibility for all mobility and networking functions [34, 1].

2.2.1.3 Roadside Unit (RSU)

The RSU is a WAVE device usually fixed along the road side or in dedicated locations, such as at junctions or near parking spaces. The RSU is a computer that have sufficient processor and storage capability to run a gateway application and to run additional safety software. It is equipped with one network device for a dedicated short range communication based on IEEE 802.11p radio technology and an antenna, it can also be equipped with other network devices so as to be used for the purpose of communication within the infrastructural network. The RSU performs three main functions which are extending the communication range of the ad hoc network by re-distributing the information to other OBUs and by sending the information to other RSUs in order to forward it to other OBUs, Running safety applications such as a low bridge warning and Providing Internet connectivity to OBUs [34, 1].

2.2.2 VANET Communication Domains

As shown in Figure 2.1, the communication between vehicles and the RSU and the infrastructure form three types of domains [17]:

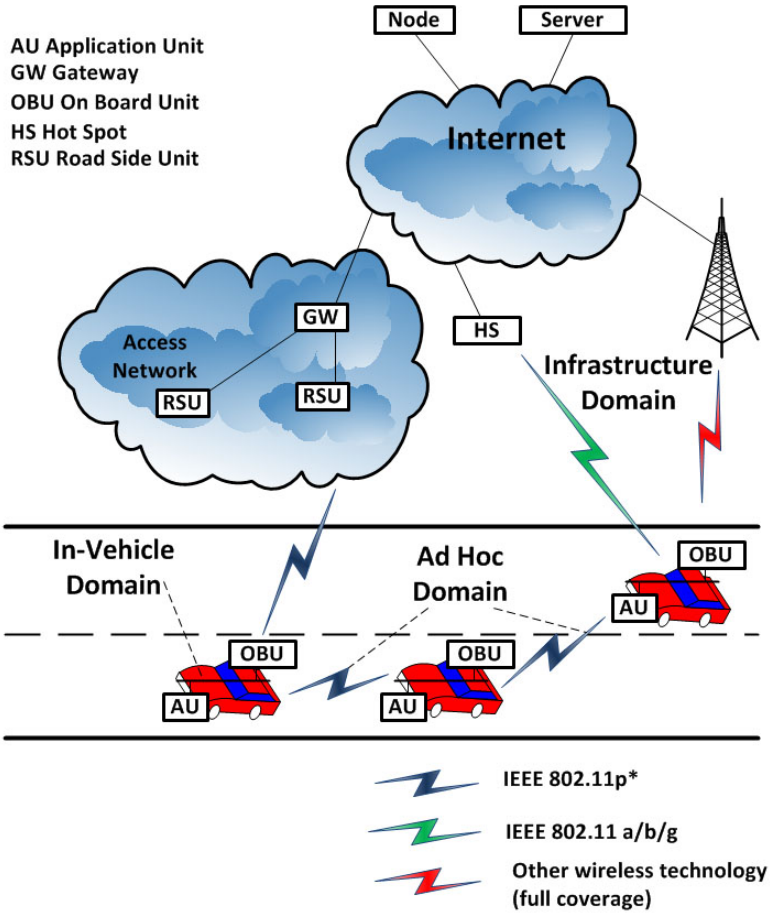

Figure 2.1: Communication domains in VANET [1]

1. **In-vehicle domain:** This domain consists of an OBU and one or multiple AUs. The connection could be wired or wireless using wireless universal serial bus (WUSB) or ultra-wideband (UWB); an OBU and an AU can reside in a

single device. The OBU provides a communication link to the AU in order to execute one or more of a set of applications provided by the application provider using the communication capabilities of the OBU [1, 34, 36].

2. **Ad hoc domain:** The ad hoc domain on VANET is composed of vehicles equipped with OBUs and a station along the road side, the RSU. According to [1, 34, 37, 36], two types of communications are available in the ad hoc domain:

    - As a component and concrete application of an ITS inter vehicle communication gained attention from researchers, academics and industry leaders, especially in the United States, Europe and Japan. Owing to its ability to improve road traffic safety, driving efficiency and to extend on board device horizons [38], vehicles communicate with other vehicles through OBUs forming an ad hoc network, which allows communication between vehicles in a fully distributed manner with decentralised coordination. Vehicle communicate with another vehicle directly if there is a direct wireless connection available between them, forming a single hop V2V communication; when there is no direct connection between them a dedicated routing protocol is used to forward data from one vehicle to another until it reaches the destination point, forming multi-hop V2V communication.

    - Vehicle communicate with an RSU in order to increase the range of communication by sending, receiving and forwarding data from one node to another or to benefit from the ability of the RSU to process special applications forming V2I communication.

3. **Infrastructural domain:** The RSU can connect to the infrastructural networks or to the Internet, allowing the OBU to access the infrastructure network; in this case it is possible that the AUs are registered with the OBU to connect to any internet based host. OBU can also communicate with other hosts for non-safety applications, using the communication of cellular radio networks (GSM, GPRS, UMTS, HSDPA, WiMax and 4G) [1, 34, 36].

2.2.3 Wireless access technology in VANET

There are numerous wireless access technologies available today, which can be used to provide the radio interface required by the vehicles in order to communicate with each other, V2V communication, or to communicate with the RSUs, V2I communication. These communication technologies intended to improve road safety, traffic efficiency and to provide driver and passenger comfort by enabling a set of safety and non-safety applications. The main technologies are Cellular systems (2/2.5/2.75/3G), Wireless Local Area Network (WLAN) or Wireless Fidelity (Wi-Fi), Mobile - WiMAX or IEEE 802.16e, Combined wireless access technologies such as Continuous Air interface for Long and Medium range (CALM M5) and the DSRC/WAVE technology which is allocated to be used specifically for VANET safety applications as illustrated bellow.

DSRC is a 75MHz licensed spectrum at a 5.9GHz band allocated by the US Federal Communications Commission (US FCC) in 1999, to be used solely for vehicle to vehicle and vehicle to infrastructure communication in the United States. In Europe and Japan the spectrum is allocated at 5.8 GHz [39]. DSRC radio technology is based on IEEE 802.11p, which originated from IEEE 802.11a and was amended for low overhead operation in the DSRC spectrum [33, 2, 32].

As shown in Figure 2.2, the 75MHz spectrum is divided into seven channels starting from channel number 172 ending with channel number 184; the capacity of each channel is 10MHz. Channel 178 is the control channel (CCH), which is used exclusively for safety communications; channels 172 and 184 are reserved for safety applications, while the other service channels (SCH) have for both safety and non-safety uses.

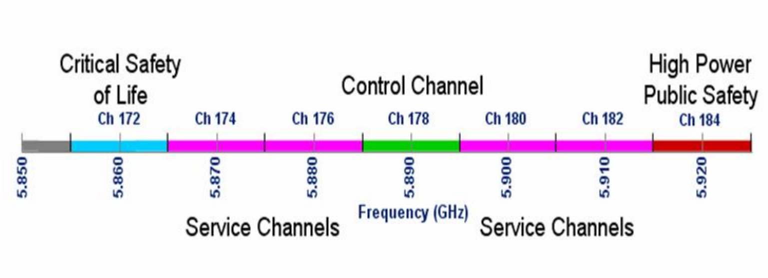

Figure 2.2: 75MHz DSRC spectrum [2]

The whole DSRC protocol stack including IEEE 802.11p (MAC and PHY layers) standardized by the IEEE 1609 working group and called Wireless Access in a Vehicular Environment (WAVE) [33, 2, 32]. DSRC/WAVE support an environment in which vehicles can be moving at speeds of up to 200Kmph, covering a communication range of 300m and reaching up to 1000m with a data rate of more than 27Mbps [40, 39].

2.2.4 VANET Applications

V2V and V2I communications allow the development of a large number of applications and can provide a wide range of information to drivers and travellers. Integrating on-board devices with the network interface, different types of sensors and Global Position System (GPS) receivers, grant vehicles the ability to collect, process and disseminate information about itself and its environment to other vehicles in close proximity to it. That has led to enhancement of road safety and the provision of passenger comfort [40, 41, 39]. As shown in Figure 2.3, VANET applications are classified according to their primary purpose into [17]:

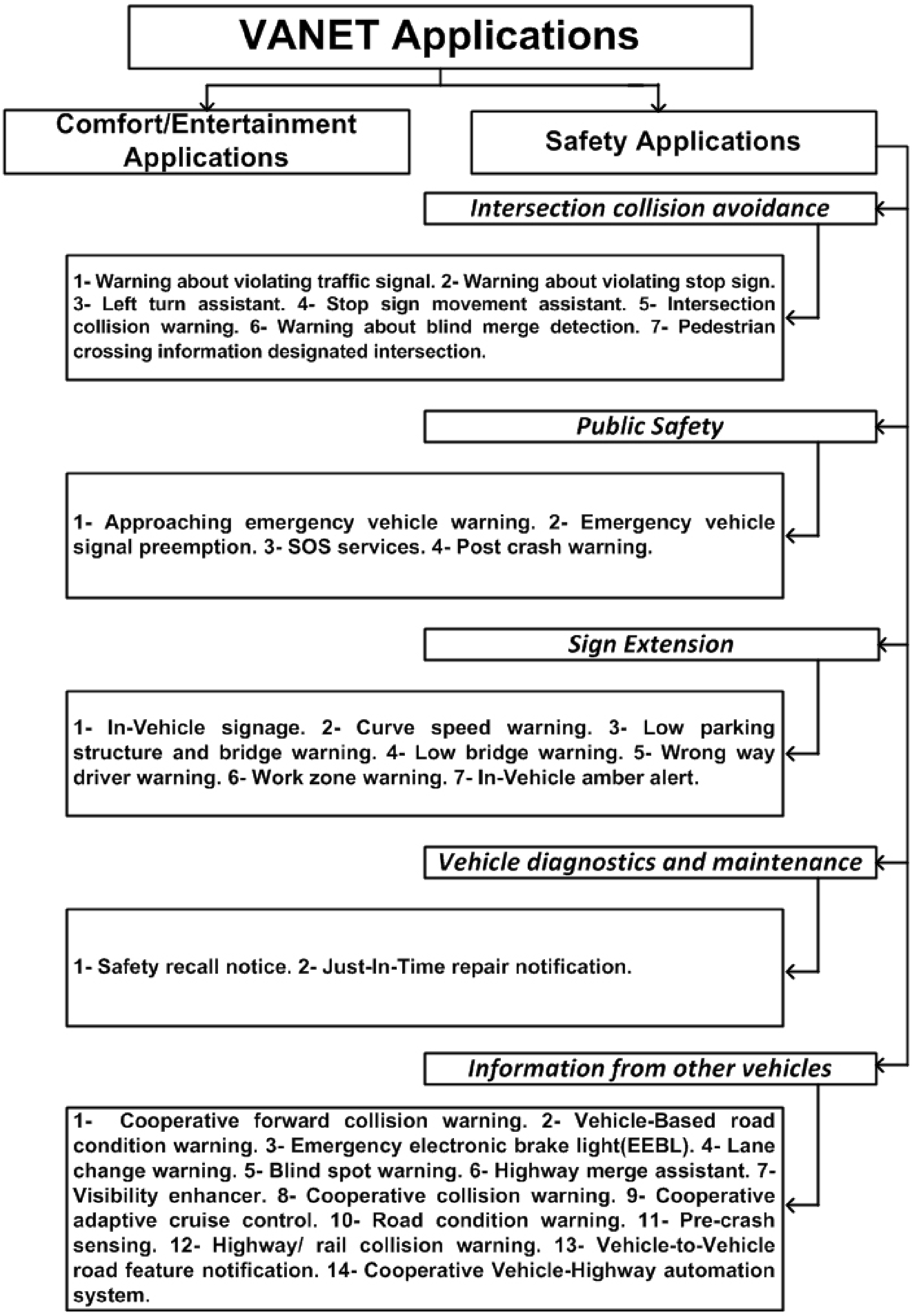

Figure 2.3: VANET applications

- **Comfort/Entertainment applications:** This category of applications is also referred to as non-safety applications, and aim to improve drivers and passengers comfort levels (make the journey more pleasant) and enhance traffic efficiency. They can provide drivers or passengers with weather and traffic information and detail the location of the nearest restaurant, petrol station or hotel and their prices. Passengers can play online games, access the internet and send or receive instant messages while the vehicle is connected to the infrastructure network.[40, 41, 32, 39].

- **Safety applications:** These applications use the wireless communication between vehicles or between vehicles and infrastructure, in order to improve road safety and avoid accidents; the intention being to save people's lives and provide a clean environment. Appendix A provides a detailed description of VANET safety applications.

Figure 2.4, depicts an example of VANET safety application using the DSRC/WAVE access technology. Where, the vehicle at the curve can disseminate a message to warn the vehicles approaching the blind spot about a traffic jam using vehicle to vehicle or vehicle to infrastructure communications, to prevent them from colliding with the traffic jam.

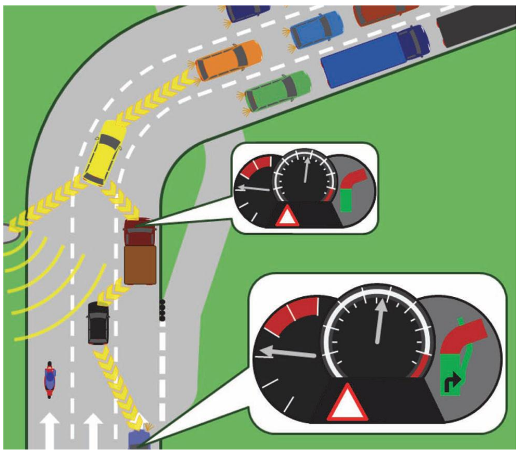

Figure 2.4: Avoid road accidents by using (V2V) and (V2I) communications [3]

Safety applications use the DSRC as a basis for wireless communication, DSRC supports V2V and V2I communications and operation at 5.9GHz in the United States, and at 5.8GHz in both Europe and Japan [38]. Some of the safety applications depend on V2V to exchange messages while other require V2I. Safety applications have as an essential requirement the ability to gather information through a vehicle's sensors, from other vehicles or both, in order to process and disseminate information in the form of safety messages to other vehicles or infrastructures depending on the application and its functions. Applying wireless communication technology in vehicles in order to communicate with other vehicles, or with the infrastructure, enables a wide range of applications and leads to an increase in the road safety level. According to [16], safety applications using V2V communication or V2I communication, or both can be categorised as fellows:

1. Intersection collision avoidance.
2. Public safety.
3. Sign Extension.
4. Vehicle diagnostics and maintenance.
5. Information from other vehicles.

The goal of this work is to design a VANET safety application utilising a context-aware system approach. The application has the potential to enhance road safety and traffic efficiency, and to save people's lives, by detecting abnormal behaviours exhibited by the driver and issuing warning messages in order to prevent accidents from happening.

2.3 Context-aware systems overview

2.3.1 Context

Many researchers have tried to define 'context'; the word context itself is originally derived from the Latin con (with or together) and texere (to weave), suggesting context is an active process describing how humans give meaning to their whole environment by weaving their experience into it [42].

The first definition of context introduced by [43], they referred to it as location and the group of people and objects all over the place. In [44], context is defined as location, weather, time and users' identities. While [45] referred to it as users' emotional state (attention, location, objects, date, time and the people around the user).

Context can be defined using synonyms (user/application's situation or environment). In [46], context is described as the information that the computer recognises about its user environment. According to [47], context is the aspects of the current situation. Salber [48] also defined a context as the information that can be sensed by the application, about the user or the application's environment

According to [49], the definition of context in the free on-line dictionary, it is that: which surrounds, and gives meaning to something else. The authors in [50] stated that, the important aspects of context are: where you are, who you are with and what resources are available. Dey et al [45] described context as the physical, social or informational state of the user.

The most accurate and clear definition for the word context among all the definitions aforementioned above is given by [51], by defining it as: *" context is any information that can be used to characterize the situation of an entity. An entity is a person, or object that is considered relevant to the interaction between a user and application, including the user and applications themselves"*.

2.3.2 Context categories

Categorising context types helps the designers of the systems to determine what type of context will be most appropriate for their applications. To this end context can be categorised into four categories as follows:

- Identity: Person name.
- time: 3:00pm, day of the week, time zone and season of the year.
- location: Information about the location of the entity
- and activity: What is accruing in the situation

The aforementioned categories can be considered as a primary context types, and acts as a source for secondary types of context. For example when we know person's identity we can acquire information related that person such as his/her phone number, email address, birth date and address. Also by determining the location of an entity we can reach a conclusion about what other objects or persons in its local environment and what activities are occurring in close proximity to that entity [51].

2.3.3 Context Attributes

According to [52], a single context atom can be described as having a known set of attributes, the most observable two are:

- Context type: Denoting the categories associated with the context, such as temperature, time, speed etc..
- Context value: Meaning the raw data gathered by the sensors, such as miles per hour.

These two attributes mentioned are nevertheless inadequate when seeking to build a context-aware system. Other attributes should be captured such as:

- Time stamp: Containing the date and time of when the context was sensed.
- Source: Containing information about how the context was gathered (i.e. the ID of the physical sensor).
- Confidence: Describing the accuracy level of the contextual information gathered; some data sources (sensors) might provide in-accurate data (i.e. location data).

2.3.4 Context sensing

Capture of contextual data is done using sensors; the word sensor refers to any source of data that can provide valuable contextual information not only to the hardware sensor[4]. According to [53], sensors can be categorised as follows:

1. **Physical sensors:** The hardware sensors are the generally used types of sensors, that can capture any physical data related to an entity. [54, 55] list different types of physical sensors as given below:
    - Light sensor: A single optical sensors such as a photodiode, IR, or colour sensor provides information about light density, reflection and concentration as well as the type of the light.

- Camera: By applying a camera's sensors, a wide range of information about the visual environment can be acquired; for instance object recognition.
- Audio sensors: Microphones can supply a remarkable information such as noise level and type of the voice (e.g. speaking, music), and can provide speech recognition by performing high level processing.
- Accelerometers: By applying this kind of sensing applications will be able to acquire information about an object's acceleration and movement.
- Location sensors: These kinds of sensors provide information about the location of an entity, GPS and GSM can be used to provide an outdoor location while indoor location can be captured by a system such as Active Badge.
- Touch sensors: These are considered to be energy savers which operate only when touched by the user, leading to reduced energy consumption.
- Temperature sensors: These provide information about the weather or body heat, and are cheap and easy to use.
- Air pressure sensors: Provide information about altitude and pressure that can indicate for example the door is closed.
- Motion detector (movement sensors): These activate only when there are moving objects in their environment.
- Magnetic field: By applying this type of sensor a system can acquire information about the device's direction and movement.
- Biosensor: These provide valuable information pertaining to skin resistance and blood pressure.
- Mechanical force sensors: These can acquire a useful information about an object's weight, the state of the user when sitting on the chair and

the force of the human hand on that object (e.g. the force of the driver's hand on the steering wheel).

- Proximity sensors: Similar to the movement sensors, they activate only when there is an object in their vicinity, in order to operate those applications that are in sleep mode; whereas no objects appear in the immediate environment.
- Humidity: By applying these sensors the air humidity can be measured, thus provide valuable information about the environment.

Several types of physical sensors can be used to capture contextual information in context-aware systems. Beigl et al [55] mentioned that the most commonly used types of sensors in ubiquitous and pervasive computing are as shown in Table 2.1:

Sensor type	Usage percentage
Object Movement	31%
Light	18%
Force	15%
Temperature	12%
Audio	12%
Humidity	6%
Proximity	6%

Table 2.1: Types of sensors used by pervasive computing

2. **Virtual sensors:** This category of sensors obtain contextual data via software applications. The system can determine the user's location by browsing emails, examining a travel-booking system or by discovering the location of the computer that the user is currently using.
3. **Logical sensors:** This category of sensors acquire information by combining data from two sources (physical and virtual sensors) in order to solve complicated problems; for example the exact location of a user in a company can

be determined by combining data from physical sensors (GPS) and from the location of the computer that he/she is currently using (virtual sensors).

Gathering contextual information is the basic level of any context-aware application. The way context-aware systems acquire contextual data can be categorised according to [4] as follows:

- **Direct sensor access:** In this approach the sensors built into the devices, the software, can directly acquire information from the sensors without any processing within the other layers, this method of sensing data is unsuitable for applications because it does not allow concurrent sensor accesses.
- **Middle-ware infrastructure:** This approach uses a method of encapsulation involving hiding low-level sensing details, thus introducing a layered architecture to a context-aware system; this approach eases the re usability of hardware sensors and allows concurrent sensor accesses.
- **Context server:** In this approach multiple users have the permission to access a remote data source. The context server is a remote component responsible for gathering the contextual information and managing user's access to this information remotely.

2.3.5 Context-aware Systems (CAS)

Context-aware systems are those systems that are capable of adapting their operations to the current context without user interaction, and are thus aimed at augmenting usability and effectiveness by taking into account the environment's contextual information [4]. In 1994 [43] first defined context-aware computing as the software that behave according to the information in its environment such as (location and the group of people and object all over the place). In [56], the author referred to context-aware systems as those systems that are able to extract (sense), interpret and respond to the current context. Dey [57] also defined context-aware systems as those systems that use contextual information in the environment to provide information and/or services to the user regarding their task.

The way a context-aware system provides information and services to users can be classified into three categories according to [48, 51]:

- **Presenting information and services to users:** Providing information to users or suggesting a selection of actions in order to perform services, GUIDE [58] is an application that use this way to provide services to users.
- **Automatically executing the service:** Applications on behalf of user take an action or reconfigure a system based on the current context, Teleport [59] is an application that automatically executes services.
- **Attaching context information for later retrieval:** This describes applications that tag captured data with relevant context information, Classroom2000 [60] is an application that uses tagging to provide services to users.

2.3.6 Context-aware system architecture

As mentioned above, context-aware systems provides information and services to users depending on the environmental contextual information. Context-aware systems incorporate three main subsystems [49]:

- **Sensing subsystem:** Which is the phase for gathering contextual information by sensors.

- **Reasoning (Thinking) subsystem:** It is the phase for employing reasoning techniques to the contextual data in order to get high level contextual information (e.g. user situation).
- **Acting subsystem:** Depending on the current situation, the systems provide services to users.

Several system architectures have been suggested in order to ease the implementation and development of context-aware systems, as shown in Figure 2.5, a conceptual layered architecture has been proposed by [9].

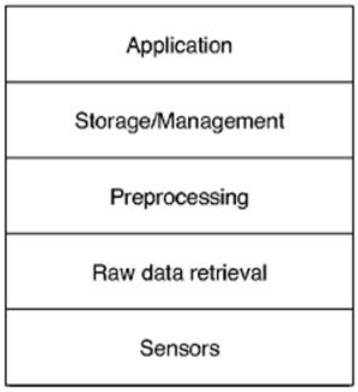

Figure 2.5: The five-layered context-aware systems Architecture [4]

The first layer (Sensors layer), consists of different types of sensors, such as physical, logical and virtual sensors. The second layer (Raw data retrieval layer) is responsible for retrieving the raw data. The third layer (Preproocessing layer) is not applied in every context-aware system [4], and when incorporated it is responsible for processing the data gathered by the sensors in order to get high level data that is meaningful and useful. The fourth layer (Storage/ Management layer) is responsible for storing and organising the data from the previous layer. Performing actual actions depending on different events and situations is implemented in the fifth layer (Application layer).

A middleware architecture has been proposed by a Context-awareness Sub-Structure project (CASS) [5], that supports the development of context-aware systems. The architecture of CASS as illustrated in Figure 2.6, is based on middleware that contains an interpreter, Contextretriever, Rule Engine and Sensorlistener. The sensorlistener stores the data gathered by the distributed sensors (Sensor nodes) in a database, retrieving stored context data accomplished by the contextretriever. An interpreter may support both the contextretriever and sensorlistenor, and a mobile computer notified about the changes in a context via Changelistener, sensors and the Locationfinder has built in communication capabilities.

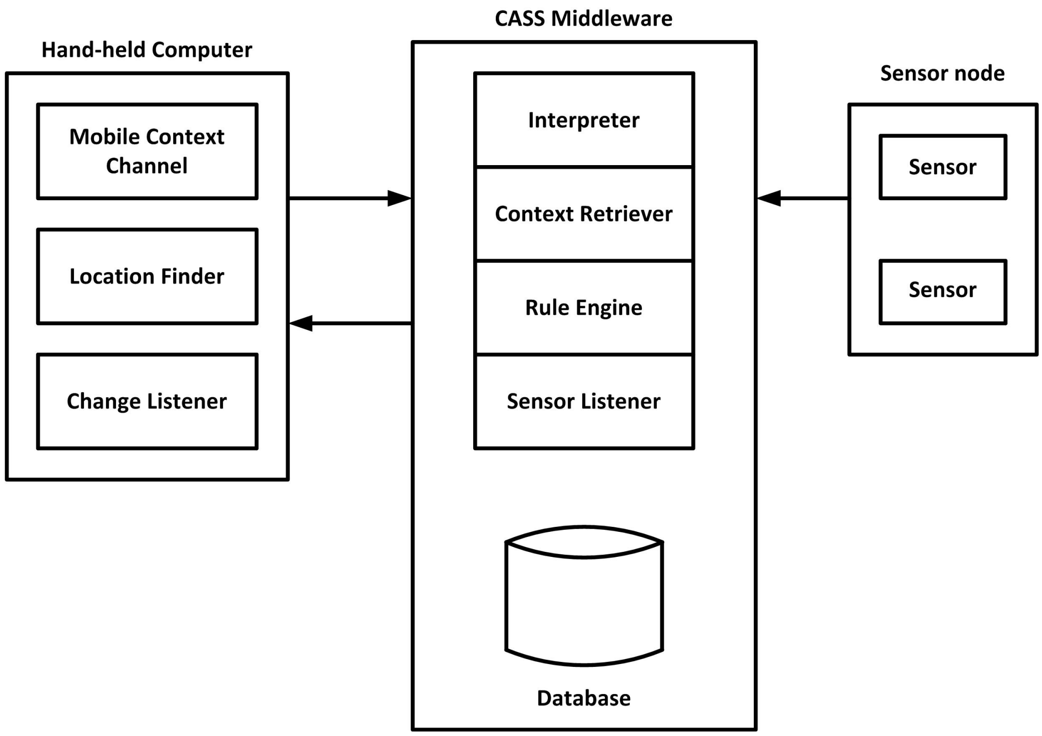

Figure 2.6: The context-awareness sub-system architecture [5]

Another architecture intended for supporting the development of context-aware mobile applications was introduced by [6], in the Service-Oriented Context-Aware Middleware project (SOCAM). As shown in Figure 2.7, SOCAM architecture consists of the following entities: context provider, context interpreter, service locating service, context database and context-aware mobile services. The central server (context interpreter) acquires the context data via the context provider and the database and provides this information to the clients.

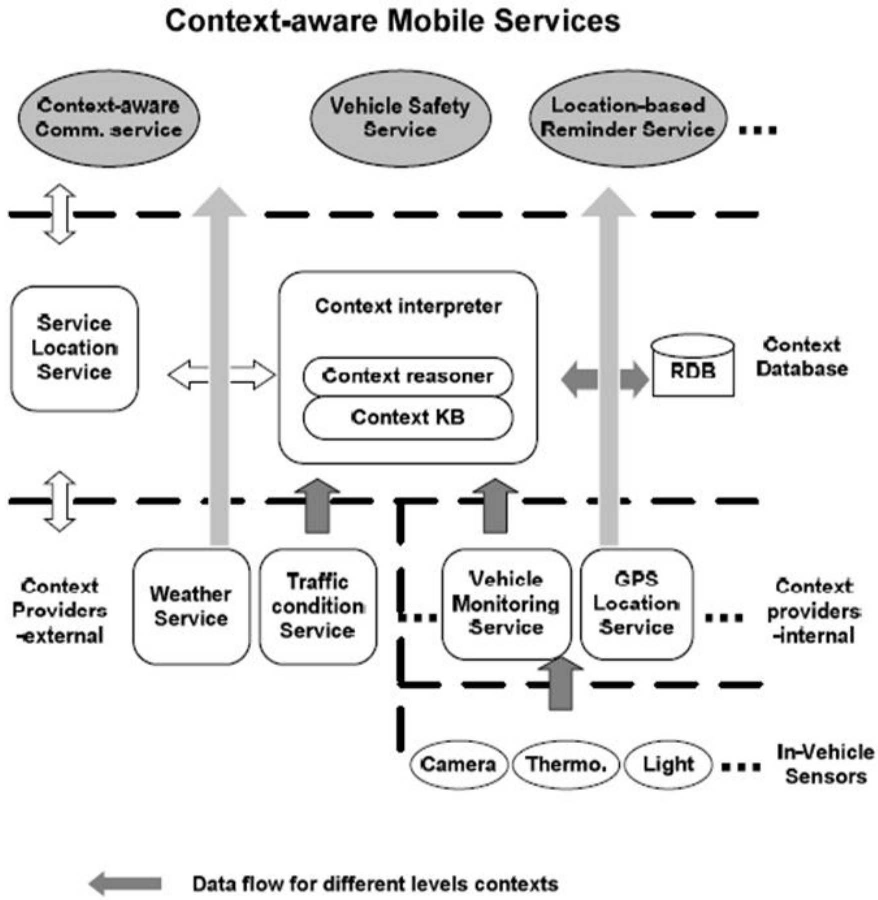

Figure 2.7: The service-oriented context-aware middleware architecture [6]

In [7], a Context Broker Architecture (CoBrA) has been proposed, which is an agent based architecture that supports the development of context-aware system in alleged intelligent space. As depicted in Figure 2.8, CoBrA architecture is composed of four components: context knowledge base, context reasoning engine, context acquisition module and privacy management module. The intelligent context broker is responsible for managing and maintaining the context model for a particular domain.

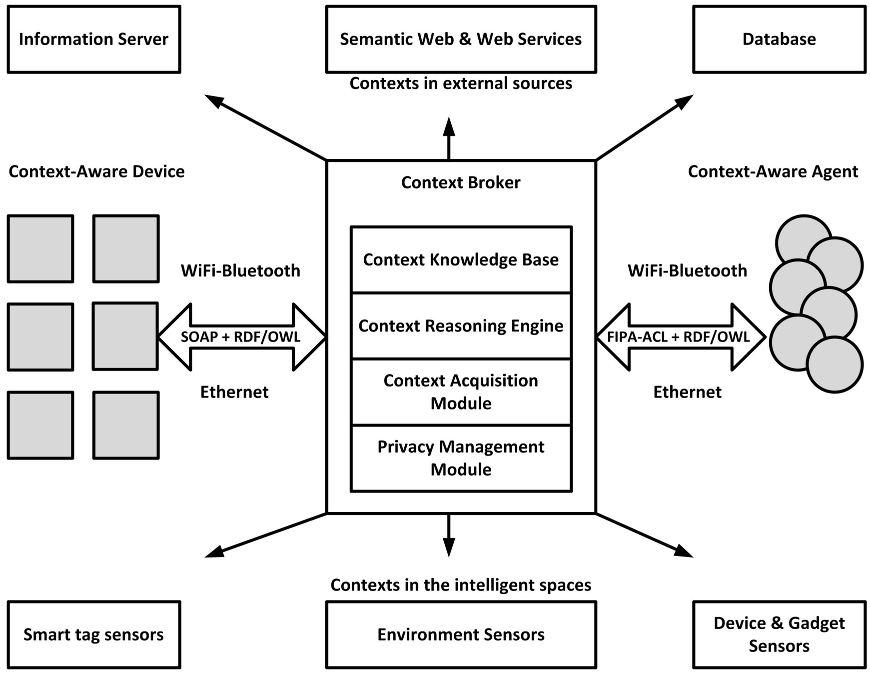

Figure 2.8: The context broker architecture [7]

The sentient object architecture, proposed by [8] was based on a Sentient Object Model. This architecture was designed to support the development of context-aware applications in an ad-hoc environment. As shown in Figure 2.9, the sentient object comprised three main components: sensory capture, context hierarchy and Inference engine. The sentient object can be both producer or consumer of other sentient objects, while it can communicate with the producers of events (sensors) or the consumers of events (actuators).

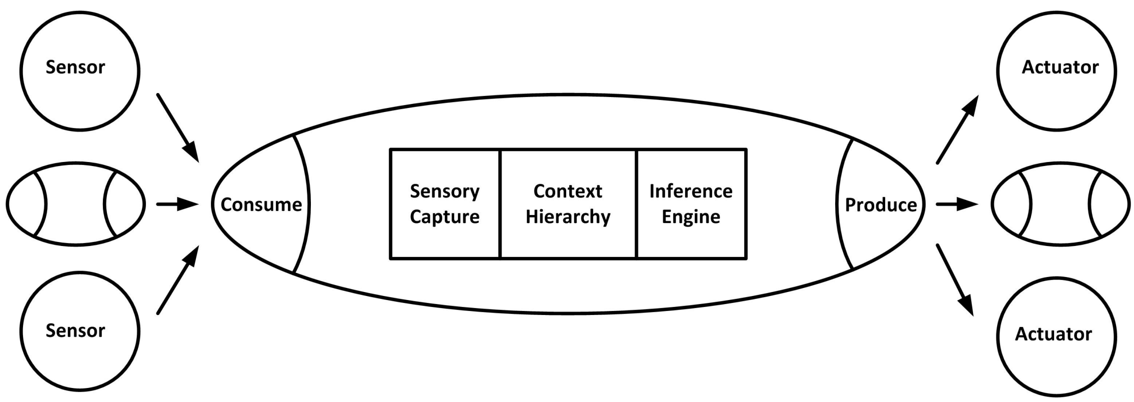

Figure 2.9: The sentient object architecture [8]

Context Toolkit is another architecture that has been suggested by [9] in order to ease the development of context-aware systems; the architecture as illustrated in Figure 2.10, consists of six main components: Sensors, Widgets, Interpreters, Aggregator and the Discoverer which is responsible for maintaining a register of available components for use by applications.

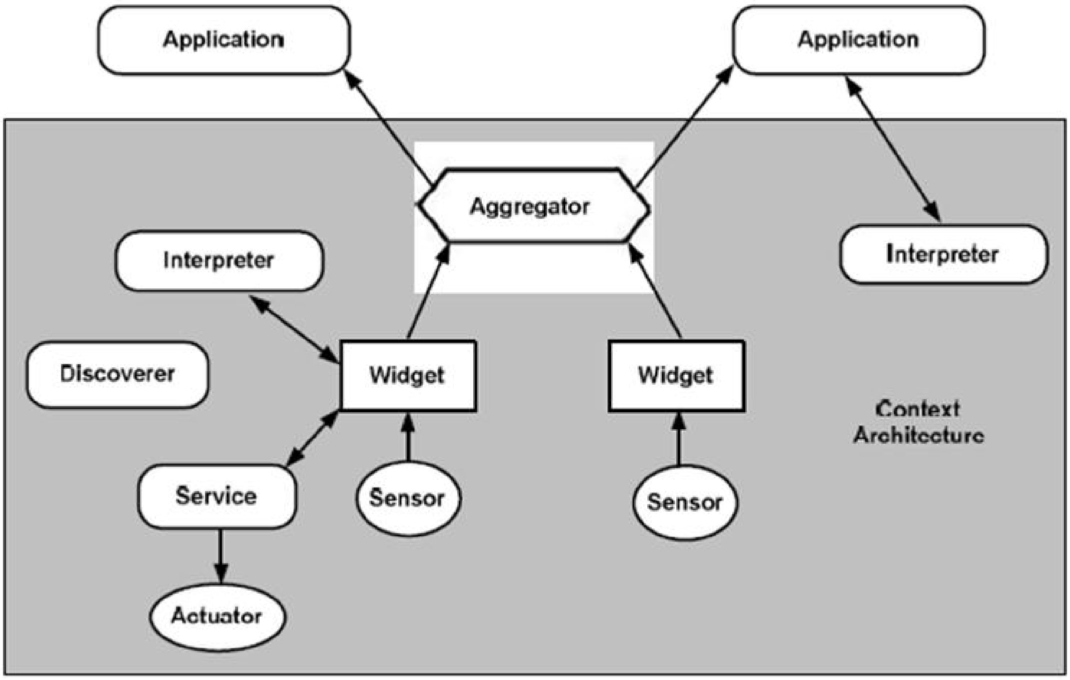

Figure 2.10: The context toolkit architecture [9]

The hydrogen project [10] proposed another architecture specialised for mobile devices and based on a five-layered conceptual model; the architecture contains three layers as depicted in Figure 2.11. The adapter layer which is responsible for retrieving context data; the management layer(the context server) responsible for offering contextual data obtained from the adapter layer to applications; and the application layer, which is responsible for implementing the appliance code.

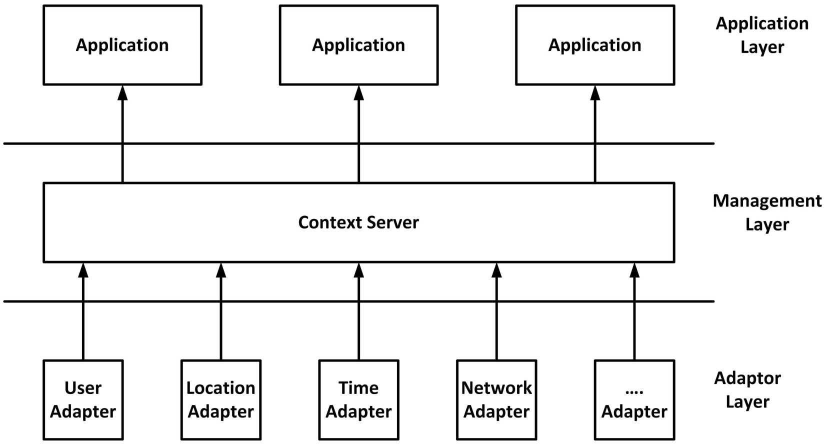

Figure 2.11: The hydrogen project architecture [10]

As described above, several context-aware systems architectures have been developed in different manners, in order to meet specific requirements. Some of them were aimed at constructing secure context-aware systems, while others sought to adapt the systems' behaviour according to their environment. Reviewing the above architectures revealed their similarity regarding the layers they incorporated (i.e. sensing layer and reasoning layer). However, they have several limitations such as: low performance processors, low storage capacities, a limited number of integrated sensors and an inability to reason under uncertainty (driver's behaviour), as no architecture from the above incorporated a mechanism for uncertainty reasoning (i.e. dynamic Bayesian network). Moreover, no architecture has been designed for the vehicle's OBU, which makes it impossible to use one of these in our system. Thus, it is essential to design a new OBU architecture for the purpose of detecting the behaviour of the driver in VANET, as the available architectures do not satisfy system requirements. Different kinds of sensors have to be integrated in order to capture information about the driver, the vehicle and the environment. In addition, uncertainty reasoning technique has to be incorporated in order to combine the collected data and deduce driver's behaviour accurately. The proposed architecture, which is designed to achieve these tasks is described in chapter 4.

2.3.7 Context Modelling and Reasoning

A context model is required to define and store the data gathered by sensors (context) in machine executable form, context modelling approaches can be summarised according to [52] into:

- **Key-Value models:** This approach is the simplest form involved in modelling a context, using a simple pair (key-value) to define the attribute and its value such as: location-in room 35, name-claudia, time-12:00 p.m.; key-value pairs are easy to manage and frequently used but lack the potential for advanced structuring in the form of an efficient context retrieval algorithm.

- **Markup schemes models:** A markup scheme is a hierarchical data structure that contains markup tags with attributes and contents. The content of markup tags are recursively defined by other markup tags.

- **Graphical models:** The Unified Modelling Language (UML) has a powerful graphical component, and is considered as a general purpose modelling tool. Owing to its generic structure it is suitable for modelling the context. This type of modelling approach is appropriate for sourcing an Entity Relationship model ER-model, which is a very valuable as a structuring tool for a relational database in system based context management architecture.

- **Object oriented models:** Using this approach to model a context offered to provide the reusability and encapsulation to tackle the problem of context dynamism. This approach encapsulates the details of context processing, and accessing contextual information is done via identified interfaces only.

- **Logical based models:** Logical based models define their context as fact, expression and rules. In logic concluding expressions are derived from a set of other expressions and across all logic based models there is a high degree of formality.

- **Ontology based models:** Ontologies are powerful tools that can be used to identify concepts and interrelations. Due to their high expressiveness and the possibilities they are suitable for modelling a context. Context Broker Architecture (CoBrA) system [7], is a superior example of an ontology being used to model a context, the system provides a set of concepts to characterise entities such as (places and persons).

Context data can be used directly by application (i.g. raw coordinates supplied by the location sensor), while it can be raw data which is the data that has to be processed prior to be used by application (i.g. identity of the building the user is in) [61]. To model a context several requirements have to be taken into account in the context model, these requirements according to [62] are as follows:

- Heterogeneity: Context models have to work with different types of sensors.
- Relationships and dependencies: Context models have to be able to capture the relationships between the contextual information.
- Timeliness: Context histories have to be captured by the context model.
- Imperfection: Owing to the heterogeneous nature of the context, data acquired by sensors can be inaccurate; context models have to resolve this problem.
- Reasoning: Context models have to support reasoning techniques that are able to:
  - Perform reasoning about certain contextual information, which is the information that can be acquired directly from the sensors (e.g. entity location).

  - Drive high level contextual information (uncertain contextual information), such as user's activity.

- Usability: Interpretation of the world concepts to modelling constructs and the manipulation of context data must be done in a straightforward way by the context model.

- Efficient context provisioning: The context model must provide efficient access to contextual information.

The aforementioned context models do not support reasoning about uncertain contextual information [63]; they are only able to define and store some types of contextual data which is the certain contextual information, such as the temperature of the room, light, weight, etc. due to their reasoning limitations (i.g ontology based model) [62, 64] or their inability to provide reasoning mechanism (i.g. graphical model) [65].

High-level contextual information such as the activity of a person (e.g. the person is sleeping, the driver is driving his/her vehicle normally) cannot be captured directly by sensors and may be incomplete and inexact information (e.g. the system cannot judge that the driver is driving his/her vehicle without being drowsy, affected by fatigue or alcohol by only accessing simple information about the vehicle such as steering angle, vehicle speed or vehicle direction) [66, 67, 68, 69]. For these reasons this contextual information is prone to uncertainty [63, 70, 64].

Reasoning about uncertainty on the one hand, leads to improvements in the quality of contextual information by performing information fusion to the sensed information that refer to the same context in order to resolve the conflict and increase the level of confidence. On the other hand, deducing high level contextual information by combining the low-level data that have sensed by different sensors. Several types of reasoning techniques have been used to reason about uncertainty such as [71, 62, 63, 72, 73]:

- **Fuzzy Logic:** As a form of data processing, Fuzzy Logic is employed by advanced electronic computer systems. In less complex information processors, the possibility that a particular event will occur is expressed as a certainty (either false or true) and it is represented by the binary digits 0 or 1. Fuzzy Logic systems, in contrast, one of its aims is the development of a methodology for the formulation and solution of problems that are too complex or too-ill defined to be analysed by conventional technique. Hence, it will breaks down the chance of the occurrence into varying degrees of truthfulness or falsehood (e.g. will occur, probably will occur, might occur, might not occur). It is suitable for performing multi-sensor fusion and demonstrating subjective context as well as for resolving conflicts between different context types. Two or more fuzzy sets can be combined to acquire a new fuzzy set, with its own individual membership function. The problem with fuzzy logic is its inability to deal with inaccurate and incomplete data [74].

- **Probabilistic Logic:** This tool permits making a logical assertion that is associated with a probability. It allows to make statements such as " The probability of C is more than 1/2" and " The probability of A is at least triple the probability of B" where C, A and B are random variables. Moreover, the probabilistic logic provide the ability to write rules in order to reason about the probabilities of an events in term of the probabilities of other events. These rules can be used to deduce high-level probabilistic contextual information and to improve the quality of contextual information. A prolog engine can be used to reason on these rules [75]. However, it does not provide adequate expressive power to capture the uncertainties and dependencies between variables, and to model the temporal aspects of the domain [76].

- **Neural Networks:** They are a network of interconnected constituents known as neurons, they are designed to mimic the way in which human brain acts and performs a tasks and functions. Neural networks perform a parallel and a non-linear computing by using their processing units (neurons) capabilities and they are sufficient in resolving the problems that needs mapping a large number of input into small number of outputs [77, 78, 79]. However, there are major problems with using this method, such as the fact that it is so hard to decide which network architecture is suitable for this application. The prediction capability of the network is less accurate than other types of reasoning techniques, such as Bayesian networks [80], and the training of the network is usually slow [73].

- **Hidden Markov Models (HMM)** It is a statistical model that represents a series of states, and the system transition from one state to another according to a transition probability. The states of the model are hidden and can not be observed directly, each state produce a set of observations (signals) that can be captured to observe the state. HMM is a subclass of Dynamic Bayesian Networks (DBN) [81, 82]. The difference between them is that DBN represent the hidden state in term of a set of random variables. While, the HMM represent the hidden state in term of single random variable. Moreover, the graph structure of DBN is more general comparing with the restricted topology of HMM [83].

- **Bayesian Networks (BNs):** They are directed acyclic graphs where the nodes represent various random events and arrows represent the relationships between the nodes (parent nodes and child node). They are efficient in representing and reasoning about uncertain contextual information and provides a different kind of reasoning, such as deducing results from causes and vice versa and they are considered as efficient tools for inferring low level information to deduce high level context. Bayesian Networks work with evidence and beliefs that represent a single time slice which means that it is not appropriate for the systems that are changing over time (the random variables at the current time are affected only by the variables at this time not by the variables at the previous time slice) [63, 75, 76, 84, 22].

- **Dynamic Bayesian Networks (DBNs):** They are considered as a set of Static Bayesian Networks interconnected by sequential time slices. The relationships between two adjacent time slices can be modelled by a Hidden Markov Model. DBNs are a general form of HMM, in which the particular state is characterised by a set of random variables rather than using a single discrete random variable. They are differ from static BNs in that they use the previous time slice with corresponding sensors readings at the current time to characterise the current state (the random variables at time slice t with the state at time t-1) [24, 22, 85, 82].

Drivers' behaviour is considered as a high-level context, so data from different kinds of sensors has to be combined (inferred) in order to characterise this context. At each time instance, the driver could be exhibiting a specific behaviour, so the system has to be able to model the behaviour at different time instances. Furthermore, uncertainty and randomness are the two main factors that appear while modelling driver behaviour due to the inaccurate readings of some sensors, and the fact that different variables need to be combined. Several methods have been used to combine the sensory data, each of which has its advantages and disadvantages. In the system for this research, Dynamic Bayesian Networks were chosen to combine data from different types of sensors to deduce the behaviour of the driver, for the following reasons:

- They are able to model time series data by taking into account the static and temporal aspects of the domain (e.g. modelling dynamic events, which are the events that evolve over time) [76, 84, 86, 24, 87, 22].
- They provide a framework that is capable of inferring data with different levels of abstraction (e.g. multiple context from different kinds of sensors) [76, 22, 88].
- They are considered to be the most reliable method of dealing with inaccurate data and unobservable physical values [62, 76, 73, 84, 86, 22].
- They are efficient for combining uncertain context from a wide range of sensors in order to deduce high-level contextual information (they reason about uncertain context) [62, 89, 84, 86].
- They are able to combine prior and current data [76, 73, 90, 84, 86].
- They have efficient algorithms for both inference and learning [89, 84, 86, 88].
- They provide algorithms for network structure learning [91, 89, 84].

2.4 Driver Behaviour Monitoring and Detection Systems

Several researchers have worked on the development of driver monitoring and detection systems using different methods. As shown in Figure 2.12, some have attempted to measure the driver's behaviour or that of the vehicle's in isolation in order to detect fatigued, drunk or drowsy drivers. Meanwhile, other researchers have tried to monitor the driver, the vehicle and the environment all together in order to detect the behaviour of the driver. Each category will be taken separately and their advantages and disadvantages discussed below.

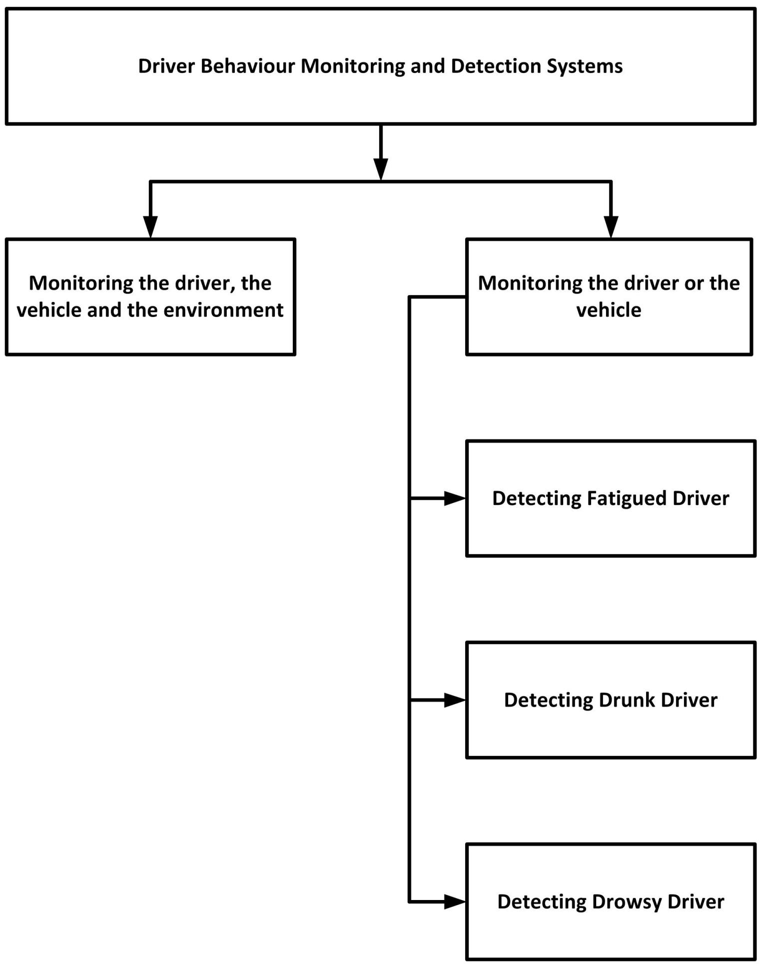

Figure 2.12: Driver Behaviour Monitoring and Detection Systems

2.4.1 Monitoring the driver, the vehicle and the environment

Sun et al [25] focused on building a context-aware smart car by developing their hierarchical model which is able to collect, reason about and react, according to contextual information about the driver, vehicle and environment, in order to provide safe driving and a comfortable driving environment.

They proposed general architecture for the smart car to collect contextual information about the situation with the traffic (e.g. relative velocity), driver situation (e.g. driver's gaze) and vehicle situation (e.g. operates normally). The smart car is able to assess the risk after collecting this information and react upon the current driving situation. Their context model classified contextual information according to degree of abstraction and semantics into three layers; which are the sensor layer (the source for contextual information), the context atom layer (is an abstraction between the real world and the semantic world) and the situation layer (complex situational information are deduced here by fusing multiple context atom).

They use the ontology in the context atom layer to interpret and share contextual information about the environmental context (e.g. weather and road surface conditions), vehicle context (e.g. engine status) and the driver's context (e.g. driver's physiological conditions and blood pressure). While, high level contextual information, such as the current state of the car is deduced in the situational layer using Petri net. Deducing the state of the car includes two parts: an offline training phase responsible for creating a pattern for each individual situation, and an online recognition of the situation based on its pattern at the current time.

They developed a software platform for the smart car as shown in Figure 2.13, which consists of four layers:

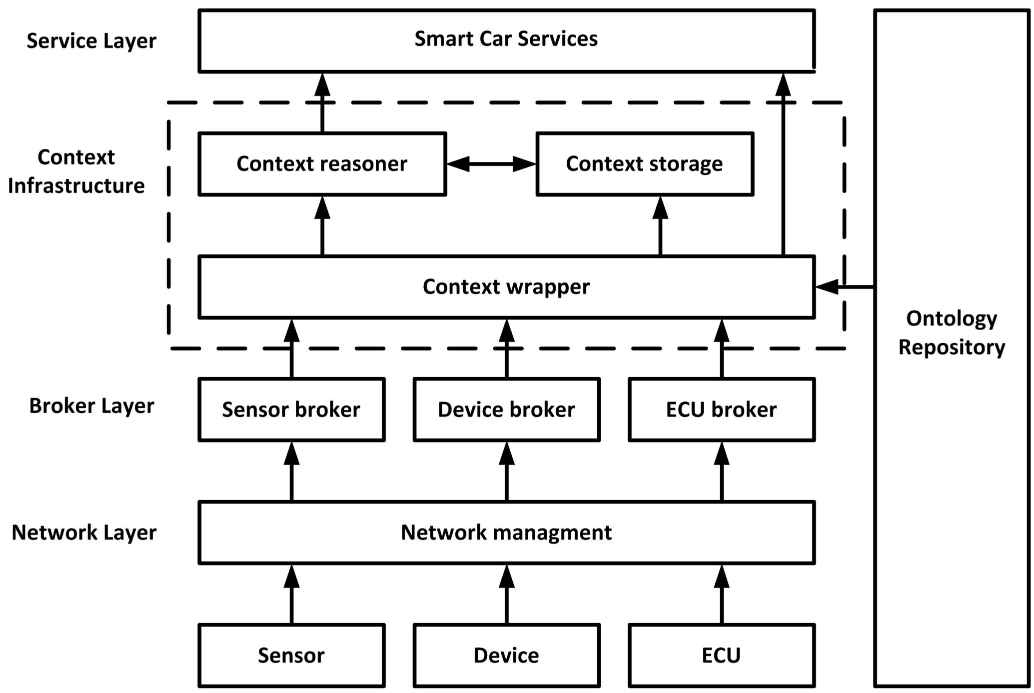

Figure 2.13: Software platform for the smart car

- Network layer: Responsible for connecting all the devices.
- Broker layer: Which contains the sensors broker responsible for registering and discovering any new sensors to be added to the model.
- Context infrastructure: The context infrastructure contains three parts: the context wrapper, is responsible for transforming the data gathered by sensors into semantic context atom; the context reasoner, which is responsible for situational training and recognition; and context storage, where historical contextual information is stored.
- Service layer: After recognising the situation a specific service should be triggered by the system, such as slowing down the speed of the car where the distance to the car in front car is detected as less than the safe limit.

However, the state of the driver was treated as an uncertain context in this system, and information fusion was carried out in the situation layer, but this did not take into account the temporal aspects of the behaviour, which is considered to be an event that evolves over the course of driving. Moreover, this system only capture information about the driver's eye movements in order to judge his or her state. This leads to inaccurate and insufficient detection of the driver's behaviour.

In [92], the authors designed their system TOlerant context-aware driver assistance system (TOCADAS), intended to reduce the number of accidents and fatalities by detecting the current driving situation and providing the driver with an appropriate service according to the detected situation.

Their contex model is similar to [25] which consist of three layers: a sensor layer, which is the source of the contextual data, the context atom layer, which is an abstraction between the real world and the semantic world, and the situation layer where the complex situational information is deduced by fusing two or more context atoms.

In the context atom layer the context model uses ontology to represent the data about the driving environment that includes static data (e.g. road signs and number of lanes), dynamic data (e.g. data that describes the motion of the vehicle) and environmental data (e.g. wind speed and temperature).

Their system uses fuzzy membership functions to map the acquired contextual information into linguistic variables and constructs a linguistic information system. Then the association rules mining based on a rough set is used to extract control rules for the system from a linguistic information system. After detecting the situation by employing the temporal context situation pattern, which a way to describe the behaviour of the vehicle that changing over time. The most appropriate service for the driver will be triggered according to the current driving situation, based on the degree of context pattern similarity.

However, this system focuses on providing the best services to the driver according to the current driving situation. The driving situation is considered to be an interaction between the vehicle and the environment only, where only information such as wheel fraction, brake signal, vehicle acceleration and outside temperature is captured, without taking into account the context related to the driver, such as driver eye movements, which can decide the fatigue level of the driver. This drawback in turn leads to inaccurate and insufficient detection of driver behaviour.

2.4.2 Monitoring the driver or the vehicle

Driver errors is considered as the main factor in most road accidents [18]. Therefore, many researchers have tried to develop fatigue, drunk or drowsy driver detection systems. The following is a summary of the main researches in this field.

[26] designed a system that uses a video camera for fatigue detection. A video camera is used to record the face of the driver, then the video is converted into frames. After locating the eyes in each frame and determining that the eyes are closed or opened for a defined period of time the system decides whether the driver is affected by fatigue or not. If the driver is judged to be affected by fatigue the system generates a warning signal to the driver.

In [93], the authors proposed a system that can predict the driver's level of fatigue and trigger a warning to the driver in real time. By acquiring more than one

visual cue including eyelid movement, gaze movement, head movement and facial expression and then using a probabilistic model to use the cues acquired to model driver's fatigue level.
Their system uses two cameras to extract visual cues from the driver. Their probabilistic model uses a Bayesian network to predict the driver's level of fatigue by using previously acquired cues and contextual information that is relevant to the driver, such as the health and history of sleep.

[94] designed a system to track the driver's head movements, face and eyes using Unscented Kalman Filter in order to detect the driver fatigue and control the vehicle using intelligent vehicle cruise control in an attempt reducing the number of accidents on road. The system tracks the eyes of the driver and if they are closed over 5 consecutive frames the cruise control is start-up to reduce the vehicle speed to be below 5km/h, the intelligent vehicle breaking control is started up to avoid accidents and the alarm will issue to the driver. The proposed system evaluated in a vehicle equipped with a camera and computer for extracting the feature and the accuracy in detecting the fatigue were 99.5%.

Dai et al [27] developed a program that works in a mobile phone and contains an accelerometer and orientation sensors placed in the vehicle to detect a drunk driver in real time. The program compares current accelerations with the typical drunk driving patterns and when the program indicates that the driver is influenced by alcohol it generates warning messages to alert the driver and send a message to inform police.

The author categorised the cues related to the drunk drivers as follows:

- Problems in maintaining the lane position: weaving, drifting, swerving, turning illegally, turning with wide radius and turning abruptly.

- Problems in controlling speed: accelerating or decelerating unexpectedly, stopping imperfectly and breaking irregularly.
- Judgement and vigilance problems: driving with tyres on lane markers, driving on the wrong side of the road, driving at night without turning the front lights on and takeing a long time to respond to traffic lights.

The probability of the driver being influenced by alcohol is 50.75%, when there is a problem in maintaining lane position; 45.70% if there is a problem in controlling speed, and 40% for judgment and vigilance problems. If more than one cue is observed then the probability of drunk driving is increased.

Their system focuses on cues such as maintaining lane position and speed control to detect drunk drivers. They map the cues regarding maintaining lane and speed control into lateral acceleration and longitudinal acceleration respectively. Using mobile phone sensors provides the capture of acceleration information (lateral and longitudinal acceleration) and the system processes this information and uses multiple round pattern matching to improve the accuracy of the system. If the system detects a drunk driving it will generate an alert message to warn the driver and the phone will automatically call the police.

[95] have developed a drunk and drowsy driver detection system combining breath and alcohol sensors in a single device. This device is able to measure the degree of alertness of the driver and detect the charged water clusters in the driver's breath to detect alcohol using breath and alcohol sensors.

The system tests the expired gas in the breath, which includes positively or negatively charged water clusters, by applying an electric field. Then breath and alcohol sensors are used to detect the breath and the alcohol in the breath of the driver.

Alcohol sensor is responsible for detecting alcohol in the driver's breath, while, detection of the drowsiness is done via a breath sensor when it detects that the conscious breathing has become unconscious breathing.

Ueno et al [28] developed a non contact system to prevent the drowsiness of a driver by detecting the eyes of the driver and checking whether they are opened or closed using a CCD camera. The system is based on capturing the face of the driver and using image processing techniques to check if the eyes are closed for long intervals. If the eyes are closed then the driver is drowsy and the system will issue a warning to the driver.

Garg et al [96] proposed a drowsiness detection and security system based on image processing and pattern recognition, tracking the eyes of the driver and extracting the Iris image by using a single camera and a temperature sensor to measure the heat from the driver's body. Their system began by capturing the driver's image and recognising the iris of the eyes. Then engine will start only if the driver is authentic. Then, the drowsiness system will start by checking if driver's eyes are opened or closed and if the heat of the body is normal or reduced. If the system recognises that the driver is drowsy by discovering the eyes are closed and the heat of the body is reduced a warning message is triggered and emitted for a 10 second period. This system also provides an opportunity to call the police via GPS using the eyes; if the driver opens one eye and closes the other the system will call the police without revealing other people around the driver.

In [97], a system for drowsy driver detection in real time driving by collecting information about the vehicle's behaviour; such as the speed of the vehicle, the vehicle's lateral position, yawing angle, steering wheel angle and the vehicle's lane position has been proposed. The system uses artificial neural networks in order to combine different indications of drowsiness and to predict if a driver is drowsy or not and issue a warning if required.

The authors in [98], used a 990nm wavelength infra-red light emitting diode and a proximity sensor fixed on the steering wheel to collect information about human pulse waves; human pulses are prone to change, especially when sleeping where it becomes more monotonous than normal. Their system used the photoplethysmograpgy (PPG) to analyse the data gathered by the sensors (the infra-red signals reflected from driver finger) and to then compare it with a predefined user's preferences to judge if the driver is drowsy or not.

The driver behaviour detection systems described above focus on the detection of the driver's status (e.g. drunk, affected by fatigue or drowsy) and the issuing of a warning messages to the driver in order to prevent road accidents. Although these systems have achieved good results in terms of improving road safety and enhancing traffic efficiency, we found that they have a number of drawbacks, such as being limited to capturing contextual information about the driver or the vehicle in isolation, without any consideration of the environmental contextual information.

The importance of including information about the driver, the vehicle and the environment is that this improves the accuracy and efficiency of the detection process; for example, capturing only information about the vehicle will fail to detect a fatigued driver who is driving on a straight road with out changing the vehicle behaviour even if his or her eyes are closed and he/she is affected by fatigue. Moreover, these systems are limited to detecting only one type of driving behaviour (i.e. fatigue), and some of them use intrusive sensing equipment to acquire the context, and hence reduce the readiness of the driver to use the systems.

There is still no comprehensive and non-intrusive system that is able to monitor the driver, the vehicle's state and the environmental changes in order to detect different styles of driving behaviour (e.g. drunk, fatigued, reckless and normal driving) by performing effective reasoning about uncertain context, and therefore accurately detecting the current behaviour of the driver and alerting him or her in order to prevent accidents from occurring.

This study is attempting to construct a comprehensive and non-intrusive driver behaviour detection system from the view point of context awareness. The system utilises a context-aware system approach to detect four types of driving behaviour: drunk, fatigued, reckless and normal behaviour, by performing reasoning about certain and uncertain contextual information. Most of the context that can be used to characterise the behaviour of the driver, including information about the driver, the vehicle and the environment, will be collected and analysed. A probabilistic reasoning technique (uncertainty reasoning), which is based on Dynamic Bayesian Networks, will play an important role in deducing high-level context (the behaviour of the driver), thus providing a flexible yet more accurate proactive driver behaviour detection system.

2.5 Summary

This chapter has presented an overview of VANET, context and context-aware systems, with an explanation of the main techniques that have been used to model and reason about certain and uncertain context, including the reasons behind opting DBN as the reasoning technique in this thesis.

A broad classification was then presented in order to illustrate the work that has been carried out in the field of driver behaviour monitoring and detection systems. This was discussed in order to explain the main idea and the mechanism of each system, with an illustration of the main drawbacks associated with them. Finally, the main differences between this work and previous work were explained.

These systems have tried to detect one type of driver behaviour (i.e. drunk), without taking into consideration all the factors that are related to this behaviour, which has the impact of reducing the accuracy and efficiency of the detection.

The next chapter (Chapter 3) will talk about the behaviour of the driver and the Dynamic Bayesian Network, which is the reasoning technique used in the work for this thesis.

Chapter 3

Fundamental Principles of Bayesian Networks

Objectives:

- Define the aim of the driver behaviour detection system
- Present an overview and definition of driver behaviour
- Describe the features and characteristics of Bayesian networks and dynamic Bayesian networks
- Illustrate the currently available software which supports the implementation of dynamic Bayesian networks
- Justify the use of GeNIe version 2.0 software

3.1 Introduction

The key concept that has been introduced in this thesis is the detection of the abnormal behaviour of the drivers in VANET, utilising a context-aware system approach, by capturing information about the driver, the vehicle and the environment, then performing reasoning about uncertain contextual information, so as to alert the driver and prevent accidents from taking place. Most of the proposed driver behaviour detection systems have tried to detect the status of the driver (e.g. drunk or fatigued) by capturing information related to the driver and the vehicle or the driver and the environment by performing diverse reasoning methods. However, the behaviour of the driver is a process that evolves over time, and this is considered to be a complex interaction between the driver, the vehicle and the environment, which means that considering the behaviour of the driver as a certain context, or capturing information about the driver or the vehicle in isolation, will lead to inaccurate and insufficient behaviour detection. Furthermore, there is still no comprehensive system that is able to detect different kinds of abnormal driving behaviour.

In order to overcome these drawbacks, a novel driver behaviour detection technique has been developed in this thesis. This technique takes into consideration the behaviour of the driver as an uncertain context, and allows the behaviour detection process to capture the static and temporal aspects related to the behaviour by performing reasoning under uncertainty (probabilistic inference), using DBN as the reasoning technique. The system is able to detect four styles of driving behaviour (i.e. drunk, fatigued, reckless and normal behaviour). The context used to infer the behaviour of the driver includes a combination of driver-related context (i.e. driver's eyes movements and level of alcohol in the driver's blood), vehicle-related context (i.e. position in lane, vehicle speed and acceleration) and environmental context (i.e. time, time zone, temperature and noise), which will lead to more accurate and robust behaviour detection.

Based on the proposed technique, a novel OBU driver behaviour detection architecture has been developed; this architecture has been designed and developed based on the concept of context-awareness. The architecture comprises the three main phases of context-aware system: sensing, reasoning and application. It illustrates how a vehicle in VANET can sense and reason about and react to information related to the driver, the vehicle and the environment in order to detect the current behaviour exhibited by the driver. Chapter four will explain this architecture in detail. While, in this chapter, definitions of normal and abnormal driving behaviour are given, and the circumstances and conditions that are required in order to implement the driver behaviour detection system are introduced. An overview of static BNs and DBNs, with their mathematical notions, inference algorithms and the main steps that have to be followed to create a DBN, is presented. Finally, an illustration is given of the main software packages that can be used to implement and evaluate DBNs.

3.2 Overview of Driver Behaviour

Several definitions of driver behaviour have been proposed in the literature. Before defining the behaviour of the driver, it is necessary to clarify what driving is. Several authors [99, 100, 101] have defined driving as the interaction between the driver, the vehicle and the environment (surrounding road information and traffic). The behaviour of the driver was defined by [102] as a sequence of actions, each of which is associated with the specific state of the driver, the vehicle and the environment, and characterised by a set of contextual information. Another definition was given by [103], who referred to driver behaviour as a sequence of internal states of the driver, each of which can be observed through capturing a set of observable features (contextual information) that is associated with it.

Yu et al [69] classified driving styles into two types: normal and abnormal, and stated that the status of the driver changed from normal to abnormal due to him or her being affected by fatigue or stress. They defined normal driving behaviour as the status when the driver is sober, focused, making the right judgements on the road and able to respond with accurate and quick reactions in case of an emergency or accident. While, abnormal driving behaviour was defined as the status when the driver is drowsy, vision is affected and the driver is making misjudgements and unresponsive actions. In [104], an aggressive driver is defined as one who drives with sudden acceleration and hard braking. In [105], aggressive driving is defined as when the driver makes a combination of moving traffic offences that may cause a danger to other drivers or property. They characterised an aggressive driver as one who exceeds the speed limit, follows the vehicle ahead too closely, performs unsafe lane changes and fails to obey traffic control rules (e.g. traffic signal).

In this study, the behaviour of the driver is defined from the perspective of context awareness as follows:

Driver behaviour is a complex and dynamic interaction between three entities: the driver, the vehicle and the environment. It is described as a transition between a sequence of states (for example normal, affected by fatigue, drunk or reckless), over the course of driving a driver will be in a particular state, which he or she may remain in for a period of time and then potentially changing to a different state. Each state can be characterised by capturing a large amount of contextual information of relevance to the interacting entities. The behaviour of the driver is considered to be normal (safe) if his or her actions associated with the current state will not lead to an accident; it is otherwise considered to be unsafe (abnormal).

The behaviour of the driver can be represented as follows:

$$B = \{S_{t=1}, S_{t=2}, ..., S_{t=n}\} \tag{3.1}$$

Where (B) is the behaviour of the driver, (S) is the state and (t) is the time. The states of the driver were classified into four classes: normal (S_n), drunk (S_d), fatigued (S_f) and reckless (S_r). According to the above definition, each state may be characterised by capturing a set of observable context (C). Therefore, the state can be represented as in the following equation:

$$S_{t=i} = \{C_1, C_2, C_3, ..., C_k\} \tag{3.2}$$

In conclusion, the behaviour of the driver is considered as the current unobservable state $(S_{t=i})$ that can be characterised by capturing a set of observable context (C_j), where $(S_{t=i})$ is the state at the time $= i$ and (C_j) is the context that need to be captured in order to characterise the state.

As shown in Table 3.1, a set of context and their possible values, which have been used to characterise the behaviour of the driver, have been specified from the previous published work. The context illustrated in this table do not constitute a complete list of all possible features. In fact, additional context can be used and analysed in order to detect the behaviour of the driver. The table presents the context type, the sensors used to acquire this context, the possible values of the context and the type of abnormality (e.g. fatigued, drunk or reckless).

Information	**Sensors**	**Values of Context**	**Driver's Status**	**Reference**
Speed	Speed sensor	Good control to the speed, Bad control to the speed	Fatigue, Drunk, Reckless	[27, 106, 107]
Vehicle heading	Adaptive hello messages	Same road direction, Opposite to the road direction	Drunk	[27]
Driver head position	Camera	Straight, Frequent tilts	Fatigue	[108, 93, 28]
Driver eyes state	Camera	Eyes are opened, Eye are 80% covered by eyelids for 20 seconds	Fatigue	[108, 93, 109, 110]
Position over the lane	Camera	Driving between the lane marker, Changing lane frequently, Driving with tyres on or our the lane	Drunk, Fatigue	[27, 106]
Intoxication	Alcohol + Breath sensors	No alcohol exist, Alcohol exist	Drunk	[95]
Time of the day	GPS	(3-5 Am,3-5 Pm) sleepy, (9-11 am, 9-11 pm)awake	Fatigue	[76]
Acceleration	Accelerometer	Moderate acceleration, Sudden acceleration/ deceleration	Fatigue, Drunk, Reckless	[27, 107]

Table 3.1: Contextual information used to detect the behaviour of the driver

Based on Table 3.1 and the previous definitions of driving behaviour [103, 102, 111, 29, 100, 105, 99], four categories of driving behaviour have been defined, as follows:

1. **Normal behaviour:** The behaviour is considered to be normal when the driver is concentrating on the driving task. This can be characterised by controlling the speed of the vehicle, avoiding sudden acceleration, driving without alcohol intoxication, maintaining a proper position between lane markers and having eyes open while driving. When the driver matches these criteria, the behaviour is considered to be normal.

2. **Drunk behaviour:** The behaviour is considered to be drunk when the driver is driving while intoxicated by alcohol. This behaviour can be characterised by a set of observable actions, such as sudden acceleration, driving without maintaining the proper lane position, driving with out controlling the speed etc, but is not limited to them.

3. **Fatigued Behaviour:** In [106], fatigue is defined as an evolving process that increases during driving, which leads to reduced effectiveness in driving. In [112, 106, 113], it is stated that a driver who is driving after a period of 17 hours without sleep behaves exactly the same as a driver who has 0.05% intoxication with alcohol, and one who is driving after a period of 24 hours without sleep behaves exactly the same as a driver who has 0.1% intoxication with alcohol. Depending on this argument, fatigued driving has been defined as driving that exhibits the same characteristics as drunk driving except that there is no alcohol in the driver's blood.

4. **Reckless behaviour:** In [107], the reckless driver is defined as a driver who drives at high speed, with a high degree of acceleration, and puts other road users at risk. A driver is classified as driving in this category when there is no alcohol intoxication and the driver's eyes are open, but the driver is exhibiting

the following behaviours (although not limited to them): driving with sudden acceleration, not maintaining the proper lane position, etc.

3.3 Assumptions

In order to attain the objectives behind developing the proposed driver behaviour detection system, a number of assumptions have to be taken into account so as to accomplish the requirements with which this thesis is concerned, as follows:

- Each vehicle is equipped with a set of sensors (the source of contextual data). These include cameras, an alcohol sensor, a speed sensor and the an accelerometer sensor.
- Each vehicle is equipped with a Navigation System (NS) and a preloaded digital map.
- Each vehicle is equipped with a VANET-based On Board Unit (OBU).
- The roads are managed by the Traffic Management Centre (TMCs) enable vehicles to collect information about the driving environment.
- Each vehicle collects information about its speed, position and direction using devices it is equipped with.

3.4 Bayesian Networks Overview

In this section, an illustration of Bayesian networks and dynamic Bayesian networks, on which the behaviour detection algorithm in this thesis will be based, is given in detail. Bayesian networks, or the so-called Bayesian belief networks, are directed acyclic graphs that represent the conditional independence between a set of random variables, and which deal with uncertain information and probabilistic inference upon receiving evidence. They can represent different types of uncertain information and the probabilistic relationship between them, then combine this information into one inference system to describe the uncertain domain. These networks consist of a set of nodes that represent the random variables and a set of arcs representing the dependencies between them [63, 11, 22, 85, 114]. Bayesian networks have been used in several expert systems and in the field of Artificial Intelligence (AI) in terms of combining data from different sensors and inferring information from them using diverse inference procedures [115, 116, 117]. Bayesian Networks have the ability to capture the static aspect of the domain, which means that only the observation at the current time slice will be taken into account during the inference process.

From the temporal perspective (the systems that evolve over time), Bayesian networks brought a new approach to model both the static and dynamic aspects of the domain, in order to characterise an event. These tools are called dynamic Bayesian networks, and are able to model the time-series data. They are considered as a set of static Bayesian Networks interconnected by sequential time slices. The relationship between two neighbouring time slices can be modelled using the first-order Markov model, which means that the event desired to infer at a time slice (t) is affected by the random variables at time slice (t), and by the variables at time slice (*t-1*) only [85, 118, 88]. The HMM can be considered a special case of DBNs, where the difference is that DBNs represent the hidden state in term of a set of random variables, while the HMM represent the hidden state in terms of a single random variable. Moreover, the graph structure of DBNs is more general than the restricted topology of HMM [83].

3.4.1 Static Bayesian Networks

Bayesian networks are a special case of graphical model that describes the probabilistic relationship between a set of random variables. They are Directed Acyclic Graph (DAG), which contain a set of nodes, X_i, that represent the random variables, and a set of arcs that represent the causal relationship between the variables. These variables can either be discrete (i.e. the nodes that take the values {high, low}) or continuous (i.e. speed and age which take real numbers {10, 20, 30 etc.} as its values) [11, 119, 116]; only the discrete variables are used in this thesis. Each node in the network has a finite set of mutually exclusive states. The node can be in one of its states with a certain probability and depending on the state of its parents' nodes; a conditional probability table for each node presents the probability distributions of the node. If node X has no parents, the table will represent the prior probability of node X [11, 85, 120]. Figure 3.1 depicts a simple network which includes three nodes (A, B, C).

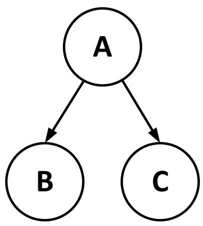

Figure 3.1: Three-node Bayesian Network

As shown in the figure above, the directed arcs from node A to nodes B and C reflect the direct influence of node A on both nodes B and C. In this case, node A is considered a parent of nodes B and C, and is called a root node because it has no parents. Nodes B and C are considered children of node A, and are called leaf nodes because they have no children. The directed arc usually points from a node called a parent node, which is the cause variable to the other, called child node, which is the effect variable. The term 'conditional independence' means that node B in Figure

3.1 is only affected by node A.

3.4.1.1 Bayesian Networks Rules

The backbone of Bayesian networks probability calculations is Bayes' theorem, which was introduced by Thomas Bayes [119, 11], as follows:

$$P(H|E) = \frac{P(E|H).P(H)}{P(E)} \tag{3.3}$$

P(H|E): The posterior probability of hypothesis H given new evidence E.
P(E|H): The likelihood of evidence E given hypothesis H.
P(H): The prior probability of hypothesis H (before receiving the evidence).
P(E): The prior probability of evidence E.

Bayesian networks are used to capture the conditional independence between a set of random variables. They can be considered a knowledge base, which can infer a belief or give a conclusion about an event in the system by propagating the beliefs all over the network upon receiving evidence. In other words, the probability of the hypothesis changes from prior to posterior probability when the system observes new evidence [121, 119, 116].

Assuming $A_i = \{a_1,\ a_2,....,\ a_i\}$ to be a set of i random variables that represent a specific domain, let $P(a_1,\ a_2,\ .....,\ a_i)$ be a joint probability distribution over the domain. Without considering the conditional independence between the variables, we can apply the chain rule for the basic probability theory, as shown in the following equation [118]:

$$P(a_1, ..., a_i) = P(a_1).P(a_2|a_1).......P(a_i|a_i, ..., a_1) \tag{3.4}$$

However, by considering the conditional independence between the random variables, the chain or the product rule for calculating the joint probability distribution in a Bayesian network can be written as follows [116, 11]:

$$P(a_1, ..., a_n) = \prod_{i=1}^{n} P(a_i|Pa(a_i)) \tag{3.5}$$

Where, *Pa(a_i)* refers to the parents of a_i. In other words, it is a set of nodes which directly influences node a_i.

As shown in Figure 3.2, the network consists of six nodes, which means that there are six random variables and six conditional probability tables.

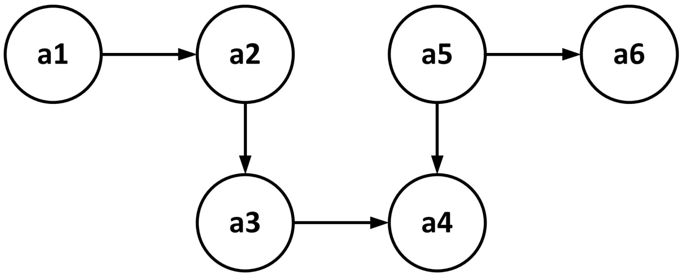

Figure 3.2: Six-node Bayesian Network

For example, node a_4 is only affected by a_3 and a_5. This means that if the values of a_3 and a_5 are known, a_4 is conditionally independent from the rest of the network. Therefore, the joint probability distribution of the network can be written as in the following equation:

$$P(a_1, a_2, a_3, a_4, a_5, a_6) = P(a_6|a_5).P(a_4|a_5, a_3).P(a_3|a_2).P(a_2|a_1).P(a_5).P(a_1) \tag{3.6}$$

In any simple network, the marginal probability or the likelihood of each node being in one of its mutually exclusive states can be calculated using the chain rule and Bayes' theorem. This process is called marginalisation [122].

3.4.1.2 Conditional Probability Table (CPT)

Each discrete variable (node) in Bayesian networks has a conditional probability table associated with it. This table defines the probability distributions of the node and quantifies the strength of the relationships between the variables. The probability distribution is the probability of the node being in one of its states. The main dilemma when using CPTs is the size of the tables. The size of any CPT depends on the number of the node's states and the number of its parents' states. As a result, for Boolean networks, the node with z parents requires 2^{z+1} probabilities in its CPT. This means that reducing the number of parent nodes will lead to a reduction in the size of CPTs [11, 119, 116]. The following two approaches may be used to obtain the values (probabilities) of the conditional probability table for each node in the network [76, 24, 123]:

1. Obtaining the values by performing statistical analysis on a huge amount of training data. Training data is obtained by performing several tests in a testbed specifically designed for the system, and collecting the output for each test or collecting records containing different cases for the same problem. A learning algorithm can then be used to learn (parameterise) the conditional probability tables from this data, so the network can best represent the domain. Several learning algorithms are available in the software packages, such as expectation maximisation (EM) [11].

2. Parameterising the network can be achieved by critically analysing a set of previously published papers and researches that are related or similar to the system. In other words, the conditional probabilities for each node can be acquired manually by studying the relationship between causes and effects in the network, with the help of work that has already been conducted in the field.

Due to the difficulties of acquiring a large amount of training data for this study, as no testbed is equipped with all the sensors required for the model and no previous studies have been done that provide all the data required to parameterise the system, this thesis will use the second approach in order to determine the values of the conditional probability tables for the network.

3.4.1.3 Reasoning with Bayesian Networks

Reasoning with Bayesian networks means calculating the posterior probability distributions for one or a set of variables (nodes) once the observations of some other nodes are known. This is done by flowing the new information all around the network without taking into account the arcs directions. This process is also called inference, belief updating or probability propagation. As shown in Figure 3.3, there are four types of reasoning in Bayesian networks, as follows [11, 124]:

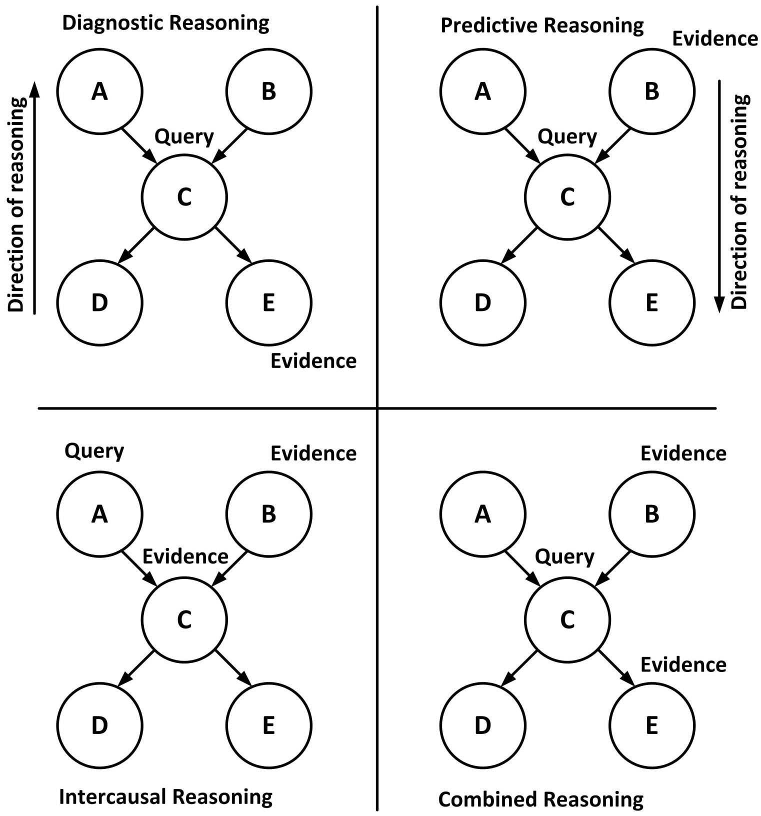

Figure 3.3: Types of reasoning in Bayesian Networks [11]

1. Diagnostic reasoning: This kind of reasoning occurs in the opposite direction to the arcs, which means the new information flows from the effect nodes to the cause nodes. Having knowledge of new information about node E increases the belief about node C being in a specific state.

2. Predictive reasoning: This type of reasoning occurs in the same direction as the network arcs, which means having new evidence about node B increases the belief about node C being in a specific state.

3. Intercausal reasoning: This is a form of reasoning about the mutual causes for a common effect. These common causes are independent in that having new information about node A does not change the belief about node B. However, having new information about node C leads to an increase in the belief about both nodes A and B. If information is known about node B, this information explains the observed node C and decreases the belief about node A. In other words, having information about the effect and one of the mutual explanatory causes will decrease the belief in the other cause. Explaining away is a special form of intercausal reasoning.

4. Combined: In some systems, the reasoning does not fit tidily into one of the above-mentioned reasoning types. In fact, a combination of them can be used in any way (e.g. a combination of diagnostic and predictive reasoning can be used). In this thesis, a combination of diagnostic and predictive reasoning will be used, as the hypothesis node (the node desired to be inferred) has both causes and effects.

Bayesian networks can be singly connected where there is only one path between any pair of nodes in the network. They can also be multiply connected in cases where more than one path between any pair of nodes exists [118, 85]. There are two types of inference algorithms in Bayesian networks: approximate inference algorithms (i.e. stochastic simulation algorithms, model simplification methods, search based methods and loopy belief propagation) and exact inference algorithms (i.e. polytree, clustering, conditioning, arc reversal, variable elimination, symbolic probabilistic inference and the differential method), both of which involve a complex computations. The exact inference is not sufficient when the network is large; hence, an approximate inference can be used. The speed of the inference process relies on several issues, such as the location of the hypothesis nodes, how highly connected the network is and the number of undirected loops in the network.

One single inference algorithm that can be used for all problem domains does not exist; different algorithms can be used for different problems. Therefore, it is important to thoroughly study the relationship between the domain of the problem and the domain of the inference algorithm in order to decide which algorithm is most efficient for the problem [11, 116]. The network that has been designed in this thesis is a singly connected network, as there is only one path between any pair of nodes, and the hypothesis node has both parents and children. Therefore, the polytree algorithm was selected to perform the inference process, because it is an exact inference algorithm and works for a singly connected networks. Moreover, it performs a combination of diagnostic and predictive reasoning. If the cross effects of the nodes are considered in our model, to determine for example, the effect of acceleration node on controlling the speed node, the network will be multiply connected. In this case, an algorithm for multiply connected networks such as clustering algorithm has to be used to perform the inference process.

Pearl's message passing algorithm (polytree)

This algorithm, also called the polytree algorithm, is used to perform the exact inference in the singly-connected networks [125, 11, 117, 116, 126]. The algorithm performs the inference using three parameters, which are: $\lambda(X)$; $\pi(X)$ and α.

Where:
$\lambda(X)$ can be computed from the messages received from node X's children.
$\pi(X)$ can be computed from the messages received from node X's parents.
α is a normalising constant which is equal to $1 / p(e)$.

Let e be a set of values for all evidence in a singly connected network. The network contains a hypothesis node X, parents of node X, $C=\{C_1,\ldots.,C_m\}$, and children of node X, $O=\{O_1,\ldots., O_n\}$. Let x_i, c_m and o_n be a set of values taken by X, C and O, respectively. For any hypothesis node X, e can be divided into two groups; the first group, e^-, denotes all the observed variable that are connected to node X via its children (including node X itself if it is observed). The second group, e^+, denotes all the observed variables which are connected to node X via its parents. The posterior probability of node X given the evidence e can be computed using the following equation:

$$P(X|e) = \alpha.\lambda(X).\pi(X) = \frac{\lambda(X).\pi(X)}{\sum\limits_{x_i} \lambda(X).\pi(X)} \tag{3.7}$$

Where:
$\lambda(X) = P(e^-|X)$
$\pi(X) = P(X|e^+)$

As node X, is a discrete node, which means it has a finite set of states, the λ*(X)* and π*(X)* have to be calculated for all the variables in the networks, and their values will be vectors whose values are associated with each of the discrete values for X.

Computing λ*(X)* has to be done for all children (O_i) of node X, as follows:

$$\lambda(X) = P(e^-|X)$$
$$= P(e_{O_1}, e_{O_2}, ..., e_{O_n}|X)$$
$$= P(e_{O_1}|X).P(e_{O_2}|X).......P(e_{O_n}|X)$$
$$= \lambda_{O_1}(X).\lambda_{O_2}(X).....\lambda_{O_n}(X)$$

The computation for each λ_{O_i}*(X)* is as follows:

$$\lambda_{O_i}(X) = P(e_{O_i}|X)$$
$$= \sum_{o_i} P(e_{O_i}, o_i|X)$$
$$= \sum_{o_i} P(e_{O_i}|o_i).P(o_i|X)$$
$$= \sum_{o_i} \lambda(o_i).P(o_i|X)$$

This indicates that to compute λ*(X)*, we need both the conditional probabilities and the λ*'s* for all children of node X. Therefore, λ*(X)* can be written as follows:

$$\lambda(X) = \prod_{i=1}^{n} \lambda_{O_i}(X) \tag{3.8}$$

Computing π*(X)* has to be done for all parents of node X (C_j), as follows:

$$\pi(X) = P(X|e^+)$$
$$= P(X|e_{C_1}, e_{C_2}, ..., e_{C_m})$$

$$= \sum_{c_1,\ldots,c_m} P(X|c_1,.....,c_m).P(c_1,...,c_m|e_{C_1},...,e_{C_m})$$
$$= \sum_{c_1,\ldots,c_m} P(X|c_1,.....,c_m).P(c_1|e_{C_1}).P(c_2|e_{C_2}).....P(c_m|e_{C_m})$$
$$= \sum_{c_1,\ldots,c_m} P(X|c_1,.....,c_m).\pi_X(c_1).\pi_X(c_2).....\pi_X(c_m)$$

This indicates that to compute $\pi(X)$, we need both the conditional probabilities and the π*'s* for all parents of node X. Therefore, $\pi(X)$ can be written as follows:

$$\pi(X) = \sum_{c_1,\ldots c_m} P(X|c_1,....,c_m) \prod_{j=1}^{m} \pi_X(c_j) \tag{3.9}$$

3.4.2 Dynamic Bayesian Networks

Dynamic Bayesian networks are directed acyclic graphs that represent the conditional independence between a set of random variables, and which deal with uncertain information and probabilistic inference upon receiving evidence. They consist of a set of nodes that represent the random variables and a set of arcs that represent the conditional independence between variables. DBNs take into account the static and temporal aspects of the domain. In other words, they are used to model the time-series data (e.g. the events that evolve over time which can not be modelled by detecting an observation on a particular time instance only). The word dynamic does not mean that the network structure or the nodes change automatically, but the dynamic systems can be modelled. They can be viewed as a set of static Bayesian networks that are interconnected by sequential time slices, each of which represents a snapshot of the evolving process at a given point in time [127, 24, 88, 82, 118, 85].

Let $[Z_1,\ Z_2,\ \ldots,\ Z_t,..]$ be a semi-endless collection of random variables. $Z_t = (C_t, X_t, O_t)$ represents the input, hidden and output variables of the model at a given time. DBNs are used to model the probability distribution over the semi-endless collection of random variables. A DBN can be defined as a pair of *(S,$\overrightarrow{S}$)*, where S is a static Bayesian network that defines the prior state distribution *P(Z1)*, and $\overrightarrow{S}$ defines *P($Z_t|Z_{t-1}$)*, which is a two-slice temporal Bayesian network, as shown in the following equation [83]:

$$P(Z_t|Z_{t-1}) = \prod_{i=1}^{N} P(Z_t^i|Pa(Z_t^i)) \tag{3.10}$$

Where Z_t^i is the i-th node at time slice *(t)* and could be a part of C_t, X_t or O_t, N is the number of nodes in the network, and *Pa(Z_t^i)* are the parents of Z_t^i. The parents may be in the previous or the same time slice only (the model is considered as a first-order Markov process). The variables in the second time slice have parameters associated with them (conditional probability distribution), while the variables in the first time slice do not have parameters associated with them. In this case, the process is stationary and the model can be explained by giving only the first two time slices.

The author in [119], extended the above definition by allowing the DBN to model K^{th}-order Markov processes; this is achieved by defining the DBN as follows:

$$P(Z_t|Z_{t-1}, Z_{t-2}, ..., Z_{t-k}) = \prod_{i=1}^{N} P(Z_t^i|Pa(Z_t^i)) \tag{3.11}$$

In the above definition, the parents can either be in the previous time slice or further in the past, the initial network can be defined implicitly in the unrolled DBN, or may be defined explicitly using the anchor nodes.

3.4.2.1 Inference

There are two methods to achieve exact inference in DBN. The first is based on the idea that the unrolled DBN can be considered the same as a static BN, so any exact BN inference algorithms can be applied. The second method is to convert the DBN into an HMM, in which case the forward-backward algorithm can be applied [119, 118, 88]. In this thesis, the first method has been selected; in terms of unrolling the DBN and considering it the same as a static BN. Then the polytree algorithm has been applied, which is illustrated in section (3.4.1.3.1), for performing the inference process.

3.4.2.2 Creating a Dynamic Bayesian Network

In order to construct a DBN, four steps have to be followed in the following sequence:

1. Defining network variables (nodes) and their states; this includes specifying the hypothesis node (e.g. the node desired to be inferred), the information nodes (e.g. contextual variables that affect the hypothesis node) and the observable nodes (e.g. contextual variables that result from the hypothesis node).

2. Drawing the causal relationships between random variables (drawing the network graph).

3. Specifying the conditional probability distributions for each node.

4. Performing the inference process in order to infer the hypothesis node.

3.5 Dynamic Bayesian Networks software

In this section, a general overview of the software and modelling tools that can be used for DBN modelling is given. The currently available software that supports DBN functionality can be summarised as follows [11, 119]:

- **The Bayes net toolbox (BNT) for Matlab:** This tool was developed during Murphy's time [128, 83]; it is a free and open-source tool that can be used with Matlab only. This tool lacks a graphical user interface and is slow, because Mathlab is slow. The DBN is represented in this tool as a first-order Markov process only, and the conditional probabilities of the nodes can be represented as either discrete or continuous. It supports different kinds of inference algorithms, such as junction tree, variable elimination, polytree and Gibbs sampling. Moreover, it provides both parameter and structural learning algorithms. However, the BNT cannot be considered a general-purpose tool because still more functionality needs to be added to the software (i.e. prediction, support for non-directed acyclic graphical models). Furthermore, it does not support the exporting and importing of other BN file formats from other packages, and the script part, which is needed to implement the network, can be very slow [11, 119].

- **Netica:** Netica was developed by Norsys [129]; it supports a DBN specification and roll-out. A compilation for the designed network has to be carried out before implementing the inference process, in terms of transferring the network into junction tree representation, where the software uses the junction tree inference algorithm. It provides parameter learning only, and has a graphical user interface. A k-order Markov model DBN can be modelled in this software. Moreover, the evidence can either be entered by hand or by importing it from a data file. However, it is only commercially available [11, 119].

- **Analytica:** This software was developed by Lumina Deicion systems [130]; it supports both BN and DBN by allowing users to choose the temporal nodes and specify the time steps, and it uses the sampling method for the inference. It additionally provides both the application program interface (API) and the graphical user interface, as well as both kinds of variables: continuous and discrete. However, it does not use the terminology of a DBN, which can make it difficult for the user to identify the features of its functionality. Analytica does not support structural or parameter learning, and is only commercially available [11, 119].

- **The graphical model toolkit (GMTk):** This tool [131] is written in C++ language and is open-source toolkit. It is specialised in DBN for automatic speech recognition (ASR) systems, and has several features, such as different inference techniques and the ability to deal with both continuous and discrete nodes. Modelling a DBN with this tool gives more expressive power, but much work is needed in order to specify the model due to the fact that it is specialised for ASR systems and lacks more functionality, which can be useful for other types of applications. Moreover, it is very hard for users who do not have sufficient information about the GMTk software development to understand the toolkit because the documentation is not complete [119].

- **BayesiaLab:** This software was developed by Bayesia company in France [132]; it supports the discrete nodes, while the continuous nodes are supported by discretisation. BayesiaLab works in two modes: modelling and validation. The first mode allows users to add temporal arcs in order to create a first-order Markov model DBN only. The inference is done in the validation mode, with the junction tree inference algorithm. Gibbs sampling is also available in this software. It supports both parameter and structural learning; however, it is commercial rather than open-source software, and is not free to use [11, 119].

- **Probabilistic network library (PNL):** PNL is an open-source toolkit written in C++ language, which was developed by Intel's research group in Saint Petersburg. It supports a DBN and diverse graphical models, and is considered to be a C++ implementation of BNT, but it does not support all the functionality provided by the BNT. It can handle the discrete nodes only, and provides both parameter and structural learning. Moreover, it uses the junction tree algorithm for the inference process. However, although the API is well-documented, which makes it a good choice for modelling a DBN, it does not provide a graphical user interface, and still lacks in much functionality compared with BNT [119].

- **GeNIe:** GeNIe was developed by Decision Systems Laboratory, at the University of Pittsburgh [133]. It is the graphical network interface of SMILE, which is a library of functions for graphical probabilistic and decision network models. In other words, GeNIe is an outer shell to SMILE. GeNIe is implemented in C++ language and runs under the Windows operating system. It supports both BN and DBN implementation by providing temporal reasoning, and allows the construction of large and complex networks with a user-friendly interface. It provides several inference algorithms for exact and approximate inference, such as clustering, polytree, logic sampling, likelihood sampling, self-importance etc. Moreover, it supports the discrete nodes and has the ability to import and export other BN file formats. Furthermore, it allows users to specify the nodes of interest as target nodes, which means that during the inference process, only the nodes of interest are guaranteed to be fully updated, which leads to a reduction in computation [11, 119].

In our conclusion, many software packages can be used for modelling and implementing a DBN, each of which has its advantages and disadvantages. However, there is no suitable software for modelling a DBN. Slow implementation, a lack of graphical user interface and complex programming to implement a DBN are disadvantages that are associated with many of the software packages. GeNIe software version 2.0 has been used in this thesis to implement the

DBN, for to the following reasons:

- It supports both BN and DBN implementation by providing temporal reasoning so the first-order Markov DBN can be implemented.
- It provides a polytree inference algorithm, which is the algorithm that has been chosen to perform the inference in this thesis.
- It provides bar charts for the nodes and shows probabilities of each state graphically.
- It is easy to construct the network and define the temporal nodes using graphical click and drop.
- The software is freely available.
- It provides a user-friendly interface.
- It supports a complete integration with Microsoft Excel; data can be copied and pasted into an internal spreadsheet view of GeNIe.

The steps for creating our proposed DBN model using GeNIe software are demonstrated in Appendix B.

3.6 Summary

This chapter presented an overview of new solution to accident prevention in VANET by detecting abnormal behaviours exhibited by the driver. Moreover, it gave an overview of the behaviour of the driver, with a new definition from the perspective of context awareness. Four kinds of driver behaviour were classified: reckless, drunk, fatigued and normal behaviour. The circumstances that are required in order to implement the driver behaviour detection system were also described.

An illustration of the main features and functionalities of BNs, such as implementation, reasoning and the available inference algorithms with a detailed description of polytree algorithm, which will be used for inference in this work has been given. Furthermore, a DBN has been illustrated as a temporal reasoning technique that can handle uncertain information and perform a probabilistic inference to model the system that evolves over time (e.g. driver's behaviour), with an explanation of its mathematical notion, inference and the way it can include evidence from the different time slices. In addition, the main existing software packages, which support the implementation of a DBN, were demonstrated, and their advantages and disadvantages outlined, with a description of the main reasons for choosing GeNIe version 2.0 as the implementation tool in this thesis.

The next chapter (Chapter 4) will present a detailed description of the proposed OBU driver behaviour detection architecture.

Chapter 4

On Board Unit Architecture Based on Context-aware System

Objectives:

- Propose a novel OBU driver behaviour detection architecture in VANET
- Explain the three phases of the architecture that have been designed based on the concept of context-aware system
- Define the components of the proposed OBU architecture
- Describe the mechanism of driver behaviour detection

4.1 Introduction

In this chapter, a novel OBU architecture in VANET based on a context-aware system, with a detailed illustration of its components is presented. The mechanism for detecting the abnormal behaviour of the driver, based on the proposed architecture, is explained. The architecture is built on a new technique for detecting abnormal behaviour exhibited by the driver. This is achieved by capturing information and reasoning about the driver, the vehicle and the environment contextual information, in order to prevent accidents from taking place by warning the driver upon detecting abnormal behaviour.

The architecture comprises three main phases: sensing, reasoning and application. These phases represent the three main subsystems of a context-aware system: sensing, reasoning and acting, respectively. The five-layered conceptual framework [1] represents the fundamental base of our architecture, where the five layers of this model are utilised to construct the components of our architecture.

As mentioned in Chapter two, the behaviour of the driver is considered to comprise an interaction between three entities: the driver, the vehicle and the environment, and is described as a high-level context (uncertain context), which means that a reasoning technique for uncertain information is required to combine the collected information and infer the current behaviour of the driver. Therefore, a DBN algorithm is included in the architecture to perform probabilistic reasoning in terms of combining the collected information and inferring the driver's behaviour. Different kinds of sensors (i.e. cameras, GPS, speed sensor, TMC etc.) are included in the ar-

[1] The figure and the illustration of the five-layered conceptual framework is depicted in Chapter (2)

chitecture in order to collect most of the available contextual information about the three entities. This will result in more accurate and sufficient behaviour detection.

4.2 Driver Behaviour Detection Mechanism

This section presents the mechanism for detecting the behaviour of the driver. The flowchart in Figure 4.1 depicts the process of driver behaviour detection. The vehicle senses the contextual information about the vehicle, the driver and the environment from sensors, including both physical and virtual sensors, such as speed sensor, accelerometer, TMC, adaptive hello message, cameras, GPS, alcohol sensor etc, which are connected to the OBU. After collecting this information from the sensors, the interpreter translates the different kinds of data into a form that can be processed by the processor of the OBU. Transforming the sensory data into a machine-processable form can be carried out by applying one of the modelling techniques, such as ontology modelling [7] (this will be out of the scope of this thesis).

After transforming the captured context into a machine-processable form, the OBU processor performs a behaviour detection algorithm (DBN algorithm) in order to reason about the uncertain context (driver behaviour) by combining the data received from the interpreter using the probabilistic inference to deduce the current behaviour of the driver. If the output of the inference is normal driving behaviour that satisfies all normal driving criteria, no action will be taken by the processor and the vehicle will sense new information. If the output of the inference is abnormal driving behaviour, such as being drunk, fatigued or reckless, the processor will choose the appropriate in-vehicle alarms and sends a signal to control unit 1 informing it to operate in-vehicle alarms, and will performing the algorithm of calculating the corrective action for other vehicles on the road according to their position, velocity and direction. After calculating the corrective action for other vehicles on the road, the processor will send a signal to control unit 2 to send warning messages (the corrective action algorithm will be out of the scope of this thesis). This process is based on a context-aware system and is a self-organising process in which sensing, reasoning and acting upon contextual information occurs instantly.

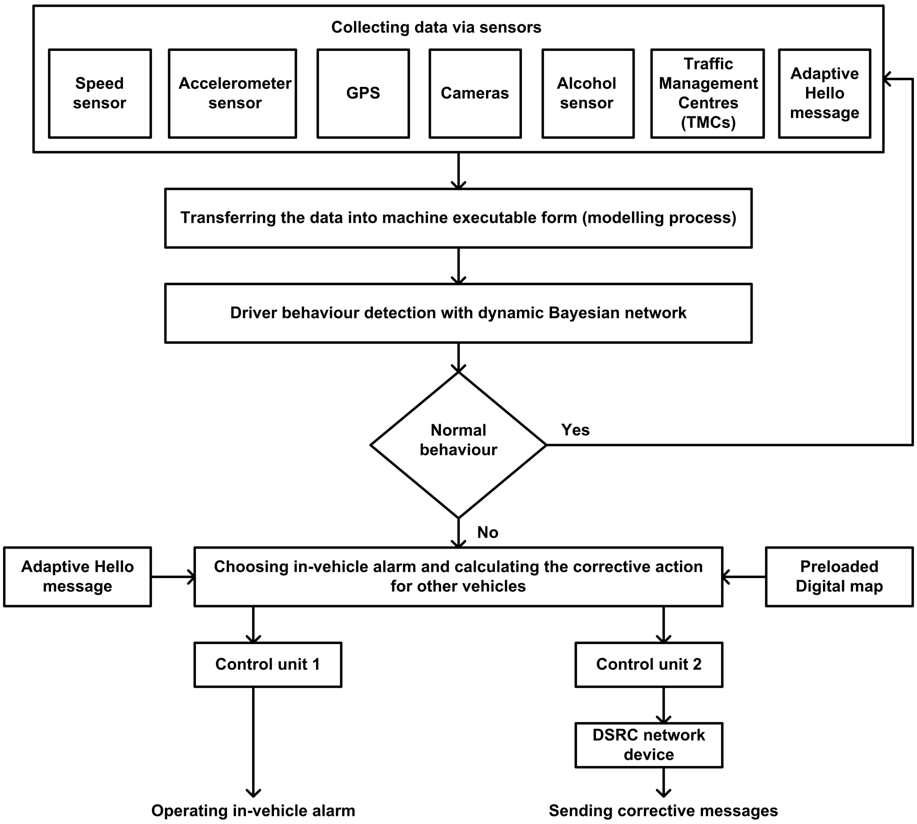

Figure 4.1: Driver behaviour detection system mechanism

4.3 OBU Architecture

This section describes all the components of the proposed OBU driver behaviour detection architecture in VANET, as shown in Figure 4.2. This figure illustrates in detail each component in the architecture, and reveals the function of each unit; particularly the way in which these components cooperate with each other to achieve the task of detecting the behaviour of the driver and warning the driver using the in-vehicle alarms. The green blocks in the architecture represent the new units that have been added to the typical components of the OBU.

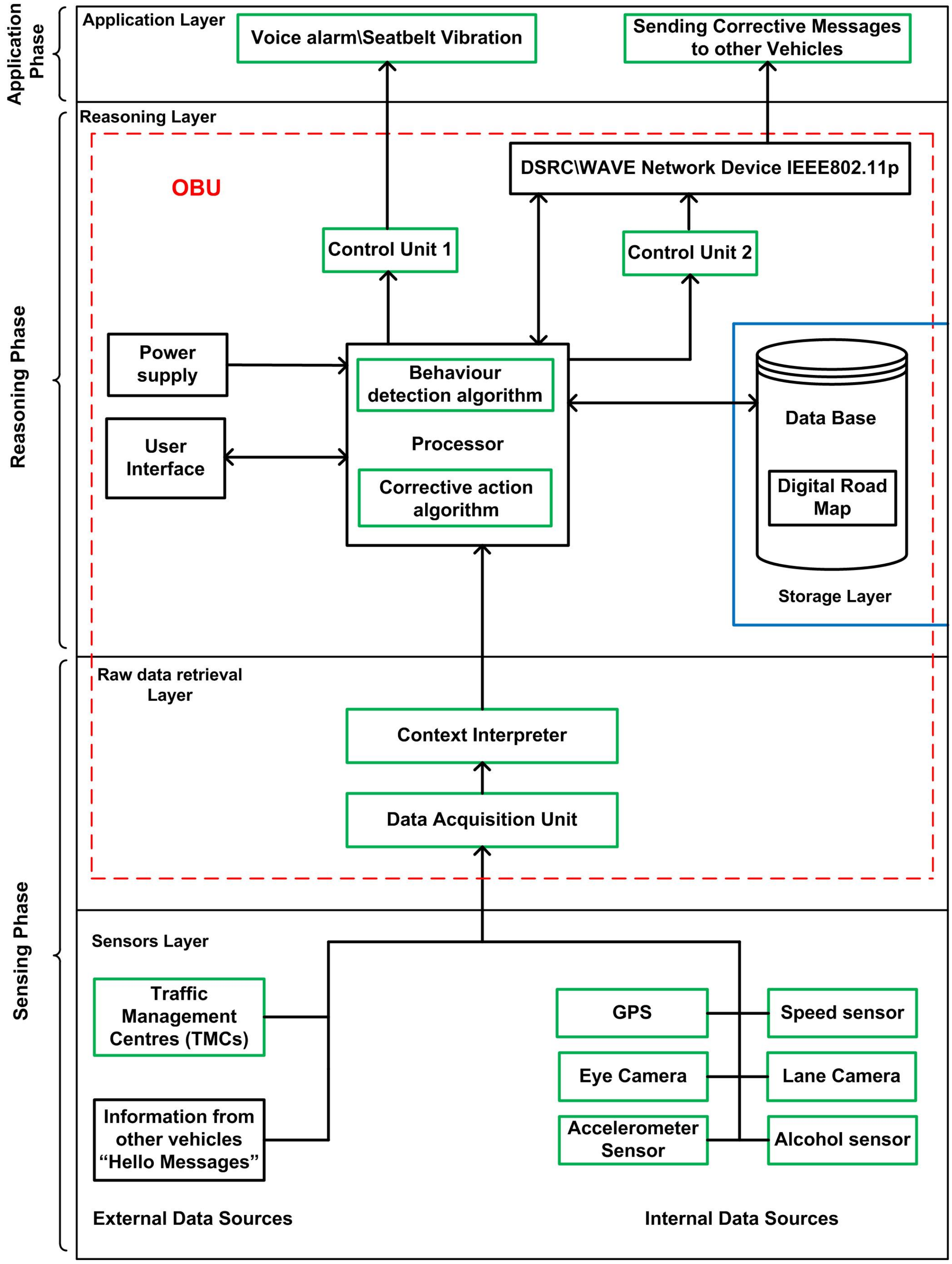

Figure 4.2: Driver behaviour detection system architecture

As depicted in Figure 4.2, the architecture utilises the five layers of the conceptual framework, and is divided into three main phases: the sensing phase, the reasoning phase and the application phase, which represent the three main subsystems of a context-aware system: the sensing, reasoning and acting subsystems, respectively. The in-vehicle alarms are operated in the third phase, which depends upon the results of the second phase, which in turn depends on receiving the information from the first phase.

The proposed architecture performs the behaviour detection process through the processes carried out by the following components:

- Sensors
- Data acquisition unit
- Context interpreter
- Processor, which performs the behaviour detection algorithm based on the DBN
- Database
- Control unit 1 and control unit 2

The *sensors* represent the sensors layer in the five-layered conceptual framework. The *data acquisition unit* and the *context interpreter* make up the raw data retrieval layer. The *processor* that performs the DBN algorithm, *control unit 1* and *control unit 2* represent the preprocessing layer, and the *database* corresponds to the storage and management layer. Finally, the *in-vehicle alarms* and the *warning messages* characterise the application layer.

4.3.1 Sensing phase

This section presents the way in which the context can be sensed, and explains the types of sensors used in the architecture. The sensing phase represents the sensing subsystem in the context-aware system, and is responsible for gathering the contextual information about the driver, the vehicle and the environment, then transforming the collected information into a machine-executable form to be processed in the next phase (reasoning phase). It is divided into two layers, as follows:

- **Sensors layer:** This layer is responsible for acquiring the context data. It consists of a set of sensors integrated into the vehicle and connected to the vehicle's OBU in which the system is operating. Different types of sensors provide different types of information according to the system requirement. As described earlier in Chapter 2, there are three types of sensors: physical, virtual and logical. Two types of data sources (sensors) have been used in this architecture in order to gather context. The internal data sources (physical sensors) refer to the set of sensors equipped within the vehicle, such as cameras, speed sensor, GPS, alcohol and accelerometer sensor, which offer information about the state of the driver's eyes, the vehicle's speed, acceleration information, lane position and level of alcohol in the driver's blood.

  External data sources (virtual sensors) are also incorporated in order to gather external information, including the traffic management centres (TMCs), which provide information relating to traffic, weather and road conditions, based on the website, dynamic message signs and highway auditory radio data [1]. External data sources also include information about other vehicles (velocity, current position and direction) collected through received hello messages [134].

As shown in Figure 4.3, this layer transmits the captured context to the raw data retrieval layer. The functionality of a context-aware system depends on the functionality of each sensor, because a failure in any of the sensors will result in the provision of inaccurate information to the system, hence leading to faulty system performance.

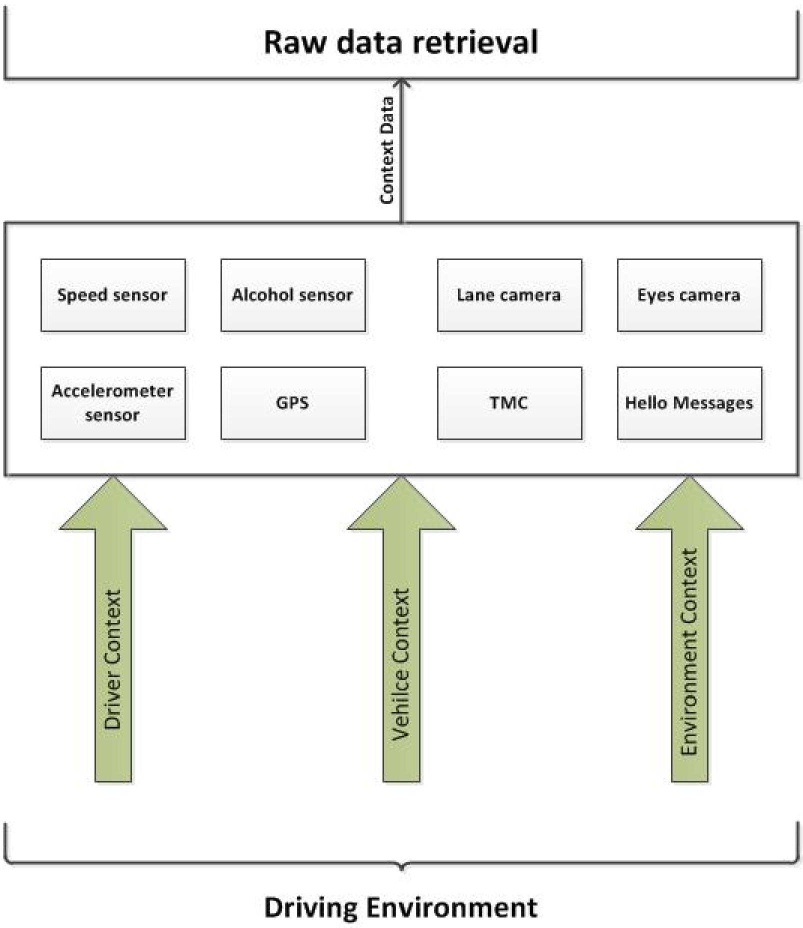

Figure 4.3: Sensor layer

The following are descriptions of each of the sensors that has been used in this system:

- **Lane camera sensor:** Camera can be used to capture and provide information about the lane position, lane deviation and departures [135].

  - **Global Positioning System (GPS):** GPS provides information about the speed limit of the road, current time and current speed, and helps to provide the current position and direction of the vehicle [31, 27].
  - **Accelerometer sensor:** The accelerometer sensor measures proper acceleration and the rate of change of the vehicle's velocity, so that it can used to provide information about whether the acceleration is normal or abnormal [136].
  - **Eye camera sensor:** The vision system is used to capture information about the driver's eyes and decide whether they are open or closed. PERCLOS is considered to be the perfect eyes measuring tool [76].
  - **Alcohol sensor:** This sensor is used to detect the level of alcohol in the driver's blood by measuring the alcohol content in the driver's breath [95].
  - **Speed sensor:** This sensor provides information about the current speed of the vehicle [27, 136]
  - **Adaptive hello messages:** Information about the vehicle position, direction and velocity can be obtained from the periodic messages that are disseminated in VANET [31].
  - **Traffic Management Centre (TMC):** Different information can be obtained from the TMC, such as information relating to traffic, weather and road conditions, based on the website, dynamic message signs and highway auditory radio data [1].

- **Raw data retrieval layer:** The purpose of applying this layer is to separate low-level sensing details from the sensors for the upper layer of the system as well as abstract contextual information that has been received from the sensors layer. This layer contains two components, as follows:

– **Data acquisition unit:** This is responsible for controlling and coordinating all sensors in the sensors layer.

– **Context interpreter:** The modelling process is carried out in this unit, in terms of transforming the data that has been received from the data acquisition unit into a machine-executable form. Several types of modelling algorithms can be applied to abstract the received sensory data (for example ontology modelling). The received data may come from different types of sensors such as camera, GPS, speed sensor etc. This component transfers the data to a form which can be processed by the reasoner (the modelling process by applying one of the available modelling technique is out of the scope of this thesis).

4.3.2 Reasoning phase

This phase characterises the reasoning subsystem in the context-aware system; it is responsible for inferring the behaviour of the driver and calculating the corrective actions for other vehicles on the road. As mentioned in Chapter 2, there are two types of contextual information: certain information, which is obtained from a single sensor, and uncertain contextual information, which can not be acquired by a single sensor, and may be incomplete or inexact. The behaviour of the driver is considered to be uncertain contextual information (high-level contextual information). Therefore, a behaviour detection algorithm based on a DBN is applied in order to perform reasoning about uncertainty by combining the information received from the previous phase (sensing phase), in order to detect the behaviour of the driver during real-time driving. The corrective action algorithm is responsible for calculating the appropriate corrective action for other vehicles on the road. The reasoning phase consists of two layers, as follows:

- **Reasoning layer:** This layer is responsible for extracting the current behaviour of the driver (e.g. fatigued, drunk, normal or reckless) and choosing the appropriate in-vehicle alarm to warn the driver and avoid road accidents. This layer comprises the following components:

  – **Processor:** The OBU processor is responsible for managing all the components of the OBU and controlling all the tasks and activities it performs. The processor performs two algorithms, as follows:

    1. **Behaviour detection algorithm:** This algorithm is designed to reason about uncertain contextual information in order to detect the current behaviour of the driver, using a DBN to combine data collected from the sensing phase, which includes information about the driver, the vehicle and the environment, in order to detect the type of behaviour being exhibited. If the behaviour of the driver is normal, no action is needed. In the case of abnormal driving behaviour (e.g. drunk, fatigued or reckless), the processor sends signals to control unit 1 to operate the appropriate in-vehicle alarm, and performs the corrective action algorithm. This thesis will focus only on the driver behaviour detection algorithm, which will be explained in detail in Chapter 5.
    2. **Corrective action algorithm:** The aim of performing this algorithm is to calculate the proactive corrective action for other vehicles on the road according to their position, velocity and direction, with the use of the preloaded digital road maps and the information collected from the adaptive hello messages. After calculating the corrective action, the processor sends a signal to control unit 2 in order to send the warning message through the wireless technology provided

by VANET (the corrective action algorithm will be out of the scope of this thesis).

- **Control unit 1:** This unit is responsible for controlling in-vehicle alarms, such as seat vibration and audio alarm, to attract the driver's attention. This unit receives the signal from the processor in the case of abnormal driving behaviour occurring.
- **Control unit 2:** After receiving the signal from the processor indicating abnormal driving behaviour, this unit sends signals to the DSRC/WAVE device in order to transmit the corrective messages to other vehicles on the road, or to the roadside unit.
- **DSRC/WAVE network device:** The OBU contains a DSRC/WAVE network device based on IEEE 802.11p [31]. It is responsible for connecting the vehicle to other vehicles' OBUs or with the roadside unit through the wireless radio frequency, based on IEEE 802.11p. The OBU can send or receive messages via this network device.
- **Power supply:** The power supply is responsible for providing power to the OBU. It is rechargeable and provides power to the OBU without any constraints.
- **User interface:** This contains the audio and video interface that allows the user to interact with the services provided by the OBU. It also provides the driver with audio alarms in order to allow him or her to concentrate on the road.

• **Storage layer:** In this layer, the data base stores digital maps of the road and historical data (past driving situations).

4.3.3 Application Phase

This phase represents the acting subsystem in the context-aware system, and is responsible for operating in-vehicle alarms to warn the driver. It is also responsible for disseminating the warning messages, including corrective actions, to other vehicles on the road in order to prevent the occurrence of accidents and to decrease the number of potential fatalities.

4.4 Summary

This chapter introduced a novel OBU driver behaviour detection architecture in VANET. The architecture has been designed based on the concept of a context-aware system, which comprises three main phases: sensing, reasoning and application phase, and utilises the five-layered conceptual framework layers. A DBN model has been integrated into the architecture in order to reason about uncertain contextual information (driver's behaviour). The motivation behind this architecture was to detect abnormal behaviour exhibited by drivers by combining information about the driver, the vehicle and the environment, and to warn the driver in order to prevent accidents from taking place.

The next chapter (Chapter 5) will present a detailed illustration of a DBN model for combining different context and inferring the behaviour of the driver by performing a probabilistic inference.

Chapter 5

A Dynamic Bayesian Network Model for Driver's Behaviour Detection

Objectives:

- Determine the objectives behind the development of our proposed DBN model
- Propose our novel DBN model for the detection of driver's behaviour
- Define the steps for the creation of the proposed DBN model

5.1 Introduction

In this chapter, a novel probabilistic framework for detecting different kinds of driver behaviour in VANET using a DBN to combine information about the driving environment and to infer the behaviour of the driver is introduced. The system is comprehensive and is able to detect four kinds of driver behaviour, which are: drunken, fatigued, reckless and normal behaviour. The behaviour of the driver is a dynamic process, which evolves over the course of driving. For example, the alcohol level in the driver's blood may be low at the beginning of the driving, but will become higher when he or she is drinking during the driving process. The driver's level of fatigue may also increase during driving [106]. This fact indicates that, in addition to the observable contextual information at the current time instant, the driver's state at the previous time instant is also considered an indicator for his or her state at the current time.

However, designing an accurate and efficient driver behaviour detection system can present several challenges. For example, the temporal aspects which are exhibited by the driver's behaviour have to be captured, the captured contextual information may be incomplete or inexact due to the inaccurate reading of some sensors; and the fact that different information needs to be combined in order to obtain the high-level context (driver's behaviour) [22, 88]. Several information fusion techniques have been proposed such as Fuzzy logic, the Dampester-Shapher theory, Neural Networks and the Kalman filter. These methods do not provide efficient expressive capabilities for capturing incomplete data, uncertainties, dependencies between the variables and the temporal aspect exhibited by the behaviour.

To tackle these dilemmas, the model that has been designed in this chapter uses a DBN technique to combine data from different types of sensors to infer the driver's behaviour due to the following reasons. Firstly, it is considered to be the most reliable method for dealing with inaccurate data and unobservable physical values. Secondly, it is able to model time series data (systems, which evolve over time). Thirdly, it is efficient at combining uncertain contextual information from a wide range of sensors to deduce high level contextual information (reason about uncertain context); and is able to combine prior data with current data [76, 84, 86, 24, 87].

As mentioned previously in Chapter 3, a DBN can be thought of as a set of static Bayesian networks interconnected by sequential time slices. In this thesis, the DBN has considered representing a first-order Markov process, which means that the hypothesis node at time slice *(t)* depends on the variables at time slice *(t)* and the hypothesis node at time slice *(t-1)* only. The information used to infer the behaviour includes information about the driver (i.e. the state of the driver's eyes), the vehicle (i.e. position in the lane) and the environment (i.e. the temperature). In conclusion, merging contextual information from different kinds of sensors and capturing the temporal aspect of the behaviour via performing a probabilistic reasoning under uncertainty using a DBN will lead to more efficient and accurate detection of four kinds of driving behaviour.

5.2 Problem definition

The main objective for designing our DBN model is to infer the unobserved high-level context (driver's behaviour) from the observed context (sensory data). The model is able to detect four styles of behaviour, which are: drunken, fatigued, reckless and normal behaviour. The behaviour of the driver in affected by many factors (information variables); whilst each behaviour reflects several observations (observable variables). It is impossible to include all factors and observations in our model and therefore, we have only chosen the most important variables, which can lead to a more accurate detection of the hypothesis node than others, as will be shown in Section (5.3.1).

As mentioned earlier in Chapter 3, the behaviour of the driver is described as a transition between sequenced states, over the course of driving a driver will be in a particular state, which he or she may remain in for a period of time and then potentially changing to a different state. Each state can be inferred by capturing and combining a large amount of context (information and observable variables) as explained in Eq. (3.2). Therefore, the behaviour of the driver is considered as the current unobservable state.

In our problem domain, assuming that *[Z_1, Z_2, ..., Z_t,..]* is a semi-endless collection of random variables, Z_t = *(C_t,X_t,O_t)* and represents the input, hidden and output variables of the driver's behaviour detection model at a certain time *t*. C_t and O_t components of Z_t denote the information and the observable variables respectively. Whilst, the X_t component of Z_t corresponds to the state S_t, we used a DBN to model the probability distribution over the semi-endless collection of random variables and to infer the current state of the driver. In this thesis, we consider the unrolled DBN as a static Bayesian network, and the hypothesis node at time slice *(t)* depends on its immediate past on time *(t-1)* as well as on the random variables at time *(t)* only as follows:

$$P(Z_t|Z_{t-1}) = \prod_{i=1}^{N} P(Z_t^i|Pa(Z_t^i))$$

Where Z_t^i is the i-th node at time slice t, N is the number of nodes in the network and $Pa(Z_t^i)$ are the parents of Z_t^i.

5.3 DBN-based driver's behaviour detection model

This section presents the steps used for the creation of our DBN driver behaviour detection model, which is able to capture the temporal aspects of the behaviour and integrate the evidence over time. As mentioned earlier in Chapter 3, there are four steps, which have to be undertaken in order to design a DBN, starting from the choice of the network's nodes to inferring the state of the hypothesis node. In the following paragraphs, a detailed explanation for these steps will be provided.

5.3.1 Defining the network variables (nodes)

The first step in creating a DBN is specifying the nodes of the network and determining their states. A DBN can deal with both discrete and continuous nodes, we only used discrete nodes in this model; this means that each node has a finite set of values.

The hypothesis node in our network is the state node which represents the current state of the driver; whilst, the contextual variables have been divided into two groups. The first group (the information nodes) represents the variables, which may affect the state node; and the second group (the observable nodes) corresponds to the information that results from the state node as follows:

- **Group 1:** This group includes the contextual variables, which affect the state node such as circadian rhythm, and the driving environment, which represent the environment related information.
  - **Circadian rhythm:** This refers to the human sleep-awake cycle, which

is considered a cause of driver fatigue. There are two periods during the day (3-5 PM and 3-5 AM) during which humans reach their peak level of fatigue [24, 76, 106]. The circadian rhythm was considered to influence the hypothesis nodes. The circadian rhythm node is affected by two nodes: time and the time zone nodes. Thus, the circadian rhythm, time and the time zone nodes were taken as information nodes in our DBN.

  - **Driving environment:** Noise and temperature are considered to have a significant influence on the driving environment, which can in turn cause fatigue. Fatigue is more likely to occur when noise and high temperatures occur inside or outside of the vehicle [24, 76]. As a result, the driving environment, noise and temperature nodes were selected as information nodes in our network.

- **Group 2:** This group denotes the contextual variables that result from the state node. It includes vehicle related information such as (the vehicle speed, position between the lane markers and acceleration) and driver-related information (the state of the driver's eyes and the level of alcohol in the blood).

  - **Controlling the speed:** Drunk and fatigued drivers struggle to control their speed due to their mental state. According to the definition of the reckless driver used in this study, the driver may violate the speed limit [27]. In consequence, this node was taken as an observable node in the network.
  - **Position in the lane:** According to [27], the drunk driver has a problem in maintaining the lane position (vehicle position between the lane markers). Hence, this node has been selected as an observable node in our DBN.

- **Acceleration:** The driver is considered to exhibit normal behaviour whilst driving with normal acceleration and is considered to exhibit abnormal behaviour, such as being drunk, fatigued or reckless while driving with sudden acceleration [27]. Therefore, the acceleration node was opted as one of the observable nodes in the network.

- **Driver's eyes movements:** Eyes movements are considered as a visual behaviour that reflects the person's level of fatigue, and eyelid movements can characterise eyes movements. The percentage of eyelid closure over the pupil over time (PERCLOS) is considered as the most reliable method for measuring a person's level of fatigue. If the eye is 80% covered by the eyelid for a period of time the person is considered fatigued. The average eye closure and opening speed (AECS) is one more measure for eyelid movements, where the fatigued person might open/close his or her eyes slowly due to tired eye muscles [137, 138, 93, 76, 24]. Thus, the eye movement, eyelid movements, PERCLOS and AECS were chosen as a set of observable nodes in our DBN.

- **Intoxication:** This refers to the amount of alcohol in the driver's blood. The permitted level of alcohol in the driver blood is 0.05%. Thus, the driver is considered to be drunk if there is alcohol intoxication of more than 0.05% in the blood [127, 95]. This node is also taken as an observable node in the network.

Having determined the hypothesis and the information and the observable nodes, their discrete states must be specified before the value (probability) for each state is chosen. Tables 5.1, 5.2 and 5.3 illustrate the possible discrete states for all nodes in our DBN.

Node name	State 1	State 2
Lane_maintenance	Good	Bad
Intoxication	Less_than_limit	More_than_limit
Acceleration	Sudden	Moderate
Controlling_speed	Good	Bad
Eyes_movements	Normal	Abnormal
Eyelid_movements	Normal	Abnormal
PERCLOS	Normal	Abnormal
AECS	Slow	Normal

Table 5.1: Observable nodes and their states

Node name	State 1	State 2
Time	Fatigue	Active
Time_zone	Change	No change
Circadian	Fatigue	Awake
Noise	High	Normal
Temperature	High	Normal
Environment	Good	Bad

Table 5.2: Information nodes and their states

Hypothesis node	State 1	State 2	State 3	State 4
State	Fatigue	Normal	Reckless	Drunk

Table 5.3: The states of the hypothesis node

5.3.2 Drawing the Network graph

The second step in designing a DBN is specifying the causal relationship between the random variables. This is carried out by drawing the network arcs. After specifying the network variables and the causal relationships between them, the network graph can be drawn in terms of the directed acyclic graph. Figure 5.1, depicts the proposed DBN model for detecting the behaviour of the driver. The hypothesis node in this network is the state node. The set of variables above the hypothesis node denotes the information nodes; whilst the variables beneath the hypothesis node represent the observable nodes. As shown in Figure 5.1, the network unrolled for two time slices, the state node at time t depends on the variables at time t and on the state node at time *t-1*. The network can be unrolled for T time slices, where the same structure will be replicated at each time slice.

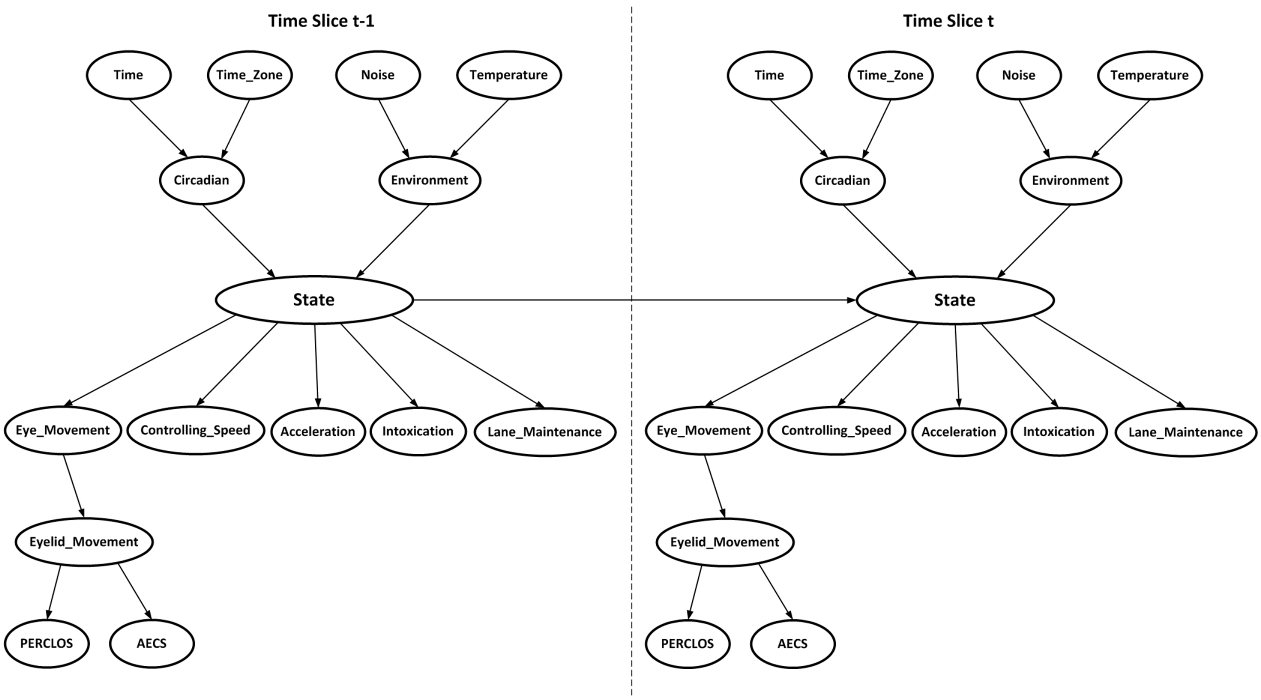

Figure 5.1: A DBN model for detecting the behaviour of the driver

5.3.3 Parameterising the network

The third step in designing a DBN is parameterising the network, which means choosing the values for the conditional probability tables for all nodes in the network. It is the stage that determines the prior probability of the root nodes and the conditional probabilities of the links in the network. Defining the probability of each node in the network refers to the probability of the node being in one of its states when the evidence is received. As mentioned in Chapter 3, specifying the values of the CPTs can be done either by performing statistical analysis of a huge amount of training data, or by critically analysing a set of previously published papers and researches, which are related or similar to the system.

There were difficulties involved in acquiring a large amount of training data for this study, as no testbed is equipped with all the sensors required for the model. Network parameters and transition distributions between time slices in the model were chosen manually. This was done according to critical analysis of a large number of reports published by the UK department for transport (DFT) and other transportation organisations, including the national highway traffic safety administration (NHTSA), as well as published papers covering features that relate to the system.

Several studies and models regarding driver behaviour detection have been proposed in the literature, no one of which have provided all the data required to parameterise the system, due to the fact that, most have focused on detecting a single style of driving behaviour, such as drunk, reckless or fatigued, using different kinds of information regarding the driver, the vehicle and the environment. The hypothesis node in this model includes four states, which are: normal, drunk, reckless and fatigued, and the contextual nodes cover information about the driver, the vehicle and the environment. A period of more than three months was spent thoroughly studying and analysing the following reports and published papers [112, 76, 27, 127, 22, 24, 97, 139, 106, 140], in order to determine the relationship between the variables and to set the parameters for each node (variable) in the network. This facilitated attainment of all the probabilities required to parameterise the DBN model.

Tables 5.4 - 5.19 illustrate the prior probabilities and the conditional probabilities for all the nodes in the network.

Time	Probability
Fatigue	0.26
Active	0.74

Table 5.4: Prior probability for Time node

Time_zone	Probability
Change	0.17
No Change	0.83

Table 5.5: Prior probability for Time_zone node

Noise	Probability
High	0.15
Normal	0.85

Table 5.6: Prior probability for Noise node

Temperature	Probability
High	0.15
Normal	0.85

Table 5.7: Prior probability for Temperature node

Time / States	Time_zone	Circadian	Probability
Fatigue	Change	Awake	0.1
	No Change	Awake	0.4
	Change	Fatigue	0.9
	No Change	Fatigue	0.6
Active	Change	Awake	0.3
	No Change	Awake	0.95
	Change	Fatigue	0.7
	No Change	Fatigue	0.05

Table 5.8: Conditional probabilities for Circadian node given its parents

Noise / States	Temperature	Environment	Probability
High	High	Good	0.06
	Normal	Good	0.2
	High	Bad	0.94
	Normal	Bad	0.8
Normal	High	Good	0.27
	Normal	Good	0.85
	High	Bad	0.73
	Normal	Bad	0.15

Table 5.9: Conditional probabilities for Environment node given its parents

Circadian	Environment	State	Probability
Awake	Good	Fatigue	0.05
		Normal	0.31
		Reckless	0.31
		Drunk	0.33
Awake	Bad	Fatigue	0.27
		Normal	0.25
		Reckless	0.25
		Drunk	0.23
Fatigue	Good	Fatigue	0.27
		Normal	0.25
		Reckless	0.25
		Drunk	0.23
Fatigue	Bad	Fatigue	0.51
		Normal	0.16
		Reckless	0.16
		Drunk	0.17

Table 5.10: Conditional probabilities for State node at time (t-1) given its parents

Lane_maintenance / States	Fatigue	Normal	Reckless	Drunk
Good	0.49	0.975	0.41	0.49
Bad	0.51	0.025	0.59	0.51

Table 5.11: Conditional probabilities for Lane_maintenance node given its parent

Intoxication / States	Fatigue	Normal	Reckless	Drunk
Less_than_limit	0.99	1	0.975	0.1
More_than_limit	0.01	0	0.025	0.9

Table 5.12: Conditional probabilities for Intoxication node given its parent

Acceleration / States	Fatigue	Normal	Reckless	Drunk
Moderate	0.3	0.925	0.374	0.3
Sudden	0.7	0.075	0.626	0.7

Table 5.13: Conditional probabilities for Acceleration node given its parent

Controlling_speed / States	Fatigue	Normal	Reckless	Drunk
Good	0.46	0.975	0.48	0.46
Bad	0.54	0.025	0.52	0.54

Table 5.14: Conditional probabilities for Controlling_speed node given its parent

Eyes_movements / States	Fatigue	Normal	Reckless	Drunk
Normal	0.05	0.99	0.99	0.05
Abnormal	0.95	0.01	0.01	0.95

Table 5.15: Conditional probabilities for Eyes_movements node given its parent

Circadian	Environment	State (t-1)	Fatigue	Normal	Reckless	Drunk
Awake	Good	Fatigue	0.7	0.1	0.1	0.1
		Normal	0.1	0.7	0.1	0.1
		Reckless	0.1	0.1	0.7	0.1
		Drunk	0.1	0.1	0.1	0.7
Awake	Bad	Fatigue	0.8	0.1	0.05	0.05
		Normal	0.4	0.4	0.1	0.1
		Reckless	0.3	0.1	0.5	0.1
		Drunk	0.2	0.1	0.1	0.6
Fatigue	Good	Fatigue	0.7	0.1	0.1	0.1
		Normal	0.5	0.4	0.05	0.05
		Reckless	0.4	0.1	0.4	0.1
		Drunk	0.4	0.05	0.05	0.5
Fatigue	Bad	Fatigue	0.8	0.1	0.05	0.05
		Normal	0.6	0.3	0.05	0.05
		Reckless	0.4	0.1	0.4	0.1
		Drunk	0.3	0.05	0.05	0.6

Table 5.16: Conditional probabilities for State node at time (t) given its parents

Eyelid_movements / Eyes_movements	Normal	Abnormal
Normal	0.95	0.01
Abnormal	0.05	0.99

Table 5.17: Conditional probabilities for Eyelid_movements node given its parent

AECS / Eye-lid_movements	Normal	Abnormal
Slow	0.05	0.97
Normal	0.95	0.03

Table 5.18: Conditional probabilities for AECS node given its parent

PERCLOS / Eye-lid_movements	Normal	Abnormal
Normal	0.95	0.02
Abnormal	0.05	0.98

Table 5.19: Conditional probabilities for PERCLOS node given its parent

5.3.4 Inferring the hypothesis node

Inferring the state of the hypothesis node is the final step in designing a DBN. The behaviour of the driver is an evolving process which dynamically evolves over the course of the time spent driving. Thus, the hypothesis node for a previous time slice is considered as an information node, which can affect the hypothesis node at the current time slice in addition to the information and observable nodes at the current time slice.

As mentioned in chapter three, the inference in DBN can be carried out by either converting the DBN to a HMM and then performing a forward backward algorithm or unrolling the network and applying any exact static BN algorithm. In this thesis, the inference process is carried out by unroll the DBN and perform an exact static BN algorithm. Since the hypothesis node in our network depends on the evidence received from both information nodes and observable nodes, a combination of diagnostic and predictive reasoning is required to infer the hypothesis node (state node). Therefore, we chose the polytree algorithm to perform the inference process, because it includes evidence from both the parents and children of the hypothesis node [126].

Figure 5.2 depicts a fragment of a DBN unrolled for T time slices. As illustrated in the figure, the hypothesis node at the previous time slice (x^l_{t-1} where l = 1, 2, 3, 4), with its different states, is considered to be an information node used to infer the hypothesis node at time t.

Let X_t be the hypothesis node (state) at time t. C^j_t (j= 1, 2, 3, 4, 5 or 6) represent the information variables at time t. While, O^k_t (k= 1, 2, 3, 4, 5, 6, 7 or 8) signifies the observable variables at time t. Let x^l_t, $c^{j,m}_t$ and $o^{k,n}_t$ (l = 1, 2, 3, 4 and m, n = 1,2) represent the values taken by X_t, C^j_t and O^k_t respectively.

Let $e_t = \{e^-_t, e^+_t\}$ be the evidence at time t, which consists of evidence from both observable and information nodes, where e^-_t represents the evidence received from node X's children and e^+_t denotes the evidence received from node X's parents. e^-_t = $\{e^{i,j}_{o,t}\}$ denotes the evidence of the ith observable node with jth states at time t. While, $e^+_t = \{e^{i,j}_{c,t}\}$ represents the evidence of ith information nodes with jth states at time t. Calculating the conditional probability of the hypothesis node (X) at time t, upon receiving the evidence e_t using a polytree algorithm can be carried out using Eq. 3.7 as follows:

$$P(X = x^l_t|e_t) = \alpha.\lambda(X).\pi(X) = \frac{\lambda(X).\pi(X)}{\sum\limits_{x^l}\lambda(X).\pi(X)} \tag{5.1}$$

$$l = 1, 2, 3, 4$$

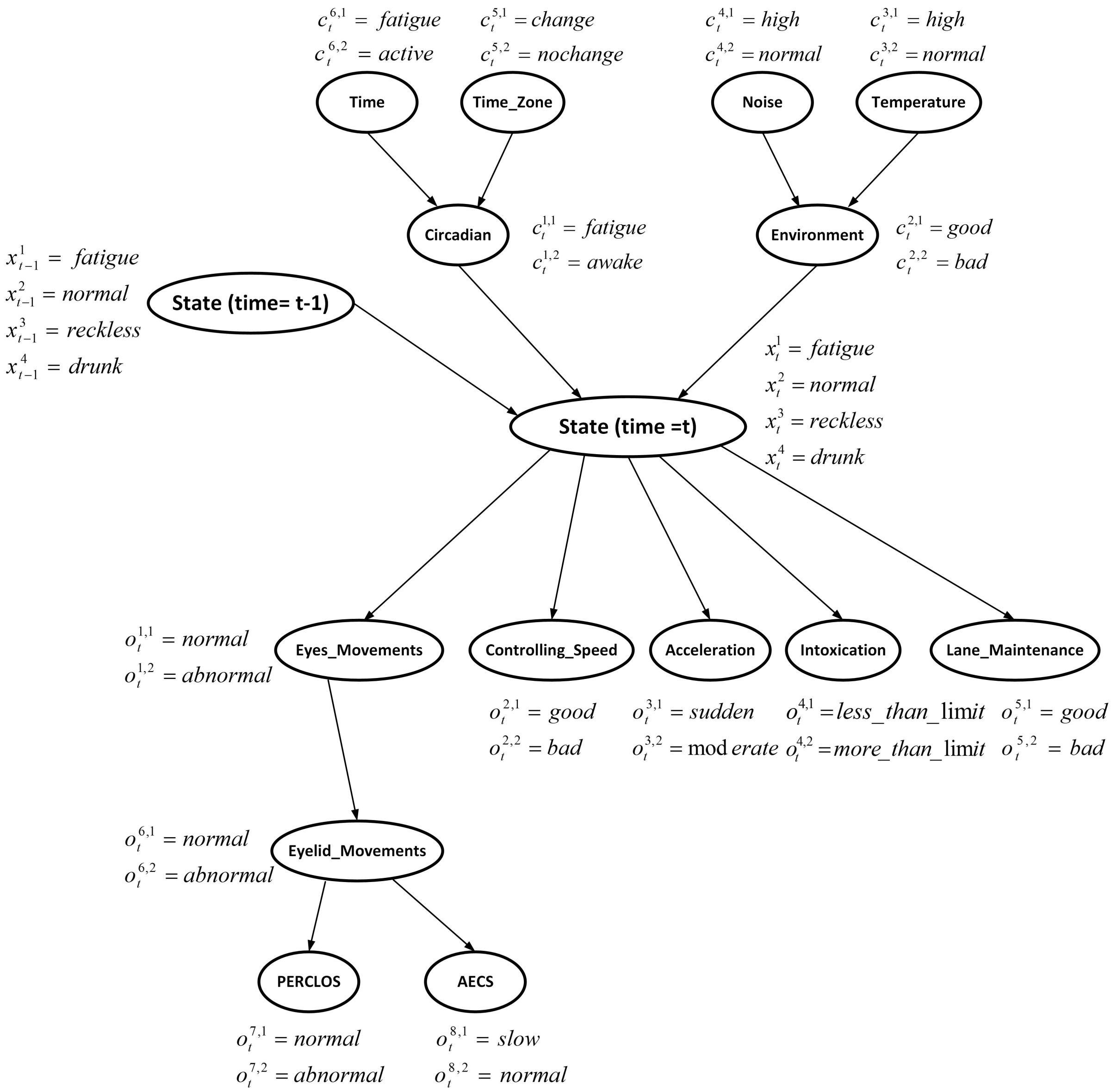

Figure 5.2: A fragment of the unrolled DBN

Where:

$\lambda(X) = P(e_t^-|X)$ referring to the conditional probability of the evidence received from the observable nodes given the occurrence of the hypothesis node at time t.

$\pi(X) = P(X|e_t^+)$ referring to the conditional probability of the hypothesis node given the occurrence of the evidence received from the information nodes at time t.

Therefore, Eq. (5.1) can be written as follows:

$$P(X = x_t^l | e_t) = \frac{P(e_t^- | X = x_t^l).P(X = x_t^l | e_t^+)}{\sum_{j=1}^{4} P(e_t^- | X = x_t^j).P(X = x_t^j | e_t^+)}$$
$$l = 1, 2, 3, 4 \tag{5.2}$$

l is the number of states of the hypothesis node

With reference to Eq. (3.8), P($e_t^-|X = x_t^l$) can be calculated as follows:

$$P(e_t^- | X = x_t^l) = \prod_{i=1}^{5} \lambda_{O_t^i}(X)$$
$$= \left(P(e_{o,t}^{1,j} | X = x_t^l)\right) \times \left(P(e_{o,t}^{2,j} | X = x_t^l)\right) \times \left(P(e_{o,t}^{3,j} | X = x_t^l)\right) \times$$
$$\left(P(e_{o,t}^{4,j} | X = x_t^l)\right) \times \left(P(e_{o,t}^{5,j} | X = x_t^l)\right) \tag{5.3}$$
$$l = 1, 2, 3, 4 j = 1, 2$$

l is the number of states of the hypothesis node

Using Eq. (3.9), $P(X|e_t^+)$ can be calculated as follows:

$$P(X = x_t^l | e_t^+) = \sum_{c_t^1, c_t^2, x_{t-1}} P(X = x_t^l | c_t^1, c_t^2, x_{t-1}) \prod_{j=1}^{3} \pi_X(c_t^j)$$
$$= \sum_{i=1}^{2} \sum_{m=1}^{2} \sum_{n=1}^{4} P\left(X = x_t^l | c_t^{1,i}, c_t^{2,m}, x_{t-1}^n\right).P\left(c_t^{1,i} | e_{c,t}^{1,i}\right).P\left(c_t^{2,m} | e_{c,t}^{2,m}\right).P\left(x_{t-1}^n\right) \tag{5.4}$$
$$l = 1, 2, 3, 4$$

l is the number of states of the hypothesis node

Eqs. (5.2), (5.3) and (5.4) explain the method of computing the conditional probability of the state node at time *t*. This was done by taking into account, during the inference process the observation and the information nodes at time *t*, in addition to considering the state node at time *t-1* as an information node.

5.4 Summary

This chapter presented a novel DBN model for detecting four styles of driver behaviour, which are drunkenness, fatigue, reckless and normal behaviour as a step towards improving road safety. It explained how the proposed model is able to reason about uncertain contextual information by combining many variables related to the driver, the vehicle and the environment (i.e. intoxication, position in the lane and the circadian rhythm ... etc.) and performs a probabilistic inference for deducing the behaviour.

The unrolled DBN is considered as a static BNs, where the hypothesis node at the previous time slice is considered to be one of the information variables when inferring the hypothesis node at the current time slice. After the network has been parameterised, the polytree inference algorithm was applied to infer the hypothesis node, which was found to be the most appropriate algorithm for our model. Using a DBN and taking into account a large amount of contextual information will lead to more accurate and efficient behaviour detection.

The next chapter (Chapter 6) will show the validation of the proposed behaviour detection model using synthetic data. Moreover, experiments with different scenarios will demonstrate the ability and the validity of our proposed model to detect different kinds of behaviour during driving.

Chapter 6

Evaluation and Experiments

Objectives:

- Show the validity and the performance of our DBN driver behaviour detection model
- Illustrate the importance of including different context during the inference process
- Present our DBN driver behaviour detection model with different scenarios

6.1 Introduction

Detecting abnormal behaviours exhibited by drivers may result in enhanced road safety and prevent accidents from happening, by alerting the driver. This chapter introduces, the validation of our proposed DBN driver behaviour detection system using synthetic data, concentrating on detection ability and validity. It explains the accuracy and effectiveness of our model when detecting different styles of behaviour (drunken, fatigue, reckless and normal).

Firstly the system has been validated by applying all possible combinations of evidence (including information and observable nodes) in order to prove detection accuracy, evaluate the effects of both types of nodes on the hypothesis node (state node) and to illustrate the importance of including context from different sources during the inference process.

In addition, three experiments have been introduced to reveal the validity and performance of the proposed system during driving. The first experiment was conducted following two scenarios to validate the system's ability to detect fatigue behaviour upon receiving evidence; the system was able to detect fatigue behaviour in both scenarios using different combinations of evidence. The second experiment illustrated the ability of the system to detect drunken behaviour during driving by applying two different scenarios. The final experiment demonstrates how the system can detect reckless behaviour, by applying several combinations of evidence in two scenarios.

6.2 Model verification using synthetic data

This section presents the validity of our model for detecting four styles of behaviour (drunken, fatigue, reckless and normal behaviour) with all possible combinations of evidence. Given the parameterised Dynamic Bayesian network, the driver behaviour inference process starts upon the reception of evidence obtained from sensors. As shown in Figure 5.2, the network consists of 10 evidence nodes (root nodes and leaf nodes), each of which has two possible states. The total number of all possible combinations of evidence is 2^{10}. As the circadian node is affected by the time and time zone nodes, and the environment node is affected by the noise and temperature nodes, the circadian and environment nodes will be treated as evidence nodes throughout this evaluation. The eyes movements' node is treated as an evidence node as it affects the eyelid movements' node, which in turn affects the PERCLOS and AECS nodes. After considering the circadian, environment and the eyes movements as evidence nodes, the total number of all possible combinations of evidence attained is 2^7; which is equal to 128 possible inputs. During the inference process, the hypothesis node at time slice t will be influenced by information and observable variables at time slice t, as well as by the hypothesis node at the previous time slice *t-1*.

We have instantiated all the possible combinations of evidence as shown in Tables 6.1 - 6.4. In our evaluation, we have used two time slices in order to include the state node at the previous time slice. Therefore, this evidence represents two time slices given the same evidence. With the aim of simplicity, we will use the following abbreviations; Eyes_movements = EM, Controlling_speed = CS, Acceleration = A, Intoxication = I, Lane_maintenance = LM, Circadian = C and Environment = E in the tables.

No.	EM	CS	A	I	LM	State	Belief
1.	abnormal	bad	sudden	morethanlimit	good	Drunk	0.9983666
2.	abnormal	good	sudden	morethanlimit	good	Drunk	0.99836406
3.	abnormal	bad	sudden	morethanlimit	bad	Drunk	0.99835497
4.	abnormal	bad	moderate	morethanlimit	good	Drunk	0.99835462
5.	abnormal	good	sudden	morethanlimit	bad	Drunk	0.99835145
6.	abnormal	good	moderate	morethanlimit	good	Drunk	0.99835107
7.	abnormal	bad	moderate	morethanlimit	bad	Drunk	0.99833838
8.	abnormal	good	moderate	morethanlimit	bad	Drunk	0.99833347
9.	normal	bad	sudden	morethanlimit	good	Drunk	0.83613338
10.	normal	good	sudden	morethanlimit	good	Drunk	0.81700362
11.	normal	bad	sudden	morethanlimit	bad	Drunk	0.74741809
12.	normal	bad	moderate	morethanlimit	good	Drunk	0.74475582
13.	normal	good	sudden	morethanlimit	bad	Drunk	0.7206437
14.	normal	good	moderate	morethanlimit	good	Drunk	0.71779062
15.	normal	bad	moderate	morethanlimit	bad	Drunk	0.62553298
16.	normal	good	moderate	morethanlimit	bad	Drunk	0.59232894
17.	abnormal	bad	sudden	lessthanlimit	good	Fatigue	0.92870167
18.	abnormal	bad	sudden	lessthanlimit	bad	Fatigue	0.92789859
19.	abnormal	good	sudden	lessthanlimit	bad	Fatigue	0.92760968
20.	abnormal	good	sudden	lessthanlimit	good	Fatigue	0.92741532
21.	abnormal	bad	moderate	lessthanlimit	good	Fatigue	0.92717897
22.	abnormal	bad	moderate	lessthanlimit	bad	Fatigue	0.92658162
23.	abnormal	good	moderate	lessthanlimit	bad	Fatigue	0.92538233
24.	abnormal	good	moderate	lessthanlimit	good	Fatigue	0.82922436
25.	normal	bad	moderate	lessthanlimit	bad	Reckless	0.99319245
26.	normal	bad	sudden	lessthanlimit	bad	Reckless	0.99145328
27.	normal	good	sudden	lessthanlimit	bad	Reckless	0.99050194
28.	normal	bad	sudden	lessthanlimit	good	Reckless	0.98572
29.	normal	good	moderate	lessthanlimit	bad	Reckless	0.92319783
30.	normal	bad	moderate	lessthanlimit	good	Reckless	0.88396473
31.	normal	good	sudden	lessthanlimit	good	Reckless	0.70458318
32.	normal	good	moderate	lessthanlimit	good	Normal	0.98157912

Table 6.1: First set of combinations of evidence

Table 6.1 illustrates the states of the observable nodes (evidence), the inference results (hypothesis node's state) and the degree of belief, given that Circadian = awake and Environment = good.

No.	EM	CS	A	I	LM	State	Belief
1.	abnormal	bad	sudden	morethanlimit	good	Drunk	0.9960862
2.	abnormal	good	sudden	morethanlimit	good	Drunk	0.99608317
3.	abnormal	bad	sudden	morethanlimit	bad	Drunk	0.99607234
4.	abnormal	bad	moderate	morethanlimit	good	Drunk	0.99607192
5.	abnormal	good	sudden	morethanlimit	bad	Drunk	0.99606815
6.	abnormal	good	moderate	morethanlimit	good	Drunk	0.9960677
7.	abnormal	bad	moderate	morethanlimit	bad	Drunk	0.99605258
8.	abnormal	good	moderate	morethanlimit	bad	Drunk	0.99604673
9.	normal	bad	sudden	morethanlimit	good	Drunk	0.8330337
10.	normal	good	sudden	morethanlimit	good	Drunk	0.81457968
11.	normal	bad	sudden	morethanlimit	bad	Drunk	0.74781225
12.	normal	bad	moderate	morethanlimit	good	Drunk	0.74526557
13.	normal	good	sudden	morethanlimit	bad	Drunk	0.7222166
14.	normal	good	moderate	morethanlimit	good	Drunk	0.71949096
15.	normal	bad	moderate	morethanlimit	bad	Drunk	0.63142033
16.	normal	good	moderate	morethanlimit	bad	Drunk	0.59970129
17.	abnormal	bad	sudden	lessthanlimit	good	Fatigue	0.98697633
18.	abnormal	bad	sudden	lessthanlimit	bad	Fatigue	0.98674408
19.	abnormal	good	sudden	lessthanlimit	bad	Fatigue	0.98665223
20.	abnormal	good	sudden	lessthanlimit	good	Fatigue	0.98627503
21.	abnormal	bad	moderate	lessthanlimit	good	Fatigue	0.98632166
22.	abnormal	bad	moderate	lessthanlimit	bad	Fatigue	0.98636925
23.	abnormal	good	moderate	lessthanlimit	bad	Fatigue	0.98579751
24.	abnormal	good	moderate	lessthanlimit	good	Fatigue	0.96163593
25.	normal	bad	moderate	lessthanlimit	bad	Reckless	0.97423789
26.	normal	bad	sudden	lessthanlimit	bad	Reckless	0.96479925
27.	normal	good	sudden	lessthanlimit	bad	Reckless	0.96530923
28.	normal	bad	sudden	lessthanlimit	good	Reckless	0.94648014
29.	normal	good	moderate	lessthanlimit	bad	Reckless	0.90277857
30.	normal	bad	moderate	lessthanlimit	good	Reckless	0.85726797
31.	normal	good	sudden	lessthanlimit	good	Reckless	0.69691923
32.	normal	good	moderate	lessthanlimit	good	Normal	0.96870979

Table 6.2: Second set of combinations of evidence

Table 6.2 illustrates the states of the observable nodes (evidence), the inference results (hypothesis node's state) and the degree of belief, given that Circadian = awake and Environment = bad.

No.	EM	CS	A	I	LM	State	Belief
1.	abnormal	bad	sudden	morethanlimit	good	Drunk	0.99099056
2.	abnormal	good	sudden	morethanlimit	good	Drunk	0.99098865
3.	abnormal	bad	sudden	morethanlimit	bad	Drunk	0.99098182
4.	abnormal	bad	moderate	morethanlimit	good	Drunk	0.99098156
5.	abnormal	good	sudden	morethanlimit	bad	Drunk	0.99097917
6.	abnormal	good	moderate	morethanlimit	good	Drunk	0.99097889
7.	abnormal	bad	moderate	morethanlimit	bad	Drunk	0.99096935
8.	abnormal	good	moderate	morethanlimit	bad	Drunk	0.99096566
9.	normal	bad	sudden	morethanlimit	good	Drunk	0.8513111
10.	normal	good	sudden	morethanlimit	good	Drunk	0.834296
11.	normal	bad	sudden	morethanlimit	bad	Drunk	0.77178978
12.	normal	bad	moderate	morethanlimit	good	Drunk	0.76937932
13.	normal	good	sudden	morethanlimit	bad	Drunk	0.74748311
14.	normal	good	moderate	morethanlimit	good	Drunk	0.74488447
15.	normal	bad	moderate	morethanlimit	bad	Drunk	0.65993848
16.	normal	good	moderate	morethanlimit	bad	Drunk	0.62891076
17.	abnormal	bad	sudden	lessthanlimit	good	Fatigue	0.97963227
18.	abnormal	bad	sudden	lessthanlimit	bad	Fatigue	0.97922767
19.	abnormal	good	sudden	lessthanlimit	bad	Fatigue	0.979085
20.	abnormal	good	sudden	lessthanlimit	good	Fatigue	0.97891846
21.	abnormal	bad	moderate	lessthanlimit	good	Fatigue	0.97881735
22.	abnormal	bad	moderate	lessthanlimit	bad	Fatigue	0.97861226
23.	abnormal	good	moderate	lessthanlimit	bad	Fatigue	0.97799117
24.	abnormal	good	moderate	lessthanlimit	good	Fatigue	0.95463368
25.	normal	bad	moderate	lessthanlimit	bad	Reckless	0.95990795
26.	normal	bad	sudden	lessthanlimit	bad	Reckless	0.94573248
27.	normal	good	sudden	lessthanlimit	bad	Reckless	0.94699258
28.	normal	bad	sudden	lessthanlimit	good	Reckless	0.92032394
29.	normal	good	moderate	lessthanlimit	bad	Reckless	0.87286516
30.	normal	bad	moderate	lessthanlimit	good	Reckless	0.81588397
31.	normal	good	sudden	lessthanlimit	good	Reckless	0.63173878
32.	normal	good	moderate	lessthanlimit	good	Normal	0.97876995

Table 6.3: Third set of combinations of evidence

Table 6.3 illustrates the states of the observable nodes (evidence), the inference results (hypothesis node's state) and the degree of belief, given that Circadian = fatigue and Environment = good.

No.	EM	CS	A	I	LM	State	Belief
1.	abnormal	bad	sudden	morethanlimit	good	Drunk	0.9939844
2.	abnormal	good	sudden	morethanlimit	good	Drunk	0.99398279
3.	abnormal	bad	sudden	morethanlimit	bad	Drunk	0.99397705
4.	abnormal	bad	moderate	morethanlimit	good	Drunk	0.99397683
5.	abnormal	good	sudden	morethanlimit	bad	Drunk	0.99397483
6.	abnormal	good	moderate	morethanlimit	good	Drunk	0.9939746
7.	abnormal	bad	moderate	morethanlimit	bad	Drunk	0.99396659
8.	abnormal	good	moderate	morethanlimit	bad	Drunk	0.99396349
9.	normal	bad	sudden	morethanlimit	good	Drunk	0.88272182
10.	normal	good	sudden	morethanlimit	good	Drunk	0.86870401
11.	normal	bad	sudden	morethanlimit	bad	Drunk	0.81603807
12.	normal	bad	moderate	morethanlimit	good	Drunk	0.81396919
13.	normal	good	sudden	morethanlimit	bad	Drunk	0.79504355
14.	normal	good	moderate	morethanlimit	good	Drunk	0.79278154
15.	normal	bad	moderate	morethanlimit	bad	Drunk	0.71693047
16.	normal	good	moderate	morethanlimit	bad	Drunk	0.68827736
17.	abnormal	bad	sudden	lessthanlimit	good	Fatigue	0.99081487
18.	abnormal	bad	sudden	lessthanlimit	bad	Fatigue	0.99063415
19.	abnormal	good	sudden	lessthanlimit	bad	Fatigue	0.99056101
20.	abnormal	good	sudden	lessthanlimit	good	Fatigue	0.99019162
21.	abnormal	bad	moderate	lessthanlimit	good	Fatigue	0.99025553
22.	abnormal	bad	moderate	lessthanlimit	bad	Fatigue	0.99034612
23.	abnormal	good	moderate	lessthanlimit	bad	Fatigue	0.98985086
24.	abnormal	good	moderate	lessthanlimit	good	Fatigue	0.97228152
25.	normal	bad	moderate	lessthanlimit	bad	Reckless	0.95443975
26.	normal	bad	sudden	lessthanlimit	bad	Reckless	0.93574734
27.	normal	good	sudden	lessthanlimit	bad	Reckless	0.93816031
28.	normal	bad	sudden	lessthanlimit	good	Reckless	0.90207112
29.	normal	good	moderate	lessthanlimit	bad	Reckless	0.87400552
30.	normal	bad	moderate	lessthanlimit	good	Reckless	0.81544095
31.	normal	good	sudden	lessthanlimit	good	Reckless	0.64784632
32.	normal	good	moderate	lessthanlimit	good	Normal	0.97085001

Table 6.4: Fourth set of combinations of evidence

Table 6.4 illustrates the states of the observable nodes (evidence), the results of inference (hypothesis node's state) and the degree of belief, given that Circadian = fatigue and Environment = bad. The data illustrated by the aforementioned tables were obtained automatically using polytree algorithm, which is provided by Genie software. The process of belief updating was carried out by setting the nodes' states according to the required input and then performing "update belief" function.

As shown by the above tables, the system is able to detect the state of the driver (drunken, fatigue, reckless and normal) accurately over time, by applying all possible combinations of evidence. This proves the validity and the accuracy of the system for detecting different styles of behaviour, where different combinations of evidence lead to different states with different degrees of belief.

6.2.1 Evaluation of the effects of information nodes

This section shows the effects of the circadian and environment nodes on the state nodes, where the state node is directly affected by these nodes. As shown in Table 5.8, the circadian node may be in one of two mutually exclusive states depending on the state of its parents time and time_zone nodes. Similarly, Table 5.9 indicates that the environment node may be in either one of its permitted states, according to the states of its parents noise and temperature nodes.

Based on the proposed system design, different states of the circadian and environment nodes lead to different degrees of belief in the state node. The system is able to detect normal driving behaviour in all possible states of the circadian and environment nodes given all possible combinations of all of the observable nodes' states (ref no. 32 in table 6.1, ref no. 32 in table 6.2, ref no. 32 in table 6.3 and ref no. 32 in table 6.4). As shown in these tables, the degree of belief in normal behaviour reaches its highest level in case where Circadian = awake and Environment = good. This also validates the detection accuracy of the proposed system. For fatigue, reckless and drunken behaviour, we have conducted several comparisons to validate the effects of the above nodes on the state node, as illustrated in the following paragraphs.

Figure 6.1 depicts a comparison between all the possible combinations of evidence, which leads to an assessment of fatigue behaviour. As seen in the figure, four curves represent the level of fatigue for all possible states of the circadian and environment nodes, given all the possible combinations of all of the observable nodes' states. The level of driver fatigue, in the case Circadian= awake and Environment= good is the lowest level in the chart, while in the case Circadian= fatigue and Environment = bad fatigue reaches its highest level. This demonstrates the effect of the environment and the circadian rhythm on the driver's level of fatigue. This further validates our driver behaviour detection model.

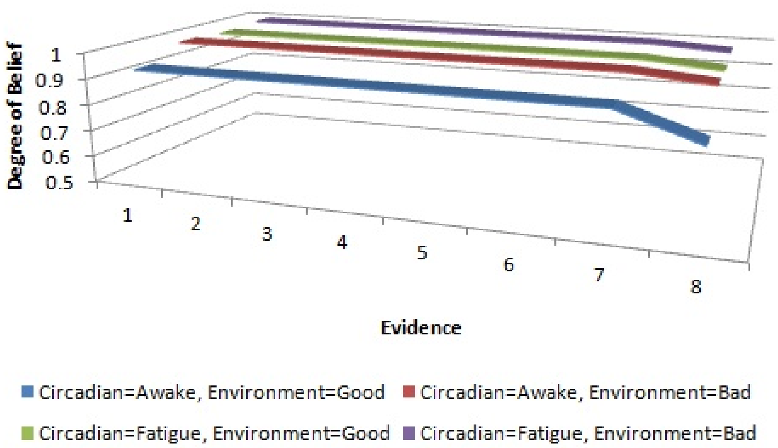

Figure 6.1: A comparison between all possible evidence of fatigue behaviour

Figure 6.2 depicts a comparison between all the possible combinations of evidence leading to an assessment of reckless behaviour. Four curves in the figure characterise the belief in reckless behaviour in all possible states of the circadian and environment nodes, given all possible combinations of all of the observable nodes' states. Reckless behaviour is more likely to be present when Environment = good and Circadian = awake, which again validates the driver behaviour detection model.

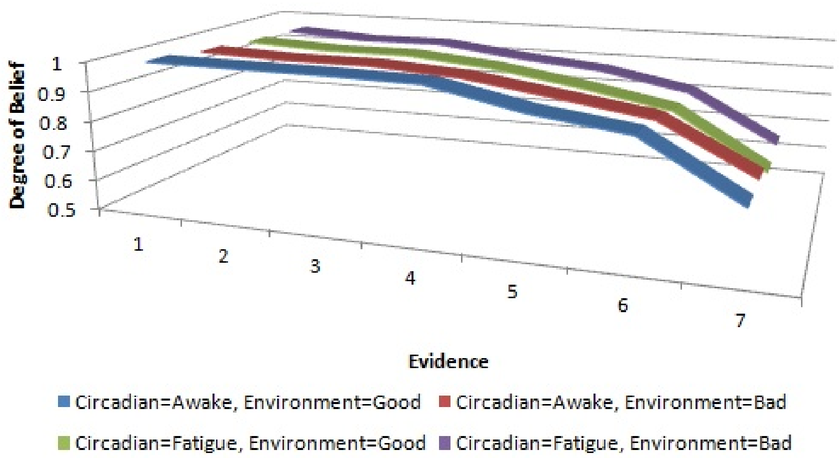

Figure 6.2: A comparison between all possible evidence of reckless behaviour

Figure 6.3 illustrates a comparison between all the possible combinations of evidence, which lead to an assessment of drunken behaviour. As shown in the figure, there are four curves showing the belief in drunken behaviour for all possible states of the circadian and environment nodes, given all possible combinations of all of the observable nodes' states. The belief in drunken behaviour is approximately the same in all cases. It reaches its highest degree when Circadian = fatigue and Environment = bad. This again proves the validity of our proposed model.

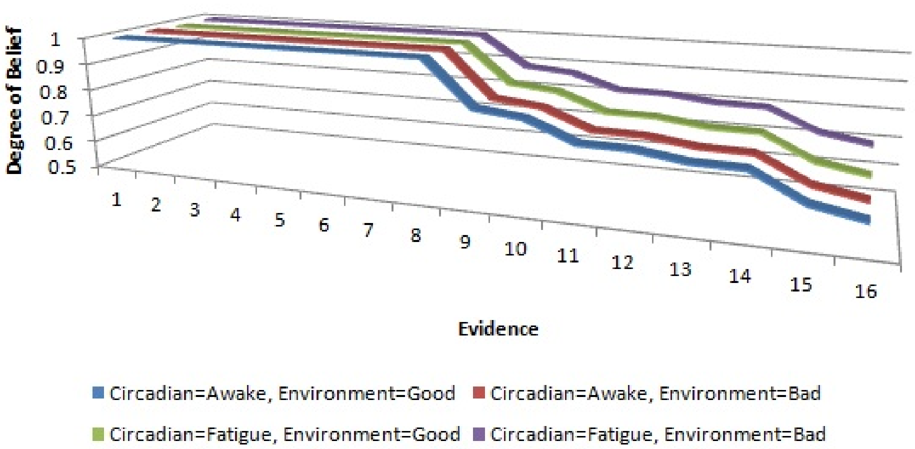

Figure 6.3: A comparison between all possible evidence of drunk behaviour

6.2.2 Evaluation of the effects of observable nodes

In this thesis, the behaviour of the driver is considered as the current unobservable state of the driver. The current state can be characterised by capturing a large amount of context. This section presents the importance of including more than one context during the inference process and shows the validity of our system in terms of detection accuracy, using context related to the driver, the vehicle and the environment.

As shown in Figure 6.4, drunken driving can be detected more accurately when the eyes of the driver are closed, as this provides a combination of intoxication and other evidence (ref nos. 1-8 in table 6.1), as compared with the case where the eyes are open combined with intoxication and other evidence (ref nos. 9-16 in table 6.1).

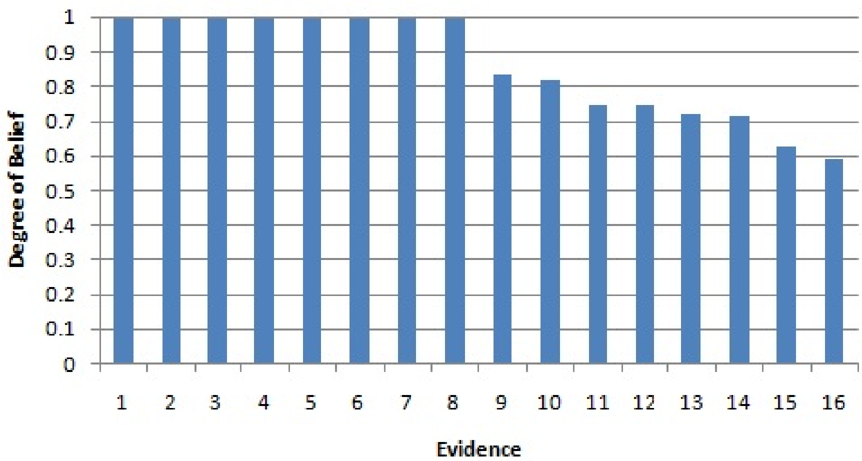

Figure 6.4: Detecting the drunken behaviour given different evidence

As depicted in Figure 6.5, when we instantiated more than one evidence in combination with eyes movements evidence (ref nos. 17-23 in table 6.1) the fatigue level of the driver was found to be more accurate than when we instantiated a single evidence regarding eyes movements (ref no. 24 in table 6.1).

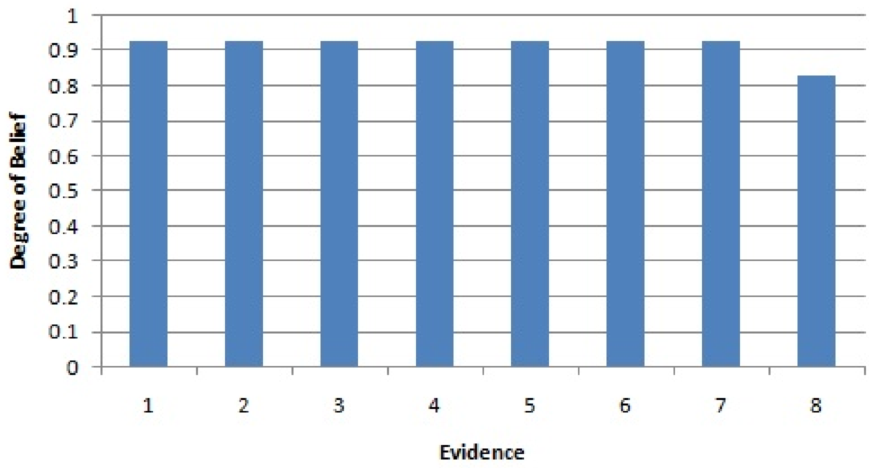

Figure 6.5: Detecting the fatigue behaviour given different evidence

As illustrated in Figure 6.6, in the case of reckless driving behaviour, when we instantiated more than one evidence (ref nos. 25-28 in table 6.1), the detection accuracy was more accurate than when we instantiated a single evidence such as bad lane maintenance, bad controlling speed and sudden acceleration (ref nos. 29-31 in table 6.1).

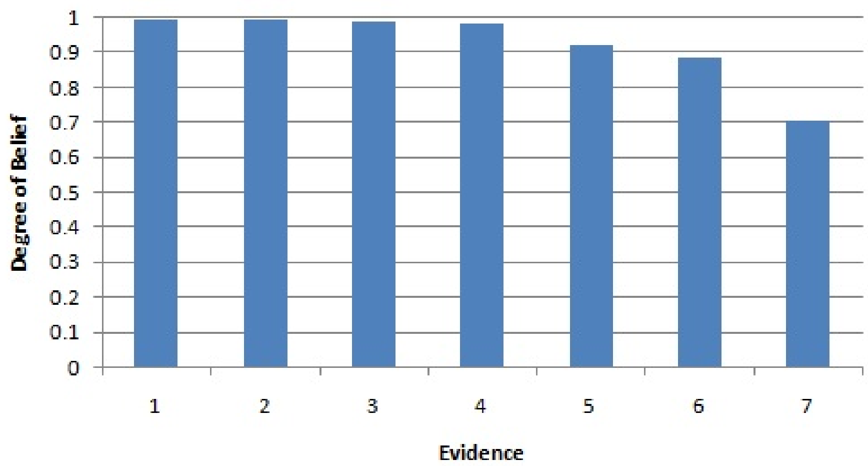

Figure 6.6: Detecting the reckless behaviour given different evidence

The system has been able to detect drunken behaviour in the case where there is only one evidence referring to this style of behaviour (intoxication more than the limit) and all other evidence refers to normal behaviour (ref no. 14 in table 6.1), for example, eyes are open, speed control is good, etc. This further demonstrates the validity of our system because a drunk driver is expected to exhibit dangerous behaviour at any time.

It can be seen from Table 6.1 that the system detected drunken behaviour in (ref no. 14 in table 6.1) with a higher degree of belief than in (ref no. 16 in table 6.1); even though another evidence (bad lane maintenance) is present in the latter. This has occurred for the following reason: this node can result from both drunken and reckless behaviour but with higher probability in the case of reckless behaviour. Therefore, the degree of belief in drunken behaviour is decreased and in reckless behaviour is increased, but the system is still able to detect the drunken behaviour because the intoxication is above the limit.

The above mentioned inference results reveal the fact that the presence of more than one evidence guarantees the occurrence of a specific behaviour, and explain the importance of combining different types of contextual information in order to deduce the behaviour of a driver. These results show the utility of the proposed driver behaviour detection system in detecting different styles of behaviour by combining context from different sensors. In the above comparisons, we have used the data in Table 6.1 where Circadian = awake and Environment = good since we have tried to show the importance of including more than one context in the inference process. This is because the situation is the same in Tables 6.2 - 6.4.

6.3 Experiments

This section shows the validity and effectiveness of our proposed DBN model in detecting different states of the driver while driving his/her car (detecting the driver's state over time). During driving, the driver might be in a particular state, which he or she may remain in for a period of time and then potentially changing to a different state. In this section, we will validate the ability of our system to detect different styles of behaviour, which are fatigue, drunken and reckless behaviour.

After constructing a DBN for a specific domain, we can use it to represent knowledge of the domain and to reason about the interpretation of certain input data. The interpretation process includes instantiating a group of variables analogous with the input data and then calculating their effects on the probability of other variables referred to as hypothesis nodes [126]. In the inference process, the DBN will be unrolled for t time slices, which is the time period that interests the decision maker. Each time slice represents a snapshot of the evolving system (driver's behaviour) and this can be defined as a period of time in which the system receives sensor readings, transfers them into a machine-processable form and then feeds them into the system. Inference (filtering) is the process of calculating the probability of the state node at time t given the evidence from the past until time t (calculating $\mathrm{P}(X_t|e_{1:t})$). The process is used to track the current state of the system continually, in order to make decisions [119].

In the following experiments, we have assumed that each time slice represents a period of one second, and during driving, the system will perform the inference process every five seconds (five time slices) in order to continuously detect the state of the driver and take the corresponding decision. If a testbed was available and equipped with the sensors required to acquire data for our proposed model, the period of time for each time slice could be changed according to the real time needed to do the sampling process and the period in which we infer the behaviour can change in relation to real driving behaviours. Usually the real data are divided into two parts; the first part is used to learn the parameters of the model, using one of the available learning algorithms, and the second is used for testing the model.

Most road accidents occur due to driver error. In fact, fatigued, drunken and reckless driving are considered the main causes, due to the dangerous actions that are associated with these states. Detecting the above kinds of driving will lead to the provision of a safe driving environment and will save people's lives. Therefore, our system concentrates on the mechanism for detecting abnormal driver behaviours. The following sections present experiments to prove the validity of the proposed model in detecting the above mentioned kinds of behaviours during driving.

The inputs used in the following scenarios might not reflect real life situations (sensor readings), but we are trying to show the capability of the proposed system in detecting the driver's behaviour, by changing the system's inputs to challenge and make sure that the system is comprehensive enough to recognise each state correctly.

6.3.1 Experiment 1: Detecting the Fatigued Driver

In this study, the driver is considered to be fatigued if he/she exhibits abnormal eyes movements and if one or all of the following criteria are satisfied in combination with abnormal eyes movement evidence:

- The driver is driving without maintaining a proper lane position.
- The driver is performing sudden acceleration.

- The speed of the vehicle exceeds the speed limit.
- There is no alcohol intoxication in the driver's blood.

Two scenarios will present the case of changing from normal to fatigue behaviour. The first scenario shows the validity of system detection in the case where eyes movements are abnormal and the driver is driving without maintaining a proper lane position. The second scenario illustrates the detection of a fatigued driver in the case where eyes movements are abnormal and the driver does not control the vehicle's speed. As shown earlier in this chapter, the level of driver's fatigue can vary according to the states of the circadian and environment nodes. We have set the states of the circadian and environment nodes to fatigue and bad respectively in both scenarios.

Scenario 1: As shown in Figure 6.7, the vehicle is moving from point 1 to point 2 on a straight two-sided road. We have divided the period of driving in which the vehicle is moving between these points into 15 equivalent time slices. Each time slice represents a period of one second during which the vehicle collects new information via sensors and feeds it to the system, the inference (filtering) process being carried out every five seconds. During time slices 1-5, the driver is driving while in the normal state. In time slices 6-10, the sensors indicate that the driver's eyes movements are abnormal which means that he/she has changed from the normal to fatigue state. From time slice 11 to 15, another indication of fatigue is present in addition to the abnormal eyes movements evidence, as the vehicle is not maintaining the proper lane position, which increases the degree of belief in fatigue.

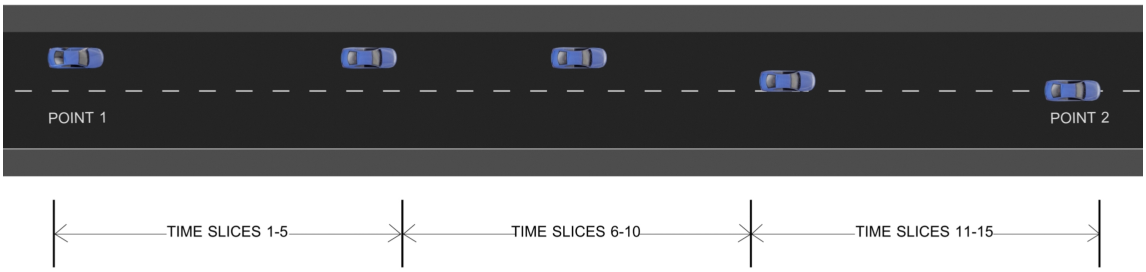

Figure 6.7: Scenario 1

Modelling the above scenario using our proposed system can be carried out by setting the states of the nodes according to the data provided by sensors. Table 6.5 illustrates the combination of evidence in time slices 1-5 where the behaviour is normal, while Tables 6.6 and 6.7 present the combinations of evidence used in time slices 6-10 and 11-15 respectively.

Node	EM	CS	A	I	LM
State	normal	good	moderate	lessthanlimit	good

Table 6.5: Evidence at time slices 1-5

Node	EM	CS	A	I	LM
State	abnormal	good	moderate	lessthanlimit	good

Table 6.6: Evidence at time slices 6-10

Node	EM	CS	A	I	LM
State	abnormal	good	moderate	lessthanlimit	bad

Table 6.7: Evidence at time slices 11-15

Figure 6.8 illustrates the inference results of the proposed DBN model upon receiving evidence in time slices 1-5, 6-10 and 11-15. It can be seen from the figure that there are four curves demonstrating the degrees of belief in the state node at different time slices. The system has detected the normal driving behaviour at time slice 5 with a belief of about 0.97. This is because all sensor readings from time slice 1 to 5 indicated normal driving behaviour. At time slice 10, the system was able to infer the fatigue state with a belief of about 0.97 with the appearance of evidence regarding fatigue (abnormal eyes movements) from time slice 6 to 10. The system was able to deduce the fatigue state at time slice 15 with a degree of belief of about 0.99 due to the appearance of more than one evidence (abnormal eyes movements and bad lane maintenance) from time slice 11 to 15.

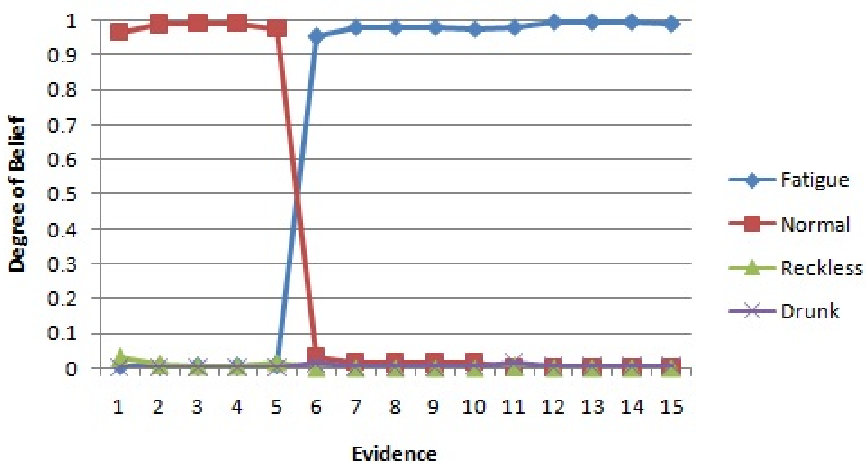

Figure 6.8: Inference results at time slices 5, 10 and 15

The above scenario presents the system ability in detecting the fatigue state during driving, using different sensor readings. Different degrees of belief can be attained according to the evidence from the sensors. As shown in the above figure, the probabilities of detection (the degrees of belief in the state node) are high which caused the curves to fluctuate sharply. This is because we have used synthetic data and we have entered hard evidence to the system. In other words, when we have instantiated the evidence nodes, the probability of each evidence has been set to

100%. For example, we have set the probability of abnormal eyes movements to 100% abnormal. The results would be different if real data were entered into the system; for example eyes movements might be 75% abnormal and 25% normal. This situation will be the same in all of the following scenarios, for the same reason.

Scenario 2: As shown in Figure 6.9, the vehicle is moving from point 1 to point 2 on a straight two-sided road. We have divided the period of driving in which the vehicle is moving between these points into 15 equivalent time slices. Each time slice represents a period of one second during which the vehicle gathers new information via sensors and feeds it into the system. The inference process will be done every five seconds. During time slices 1-5, the driver's driving behaviour is normal. In time slices 6-10, the sensors indicate that the driver's eyes movements are abnormal which means that the driver has changed from the normal to fatigue state. During time slices 11 to 15, another indication of fatigue is presented as the vehicle's speed exceeds the road speed limit, which increases the belief of fatigue.

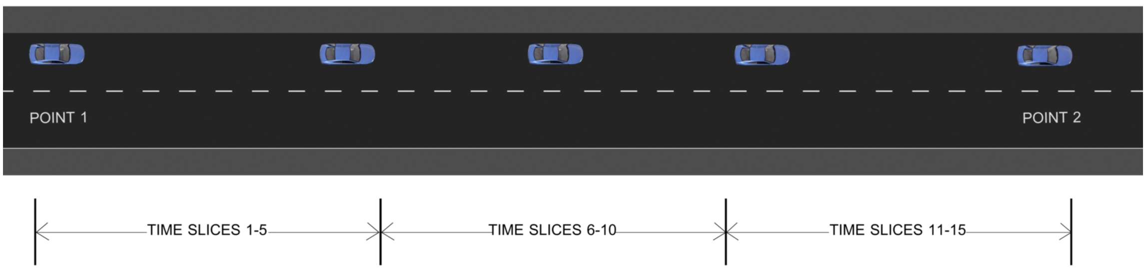

Figure 6.9: Scenario 2

Modelling the above scenario using our proposed system can be done by setting the states of the nodes according to the data provided by sensors. Table 6.8 illustrates the combination of evidence in time slices 1-5 when the behaviour is normal, while Tables 6.9 and 6.10 present the combinations of evidence used in time slices

6-10 and 11-15 respectively.

Node	EM	CS	A	I	LM
State	normal	good	moderate	lessthanlimit	good

Table 6.8: Evidence at time slices 1-5

Node	EM	CS	A	I	LM
State	abnormal	good	moderate	lessthanlimit	good

Table 6.9: Evidence at time slices 6-10

Node	EM	CS	A	I	LM
State	abnormal	bad	moderate	lessthanlimit	good

Table 6.10: Evidence at time slices 11-15

Figure 6.10 depicts the inference results of the proposed DBN model upon receiving evidence in time slices 1-5, 6-10 and 11-15. As shown in the figure, there are four curves representing the degrees of belief in the state node at different time slices. The system detected the normal driving behaviour at time slice 5 with a belief of about 0.97. This is because all sensor readings indicated normal driving behaviour from time slice 1 to 5. The inference results at time slice 10 illustrate the degree of belief of the system in the fatigue state, the system was able to detect the fatigue state with a belief of about 0.97 when the evidence regarding fatigue appeared (abnormal eyes movements) from time slice 6 to 10. The belief in the fatigue state at time slice 15 has reached about 0.99 due to the appearance of more than one fatigue evidence (abnormal eyes movements and bad speed control) from time slice 11 to 15.

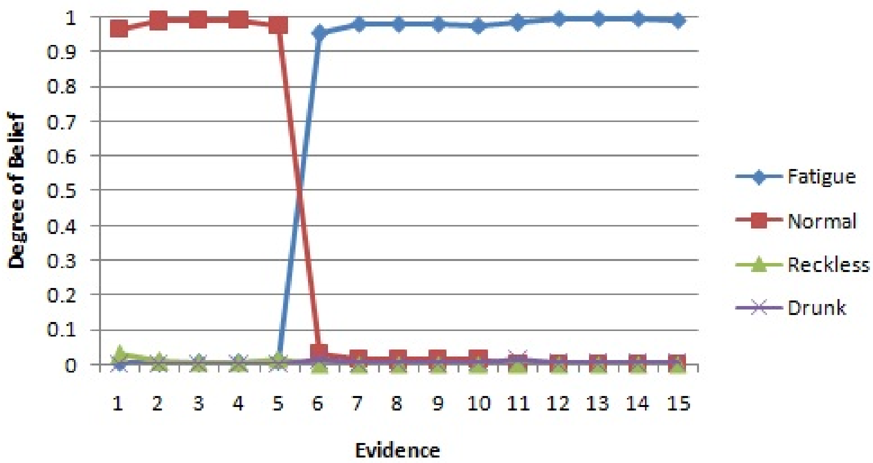

Figure 6.10: Inference results at time slices 5, 10 and 15

6.3.2 Experiment 2: Detecting the Drunk Driver

In this study, the driver is considered to be drunk if he/she exhibits the same characteristics as a fatigued driver, as illustrated in the previous experiment; but in addition, there is alcohol intoxication in the driver's blood.

In the following, two scenarios will describe the case of changing from normal to drunken behaviour. The first scenario illustrates the validity of our proposed system in detecting the drunk driver in the case where there is alcohol intoxication above the limit, eyes movements are abnormal and the driver fails to maintain the proper lane position. Meanwhile, the second scenario shows the ability of the system to detect the drunk driver in the case where there is alcohol intoxication, eyes movements are abnormal and the driver fails to control the vehicle's speed. Different degrees of belief can be achieved according to the states of the circadian and environment nodes. In this experiment, we have set the states of the circadian and environment nodes to fatigue and bad respectively in both scenarios.

Scenario 1: In this scenario, we will consider the case depicted in Figure 6.7, but with the following sensor readings: during time slices 1-5, the driver is driving while in the normal state; in time slices 6-10, the sensors indicate that there is alcohol intoxication above the limit which means that the driver changed from the normal to drunk state; from time slice 11 to 15, two other indications of drunkenness have been presented since the vehicle is not maintaining the proper lane position and the eyes movements are abnormal, which increases the degree of belief in drunken behaviour.

Modelling the above scenario using our DBN model can be carried out by setting the states of the nodes in each time slice according to sensor readings illustrated in this scenario. Table 6.11 presents the combinations of evidences in time slices 1-5 where the behaviour is normal, while Tables 6.12 and 6.13 illustrate the combinations of evidence used in time slices 6-10 and 11-15 respectively.

Node	EM	CS	A	I	LM
State	normal	good	moderate	lessthanlimit	good

Table 6.11: Evidence at time slices 1-5

Node	EM	CS	A	I	LM
State	normal	good	moderate	morethanlimit	good

Table 6.12: Evidence at time slices 6-10

Node	EM	CS	A	I	LM
State	abnormal	good	moderate	morethanlimit	bad

Table 6.13: Evidence at time slices 11-15

Figure 6.11, illustrates the inference results of the proposed DBN model upon receiving evidence in time slices 1-5, 6-10 and 11-15. As shown in the figure, there are four curves signifying the inference results for all of the possible states of the hypothesis node at different time slices. The system detected the normal driving behaviour at time slice 5 with a belief of about 0.97. This is because all sensor readings from time slice 1 to 5 correspond to normal driving behaviour.

It can be clearly seen from Figure 6.11 that at time slice 10 the system was able to detect the drunken behaviour with a belief of about 0.91 when only one evidence regarding drunken behaviour had appeared (intoxication is above the limit). Simultaneously, the curve that demonstrates reckless behaviour grew slightly to reach about 0.10. This occurred due to the conditional probability distributions of the intoxication node, as illustrated in Table 5.12: given that the intoxication evidence above the limit, the probability of the state node being in its drunken state is 0.9 and the probability of being in its reckless state is 0.025. This led to the rise in the curve of reckless behaviour and the lowering of the curve that represents drunken behaviour.

Performing the inference at time slice 15 resulted in detection of drunken behaviour with a degree of belief of about 0.99. This happened due to the appearance of more than one drunken behaviour evidence (abnormal eyes movements, intoxication is more than the limit and the driver fails to maintain the proper lane position) from time slice 11 to 15 which highly increased the belief in drunken behaviour and decreased the belief in other states.

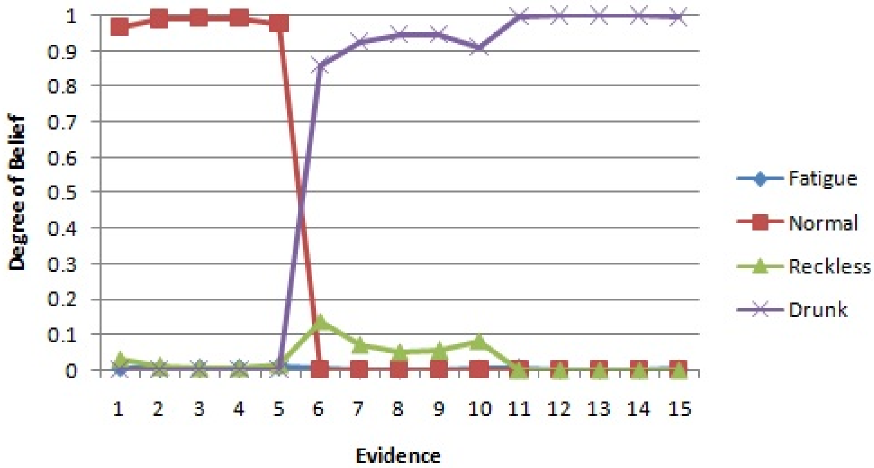

Figure 6.11: Inference results at time slices 5, 10 and 15

Scenario 2: In this scenario, we will consider the case illustrated in Figure 6.9, but with the following sensor readings: during time slices 1-5, the driver is driving while in the normal state; in time slices 6-10, the sensors indicate that there is alcohol intoxication over the limit and the eyes movements are abnormal, which means that the driver changed from the normal to drunken state; from time slice 11 to 15, another indication of drunkenness has been presented as the vehicle's speed exceeds the limit , which increases the degree of belief in drunken behaviour.

Modelling the above scenario using our DBN model is achieved by setting the states of the nodes in each time slice according to sensor readings illustrated in this scenario. Table 6.14 presents the combination of evidence in time slices 1-5 where the behaviour is normal, while Tables 6.15 and 6.16 illustrate the combinations of evidence used in time slices 6-10 and 11-15 respectively.

Node	EM	CS	A	I	LM
State	normal	good	moderate	lessthanlimit	good

Table 6.14: Evidence at time slices 1-5

Node	EM	CS	A	I	LM
State	abnormal	good	moderate	morethanlimit	good

Table 6.15: Evidence at time slices 6-10

Node	EM	CS	A	I	LM
State	abnormal	bad	moderate	morethanlimit	good

Table 6.16: Evidence at time slices 11-15

As depicted in Figure 6.12, four curves categorise the degrees of belief in the state node at different time slices. The figure illustrates the inference results of the proposed DBN model upon receiving evidence in time slices 1-5, 6-10 and 11-15. It can be seen from the figure that the system has detected the normal driving behaviour at time slice 5 with a belief of about 0.97. This is because all sensor readings corresponded to normal driving behaviour from time slice 1 to 5. The system was able to detect the drunken behaviour at time slice 10 with a belief of about 0.99 when the evidence regarding drunken behaviour had appeared (intoxication is above the limit and eyes movements are abnormal) from time slice 6 to 10.

The inference results at time slice 15 correspond to detecting the drunken behaviour with a degree of belief of about 0.99, due to the appearance of more than one drunken behaviour evidence (abnormal eyes movements, intoxication is above the limit and the driver fails to control the vehicle's speed) from time slice 11 to 15.

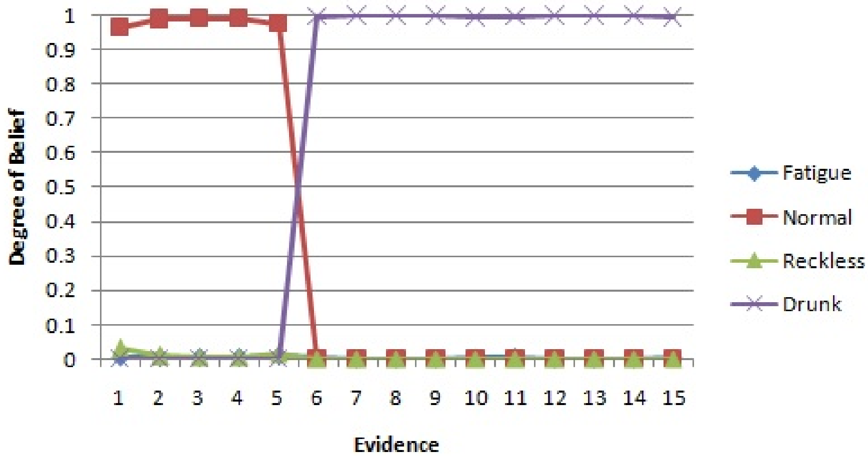

Figure 6.12: Inference results at time slices 5, 10 and 15

The above scenarios present the system ability to detect drunken behaviour during driving using different sensor readings. Different degrees of belief have been reached according to the evidence received from sensors.

6.3.3 Experiment 3: Detecting the Reckless Driver

The driver is classified in this study as driving in this category if the eyes movements of the driver are normal, there is no alcohol intoxication but if one or more than one of the following criteria are satisfied:

- Driving with sudden acceleration.
- Driving with out maintaining the proper lane position.
- The speed of the vehicle exceeds the speed limit.

Two scenarios will present the case of changing from normal to reckless behaviour. The first scenario shows the validity of the system in detecting the reckless driver when driving without maintaining the proper lane position and without controlling the vehicle speed. Meanwhile, the second scenario illustrates detection of the reckless driver in the case where the driver is performing sudden acceleration and does not maintain the proper lane position. We have set the states of the circadian and environment nodes to awake and good respectively in both scenarios.

Scenario 1: As depicted in Figure 6.13, the vehicle is moving from point 1 to point 2 on a straight two-sided road. The period of driving in which the vehicle is moving between these points were divided into 15 equivalent time slices. Each time slice represents a period of one second during which the vehicle collects new information via sensors and feeds this information to the system. The inference process is carried out every five time slices (five seconds). During time slices 1-5, the driver is driving while in the normal state. In time slices 6-10, the sensors indicate that the vehicle's position in lane is bad, while eyes movements are normal and intoxication is less than the limit. That means that the driver has changed from the normal to reckless state. From time slice 11 to 15, another indication of reckless behaviour appears as the driver fails to control the vehicle's speed, which increases the degree of belief in reckless behaviour.

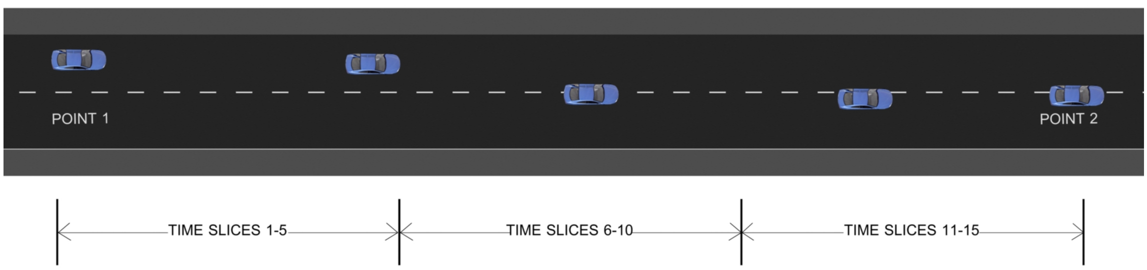

Figure 6.13: Scenario 1

Modelling the above scenario using our DBN model can be carried out by setting the states of the nodes in each time slice according to the sensor readings illustrated in this scenario. Table 6.17 presents the combination of evidence in time slices 1-5 where the behaviour is normal, while Tables 6.18 and 6.19 illustrate the combinations of evidence used in time slices 6-10 and 11-15 respectively.

Node	EM	CS	A	I	LM
State	normal	good	moderate	lessthanlimit	good

Table 6.17: Evidence at time slices 1-5

Node	EM	CS	A	I	LM
State	normal	good	moderate	lessthanlimit	bad

Table 6.18: Evidence at time slices 6-10

Node	EM	CS	A	I	LM
State	normal	bad	moderate	lessthanlimit	bad

Table 6.19: Evidence at time slices 11-15

Figure 6.14 illustrates the inference results of the proposed DBN model upon receiving evidence in time slices 1-5, 6-10 and 11-15 respectively. There are four curves in the figure demonstrating the inference results at different time slices. As illustrated in Figure 6.14, the system has detected the normal driving behaviour at time slice 5 with a belief of about 0.98. This is because all sensor readings from time slice 1 to 5 correspond to normal driving behaviour. At time slice 10, the system was able to detect the reckless behaviour with a belief of about 0.95 when the evidence (bad lane maintenance) regarding reckless behaviour appeared from time slice 6 to 10. The degree of belief in the reckless state at time slice 15 was about 0.99 due to the appearance of more than one reckless behaviour evidence (bad lane maintenance and the driver fails to control the vehicle's speed) from time slice 11 to 15.

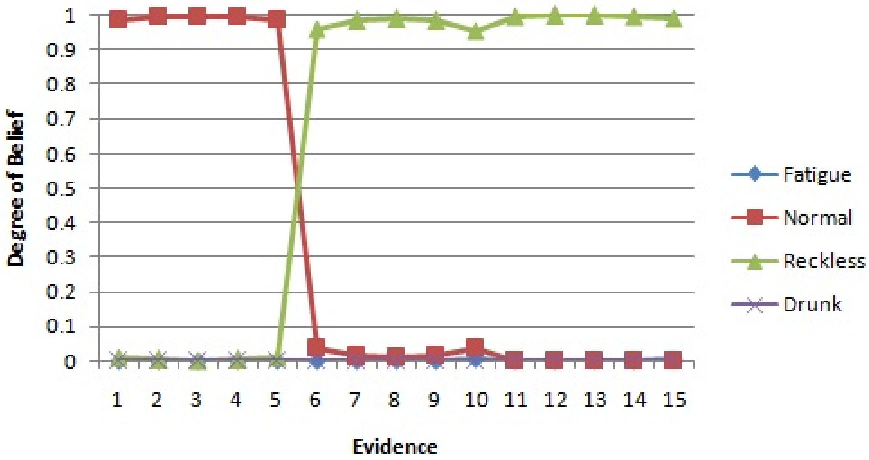

Figure 6.14: Inference results at time slices 5, 10 and 15

Scenario 2: As depicted in Figure 6.15, the vehicle is moving from point 1 to point 2 on a straight two-sided road. During time slices 1-5, the driver is driving while in the normal state. In time slices 6-10, the sensors indicate that the driver is performing sudden acceleration, while eyes movements are normal and intoxication is less than the limit, which means that the driver has changed from the normal to reckless state. From time slice 11 to 15, another indication of reckless behaviour appears as the vehicle does not maintain the proper lane position. This increases the degree of belief in reckless behaviour.

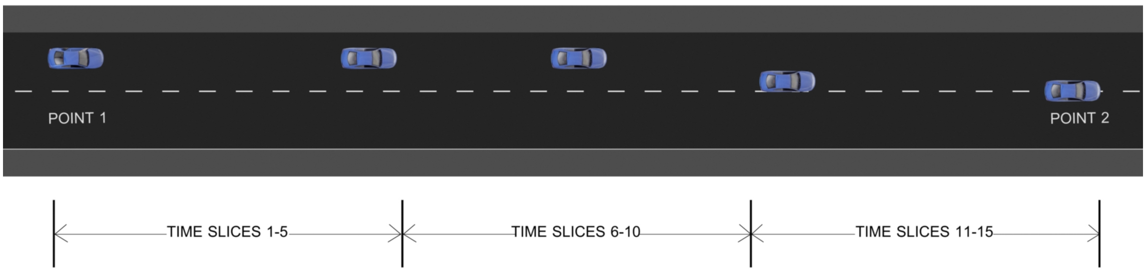

Figure 6.15: Scenario 1

Modelling the above scenario using our DBN model can be carried out by setting the states of the nodes in each time slice according to the sensor readings illustrated in this scenario. Table 6.20 presents the combination of evidence in time slices 1-5 where the behaviour is normal, while Tables 6.21 and 6.22 illustrate the combinations of evidence used in time slices 6-10 and 11-15 respectively.

Node	EM	CS	A	I	LM
State	normal	good	moderate	lessthanlimit	good

Table 6.20: Evidence at time slices 1-5

Node	EM	CS	A	I	LM
State	normal	good	sudden	lessthanlimit	good

Table 6.21: Evidence at time slices 6-10

Node	EM	CS	A	I	LM
State	normal	good	sudden	lessthanlimit	bad

Table 6.22: Evidence at time slices 11-15

Figure 6.16 illustrates the inference results of the proposed DBN model upon receiving evidence in time slices 1-5, 6-10 and 11-15 respectively. As shown in the figure, the system has detected the normal driving behaviour at time slice 5 with a belief of about 0.98. This is because all sensor readings corresponded to normal driving behaviour from time slice 1 to 5. It can be clearly seen from the figure that at time slice 10 the system was able to detect the reckless behaviour with a belief of about 0.80 when only one evidence regarding the reckless behaviour had appeared from time slice 6 to 10 as the driver was performing sudden acceleration. At the same time slice, the curve that represents normal driving behaviour dropped to about 0.18.

The lowest degree of belief in reckless behaviour was reached when we entered evidence regarding sudden acceleration only in time slices 6-10. If we compare the belief in reckless behaviour with the previous scenario at time slice 6-10 when we entered bad lane maintenance evidence, the belief in reckless behaviour was found to be higher and the belief in normal behaviour was lower. This is due to the conditional probability distributions of the acceleration and lane maintenance nodes. As shown in Table 5.13, given sudden acceleration evidence, the probability of normal behaviour is 0.075, while it can be seen from Table 5.11 that given bad lane maintenance evidence, the probability of normal behaviour is 0.025. This has led to a lower probability of reckless behaviour in the latter scenario and a higher probability in the former.

Moreover, the figure illustrates the degree of belief in the state node at time slice 15, when the degree of belief in reckless behaviour was about 0.99, due to the appearance of more than one reckless behaviour evidence (bad lane maintenance and the driver is performing sudden acceleration) from time slice 11 to 15, which decreased the belief in normal and greatly increased the belief in reckless behaviour.

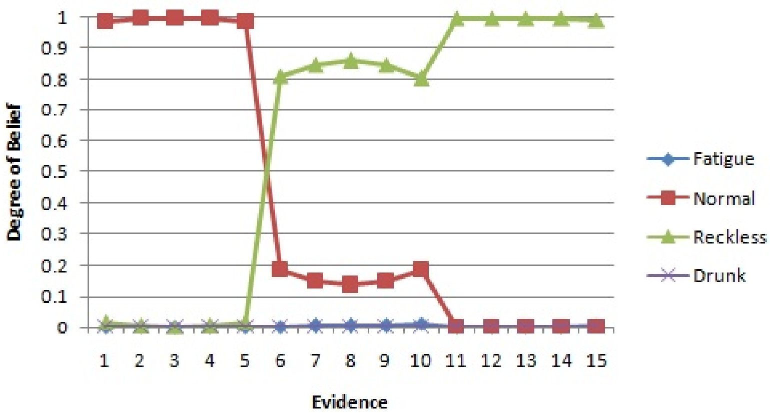

Figure 6.16: Inference results at time slices 5, 10 and 15

These scenarios demonstrate the system ability in detecting reckless behaviour during driving using different sensor readings. Different degrees of belief can be reached according to the evidence received from sensors.

All the above experiments show the ability of our proposed system to detect different styles of behaviour during driving. The system has been able to detect normal driving behaviour in all scenarios. Fatigue, drunken and reckless behaviour were detected using different combinations of evidence regarding these kinds of behaviours. Different combinations of evidence have led to different degrees of belief in the state node, given different states for the circadian and environment nodes.

6.4 Summary

In this chapter, the validity of the proposed system in detecting different styles of behaviour during driving has been presented. The importance of including context from several sources has been shown.

Three experiments each with two scenarios have been introduced to validate the system ability to detect different kinds of behaviour during driving (over time). The first experiment demonstrated two scenarios to show the system ability in detecting fatigue behaviour upon receiving evidence during driving. The second experiment has illustrated the validity of the proposed system in detecting drunken behaviour during driving using different sensor readings (different scenarios), while the last experiment, has shown the validity of the system in detecting reckless behaviour during driving, with two scenarios.

Monitoring and detecting the abnormal behaviours exhibited by drivers is important for enhancing road safety and preventing accidents from happening. The validation results of our proposed DBN model have shown the ability of the system to detect four styles of behaviour during driving, which are normal, reckless, fatigue and drunken behaviour. The results demonstrate the ability of the proposed DBN model to deduce the driver's behaviour by taking into account various sensor readings over time. The importance of including more than one context during the inference process was proven, the degrees of belief reaching their highest with the presence of more than one context.

Chapter 7

Conclusion and Future Work

Objectives:

- Summarise the work in this thesis
- Measure of success
- Propose future work that follows on from this thesis

7.1 Conclusion

VANET is an emerging application of MANET; vehicles in VANET are the nodes in the network and communicate with each other, or with the road side unit (RSU), using dedicated short range communication (DSRC). VANET forms of communication have enabled the introduction of a wide range of safety and non-safety applications. Safety applications have the potential to enhance the safety of passengers, and therefore represent a promising area of VANET; as such they are attracting the interests of car manufacturers, researchers and governments. Monitoring and detecting the behaviour of drivers is vital to ensuring road safety by alerting the driver and other vehicles on the road of cases of abnormal driving behaviour. Driver's behaviour is affected by multiple factors related to the driver, the vehicle and the environment. While driving, drivers may be in different states and it is therefore important to capture the static and dynamic aspects of such behaviour and to take into account contextual information related to that behaviour.

The goal of the work that was carried out in this thesis was to find a fundamental solution to improve road safety, prevent accidents from happening and provide a safe driving environment by developing a robust driver behaviour detection system in VANET by utilising a context-aware system approach. Developing such a system could accomplish the goal of preventing road accidents and save people's lives, hence, enlarging the scope and effectiveness of VANET safety applications.

Through the research presented in this thesis, normal and abnormal driving behaviours (i.e. fatigue, drunk and reckless) were defined in chapter 3 from the perspective of a context-aware system. A five-layer context-aware architecture for VANET was introduced in chapter 4; this can detect the behaviour of the driver by capturing information about the driver, the vehicle and the environment. The architecture comprises three main phases, which are: the sensing, reasoning and application phase, all of which represent the three main subsystems of the context-aware system. In the sensing phase, the system senses information about the driver, the vehicle and the environment. The reasoning phase is responsible for performing reasoning under uncertainty and over time, in order to deduce the current behaviour of the driver effectively. Finally, the application phase is responsible for operating in-vehicle alarms and disseminating warning messages.

Performing reasoning over time and under uncertainty requires efficient reasoning technique to combine information from different sensors and to deduce the driver's behaviour. Therefore, in chapter 5, a DBN model was presented in order to perform this task. The DBN model was able to detect four styles of behaviour: fatigue, drunk, reckless and normal behaviour. The behaviour of the driver is an evolving process; developing over the course of driving, which means the driver state at a previous time is considered an influential factor that affects the state at the current time. The DBN model combines information from different sources and takes into account the static and the temporal aspect (the driver's state at the previous time) of behaviour during the inference process.

The proposed model was validated in chapter 6 using synthetic data. The results from the validation have demonstrated the ability and the effectiveness of the proposed model for detecting different kinds of behaviour while driving, using different sensor readings. Different degrees of belief were presented according to the data inputted. Moreover, the results revealed the fact that including more than one context when inferring behaviour guarantees the detection of specific behaviours.

7.2 Measure of Success

The results in this thesis began with a set of aims labelled as the measure of success; these were illustrated in chapter 1. This section concentrates on each of these criteria to determine the degree to which the research has been successful, as shown below:

- *The research questions specified in chapter one have been met as follows:*
  - *What kind of information is needed to detect different styles of driver behaviour accurately?*

    Several factors (context) can be combined to deduce the behaviour of the driver; the most significant contributory factors have been chosen and explained in detail in chapter 5. This includes information about the driver (i.e. eyes movements), the vehicle (i.e. position in lane) and the environment (i.e. temperature).
  - *How can we design an effective driver behaviour detection system architecture for VANET by utilising a context-aware system approach?*

    A novel OBU architecture has been introduced in chapter 4, designed based on the concept of a context-aware system and built utilising a new technique for detecting the behaviour of drivers in VANET. The architecture comprises three phases: the sensing, reasoning and application phases. This process is based on a context-aware system and is a self-organising process, in which sensing, reasoning and acting upon contextual information occurs instantly.

  - *How can we design an efficient driver behaviour detection model that can perform reasoning over time (temporal reasoning) and under uncertainty?*

    The target of detecting different styles of driving behaviours includes normal, fatigue, drunk and reckless was accomplished by introducing a novel DBN model in chapter 5. The proposed DBN model combined contextual information about the driver, the vehicle and the environment and performed reasoning over time and under uncertainty, in order to deduce the above styles of a driver's behaviour while driving effectively.

- *A study presenting how our proposed architecture can be applied in VANET, in order to detect the abnormal behaviours exhibited by drivers has to be conducted.*

  An illustration is given in chapter 4 to show how the proposed OBU architecture utilises the components of the OBU and how the newly added components interact with original components in order to detect the behaviour of a driver.

- *An analysis of why DBN was chosen from among other reasoning techniques, and a determination of the advantages of this technique must be performed.*

  An extensive study has been carried out in chapter 2 to inspect available reasoning techniques. As a result, the DBN was chosen from among these techniques, and the reasons for this were demonstrated in that chapter. A detailed description of this technique showing advantages and capabilities has been illustrated in chapter 3.

- *A study showing how our proposed driver behaviour detection model is different from others has to be carried out.*

  A thorough study was conducted in chapter 2 regarding currently available driver's behaviour detection systems in order to present their limitations and illustrate the differences between these systems and our proposed system.

7.3 Future work

Vehicular ad hoc networks remains an interesting research area in the field of wireless communications and networking attracting a large number of researchers. Many challenges in this field still need to be overcome; currently the focus of this is on proposing appropriate solutions to diminish the problems that prevent development in VANET.

In accordance with the work carried out in this thesis, future concerns needing to be taken into consideration are as follows:

- Develop a context interpreter unit in order to transfer the data collected by sensors into a machine processable format using one of the available modelling techniques (i.e. ontology).
- Extend the model structure by adding more variables (information and observable variables) in order to increase the detection accuracy of the model (i.e. driver's age, weather conditions, etc.).
- The focus of the research presented in this thesis is not on designing a high-fidelity driver behaviour detection model, but on introducing a theoretical model that is able to deduce the behaviour of the driver in a principled way.

Thus, more research is needed to enhance the parameters (conditional probability distributions) of the proposed DBN model.

- Technological devices such as sensors, high performance processors and memories that have a small size and high storage capacity are available in the market, this makes it possible for the proposed system to be implemented in a real vehicle. Therefore, more research is needed to examine the requirements for implementing the model (i.e. the size and format of the database required to store the information that gathered by sensors, the processor required to perform the behaviour detection algorithm, etc.).

- One important future direction will involve designing a corrective action algorithm, the aim of this algorithm would be to alert drivers and calculate corresponding corrective actions for other vehicles on the road in cases where a driver is driving in reckless, drunken or fatigued behaviour. Corrective actions can be disseminated using the wireless access technology provided by VANET to prevent accidents from happening. Generating corrective actions for other vehicles requires combining data about the current traffic situation. This data can be collected from TMC, digital road maps and adaptive HELLO messages that are periodically disseminated in VANET. This data includes information about the road structure, other vehicles' positions, other vehicles' direction, weather conditions, etc.

  However, the process of designing a corrective action algorithm might face several challenges as follows:

  1. **Timing:** The warning messages have to reach the other vehicles in time to allow them to implement corrective actions; e.g. so the driver can

divert his/her car to a safe lane or on the hard shoulder in a timely manner.

2. **Relevancy:** The warning message has to reach only those vehicles that are on the same road or travelling in the same direction, but not vehicles moving in the opposite direction.

3. **Security:** The warning message has to be authenticated and secured; e.g. so only the unreliable vehicle can generate them.

4. **Usability:** The algorithm has to generate practical corrective actions to allow other vehicles to avoid unreliable driver; e.g. the corrective action would be to move to the left side or the right side.

Bibliography

[1] S. Olariu and M. C. Weigle. *Vehicular Networks: From Theory to Practice*. Chapman & Hall/CRC, 2009.

[2] D. Jiang and L. Delgrossi. Ieee 802.11p: Towards an international standard for wireless access in vehicular environments. In *Proceedings of the IEEE Vehicular Technology Conference, VTC Spring 2008*, pages 2036 –2040, may 2008.

[3] H. Hartenstein and K.P. Laberteaux. A tutorial survey on vehicular ad hoc networks. *IEEE Communications Magazine*, 46(6):164 –171, june 2008.

[4] M. Baldauf, S. Dustdar, and F. Rosenberg. A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, 2(4):263–277, 2007.

[5] P. Fahy and S. Clarke. Cass–a middleware for mobile context-aware applications. In *Workshop on Context Awareness, MobiSys*, 2004.

[6] T. Gu, H. K. Pung, and D. Q. Zhang. A middleware for building context-aware mobile services. In *Proceedings of the IEEE Conference on Vehicular Technology (VTC)*, volume 5, pages 2656 – 2660, may 2004.

[7] H. Chen, T. Finin, and A. Joshi. An ontology for context-aware pervasive computing environments. *Knowledge Engineering Review, Special Issue on Ontologies for Distributed Systems*, 18:197–207, 2004.

[8] G. Biegel and V. Cahill. A framework for developing mobile, context-aware applications. In *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications*, pages 361 – 365, march 2004.

[9] A. K. Dey, G. D. Abowd, and D. Salber. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Hum.-Comput. Interact.*, 16(2):97–166, 2001.

[10] T. Hofer, W. Schwinger, M. Pichler, G. Leonhartsberger, J. Altmann, and W. Retschitzegger. Context-awareness on mobile devices - the hydrogen approach. In *Proceedings of the 36th Annual Hawaii International Conference on System Sciences*, pages 10 pp.+, 2003.

[11] K. B. Korb and A. E. Nicholson. *Bayesian Artificial Intelligence*. CRC Press, Inc., Boca Raton, FL, USA, 2003.

[12] H. Hartenstein and K. Laberteaux. *VANET: vehicular applications and inter-networking technologies*. Wiley Online Library, 2009.

[13] A. Buchenscheit, F. Schaub, F. Kargl, and M. Weber. A vanet-based emergency vehicle warning system. In *Proceedings of the 2009 IEEE Vehicular Networking Conference (VNC)*, pages 1 –8, oct. 2009.

[14] UK Department for Transport. Reported road casualties in great britain: Quarterly provisional estimates q1 2011. `http://www.dft.gov.uk/statistics/releases/road-accidents-and-safety-quarterly-estimates-q1-2011` Retrieved on 19.9.2011.

[15] Y. Qian and N. Moayeri. Design of secure and application-oriented vanets. In *Proceedings of the IEEE Vehicular Technology Conference*, pages 2794 –2799, may 2008.

[16] National Highway Traffic Safety Administration. Vehicle safety communications project task 3 final report, identify intelligent vehicle safety applications enabled by dsrc. Technical report, U.S. Department of Transportation, Tech. Rep. DOT HS 809 859, 2005.

[17] S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan. A comprehensive survey on vehicular ad hoc network. *Journal of Network and Computer Applications*, 2013. `http://dx.doi.org/10.1016/j.jnca.2013.02.036`.

[18] A. Rakotonirainy. Design of context-aware systems for vehicle using complex systems paradigms. In *Workshop on safety and context in conjunction with CONTEXT'05*, pages 464–475, Paris, 2005.

[19] UK Department for Transport. Reported road casualties in great britain: 2011 annual report. `https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/9277/rrcgb2011-04.pdf` Retrieved on 04.04.2013.

[20] A. Stefanowitsch. Cognitive linguistics as a cognitive science. *Bi-directionality in the cognitive sciences*, pages 295–310, 2011.

[21] `http://en.wikipedia.org/wiki/Cognitive_science` Retrieved on 04.04.2013.

[22] X. Li and Q. Ji. Active affective state detection and user assistance with dynamic bayesian networks. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 35(1):93 – 105, jan. 2005.

[23] R.W. Picard, E. Vyzas, and J. Healey. Toward machine emotional intelligence: analysis of affective physiological state. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(10):1175 –1191, oct 2001.

[24] G. Yang, Y. Lin, and P. Bhattacharya. A driver fatigue recognition model based on information fusion and dynamic bayesian network. *Information Sciences*, 180(10):1942–1954, May 2010.

[25] J. Sun, Y. Zhang, and K. He. Providing context-awareness in the smart car environment. In *Proceedings of the 2010 IEEE 10th International Conference on Computer and Information Technology (CIT)*, pages 13 –19, 2010.

[26] M.S. Devi and P.R. Bajaj. Driver fatigue detection based on eye tracking. In *Proceedings of the First International Conference on Emerging Trends in Engineering and Technology (ICETET'08)*, pages 649 –652, july 2008.

[27] J. Dai, J. Teng, X. Bai, Z. Shen, and D. Xuan. Mobile phone based drunk driving detection. In *Proceedings of the 4th International Conference on-NO PERMISSIONS Pervasive Computing Technologies for Healthcare (PervasiveHealth)*, pages 1 –8, march 2010.

[28] H. Ueno, M. Kaneda, and M. Tsukino. Development of drowsiness detection system. In *Proceedings of the Vehicle Navigation and Information Systems Conference*, pages 15 –20, aug-2 sep 1994.

[29] M. Miyaji, M. Danno, and K. Oguri. Analysis of driver behavior based on traffic incidents for driver monitor systems. In *Proceedings of the 2008 IEEE Intelligent Vehicles Symposium*, pages 930 –935, june 2008.

[30] M. Mauve, J. Widmer, and H. Hartenstein. A survey on position-based routing in mobile ad hoc networks. *IEEE Network*, 15(6):30 –39, nov.-dec. 2001.

[31] M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan. Context aware architecture for sending adaptive hello messages in vanet. In *Proceedings of the 4th ACM International Workshop on Context-Awareness for Self-Managing Systems*, pages 65–68. ACM, 2010.

[32] H. Moustafa and Y. Zhang. *Vehicular Networks: Techniques, Standards, and Applications*. CRC Press, Boston, MA, USA, 1st edition, 2009.

[33] D. Jiang, V. Taliwal, A. Meier, W. Holfelder, and R. Herrtwich. Design of 5.9 ghz dsrc-based vehicular safety communication. *IEEE Wireless Communications*, 13(5):36 –43, october 2006.

[34] Car 2 Car communication Consortium. Car 2 car communication consortium manifesto. Technical report, Car 2 Car communication Consortium, 2007.

[35] Trial-use standard for wireless access in vehicular environments (wave) - resource manager. *IEEE Std 1609.1-2006*, pages 1 –71, 18 2006.

[36] A. Festag, G. Noecker, M. Strassberger, A. Lübke, B. Bochow, M. Torrent-Moreno, S. Schnaufer, R. Eigner, C. Catrinescu, and J. Kunisch. NoW–network on wheels: Project objectives, technology and achievements. In *Proceedings of the 6th International Workshop on Intelligent Transportation*. TU Hamburg, 2008.

[37] M.L. Sichitiu and M. Kihl. Inter-vehicle communication systems: a survey. *IEEE Communications Surveys & Tutorials*, 10(2):88 –105, 2008.

[38] J. Luo and J. Hubaux. A survey of inter-vehicle communication. Technical report, School of Computer and Communication Sciences. EPFL, Lausanne, Switzerland, Tech. Rep. IC/2004/24, 2004.

[39] Y. Toor, P. Muhlethaler, and A. Laouiti. Vehicle ad hoc networks: applications and related technical issues. *IEEE Communications Surveys & Tutorials*, 10(3):74 –88, quarter 2008.

[40] J. Jakubiak and Y. Koucheryavy. State of the art and research challenges for vanets. In *Proceedings of the 5th IEEE Consumer Communications and Networking Conference, CCNC 2008*, pages 912 –916, jan. 2008.

[41] L. Wischhof, A. Ebner, and H. Rohling. Information dissemination in self-organizing intervehicle networks. *IEEE Transactions on Intelligent Transportation Systems*, 6(1):90 – 101, 2005.

[42] C. Bolchini, C. A. Curino, E. Quintarelli, F. A. Schreiber, and L. Tanca. A data-oriented survey of context models. *ACM Sigmod Record*, 36(4):19–26, 2007.

[43] B.N. Schilit and M.M. Theimer. Disseminating active map information to mobile hosts. *IEEE Network*, 8(5):22 –32, 1994.

[44] M. Brown. Supporting user mobility. In *Proceedings of IFIP World Conference on Mobile Communications*, pages 69–77. Chapman and Hall, 1996.

[45] A. K. Dey, G. D. Abowd, and A. Wood. Cyberdesk: a framework for providing self-integrating context-aware services. In *Proceedings of the 3rd international conference on Intelligent user interfaces*, pages 47–54. ACM, 1998.

[46] N. S. Ryan, J. Pascoe, and D. R. Morse. Enhanced reality fieldwork: the context-aware archaeological assistant. In *Computer Applications in Archaeology 1997*. Tempus Reparatum, 1998.

[47] R. Hull, P. Neaves, and J. Bedford-Roberts. Towards situated computing. In *Proceedings of the First International Symposium on Wearable Computers, 1997. Digest of Papers*, pages 146 –153, oct. 1997.

[48] D. Salber. Context-awareness and multimodality. In *Colloque sur la multimodalité, IMAG, Grenoble*, 2000.

[49] S. Loke. *Context-Aware Pervasive Systems: Architectures for a New Breed of Applications*. Auerbach Publications, 2006.

[50] B. Schilit, N. Adams, and R. Want. Context-aware computing applications. In *Proceedings of the Workshop on Mobile Computing Systems and Applications*, pages 85 –90, dec 1994.

[51] A.K. Dey and G.D. Abowd. Towards a better understanding of context and context-awareness. In *Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing HUC*, pages 304–307. Springer-Verlag, 1999.

[52] T. Strang and C. Linnhoff-Popien. A context modeling survey. In *Proceedings of the First International Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp*, 2004.

[53] J. Indulska and P. Sutton. Location management in pervasive systems. In *Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003 - Volume 21*, ACSW Frontiers '03, pages 143–151. Australian Computer Society, Inc., 2003.

[54] A. Schmidt and K. V. Laerhoven. How to build smart appliances? *IEEE Personal Communications*, 8(4):66 –71, 2001.

[55] M. Beigl, A. Krohn, T. Zimmer, and C. Decker. Typical sensors needed in ubiquitous and pervasive computing. In *Proceedings of the First International Workshop on Networked Sensing Systems (INSS)*, pages 153–158, 2004.

[56] K.K. Khedo. Context-aware systems for mobile and ubiquitous networks. In *Proceedings of the International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies*, page 123, april 2006.

[57] A. K. Dey. Understanding and using context. *Personal Ubiquitous computing*, 5(1):4–7, January 2001.

[58] N. Davies, K. Mitchell, K. Cheverst, and G. Blair. Developing a context sensitive tourist guide. In *Proceedings of the First workshop on human computer interaction with mobile devices, GIST Technical Report G*, volume 98, 1998.

[59] M. Brown. Supporting user mobility. In *Proceedings of IFIP World Conference on Mobile Communications*, pages 69–77. Chapman and Hall, 1996.

[60] D. Abowd, A. K. Dey, R. Orr, and J. Brotherton. Context-awareness in wearable and ubiquitous computing. *Virtual Reality*, 3(3):200–211, 1998.

[61] K. Henricksen, J. Indulska, and A. Rakotonirainy. Modeling context information in pervasive computing systems. In *Proceedings of the First International Conference on Pervasive Computing*, pages 167–180. Springer-Verlag, 2002.

[62] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni. A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, 6(2):161–180, 2010.

[63] B. A. Truong, Y. Lee, and S. Lee. Modeling and reasoning about uncertainty in context-aware systems. In *Proceedings of the IEEE International Conference on e-Business Engineering (ICEBE'05)*, pages 102 –109, oct. 2005.

[64] B. A. Truong, Y. Lee, and S. Lee. Modeling uncertainty in context-aware computing. In *Proceedings of the Fourth Annual ACIS International Conference on Computer and Information Science*, pages 676 – 681, 2005.

[65] Z. Zainol and K. Nakata. Generic context ontology modelling: A review and framework. In *Proceedings of the 2nd International Conference on Computer Technology and Development (ICCTD)*, pages 126 –130, nov. 2010.

[66] H. Berndt, J. Emmert, and K. Dietmayer. Continuous driver intention recognition with hidden markov models. In *Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems, ITSC 2008*, pages 1189 –1194, oct. 2008.

[67] V. Cherfaoui, J.-C. Burie, C. Royere, and D. Gruyer. Dealing with uncertainty in perception system for the characterization of driving situation. In *Proceedings of the 2000 IEEE Conference on Intelligent Transportation Systems*, pages 83 –88, 2000.

[68] G. Agamennoni, J.I. Nieto, and E.M. Nebot. A bayesian approach for driving behavior inference. In *Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 595 –600, june 2011.

[69] D. Yu, J. Ma, and J. Yang. Prediction of un-safety driving status in automotive vehicle based on hidden markov. *PRZEGLAD ELEKTROTECHNICZNY (Electrical Review)*, 88(1b):115–118, 2012.

[70] W. Dargie. The role of probabilistic schemes in multisensor context-awareness. In *Proceedings of the Fifth Annual IEEE International Conference on Perva-*

sive Computing and Communications Workshops (PerCom Workshops'07), pages 27 –32, march 2007.

[71] J. Hong, E. Suh, and S. Kim. Context-aware systems: A literature review and classification. *Expert Systems with Applications*, 36:8509–8522, 2009.

[72] Y. Vanrompay and Y. Berbers. Opinion nets for reasoning with uncertain context information. *Contexts and Ontologies Representation and Reasoning*, page 71, 2007.

[73] H. Wu. *Sensor Fusion for Context-Aware Computing Using Dempster-Shafer Theory*. PhD thesis, Robotics Institute, Carnegie Mellon University, December 2003.

[74] R. R. Yager and L. A. Zadeh, editors. *An Introduction to Fuzzy Logic Applications in Intelligent Systems*. Kluwer Academic Publishers, Norwell, MA, USA, 1992.

[75] A. Ranganathan, J. Al-Muhtadi, and R.H. Campbell. Reasoning about uncertain contexts in pervasive computing environments. *IEEE Pervasive Computing*, 3(2):62 –70, april-june 2004.

[76] Q. Ji, P. Lan, and C. Looney. A probabilistic framework for modeling and real-time monitoring human fatigue. *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans*, 36(5):862 –875, sept. 2006.

[77] S. S. Haykin. *Neural networks: a comprehensive foundation*. Prentice Hall Englewood Cliffs, NJ, 2007.

[78] M. T. Hagan, H. B. Demuth, and M. Beale. *Neural network design*. PWS Publishing Co., Boston, MA, USA, 1996.

[79] C. M. Bishop. *Neural Networks for Pattern Recognition.* Clarendon Press, Oxford, January 1995.

[80] J. Petzold, A. Pietzowski, F. Faruk Bagci, W. Trumler, and T. Ungerer. Prediction of indoor movements using bayesian networks. In *Proceedings of Location- and Context-Awareness (LoCA)*, 2005.

[81] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257 –286, feb 1989.

[82] Z. Ghahramani. An introduction to hidden Markov models and Bayesian networks. *International Journal of Pattern Recognition and Artificial Intelligence*, pages 9–42, 2001.

[83] K.P. Murphy. *Dynamic bayesian networks: representation, inference and learning.* PhD thesis, University of California, 2002.

[84] G. Rigas, C.D. Katsis, P. Bougia, and D.I. Fotiadis. A reasoning-based framework for car driver's stress prediction. In *Proceedings of the 16th Mediterranean Conference on Control and Automation*, pages 627 –632, june 2008.

[85] V. Mihajlovic and M. Petkovic. Dynamic Bayesian Networks: A State of the Art. *CTIT technical reports series*, 2001.

[86] K. Livescu, O. Cetin, M. Hasegawa-Johnson, S. King, C. Bartels, N. Borges, A. Kantor, P. Lal, L. Yung, A. Bezman, S. Dawson-Haggerty, B. Woods, J. Frankel, M. Magami-Doss, and K. Saenko. Articulatory feature-based methods for acoustic and audio-visual speech recognition: Summary from the 2006 jhu summer workshop. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, volume 4, pages IV–621 –IV–624, april 2007.

[87] V. Pavlovic, J.M. Rehg, Tat-Jen Cham, and K.P. Murphy. A dynamic bayesian network approach to figure tracking using learned dynamic models. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 94 –101 vol.1, 1999.

[88] Y. Zhang, Q. Ji, and C.G. Looney. Active information fusion for decision making under uncertainty. In *Proceedings of the Fifth International Conference on Information Fusion*, volume 1, pages 643 –650 vol.1, 2002.

[89] R. EL Kaliouby and P. Robinson. Real-time inference of complex mental states from facial expressions and head gestures. In *Real-Time Vision for Human-Computer Interaction*, pages 181–200. Springer US, 2005.

[90] H. Gu and Q. Ji. Facial event classification with task oriented dynamic bayesian network. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 870 –875, june 2004.

[91] N. Oliver and E. Horvitz. A comparison of hmms and dynamic bayesian networks for recognizing office activities. In *in User Modeling 2005*, pages 199–209. Springer, 2005.

[92] I. Bae and S. Olariu. A tolerant context-aware driver assistance system for vanets-based smart cars. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM 2010)*, pages 1–5, dec. 2010.

[93] Z. Zhu and Q. Ji. Real time and non-intrusive driver fatigue monitoring. In *Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems*, pages 657 – 662, oct. 2004.

[94] Z. Zhang and J. Zhan. Driver fatigue detection based intelligent vehicle control. In *Proceedings of the 18th International Conference on Pattern Recognition, ICPR 2006*, volume 2, pages 1262 –1265, 0-0 2006.

[95] M. Sakairi and M. Togami. Use of water cluster detector for preventing drunk and drowsy driving. In *Proceedings of the 2010 IEEE Sensors Conference*, pages 141 –144, nov. 2010.

[96] R. Garg, V. Gupta, and V. Agrawal. A drowsy driver detection and security system. In *Proceedings of the International Conference on Ultra Modern Telecommunications Workshops (ICUMT '09)*, pages 1 –8, oct. 2009.

[97] D. Sandberg and M. Wahde. Particle swarm optimization of feedforward neural networks for the detection of drowsy driving. In *Proceedings of the IEEE International Joint Conference on Neural Networks, IJCNN 2008. (IEEE World Congress on Computational Intelligence)*, pages 788 –793, june 2008.

[98] H. Park, S. Oh, and M. Hahn. Drowsy driving detection based on human pulse wave by photoplethysmography signal processing. In *Proceedings of the 3rd International Universal Communication Symposium*, pages 89–92. ACM, 2009.

[99] N. Oliver and A.P. Pentland. Driver behavior recognition and prediction in a smartcar. In *Proceedings of the SPIE Aerosense-Enhanced and Synthetic Vision*, volume 4023, pages 280–290, 2000.

[100] N. Oliver and A.P. Pentland. Graphical models for driver behavior recognition in a smartcar. In *Proceedings of the IEEE Intelligent Vehicles Symposium, 2000*, pages 7 –12, 2000.

[101] D. Mitrovic. Reliable method for driving events recognition. *IEEE Transactions on Intelligent Transportation Systems*, 6(2):198 – 205, june 2005.

[102] M. Helander. Applicability of drivers' electrodermal response to the design of the traffic environment. *Journal of applied Psychology*, 63(4):481, 1978.

[103] A. Pentland and A. Liu. Modeling and prediction of human behavior. *Neural computation*, 11(1):229–242, January 1999.

[104] Y.L. Murphey, R. Milton, and L. Kiliaris. Driver's style classification using jerk analysis. In *IEEE Workshop on Computational Intelligence in Vehicles and Vehicular Systems, (CIVVS'09)*, pages 23 –28, 30 2009-april 2 2009.

[105] The National Highway Traffic Safety Administration. Define aggressive driving. `http://www.nhtsa.gov/people/injury/enforce/aggressdrivers/aggenforce/define.html` Retrieved on 04.8.2012.

[106] P. Jackson, C. Hilditch, A. Holmes, N. Reed, N. Merat, and L. Smith. Fatigue and road safety: A critical analysis of recent evidence. Technical report, Road Safety Web Publication, 2011.

[107] T. Bar, D. Nienhuser, R. Kohlhaas, and J.M. Zollner. Probabilistic driving style determination by means of a situation based analysis of the vehicle data. In *proceedings of the 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1698 –1703, 2011.

[108] H. Singh, J.S. Bhatia, and J. Kaur. Eye tracking based driver fatigue monitoring and warning system. In *Proceedings of the 2010 India International Conference on Power Electronics (IICPE)*, pages 1 –6, jan. 2011.

[109] I.G. Daza, N. Hernandez, L.M. Bergasa, I. Parra, J.J. Yebes, M. Gavilan, R. Quintero, D.F. Llorca, and M.A. Sotelo. Drowsiness monitoring based on driver and driving data fusion. In *Proceedings of the 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1199 –1204, oct. 2011.

[110] L.M. Bergasa, J. Nuevo, M.A. Sotelo, R. Barea, and M.E. Lopez. Real-time system for monitoring driver vigilance. *IEEE Transactions on Intelligent Transportation Systems*, 7(1):63 –77, march 2006.

[111] T. Wakita, K. Ozawa, C. Miyajima, K. Igarashi, K. Itou, K. Takeda, and F. Itakura. Driver identification using driving behavior signals. In *Proceedings of the 2005 IEEE Conference on Intelligent Transportation Systems*, pages 396 – 401, sept. 2005.

[112] Transport Accident Commission. Reducing fatigue - a case study. `http://www.tac.vic.gov.au` Retrieved on 23.7.2012.

[113] D. Dawson, K. Reid, et al. Fatigue, alcohol and performance impairment. *Nature*, 388(6639):235, 1997.

[114] F. V. Jensen and T. D. Nielsen. *Bayesian Networks and Decision Graphs*. Springer Publishing Company, 2nd edition, 2007.

[115] F. V. Jensen. *Introduction to Bayesian Networks*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1st edition, 1996.

[116] H. Guo and W. Hsu. A survey of algorithms for real-time bayesian network inference. In *the joint AAAI-02/KDD-02/UAI-02 workshop on Real-Time Decision Support and Diagnosis Systems*, 2002.

[117] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[118] V. I. Pavlovic. *Dynamic Bayesian Networks for Information Fusion with Application to Human-Computer Interfaces*. PhD thesis, University of Illinois at Urbana-Champaign, 1999.

[119] J. Hulst. Modeling physiological processes with dynamic Bayesian networks. Master's thesis, Delft University of Technology, 2006.

[120] S. J. Russell, P. Norvig, J. F. Candy, J. M. Malik, and D. D. Edwards. *Artificial intelligence: a modern approach*. Prentice hall Englewood Cliffs, Upper Saddle River, NJ, USA, 1995.

[121] A. Singhal and C. Brown. Dynamic bayes net approach to multimodal sensor fusion. In *Proceedings of the SPIE - The International Society for Optical Engineering*, pages 2–10, 1997.

[122] R. Sterritt, A.H. Marshall, C.M. Shapcott, and S.I. McClean. Exploring dynamic bayesian belief networks for intelligent fault management systems. In *proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, volume 5, pages 3646 –3652, 2000.

[123] E. Charniak. Bayesian networks without tears. *AI MAGAZINE*, 12(4):50–63, 1991.

[124] K. Murphy, S. Mian, et al. Modelling gene expression data using dynamic bayesian networks. Technical report, Computer Science Division, University of California, Berkeley, CA, 1999.

[125] T. A. Stephenson. An introduction to bayesian network theory and usage. *research report, idiap-rr 00-03, Dalle Molle Institute for Perceptual Artificial Intelligence*, 2000.

[126] J. Pearl. Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, 29(3):241–288, September 1986.

[127] W. Qing and Y. WeiWei. A driver abnormality recondition model based on dynamic bayesian network for ubiquitous computing. In *Proceedings of the*

3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), volume 1, pages V1–320 –V1–324, aug. 2010.

[128] Bayes net toolbox for matlab. `https://code.google.com/p/bnt/` Retrieved on 25.11.2012.

[129] Netica. `http://www.norsys.com/` Retrieved on 25.11.2012.

[130] Analytica, lumina decision systems. `http://www.lumina.com/` Retrieved on 25.11.2012.

[131] J. Bilmes and G. Zweig. The graphical models toolkit: An open source software system for speech and time-series processing. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 4, pages IV–3916 –IV–3919, may 2002.

[132] Bayesialab. `http://www.bayesia.com` Retrieved on 25.11.2012.

[133] Genie 2.0, decision systems laboratory. `http://genie.sis.pitt.edu/` Retrieved on 25.11.2012.

[134] M. M. Al-Doori. *Directional Routing Techniques in VANET*. PhD thesis, De Montfort University, 2011.

[135] S. Tokoro, K. Moriizumi, T. Kawasaki, T. Nagao, K. Abe, and K. Fujita. Sensor fusion system for pre-crash safety system. In *proceedings of the 2004 IEEE Intelligent Vehicles Symposium*, pages 945 – 950, june 2004.

[136] K. Kobayashi, K.C. Cheok, and K. Watanabe. Estimation of absolute vehicle speed using fuzzy logic rule-based kalman filter. In *Proceedings of the 1995 American Control Conference*, volume 5, pages 3086 –3090 vol.5, jun 1995.

[137] D.F. Dinges, M.M. Mallis, G. Maislin, IV Powell, et al. Evaluation of techniques for ocular measurement as an index of fatigue and the basis for alertness management. Technical report, National Highway Traffic Safety Administration, Tech. Rep. DOT HS-808 762, 1998.

[138] W.W. Wierwille, SS Wreggit, CL Kirn, LA Ellsworth, and R.J. Fairbanks. Research on vehicle-based driver status/performance monitoring; development, validation, and refinement of algorithms for detection of driver drowsiness. final report. Technical report, National Highway Traffic Safety Administration, Tech. Rep. DOT HS-808 247, 1994.

[139] E. L. Co, K. B. Gregory, J. M. Johnson, and M. R. Rosekind. Crew factors in flight operations xi: A survey of fatigue factors in regional airline operations. Technical report, NASA, Moffett Field, California, NASA Ames Research Center, 1999.

[140] D.H. Harris. Visual detection of driving while intoxicated. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 22(6):725–732, 1980.

[141] Z.Y. Rawashdeh and S.M. Mahmud. Intersection collision avoidance system architecture. In *Proceedings of the 5th IEEE Consumer Communications and Networking Conference, CCNC*, pages 493 –494, jan. 2008.

Appendix A

VANET Applications

This Appendix contains a full description of VANET applications, which comprises safety and comfort applications.

V2V and V2I communications allow the development of a large number of applications and can provide a wide range of information to drivers and travellers. Integrating on-board devices with the network interface, different types of sensors and GPS receivers, grant vehicles the ability to collect, process and disseminate information about itself and its environment to other vehicles in close proximity to it. That has led to enhancement of road safety and the provision of passenger comfort [40, 41, 39].

VANET applications are classified according to their primary purpose into:

1. **Comfort/Entertainment applications:** This category of applications is also referred to as non-safety applications, and aim to improve drivers and passengers comfort levels (make the journey more pleasant) and enhance traffic efficiency. They can provide drivers or passengers with weather and traffic information and detail the location of the nearest restaurant, petrol station or

hotel and their prices. Passengers can play online games, access the internet and send or receive instant messages while the vehicle is connected to the infrastructure network.[40, 41, 32, 39].

2. **Safety applications:** These applications use the wireless communication between vehicles or between vehicles and infrastructure, in order to improve road safety and avoid accidents; the intention being to save people's lives and provide a clean environment.

Applying wireless communication technology in vehicles in order to communicate with other vehicles, or with the infrastructure, enables a wide range of applications and leads to an increase in the road safety level. According to [16], safety applications using V2V communication or V2I communication, or both can be categorised as fellows:

1. Intersection collision avoidance
2. Public safety
3. Sign extension
4. Vehicle diagnostics and maintenance
5. Information from other vehicle

1. **Intersection collision avoidance:** Improving intersection collision avoidance systems will lead to the avoidance of many road accidents; this system is based on I2V or V2I communication. The sensors at infrastructure gather, process and analyse the information from the vehicles moving close to the intersection [141], depending on the analysis of data; if there is a probability of an accident or a hazardous situation, a warning message is sent to the vehicles

in the intersection area to warn them about the possibility of the accident so that they can take appropriate action to avoid it.

There are many applications that fall under intersection collision avoidance systems umbrella, all of them use a minimum frequency of 10Hz, relying on I2V communication and using periodic safety messages with a communication range of 200-300m, these applications are as follows:

- Warning about violating traffic signal: This application is designed to send a warning messages to vehicles to warn the drivers about a dangerous situation (accident) that would occur happen if the vehicle does not stop; when the traffic signal is running and indicating a stop, the message that is sent depends on several factors; such as traffic status, timing, the vehicle's speed, the vehicle's position and the road surface [141, 1].
- Warning about violating stop sign: This application is designed to send a warning messages to a vehicles to warn the driver about the current distance between the vehicle and the stop sign and the speed required to prevent the necessity of hard breaking, so as to prevent the vehicle from violating of a stop sign, which will then lead to the prevention of a hazardous situation [1, 16].
- Left turn assistant: The aim of this application is to help the driver to make a left turn at an intersection in a safe way, as shown in figure A.1 by sending the information collected about the traffic status on the opposite side of the road to the vehicle wanting to make the left turn. This information is collected by road sensors or by in-vehicle sensors and is then sent to vehicles, either directly from roadside infrastructure, or by the vehicles requesting the information via in-vehicle systems to allow the driver to decide whether turn left or not.

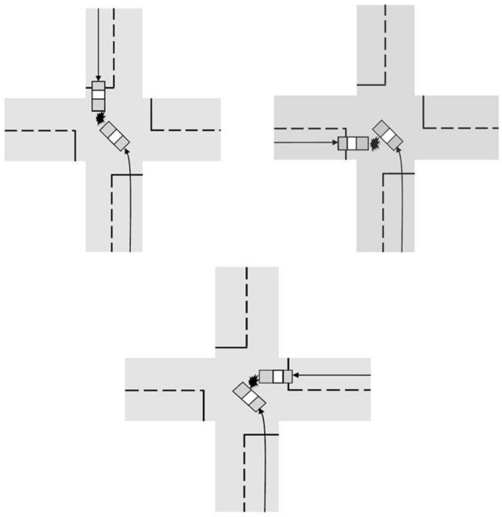

Figure A.1: Left turn driver assistant to avoid accidents [1]

- Stop sign movement assistant: The aim of this application is to warn drivers about hazardous situations that may occur if thier vehicles pass by an intersection. This is achieved by collecting data from road sensors and in-vehicle sensors and sending this information to the vehicles trying to pass the intersection; this means the driver will know if there are other vehicles approaching the intersection at the same time, and should lead to the prevention of accidents at intersections. This application relies on both V2V and V2I types of communications.

- Intersection collision warning: This application collects the information about the road intersection via sensors and in-vehicles sensors and analyses this information, if there is a probability of an accident occurring the system will generate and send a warning messages to all the vehicles approaching the intersection. The data gathered by the sensors includes vehicle velocity, position, acceleration and road surface information.

- Warning about blind merge detection: This application aims to prevent a collision at the merge point where the visibility is poor. The system will alert vehicles trying to merge if there is an unsafe situation, at the same time it will warn the remaining vehicles on the road. The system collects and processes the data at the intersection and if there is an unsafe situation detected it will generate a warning messages to vehicles.

- Pedestrian crossing information designated intersection: The main goal of this application is to warn drivers if there is a pedestrian crossing the road, by collecting information about the walkers via sensors installed in the walk side. After collecting this information the sensors can send it to the system; meanwhile at the same time the system has the ability to collect data if somebody has pressed the walk button located at the crossing signal, as shown in figureA.2, after the system has processed all the data and there is a possibility of collision found it will send a warning messages to the vehicles approaching the walk side area.

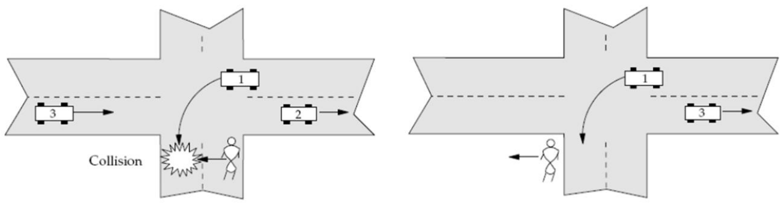

Figure A.2: Pedestrian crossing warning [12]

2. **Public safety:** Public safety applications aim to aid drivers when an accident has occurred and to support emergency teams by minimising their travel time and provide their services, most of the emergency vehicles response time are wasted in their way to the destination. The average time for the emergency vehicle to response is 6-7 minutes, while in some cases this can be as much as 25 minutes.

   The frequency used by this applications is 1Hz relying on I2V communication, V2V communication or both and useing event-driven safety messages with a communication range of 300-1000m[1].

The most familiar applications within this category are:

- Approaching emergency vehicle warning: This system is designed to satisfy the requirements to provide a clear road to allow emergency vehicles to reach their destinations without waiting in traffic, as shown in figure A.3, the system accomplished this task by disseminating alert messages relying on one way V2V communication between vehicles travelling on the same route in an attempt to clear the road clear for the emergency vehicle, this message contains information about the emergency vehicle's velocity, direction, lane information and path.

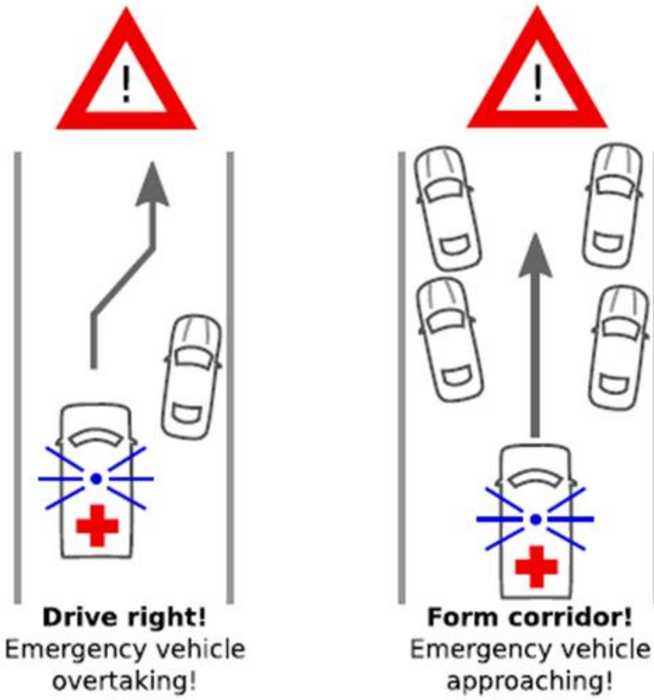

Figure A.3: Approaching emergency vehicle warning [13]

- Emergency vehicle signal preemption: Available infrastructures at each intersection support emergency vehicles by sending messages to all traffic lights on the route to the destination using V2I communication. this sets all the lights to green when the emergency vehicle arrives at the traffic signals, minimising the response time for the emergency vehicle, and reducing the possibility of an accident occurring involving it.

- SOS Services: The SOS system works in conditions where a life threatening situation occurs; by sending SOS messages in the case of accidents. The SOS signal can be trigged either automatically by the system or a driver. Both types of communication (V2V and V2I) can be used to serve the system, depending on the situation for instance, the signal could be sent to the nearest infrastructure point directly, alternatively it depends upon the vehicles in range repeating the signal and delivering it to the nearest infrastructure.
- Post crash warning: This application aims to prevent potential accidents before they happen; a vehicle which is disabled because of foggy weather or due to an accident sends a warning messages to other vehicles coming travelling in the same direction, or the opposite direction by using both types of communications (V2I and V2V) to inform them about its location, heading, direction and status information.

3. **Sign Extension:** The main goal of this application is to alert inattentive drivers to signs that are placed on the side of the road while driving in order to prevent accidents.
   Most of the sign extension applications use a minimum frequency of 1Hz relying on I2V communication and the use of periodic safety messages with a communication range of 100-500m, these applications can be classified as follows:

   - In-vehicle signage: This application relies on the RSU being fixed in a specific area; for example in a school zone, hospital zone or animal passing area to send alert messages to vehicles approaching the zone.

- Curve Speed Warning: This application relies on the RSU being fixed before the curve to disseminate messages to approaching vehicles alerting them about the location of the curve, the speed required to negotiate the curve safely and the road conditions.
- Low Parking Structure and Bridge Warning: This application is designed to alert the driver regarding the minimum height of the park they are trying to enter, by sending a warning messages to the vehicle via an RSU installed close to the parking facility, then the OBU can determine whether it is safe to enter the structure.
- Low Bridge Warning: This application is designed to alert the driver to the height of the bridge they are trying to pass under, by sending warning messages to the vehicle via an RSU installed close to the bridge, then the OBU can determine whether there is sufficient clearance.
- Wrong Way Driver Warning: This system is designed to alert a vehicle if it is travelling in the wrong direction. By using V2V communication a vehicle travelling the wrong way can alert the other vehicles around it via warning messages to prevent accidents occurring.
- Work zone warning: This system relies on the RSU installed closed to the work zone in order to warn approaching vehicles about the work zone area, sending warning messages using I2V communication.
- In-Vehicle Amber Alert: This system depends on I2V communication and send Amber warning messages (America's missing: Broadcast emergency response) to vehicles; this messages is disseminated when the police confirm that there is a vehicle involved in the crime and it is issued to all vehicles in the area, except for the suspect vehicle.

4. **Vehicle diagnostics and maintenance:** This application aims to send notification messages to vehicles in order to remind drivers about safety defects and that it is time for the vehicle to receive maintenance.

   These applications rely on I2V communication and use event-driven safety messages with a communication range of 400m, these applications can be classified into:

   - Safety recall notice: A message sent to vehicles to remind the drivers when a recall is issued.
   - Just-in-time repair notification: In this system if there is a fault within the vehicles, the OBU will send a messages to the infrastructure using V2I communication, the vehicles will receive a reply message containing instructions from the support centre to tackle this problem using I2V communication.

5. **Information from other vehicles:** This type of application relies on V2V communication, I2V communication or both to perform applications functions by a frequency of 2-50Hz and event-driven or periodic messages requiring a communication range of 50-400m.

   Information from other vehicles applications can be classified as follows:

   - Cooperative forward collision warning: This system accomplishes the goals necessary to assists a vehicle in avoiding becoming involved in an accident with the vehicle travelling ahead of it. The system uses V2V communication with a multi hop technique in order to send warning messages to a driver about the situation. These messages include information (position, direction, velocity and acceleration), each vehicle processes this information after receiving it to decide on the danger level then forward it to other vehicles.

- Vehicle-based road condition warning: This application is based on V2V communication; the vehicle collects sufficient information about the road status via the vehicle's sensors, after collecting road information the in-vehicle unit processed this data to determine the road situation in order to initiate a warning to the driver or send a warning messages to other vehicles.

- Emergency electronic brake lights (EEBL): This system aims to warn other vehicles on the road if there is going to be a need for sudden hard breaking or in case of foggy weather where visibility has become very poor and break lights are not bright enough to be recognised by other drivers; as shown in figure A.4, by using only V2V communication vehicles can disseminate the message to other vehicles on the road and alert them to the need for hard breaking ahead.

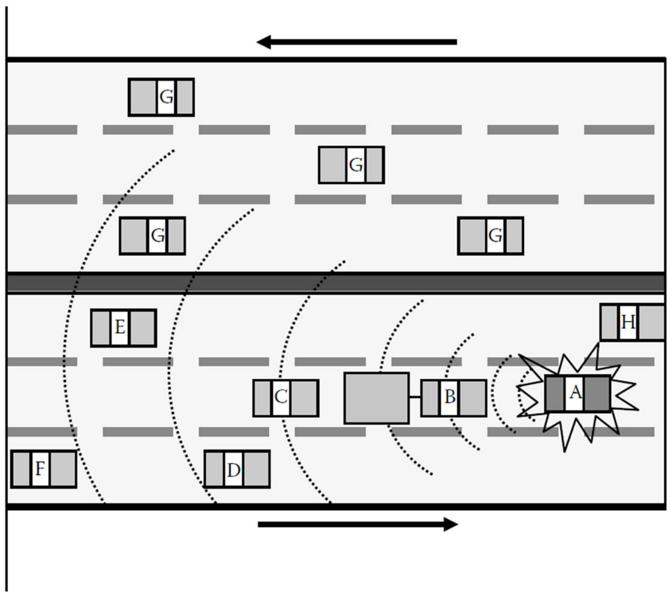

Figure A.4: Emergency electronic break light system [1]

- Lane change warning: As shown in figure A.5, this application is designed to avoid crashes that might occur due to unsafe lane changing decisions being made by the driver. The system collects data about the vehicle and the surrounding vehicles; such as speed, direction and vehicle position, and when the driver decides to change his/her current lane the system processes the data collected and evaluate whether the decision will lead to an accident. The system then issues a warning to alert the driver about the potentially dangerous situation and uses V2V communication to alert other vehicles.

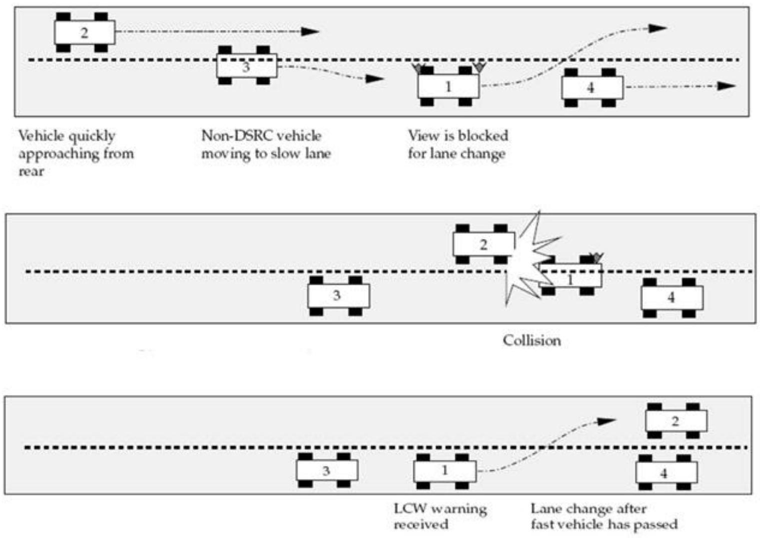

Figure A.5: Lane change warning system [12]

- Blind spot warning: This application alerts the driver if he/she decides to change lane and there is a vehicle in the blind spot; it uses V2V communication to send a warning messages to other vehicles on the road.
- Highway merge assistant: This application prevents accidents from occurring when a vehicle is attempting to merge on the highway. If the vehicles is moving on a ramp or there are other vehicles in the vehicle's blind spot then the system start to sends a warning messages to other

vehicles informing them about the speed, position and direction of the vehicle in order to take appropriate action to prevent the accident.

- Visibility enhancer: Bad weather conditions such as fog, rain and snow lead to poor visibility for the drivers, and this system assists the driver by sensing bad weather conditions and warn the driver about this conditions and warning the driver and other vehicles on the road about them..

- Cooperative collision warning: The main goal of this application is to warn the driver about any accidents that have been predicted, by relying on V2V communication. The system exchange messages between vehicles containing information about surrounding vehicles; describing their direction, position, acceleration, yaw-rate and velocity. The in-vehicle unit processes this information in combination with information about the vehicle itself; if there is a possibility of an accident the system warns the driver.

- Cooperative adaptive cruise control: This application adjusts the speed of the vehicle depending on the speed of the vehicles ahead and those behind; it uses V2V communication to exchange messages between the vehicles detailing their position, direction, speed, yaw-rate and acceleration. Meanwhile, the system utilises I2V communication to acquire the speed limit of the road.

- Road condition warning: This system is concerned with alerting vehicles about poor road conditions caused by ice or other substances causing the road to be slippery, in order to prevent accidents. The road side sensors on the system collect data regarding the road to determine if there are any unsafe conditions, then disseminates warning messages to vehicles to adjust, suggesting they adjust their speed to avoid accidents.

- Pre-crash sensing: The main goal of this system is to predict a situation in which an accident is about to happen, information can be collected from sensors, and additional data that can be acquired from other vehicles using V2V communication, this system increase the level of safety for peoples inside vehicles.
- Highway/ rail collision warning: This application aims to prevent vehicles from becoming involved in accidents with trains, by using RSUs placed at intersections to notify approaching vehicles to prevent them colliding with trains; another method is to receive messages directly from the train to warn vehicles to take corrective action.
- Vehicle-to-Vehicle Road feature notification: This system is designed to collect information about the road infrastructure using V2V communication and disseminating this information to other vehicles on the road to be used by other VANET applications.
- Cooperative Vehicle-Highway Automation System: This system controls the velocity and position of vehicles to travel on the highway as a platoon, relying on V2V communication and using V2I communication. The system collects information about the vehicle and merges the data with information regarding its position and map data in order to control the vehicle's movements and enhance the traffic flow on the highway.

Appendix B

Using GeNIe 2.0 to implement our DBN model

This Appendix contains a full description of how we use GeNIe 2.0 to implement our DBN model.

GeNIe version 2.0 [133, 11, 119] is a development tool for implementing DBNs. It was developed at the University of Pittsburgh, and it supports both BN and DBN implementation by providing temporal reasoning as well as supporting many inference algorithms such as the polytree algorithm, which we used to infer the hypothesis node in our network. Figure B.1 depicts the working environment of GeNIe software. The following sections will illustrate how we have used the GeNIe software to implement our DBN.

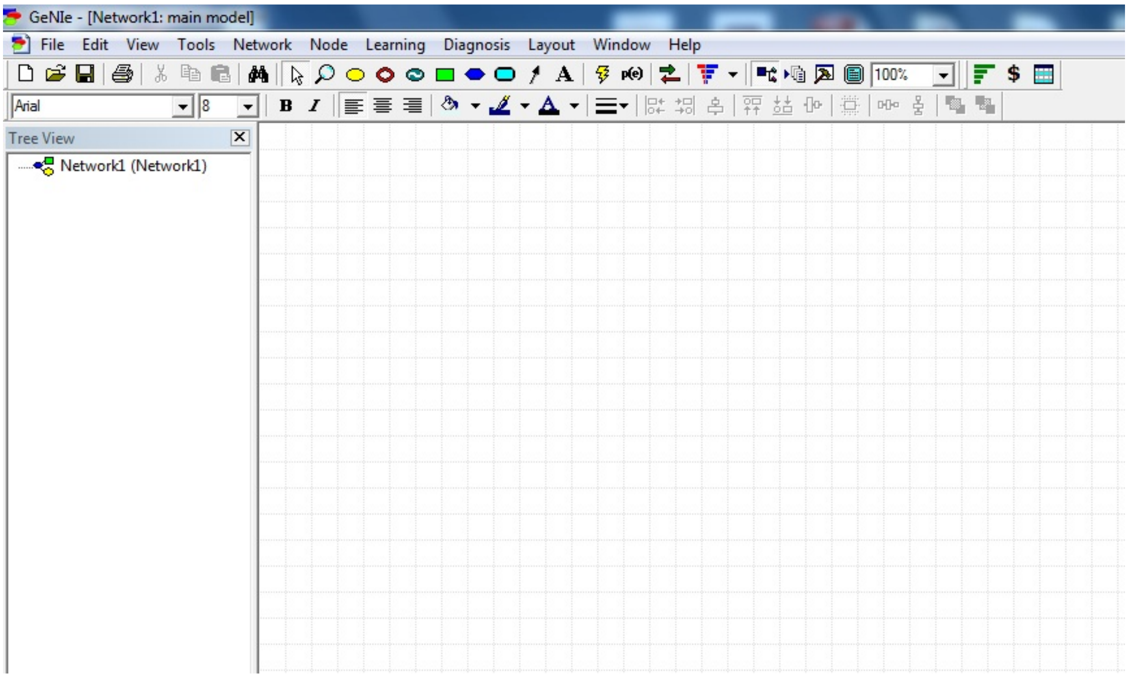

Figure B.1: Overview of GeNIe 2.0 environment

B.1 Creating network nodes

This section presents the steps for creating the nodes of the networks in GeNIe. There are two types of node, which are static and temporal nodes. In this thesis, we only use the temporal nodes which change their values or develop over time. Before inserting the temporal nodes, the temporal plate has to be inserted in order to use this type of nodes. As shown in Figure B.2, adding the plate will divide the working environment into four parts that are as follows:

- **Temporal plate:** The temporal nodes of the network have to be inserted in this part. It contains the number of time steps (time slices) which can be changed by double-clicking on it to indicate the number of time slices involved in the inference process.

- **Contemporals:** The nodes that do not change their values over time and remain steady (i.e. the sex of the driver) have to be inserted in this part. These nodes are also called static nodes.
- **Initial conditions:** Inserting the nodes required only in the first time slice during the inference process.
- **Terminal conditions:** Inserting the nodes required only in the last time slice during the inference process.

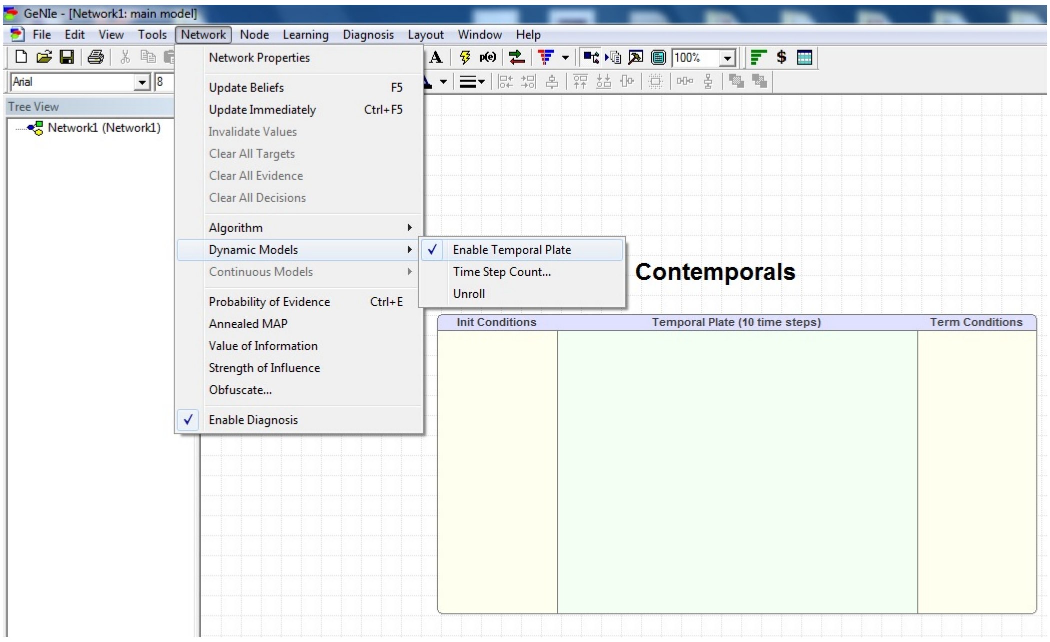

Figure B.2: Adding the temporal plate in GeNIe

Having inserted the temporal plate, the next step is to insert the nodes of a DBN. Inserting a node can be done by clicking on *Tools* → *Chance* then clicking inside the temporal plate's temporal plate, as shown in Figure B.3. After inserting the node, the node's name, type and states have to be specified via the *Identifier*, the *Diagnostic type* and the *State name* respectively as shown in Figure B.4.

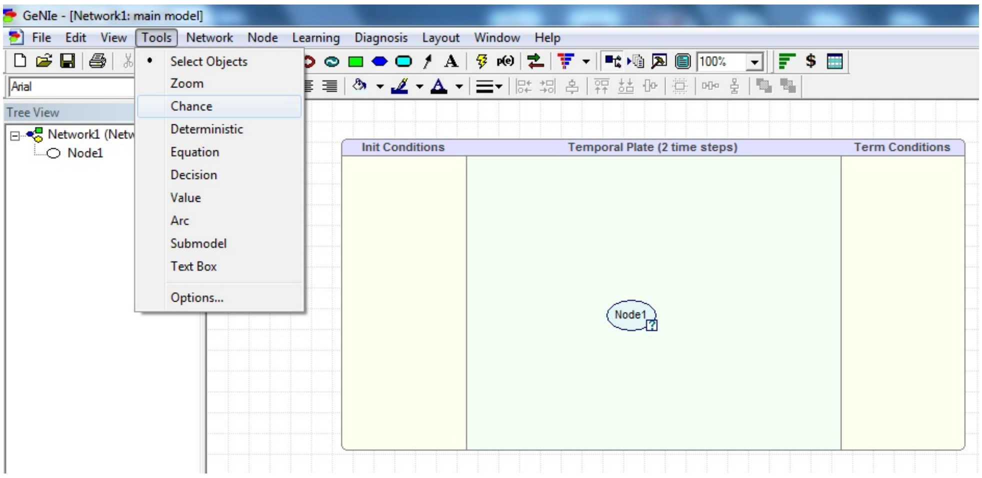

Figure B.3: Inserting a node in GeNIe

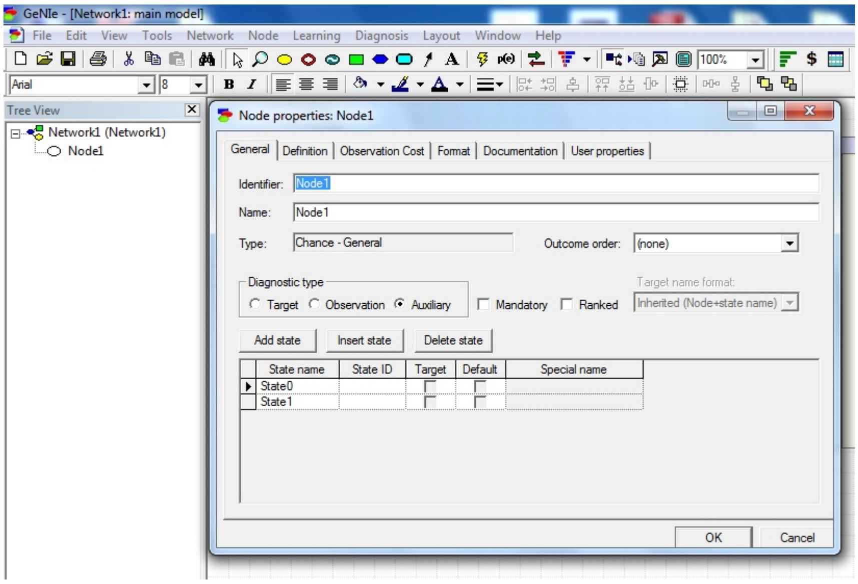

Figure B.4: Specifying the node's properties in GeNIe

B.2 Network arcs and conditional probability tables

Network arcs represent the causal relationships between random variables. This section shows the steps for adding network arcs between random variables. Two types of arcs exist in GeNIe, which are normal arc, which denotes the effect of one variable to another in the same time slice; and the temporal arc, which represent the relation between the variables within different time slices. Inserting a network arc can be done by clicking on *Tools* → *Arc* then specifying the parent node and the child node, which can be the same node. After selecting the child node, a menu will appear to allow the network designer to specify the type of the arc, either normal or temporal, with different orders. Setting the order to a specific number means specifying the order of a DBN, for example, choosing *order 1* indicates that the node will depend on its past on the previous time slice only (first order Markov process). As mentioned earlier in this chapter, the hypothesis node in our DBN only depends on its immediate past. Therefore, we have set the temporal arc of the hypothesis node in our network to *order 1*. Inserting normal and temporal arcs can be carried out as depicted in Figure B.5.

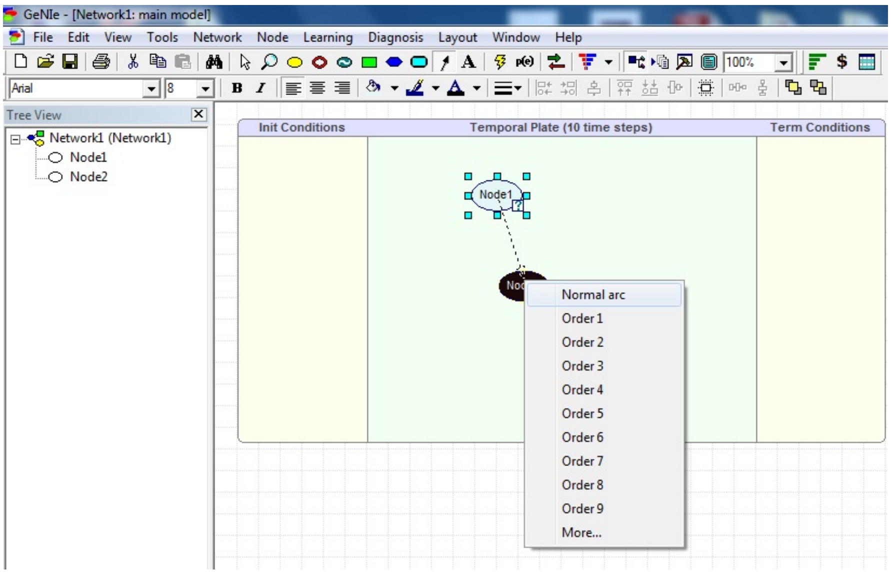

Figure B.5: Adding network arcs in GeNIe

The step after drawing the network arcs is filling the conditional probability tables for all nodes in the network. Setting the CPT for any node is done by double-clicking on any node then selecting the *definition* tab and entering the conditional probability distributions for the node as shown in Figure B.6.

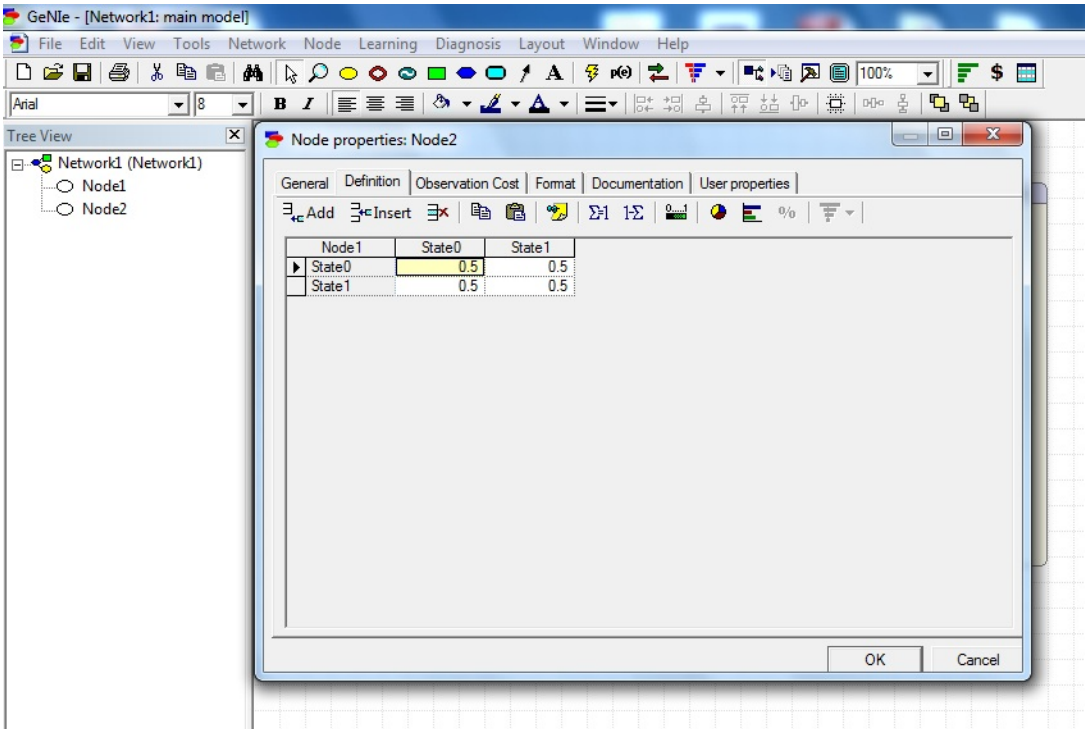

Figure B.6: Filling the conditional probability tables in GeNIe

The structure of our DBN, which is implemented using GeNIe version 2.0 software, is shown in Figure B.7.

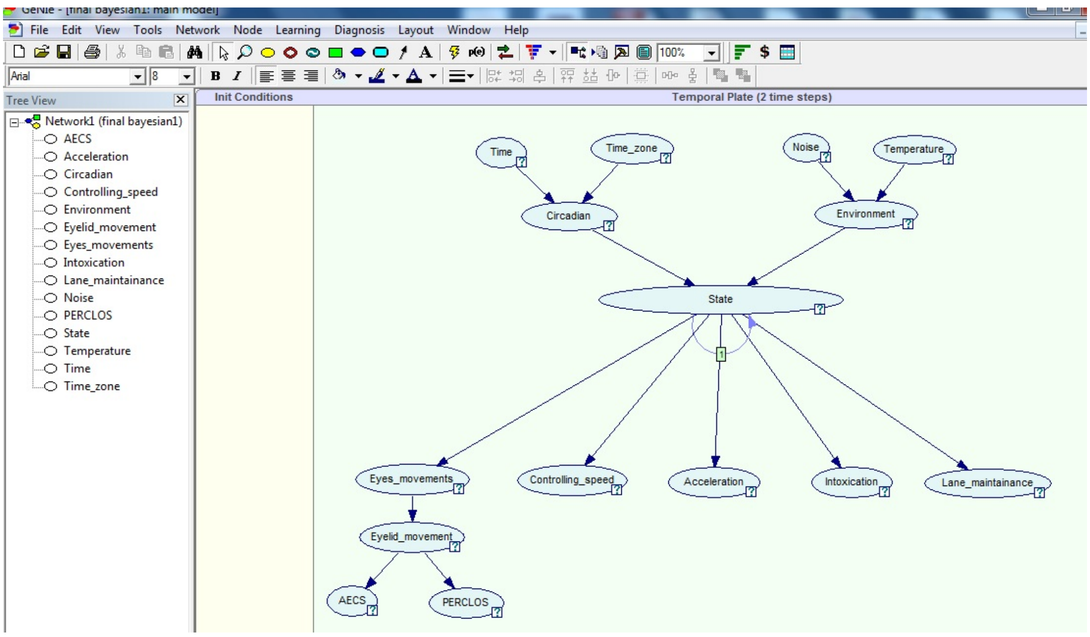

Figure B.7: DBN Structure Implemented using GeNIe

B.3 Inference and Results

As mentioned earlier in Chapter 3, GeNIe 2.0 supports different kinds of inference algorithms (i.e. the polytree algorithm). The designer has to select the appropriate inference algorithm before performing the inference process, and the selection of the algorithm depends on the problem requirements. Selecting the desired algorithm is done by clicking on *Network* $\rightarrow$ *Algorithm* then selecting the desired one (we have selected the polytree algorithm to perform the inference in our network). After choosing the inference algorithm, the inference process can be carried out by right-clicking on the network area and selecting *'Update Beliefs'*. The updated beliefs of the network can be gained by double-clicking on the hypothesis node then selecting the *value* tab as shown in Figure B.8.

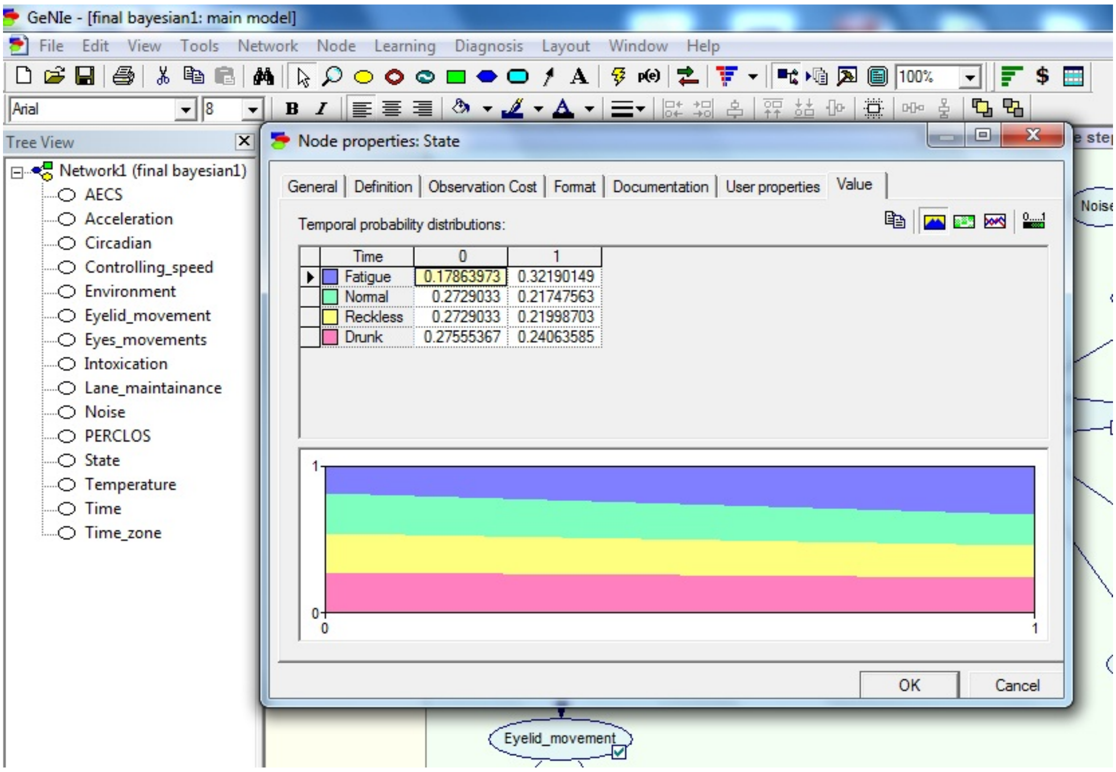

Figure B.8: The representation of the updated beliefs in GeNIe