American University in Cairo

AUC Knowledge Fountain

2-1-2016

Fuzzy logic traffic signal controller enhancement based on aggressive driver behavior classification

Shaimaa Mohammed Hegazy

Follow this and additional works at: https://fount.aucegypt.edu/etds

THE AMERICAN UNIVERSITY IN CAIRO

Fuzzy Logic Traffic Signal Controller Enhancement

Based on Aggressive Driver Behavior Classification

A thesis submitted to the

Department of Computer Science and Engineering

In partial fulfillment of the requirements for the

Degree of Master of Science

by

Shaimaa Hegazy

under supervision of

Dr. Mohamed Moustafa

Fall 2016

This thesis is dedicated to My Husband.

For his endless love, support, patience and encouragement.

ACKNOWLEDGEMENT

ABSTRACT

The rise in population worldwide and especially in Egypt, together with the increase in the number of vehicles present serious complications regarding traffic congestion and road safety. The elementary solution towards improving congestion is to expand road capacities by building new lanes. This, however, requires time and effort and therefore new methodologies are being implemented. Intelligent transportation systems (ITS) try to approach traffic congestion through the application of computational and engineering techniques. Traffic signal control is a branch of intelligent transportation systems which focuses on improving traffic signal conditions. A traffic signal controllers' main objective is to improve this assignment in a way which reduces delays. **This research proposes a new approach to enhancing traffic signal control and reducing delays of a single intersection, through the integration of an aggressive driving behavior classifier.** Previous approaches dealt with traffic control and driver behavior separately, and therefore their successful integration is a new challenging area in the field. Multiple experiment sets were conducted to provide an indication to the effectiveness of our approach. Firstly, an aggressive driver behavior classifier using feed-forward neural network was successfully built utilizing Virginia Tech 100-car naturalistic driving study data. Its performance was compared against long short-term memory recurrent neural networks and support vector machines, and it resulted in better performance as shown by the area under the curve. To the best of our knowledge, this classifier is the first of its kind to be built on this 100-car study data. Secondly, a representation of aggressive driving behavior was constructed in the simulated environment, based on real life data and statistics. Finally, Mamdani's fuzzy logic controller was modified to accommodate for the integration of the aggressive behavior classifier. The integration results were encouraging and yielded significant improvements at higher traffic flow volumes when compared against the built Mamdani's controller. The results are promising and provide an initial step towards the integration of driver behavior classification and traffic signal control.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 INTRODUCTION

In this chapter, the problem of traffic congestion and traffic-related accidents is discussed. One of the solutions proposed to solve this problem uses traffic signal control. It is pointed out that finding a suitable methodology to reduce delays while increasing safety is a challenging issue. The objective of the research is then pointed, which is mainly to enhance traffic signal control using computational intelligence techniques. One way of improvement is through incorporating driver's behavior classification to achieve the objective of minimizing intersection delays. Finally, the outline of the thesis is presented.

1.1   Problem Definition

Transportation constitutes a major element in our everyday life. Transportation of people and goods usually takes place through the road network infrastructure in most of the countries. The increase in population worldwide and especially in Egypt, together with the increase in the number of vehicles, poses severe problems regarding traffic congestion and road safety. According to the World Health Organization (WHO), the number of registered vehicles worldwide increased by 15% between 2010 and 2013 [1]. When multiple vehicles attempt to use some common transportation infrastructure having a limited capacity traffic congestion occurs. Increased traffic congestion in its worst case results in degraded use of the transportation infrastructure [2]. Traffic congestion, thereby, means increased traffic delay leading to a waste of resources such as time and fuel. This in turn leads to decreased human productivity, and a great negative effect on the national economy. A World Bank report [3] estimated the annual costs of congestion in Egypt to be up to US$8.0 billion, which accounts for almost four percent of Egypt's Gross Domestic Product (GDP). In addition to its effects on productivity, environment and the GDP, traffic congestion is also deadly. The European Union [4] identified road death in Egypt as the second most common cause of death. It has been estimated that around 1.2 million

1

people die each year from road traffic crashes [5] with thousands more injured, some of them suffering long-term disabilities as a result [6]. Without a proper action to reduce these deaths, it is expected that traffic crashes would become the "7th leading cause of death by 2030" [7]. These statistics illustrate the urgent need for a solution regarding the road infrastructure in order to reduce congestion and increase the safety of the passengers and pedestrians.

To reduce congestion, several solutions exist. The elementary solution is to expand road capacity by adding more lanes to the existing roadways in order to accommodate the increased demand or to build entirely new traffic. Infrastructure expansion, however, is not feasible due to budget restrictions and the extremely huge efforts required. Some infrastructure changes have been recently applied in Egypt, such as the Cairo subway, ring road and some bridges [8]. Despite these changes, Egypt is still suffering from accidents, delays, and bad road conditions. It is also quite important to change the aggressive road user behavior in order to reduce crashes and delays, but this is not always easy. Another method of increasing traffic network efficiency and safety is through introducing efficient traffic control systems that integrate driver behavior analysis and classification. These traffic control systems rely on technological advancements to reduce or solve traffic problems.

Intelligent Transportation Systems (ITS) refer to a collection of techniques aimed at improving transportation systems using intelligent computing techniques. The goals of ITS include among others: reducing congestion, enhancing public safety, improving access to travel information, and reducing harmful environment impacts [9]. ITS fields comprise sensors, communication and traffic control technologies. Improving vehicle flow at intersections and increasing intersection safety are of the significant objectives in traffic control and this research.

Traffic signal aims to allocate right-of-way to a single or multiple non-conflicting traffic movements at a given time; by switching on the green signal for a certain period of time. One of traffic signal control's objectives is to reduce the average delay of a group of vehicles [10]. Three

2

categories of traffic signal control exist: pre-timed, actuated, and adaptive control. Pre-timed control uses a certain preset signal timing obtained from historical traffic patterns [11], where this pattern is repeated throughout the day. Historical data is, however, inaccurate and changes over time leading to the inaccuracy and inefficiency of the controller. Moreover, pre-timed signal control does not account for the changes in traffic volume over time, which would require an update in the used historical patterns. Actuated controllers then emerged to tackle the problems introduced by pre-timed controllers. In actuated controllers, green time is modified according to the detection of passing cars in the road having right-of-way. The downside of actuated controllers is that they do not take into consideration the number of cars in the queuing side resulting in lack of optimal usage of the intersection [12]. Finally, adaptive signal control uses computational intelligence techniques to adjust signal timings by considering the conditions of the whole intersection [11]. Therefore, adaptive signal control addresses the inefficiencies inherent in the other signal control techniques.

The mentioned large number of traffic accidents does indeed increase vehicle delays and therefore it is also significant to consider the intersection safety as a method to reducing delays. One proposed method of improving intersection safety is by detecting aggressive drivers' behaviors and adjusting the traffic signal accordingly. Driver's behavior is of utmost importance and has been studied widely by researchers in different areas including statistics, psychology, engineering and signal control. Studying driver's behavior is important as it might lead to the early prediction or detection of traffic delay causes. Traffic signal control techniques can then be adjusted accordingly leading to an enhancement in the waiting time of vehicles in intersections. Integrating the study of drivers' behavior together with traffic signal control techniques is a further step in ITS, aiming to reduce intersection delays.

Despite the intensive research and work done in Intelligent Transportation Systems, and particularly those related to traffic signal control, lack appears in the area of improving traffic

3

delay and safety by incorporating the driver's behavior. Recognizing driver's behavior and early classification of aggressive behavior might be a new key to reducing intersection delays.

1.2   Motivation and Objective

In this research, improving traffic conditions is aimed through incorporating aggressive driver behavior classification within traffic signal control, in order to reduce traffic intersection delays. No previous work has been found which illustrates the importance of incorporating driver's behavior into traffic signal control. The first objective is to successfully build and test an aggressive driver behavior classifier, which would receive vehicle trajectory data and classify drivers into either being aggressive or non-aggressive. Secondly, Mamdani's fuzzy logic traffic controller [13] is to be built and simulated under normal intersection conditions with no aggressive drivers. Its performance shall be compared against static and vehicle actuated controllers under the same conditions. Thirdly, some form of aggressive driving shall be introduced into the intersection in order to show how aggressive driving increases vehicle delays at intersections. Finally, the aggressive driver behavior classifier built shall be integrated with Mamdani's fuzzy logic traffic controller in order to improve the intersection conditions with focus on vehicle delays.

1.3   Organization of the Proposal

The remainder of the proposal is structured as follows. Chapter 2 starts by providing background about traffic signal control and some of the different classification algorithms, and then moves into reviewing the literature on traffic signal control and driver behavior. Chapter 3 discusses the proposed traffic signal control architecture, its key features and compares it with state-of-the-art architectures. In chapter 4, the experiments conducted are described and their results are shown. Finally, the thesis ends with chapter 5 which provides the conclusion and outlines some of the future work.

CHAPTER 2 BACKGROUND AND PREVIOUS WORK

In this chapter, background related to intelligent transportation systems, traffic lights and classification algorithms is first discussed and then a literature review is presented. Firstly, intelligent transportation system is introduced together with some of its current applications. Traffic lights are then discussed, giving a brief history about them and afterwards moving into the important terminology in the field of traffic light optimization and safety. Traffic simulation, which is required to test developed traffic control systems, is considered afterwards together with the different simulators which can be used and how to choose a good simulator. Then some of the machine learning classification algorithms are described and their theories reviewed. Next, related work on traffic control is surveyed for the sake of understanding the nature of research performed on traffic signal control systems and the most appropriate work to use as a starting point for our research. Related work on driver behavior is likewise studied in order to comprehend how driver behavior is related to intersection safety and different ways of incorporating it into traffic signal control systems to achieve increased traffic intersection safety and reduced delay.

2.1   Intelligent Transportation Systems

The need for mobility using vehicles is ever-increasing and has over-flooded major cities leading to an expansion in the problem of traffic congestion and traffic-related accidents. This reveals the importance of improving the current transportation systems, calling for the development of more efficient and safer mobility systems. As a result, intelligent transportation systems (ITS) emerged in an ever-evolving attempt to enhance traffic and transportation conditions.

ITS integrates information systems, telecommunications, sensors and mathematical models on one hand with the conventional transportation engineering on the other hand in order to design, develop and maintain transport systems [14], [15]. Improving safety, security,

quality and efficiency of transportation systems while optimizing the use of resources are the aims of this integration [15]. In [14], ITS is divided into six areas: Advanced Traffic Management Systems (ATMS), Advanced Traveler Information Systems (ATIS), Advanced Vehicle Control Systems (AVCS), Commercial Vehicle Operations (CVO), Advanced Public Transportation Systems (APTS), and Advanced Rural Transportation Systems (ARTS). In ATMS, traffic congestion is predicted and alternative routing instructions are provided to vehicles in order to improve the efficiency of the highway network. Dynamic traffic control is expected to respond in real-time to the changing traffic conditions. ATIS is supposed to provide information to travelers including the location of incidents, road conditions and optimal routing. AVCS includes systems such as collision warning systems which indicate the possibility of a collision. It is aimed at making travel both safer and more efficient. CVO, on the other hand, is concerned with the private operators of trucks, vans and taxis in order to improve their productivity and efficiency. Enhancing the accessibility of information to users of public transportation goes under APTS. Finally, ARTS focuses on low-density roads [14].

ITS is, therefore, a wide research area aiming to enhance the traffic conditions by targeting several traffic problems using advanced technological techniques. In this thesis, the focus is on enhancing traffic signal control systems, which is a part of ATMS in order to efficiently reduce delays in a single intersection. This is to be achieved by incorporating driver's behavior classification into traffic signal control.

2.2   Traffic Light Optimization and Safety

Traffic lights date back to 1868, where manually operated semaphores where firstly used in London [16]. The railway engineer, J. P. Knight installed them outside the British Houses of Parliament [17]. The first traffic signal in the United States used red and green lights and was

developed in order to prevent accidents in 1912 in Salt Lake City, Utah [18]. The first automatic traffic signal which used colored lights was then patented by William Ghigliera in 1917 [19]. Toronto was the first city in the world to have a computerized traffic signal system in the early 1960s [20]. The main purpose of traffic signal is to allocate right-of-way to one traffic movement or several non-conflicting movements at a time in order to move vehicles through an intersection safely and efficiently. This is achieved by turning on the green signal for a specified period of time [10]. In this section, some of the terminologies related to traffic signal control and traffic safety are presented.

2.2.1 Cycle

A **cycle** represents a complete sequence of signal indications for a given intersection. The time required for the completion of this sequence is known as **cycle length**. A **cycle failure** takes place when all queued vehicles cannot be served by one green indication. A single cycle might consist of multiple intervals. An **interval** is the duration during which an indication does not change its state. One or more timing parameters indicate the duration of an interval, for example, some of the parameters which control the green interval duration are minimum green time and maximum green time. **Minimum green parameter** is first portion of the green interval during which drivers react to the start for the green interval and therefore no vehicles leave the intersection. **Maximum green parameter**, on the other hand, is the maximum amount of time that can be given to the green signal. It is used to limit the delay of the other movements in the cycle. Some time is lost during each cycle where no any movement effectively uses the intersection; this is referred to as **total lost time**. [16]

2.2.2 Phase

According to the Signal Timing Manual [16], a **phase** is the timing given to control one or more movements. In other words, it is the "right-of-way, yellow change, and red clearance

www.manaraa.com

intervals in a cycle that are assigned to an independent traffic movement" [16]. The order of phases in a given cycle is known as the **phase sequence**. The setting and ordering of phases in a cycle is critical, and therefore it is sometimes useful to design and use a **phase diagram**, which is a graphical representation of the sequence of phases [16]. Figure 1 is an example of a phase diagram for a given intersection, showing the assignments for an eight-phase controller operation. It assigns odd number phase to left-turn movements while allocating even numbers to through movements.
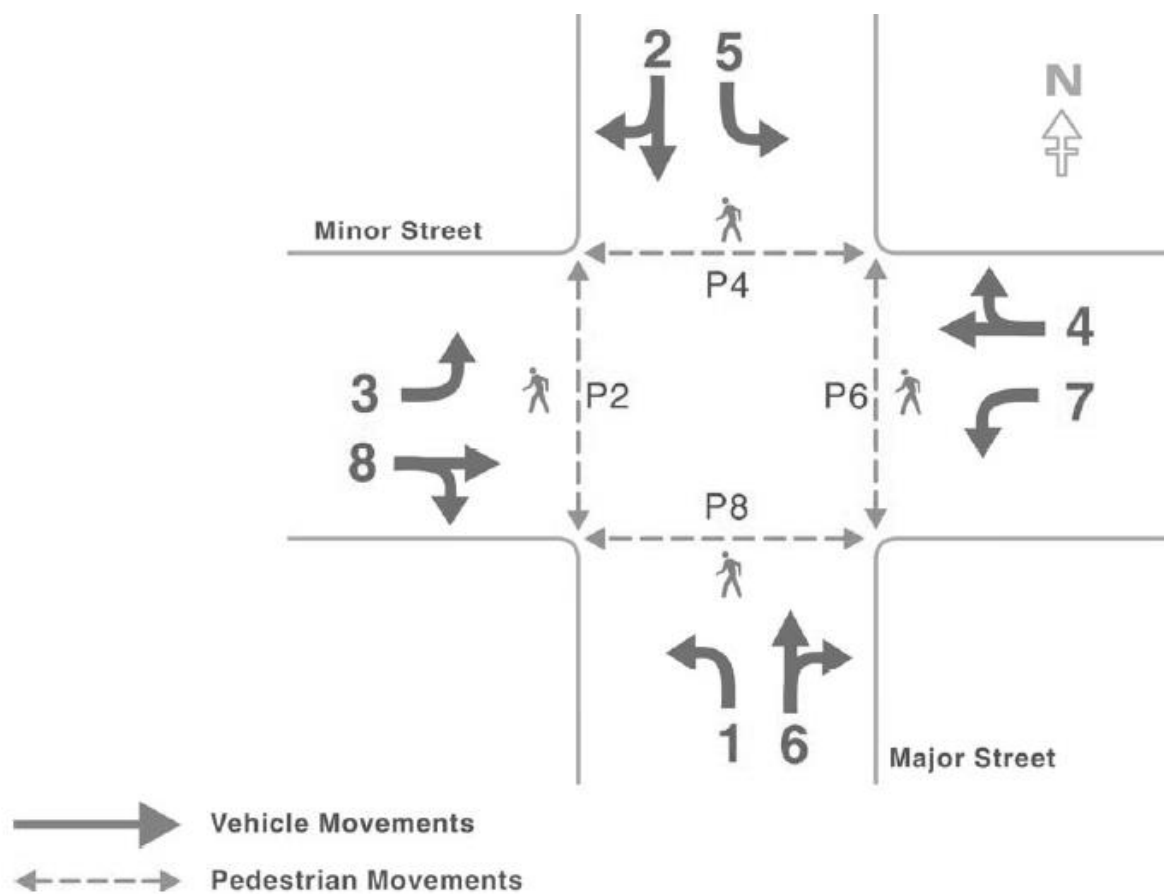
Figure 1 - Phase Diagram [16]

2.2.3   Dilemma Zone

The area in which a driver finds difficulty deciding "whether to stop or proceed through an intersection at the onset of the yellow-signal indication" is known as the **dilemma zone** [21]. Two types of dilemma zone exist; type I occurs when the time given to the yellow and red

clearance times is not sufficient to clear the intersection before the beginning of a conflicting phase. Type II, which is referred to as option zone or indecision zone, occurs when different decisions on whether to go or stop are made by different drivers at the onset of the yellow indication [16].

2.2.4 Aggressive Behavior

A formal definition of aggressive driving was offered by Tasca in [22] as the driving behavior that is "deliberate, likely to increase the risk of collision and is motivated by impatience, annoyance, hostility and/or an attempt to save time". He suggested certain behaviors that would go under aggressive behavior, such as tailgating, improper passing, running red lights, driving at very high speeds, etc. Designing a model that would classify drivers' behaviors into aggressive and non-aggressive is an essential part of a crash assesment and/or prevention module. This model could then be integrated into traffic control systems at intersections in order to reduce intersection delays.

2.3   Classification Algorithms

One of the aims of this thesis is to implement an aggressive driver behavior classifier based on driver patterns and trajectories. Three different classification algorithms are briefly described here which shall be tested in the proposed architecture.

2.3.1 Artificial Neural Networks (ANN)

An artificial neural network (ANN) is a machine learning algorithm built to mimic the human biological nervous system [23]. The basic processing unit of ANNs is the artificial neuron, which simulates the function of natural neurons. An artificial neuron receives several weighted inputs, sums them together and passes them through a transfer function which decides

9

www.manaraa.com

its output. The learning capacity of the neuron is adapted by modifying the input weights through a learning algorithms. The basic mathematical function of a neuron is shown in Equation 1, where x is a neuron with n inputs (x_0, x_1,…, x_n) and one output (y), and w is the weight associated with each input element. g is the transfer/activation function that determines the neuron's output. The output is usually either between 0 and 1 or between -1 and 1. An illustration of the artificial neuron is given in Figure 2.

$$y(x)=g\left(\sum_{i=0}^{n} w_i x_i\right)$$

Equation 1 – Mathematical function of a neuron
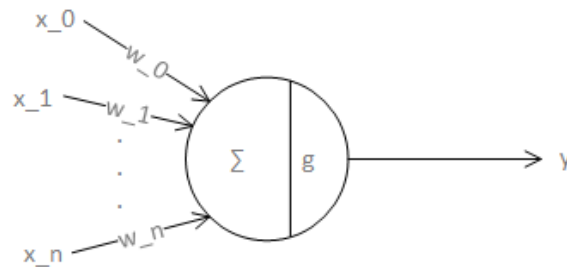
Figure 2 - Artificial neuron illustration

There are several activation functions, such as: threshold, sigmoid and hyperbolic tangent functions. In our experiments, the hyperbolic tangent sigmoid and softmax functions are used. The hyperbolic tangent function, shown in Equation 2, is a sigmoidal function whose output ranges from -1 to 1. It is advantageous because its derivative is easily calculated and this is quite important for backpropagation learning algorithms [24]. The softmax function – Equation 3 – outputs a normalized output (form 0 to 1) that can be used as class posterior probability for multi-class classification.

$$a=\text{tansig}(n)=\frac{2}{1+e^{-2n}}-1$$

Equation 2 - Hyperbolic tangent function

$$a=\text{softmax}(n)=\frac{e^n}{\text{sum}(e^n)}$$

An artificial neural network is made up of several layers, where each layer might contain several artificial neurons, and layers are connected to each other. Figure 3 shows a typical structure of a fully connected feedforward neural network with one hidden layer. Solving non-linear problems is usually achieved using neural networks with one or more hidden layers.

Input Layer   Hidden Layer   Output Layer
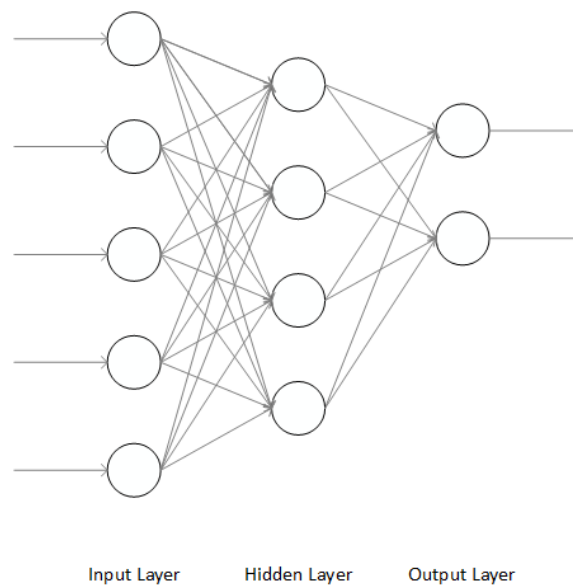
Figure 3 - A fully connected feedforward neural network

2.3.2   Long Short-Term Memory Recurrent Neural Networks (LSTM-RNN)

Besides feedforward networks, there are the recurrent neural networks (RNN), which are a category of artificial neural networks that can remember previous inputs and use them together with the current input to decide the network output. This is done by having a feedback loop from previous outputs to the current inputs, making them suitable for sequence learning tasks. However, they are only able to learn problems that involve short time lags, making them unable to process long sequences [25].

For tasks involving long-term dependencies, a specific type of RNN called Long Short-Term Memory (LSTM), which were introduced by [26], works better. The main idea behind LSTM is the memory cell, which contains four main elements: an input gate, a forget gate, an output gate and a self-recurrent connection, which ensures that the memory cell state can "remain constant from one time step to another" [27]. An illustration of an LSTM memory cell is provided in Figure 4. These gates allow information to be stored in the cell, written to it or read from. The input's gate role is to determine if the incoming signal is to alter the memory cell state, while the output gate determines if the memory cell should have an effect on the other neurons. Finally the forget gate allows the cell to remember or forget its previous state by controlling the self-recurrent connection. Since the data used for behavior classification is a time-sequence data, LSTM acts as a good candidate for the classification task.
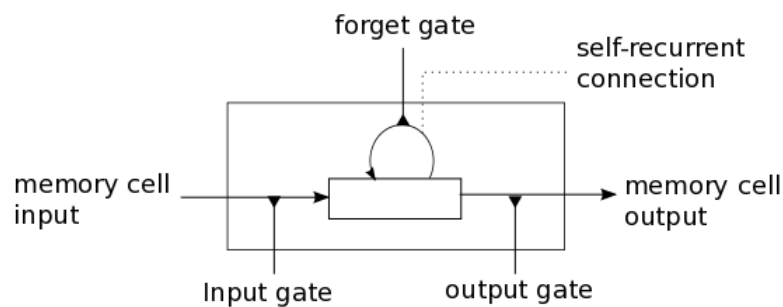
Figure 4 - Basic structure of an LSTM memory cell [27]

2.3.3 Support Vector Machines

Support Vector Machines are a class of supervised machine learning algorithms introduced by [28]. When SVM is used for classification, the aim is to maximize the margin between the training data points and the decision boundary. It is well suited to two-class classification problems, and works by finding the best hyper-plane separating points of one class from the other, such that the margin between the two classes is maximized. Figure 5 shows an illustration of the SVM idea on a simple example. This simple illustration assumes that the data is linearly separable, however, if the data is not, there is also another way that SVMs would work.

12

This is usually achieved by using a kernel that maps the data into a richer feature space and then constructing the hyper-plane in that space. Some of the common kernels are: polynomial, Gaussian radial basis function and hyperbolic tangent kernels.
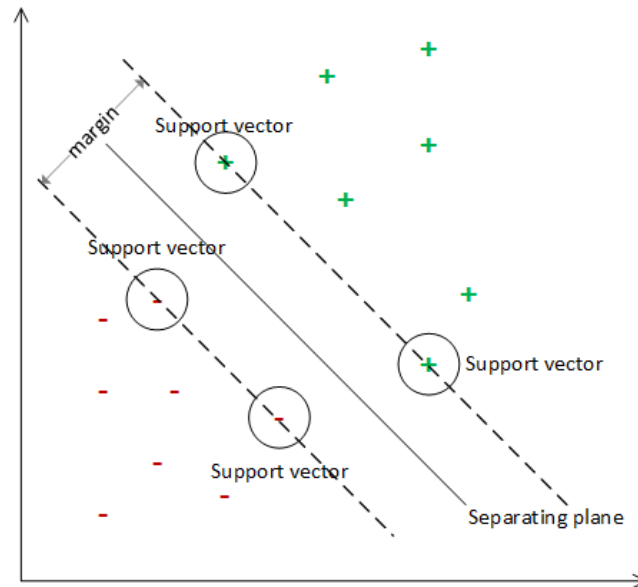
Figure 5 - Illustration of an SVM separating plane

2.4   Traffic Simulation

In the field of traffic signal control, simulating traffic is an important step to testing new proposed approaches. It is true that field testing yields more accurate and realistic results and patterns of traffic flow; however, it is more expensive and might sometimes involve hazards. Therefore simulation is the most common tool used by traffic signal designers to simulate and test their approaches.

There are two types of traffic simulators: macroscopic and microscopic simulators [29], which differ according to the level of details they provide to describe traffic and vehicle flows. In **macroscopic models**, vehicles are represented as an aggregated traffic flow. Macroscopic models consider traffic flows as a fluid process whose state is characterized by some aggregate

macroscopic variables such as *density*, *volume*, and *speed* [30]. On the other hand, **microscopic models** can describe each vehicle as a separate object with a given unique characteristic, and therefore, they can provide complex interactions among vehicles and between drivers [29]. A number of traffic simulators exist and some of which have been compared in the literature. A brief description of some is given.

Generic Mobility Simulation Framework (GMSF) was developed at ETH Zurich as part of a master thesis and is a microscopic traffic simulator. GMSF is freely available [31]. Several mobility models can be simulated, such as GIS-based mobility model, Mutli-agent microscopic traffic simulator (MMTS) model, Manhattan model and Random waypoint model. In the GIS-based mobility model, vehicles describe steady-state random trips which are constrained to the road topology of the uploaded map. Vehicles keep a safe distance from the vehicles in from of them. MMTS is a model developed also at ETH Zurich, and it realistically simulates the behaviors of all individuals in a certain area according to data collected by census. The Manhattan model, on the other hand, uses a grid layout as the road topology, and it simulates randomly accelerating and decelerating vehicles. Finally, the random waypoint model was designed for ad hoc networks, and does not usually suit realistic vehicular modeling. [31]

VanetMobiSim is Java-based and it focuses on vehicular mobility. It contains motion models at both the microscopic and macroscopic levels [32]. Road topologies can be either defined by the user or imported from road map databases. VanetMobiSim provides two traffic demand modules: trip generating module and path computation module. In the trip generating module, a set of start and stop points are defined on the road topology. The path computation module is then responsible for selecting the best path between the start and stop points. One of the advantages of VanetMobiSim is that it contains an incident generator feature, which allows the user to specify the class of incident together with its location and duration [29]. VanetMobiSim can also be downloaded freely.

14

VISSIM is a commercial microscopic traffic simulator which provides a user-friendly modeling interfaces. It also provides second-by-second information about the state of the vehicles in the network, with a higher accuracy than the other simulators [33].

Simulation of Urban MObility (SUMO) is an open-source microscopic vehicular traffic simulator developed at German Aerospace center [34]. It is designed to handle large traffic networks, where it provides support for different vehicle types, multi-lane streets, right-of-way rules at intersections and a graphical user interface. SUMO also provides an interface for traffic signal control called Traffic Control Interface (TraCI), which allows to "retrieve values of simulated objects and to manipulate their behavior on-line" [35].

In this section, four of the widely used simulators were discussed. When choosing a traffic simulator, several points have to be considered. Table 1 compares the four simulators by considering the simulator type, microscopic or macroscopic; the availability of the simulator, if it is freely available or commercial; the mobility models provided by the simulator; if a graphical user interface (GUI) is available with the simulator; and the possibility of simulating and controlling traffic signals. For the purposes of the thesis, SUMO appears to provide most of the features required; especially due to the availability of online traffic signal simulation and control.

Table 1 - Traffic Simulators Comparison

	GMSF	VanetMobiSim	VISSIM	SUMO
Simulator type	Microscopic	Microscopic and macroscopic	Microscopic	Microscopic
Availability	Freely available	Freely available	Commercial	Open-source
Mobility models	▪ GIS-based model ▪ MMTS model ▪ Manhattan model. ▪ Random waypoint model	▪ Vehicular mobility	--	--
GUI-Interface	--	--	--	Exists
Traffic Control	--	--	--	TraCI allows online simulation of traffic and manipulation of objects
Other features		Contains an incident generator feature	Information about vehicles is more accurate.	

2.5   Related Work

2.5.1   Related Work on Traffic Control

Traffic signals are used to assign right of way to one or more non-conflicting movements in a given intersection. Controlling traffic signals is essential in order to ensure reduced delay. Three main methodologies exist for controlling the traffic signal: pre-timed, actuated or adaptive control. **Pre-timed control**, which is the easiest traffic control method, repeats a certain preset signal timing obtained from historical traffic patterns [11]. It does not adapt to any changes in the traffic and therefore it is only suitable for fairly stable and regular traffic flows [12]. The **actuated controllers** were introduced afterwards to mitigate the problem of not adapting to the coming traffic. In actuated controllers, the green time is adjusted according to the detection of passing cars in the approach having the right-of-way. As long as there are cars passing, the green time is extended, until a certain predetermined maximum green time limit is reached. For example if the green time was about to end, but a vehicle was detected, it is extended for 5 or 10 more seconds. The problem of this methodology; however, is that it does not take into account the number of cars in the queuing side, and therefore optimal usage of the intersection cannot be achieved [12]. Finally, **adaptive control** tries to target the problems of the previous approaches by the use of computational intelligence. It considers the conditions of the whole intersection [11] and can adjust the traffic signal timings as well as the traffic phases. For the purposes of the thesis, we are interested in the third type; adaptive control.

Using Fuzzy Logic to Control Traffic Signals

Of the most widespread computational intelligence techniques used in traffic signal control is the fuzzy logic. This can be attributed to the stochastic and random characteristics of traffic, which can best be represented and handled using fuzzy logic systems. This is because fuzzy logic allows describing and modeling imprecision and uncertainty; by assigning degree of

truth rather than sharp crisp values. Pappis and Mamdani [13] were the first to introduce the idea of using fuzzy logic to control traffic signal in a single intersection made of two one-way streets in order to reduce the average delay of vehicles. They used as their input variables: 1) the number of vehicles arriving at the approach having the right of way (also known as **arrivals**); and 2) the **queue** building up on the other approach. Their output variable was the extension in seconds to be given to the green signal. When compared against the conventional vehicle-actuated controller, their approach yielded an overall improvement in the delay by 10% ~ 20%.

Further improvements were made on Mamdani's approach. In [36] they extended Mamdani's approach by considering four phases but without right or left turns, in addition to that, they developed a software to simulate the situation of the isolated traffic junction by considering a varying flow density. They also considered the arrivals and queue as their inputs and the extension of the green time as the output. Compared against the conventional fixed-time controller and the vehicle actuated controller, their controller produced an improvement of about 20%.

Askerzade and Mahmood [37] then introduced two major improvements by considering right and left turns and testing different defuzzification methods. Their results show an improvement of 5% to 51.4% using Sugeno-WK defuzzification method under different traffic conditions.

Further extension of the intersection was introduced in [38], who considered an intersection with 12 main direction flows, turning main direction and two-way arterial road. Their fuzzy rule set performs multiple state controls in order to choose the best state control to implement (two-state, three-state or four-state control). They used 10 fuzzy input variables and 2 output variables: extension of time; and selection of next state. However, they lack the documentation of their results.

Optimizing more Objectives

Considering only the extension time is not very efficient, as there are other parameters in traffic control which might lead to more efficient optimization of the overall intersection performance. In [39], the researchers targeted the implementation of a phase sequencer and developed their own simulation software simulating heavy traffic conditions. Results show more effective use of the intersection space (i.e. greater throughput) and less delay than pre-timed controller. However, they found out that the addition of the phase sequencer yielded no significant difference.

Niitkymaki et al. [40]–[43] widely considered fuzzy logic for controlling traffic in their FUSICO-research project in Finland in the Helsinkin University of Technology. They successfully tested fuzzy logic based traffic signal control in urban intersections. Their approach yielded less delays and less percentage of stops than the vehicle-actuated controller. Also when compared against the Pappis-Mamdani extension principle, the percentage of stops was less. After the initial successful implementation, they added signalized pedestrian crossing in their implementation to increase the safety of the intersection [43]. It was found that the performance was better under high traffic density.

Murat and Gedizlioglu in [44] aimed to improve the phase sequencing by introducing a fuzzy-logic based phase sequencer in an isolated intersection and under uneven traffic conditions. Compared with the traffic actuated model, similar results were achieved for the two phasing controlled situation and little traffic volumes. However, under larger variation and traffic volumes, the delays of vehicles were decreased by about 15%. The three phasing controller, on the other hand, decreased delays and increased the capacity for both equal and different traffic volumes. Murat and Gedizlioglu then improved the performance in [45] by considering up to four phases. They compared the results against the FUSICO project and the average delay was almost the same for the two phase case. For the three-phase and four-phase cases, performance

18

was compared against the traffic actuated and the vehicle actuated controller. The best performance was achieved when the variation and values of traffic volumes were high, as the delay was reduced by 23%. For the four phased case, also at high traffic volumes, delay was decreased by 20%.

Another approach handling the phase sequencing was introduced in [46] by considering queue length of the crucial flow phase and the maximum queue length. Their proposed fuzzy control was superior to the general fuzzy control and actuated control under increased traffic volume counts.

In the previously mentioned papers, less concern was given to addressing traffic congestion, and therefore Kim [47] aimed to deal with traffic congestion and adaptively control the cycle of traffic. He considered the number of vehicles entering an intersection from other intersections by the second, which is the volume of vehicles. He compared the results against the pre-timed and Mamdani's controllers; where he found out that little difference was achieved at light traffic. With various traffic flows, on the other hand, the number of passed vehicles increased (19.4% against the pre-timed controller and 2.7% against Mamdani's approach). The average delay time was also reduced (21.7% against the pre-timed controller and 36.9% against Mamdani's approach). Finally, the degree of saturation was improved (7.3% against the pre-timed controller and 11.7% against Mamdani's approach).

Of the interesting applications of fuzzy logic in the real world is the research done in Kuwait [48], where they developed a fuzzy logic traffic system that handles the two-way intersections. They applied and tested their system using real data collected from signalized intersection in the State of Kuwait. Good results were obtained compared to their existing system, when considering heavy traffic volumes.

Use of other Computational Intelligence Techniques

In order to improve the performance of fuzzy logic systems, it could be beneficial to combine it with other computational intelligence techniques. Optimizing the membership functions usually results in better fuzzy logic systems. Wei et al. [49] used multi-objective genetic algorithms (MOGA) to find the set of optimal parameters of the membership function. Compared against the traditional vehicle-actuated control, lower delays were achieved (15 – 30% depending on the traffic conditions) and lower percentage stops were also achieved (15 – 25% depending on the traffic conditions). Another approach introduced by Cheng and Yang [50] was to use genetic algorithms to improve some of the fuzzy rules in their rule set. Compared against fixed-time and actuated controllers, their approach produced the least total stop delay.

Stepwise evolution algorithm was used in [51] to learn both the logic rules and membership function. They also used the queue length of cars as an input variable. The results were almost similar to the results of the optimal control under given traffic patterns.

More focus was then given to multiple-stage and more complicated fuzzy controllers. Yang et al. [52] developed a two stage fuzzy controller with status variable selection. They introduced the rolling framework and optimized the parameters of membership functions and controller rules using hybrid genetic algorithm. Compared to fixed-time and actuated controllers, they obtained good results in varying traffic flows. Under low saturation, delays were minimized and traffic flow was smooth. In medium-high saturation levels, the throughput was increased and delay decreased. Finally under high saturation conditions, the throughput was also increased satisfactorily and the operation efficiency was high.

Neuro-fuzzy system was used in [53] in order to estimate the number of vehicles waiting in queue and they achieved an accuracy of estimation more than 90%. They used MOGA to minimize delays and minimize number of vehicle stops. They accomplished lower vehicle delays

20

(15-30%) and lower percentage of stopped vehicles (15-25%) when compared with the traditional extension principle (vehicle-actuated approach).

Reinforcement learning was used in [54] with a fuzzy neural network to control a two-phase controlled intersection with 2 lanes per streets. After learning, the vehicle delays were minimized by 3-6%. However it was found that it is successful at constant traffic volumes only.

Conclusion

To sum up, in this section, the previous approaches to handle traffic signals were reviewed. It can be concluded that the majority of the approaches use fuzzy logic systems to handle traffic signal control. Objective set selection is an important factor when considering fuzzy logic. Basic fuzzy logic control systems focus on using arrival and queue as parameters for controlling the traffic signals. Advanced systems, however, integrate other parameters in order to improve the accuracy. Examples include integrating a phase sequencer to increase the efficiency. Fuzzy logic can also be augmented with other computational intelligence methodologies to achieve better results.

2.5.2 Related Work on Driver Behavior

Improving intersection safety is of utmost importance due to the large number of intersection-related accidents and fatalities. Improving intersection safety results in fewer accidents at intersections and therefore also reduces delays, since accidents are shown to result in significant delay at intersections. In this research, the focus is on reducing intersection delays through classifying aggressive driver behavior and adjusting the traffic signal accordingly. This section focuses on reviewing the literature on the driver's behavior.

Driver's behavior has been studied by a myriad of researchers in different areas ranging from statistics for creating a model in order to predict driver's behavior, to psychology in order to understand the driver's behavior pattern, to Engineering with the purpose of studying the

dilemma zone and creating driver behavior classification algorithms, and finally in signal control to adapt traffic lights accordingly. The important and relevant work in each of these areas is reviewed in this section.

Driver Behavior Studies

Understanding how the driver behaves at intersections in response to the changing conditions is crucial in order to set the required conditions so as to prevent accidents and reduce delays. Researchers have studied drivers' behavior under different circumstances, such as studying red light violators, and driver patterns during and in response to changing the yellow phase timing.

Researchers in [55] aimed to examine the characteristics of red light violators in order to identify factors that strongly influence red light running behavior. In their study, 47,000 violation records captured by photo enforcement from 11 signalized intersections in California, were examined to understand red light runners' behavior. The results showed that the drivers' age, driving time and driving speeds were the most influential factors. Drivers younger in age had higher tendency to run the red light, and this was usually associated with speeds above the posted speed. Peak times (from 8 p.m. to 5 a.m.) witnessed higher violation counts, again with higher speeds. Finally, high traffic-volume intersections had fewer violations with speeds above the posted speed limits. It was also shown that longer clearance interval – which consists of yellow and all-red times – is associated with greater probability of red-light running. Red-light running behavior was also examined in [56], in which they established that red-light runners were younger. They further found that red-light runners were less likely to use safety belts and drove smaller and older vehicles.

On the other hand, researchers in [57] studied the conditions under which drivers would rather choose to stop correctly at the intersection rather than running the red-light. These

conditions were found to be: greater travel time to the intersection at the onset of the yellow signal, a shorter yellow interval, longer cycle length and the absence of vehicles in adjacent lanes. Of those factors, the most significant was the estimated travel time at the onset of the yellow interval.

Moving from examining red-light runner behavior to studying behavioral patterns during the yellow phase, a field observation of 1123 drivers at six signalized intersections in Maryland was conducted and drivers at each intersection were classified into: aggressive pass, conservative stop and normal groups, in [58]. This classification was based on the drivers' responses (stop or pass) and their distance from the stop line at the onset of the yellow signal. The average flow speed was the most critical factor implying that drivers are likely to take aggressive passing actions at the inception of the yellow phase.

According to [59], an improperly timed clearance interval can create a dilemma zone, in which drivers approaching the intersection cannot decide whether to proceed or stop safely at the onset of the yellow interval. This dilemma zone, consequently, might result in serious right-angle or rear-end crashes. In [56], it was shown that the length of the clearance interval influences the risk of red-light violations. A study to examine the effect of re-timing the change intervals on possible crashes was conducted in [60]. The need for this study arouse due to the lack of universal practice for selecting the duration of the change intervals and because of the lack of information about the effect of changing the traffic signal timing on crash risk. The traffic signal change interval was changed to fit with the values recommended by the Institute of Transportation Engineers (ITE) in the United States. According to the Manual on Uniform Traffic Control Devices, the yellow change interval should range from 3 to 6 seconds, and it has been previously reported that shorter change interval (mainly short yellow signals), resulted in an increased proportion of drivers entering the intersection and not clearing it during the change interval. Their study in [60] revealed that adjusting traffic signal change intervals to values close

23

to those associated with the ITE resulted in a reduction in the risk of multiple-vehicle crashes, principally those which result in injuries.

The effect of yellow timing was also studied in [61] but from the aspect of red light runners. Their research indicated that an increase of 0.5 to 1.5 seconds in the yellow duration decreases the frequency of red-light running by at least 50 percent. Moreover, it was shown that by time drivers adapt to this increase in the yellow duration; however, this adaption has no effect on the benefit on increasing the yellow duration.

The research on driver behavior indicated the most likely factors influencing red-light runners and aggressive passing behavior. It also showed the effect of changing traffic signal timing on reducing such aggressive and injury-causing behaviors. This is of importance to the current research, as it gives us a step forward in understanding the relationship between signal control and drivers' behaviors.

Driver Behavior Models

Studying drivers' characteristics and their attitudes in intersections gives a chance to building behavior models in order to more clearly comprehend and classify drivers' behaviors. Several studies have attempted to build models for describing and predicting the drivers' behaviors. Some of these models are described in this section.

Lane changing behavior of vehicles has been analyzed by some researchers. In [62], Hidden Markov Models (HMMs) were used to characterize and detect the driving maneuvers. Their focus was on lane changing behavior, which they classified into three categories: emergency lane change, normal lane change and lane keeping behavior. The data used to build the model were the steering angle, the steering force and the steering angle velocity. They were successful in their attempt of using HMMs to build driver recognition system for lane changing behavior. Another endeavor to analyze and infer lane changing behavior was carried out in [63],

24

in which a driver intent inference system was proposed, which depends on lane positional information, vehicle parameters and driver head motion. A sparse Bayesian learning methodology was used to analyze the data for lane change intent. Moreover, a vision system was used to approximate the driver intentions.

Besides detecting the lane change intent, some work was done to analyze and predict the driver's actions. For example, predicting driver's actions from an initial set of movements was achieved with 95 percent accuracy in [64], in which they adapted the expectation-maximization method with the hidden Markov Model to determine the human's intent. A Bayes net was again used in [65], but this time to make predictions about both the driver's intent and the future trajectory of the vehicle. The speed profile of a driver approaching the intersection was identified as an important feature for approximating the driver's intent. Their approach worked by comparing the trajectory of the past few seconds with the simulated driver behavior in order to estimate the posterior probabilities of the different driver intents. They classified driver intents into four categories: going straight, stopping at the stop line, turning right, and turning right but stopping at pedestrian crossing.

It is important to take into account the context of the driver at the onset of making decisions, and therefore [66] developed a machine learning framework for modeling and recognizing the driver maneuvers with an emphasis on how the driver's performance is affected by the context. Multi-modal four video signals captured the contextual traffic, the driver's head and driver's viewpoint to reflect the driving context. They also had a real-time acquisition system which records the car's brake, gear, steering wheel angle, speed and acceleration throttle signals. Graphical models, HMMs and potentially extensions were finally trained to create models for the following seven driver maneuvers: passing, changing lanes right and left, turning right and left, starting and stopping. Their models had a prediction power of 1 second before the maneuver starts taking place.

In summary, it has been shown that statistical models are successful in building models to describe and predict the driver's behavior, be it in predicting the stop/go behavior, lane-change behavior, or the general driver's intent.

Driver Behavior Classification

Driver behavior classification was attempted in some research. Support Vector Machine (SVM) classifier with a Bayesian Filter was tested by [67] to distinguish between the compliant drivers and violators based on the vehicle speed, acceleration and distance to intersection. They also used an HMM-based classifier which uses the expectation-maximization algorithm. The features used for this classifier were: vehicle speed, longitudinal acceleration, estimated time of arrival, range to intersection and the required deceleration parameter. The algorithms were successfully validated on naturalistic data and were shown to be effective in classifying drivers. The SVM-BF produced lower false positive rates on their dataset and thus it would be a good option if driver annoyance is the main concern. On the other hand, the HMM had a higher true positive rate when the rate of false positives was allowed to increase past 10%, making it more suitable if detection accuracy is a major concern. Support Vector Machines and Bayesian Filters were also used in [68] to classify "agents" into dangerous and harmless agents. An agent in their research referred to other drivers in the environment of the current driver (host). They defined a dangerous agent's goal as causing damage to the current vehicle or putting it in a risky situation, whereas a harmless agent's goal is to minimize time or distance. They identified the most critical features used in classification to be: the relative distance between the host and the suspicious vehicle, the heading of the suspicious vehicle relative to the host vehicle and the speed of the suspicious vehicle. Testing their classifier in 60 different scenarios, the classifier achieved a coverage of 100 percent and a precision of 77 percent.

Other researchers focused on the analysis of vehicle trajectory patterns. Self-organizing feature map neural networks were used by [69] to learn the characteristics of normal vehicle

trajectories. They represented trajectories using 2D image coordinates, together with first and second order motion information and a moving average smoothing. Training of the neural network was carried out on normal trajectories using video footage of a car park in normal conditions. The neural network can then be used in surveillance, whereby it marks object trajectories different from those encountered during the training phase as abnormal behavior. They achieved correct classification rate of 92% on 39 tested trajectories. Their major limitation lies in the difficulty in building prior knowledge into self-organizing networks and therefore the system was unable to differentiate between normal paths that have not been seen before and accurately suspicious trajectories. In [70] a whole vehicle trajectory was used as an input, unlike previous methods which depended only on individual flow vectors. They proposed a new method for learning patterns for anomaly detection and activity prediction using fuzzy self-organizing neural network (fuzzy SOM). They showed that the fuzzy SOM outperforms both Kohonen self-organizing feature map (SOFM) and vector quantization in terms of both speed and accuracy. In addition they achieved inspiring performance in anomaly detection (88.75% recognition rate) and activity prediction.

Driver classification research indicated that it is possible to classify drivers according to their behavior at intersection using data derived either using vehicle detectors or video cameras. It has also been shown that the two most important factors for classifying drivers are the speed and distance to intersection at the onset of the yellow signal.

Conclusion

This section has attempted to provide a review of the literature on driver's behavior. Research on driver's behavior was classified into three categories: driver behavior studies, driver behavior models and driver behavior classification. Driver behavior studies focused on determining driver behaviors which lead to red-light running and crashes at intersections. Statistical models to study and predict drivers' behavior were studied and built in the driver

behavior model research. Finally, some systems attempted to classify driver behavior at intersection into compliant drivers and violators. None of the previous research, however, tackled the classification of aggressive behavior using neural networks, and therefore this research focuses on using two different types of neural networks and compares them with SVMs.

2.6   Summary

Research work done on traffic signal control was reviewed, as well as that carried out on driver behavior. From the work done on traffic signal control, it was concluded that fuzzy logic control systems form the basis for most of the signal control systems. Variations in different fuzzy logic-based control systems arise from choosing different parameter sets, and augmenting fuzzy logic with other computational intelligence techniques. It is the concern of most traffic signal control systems to reduce traffic delays. Some systems also tackle improving traffic congestion, and others focus on improving safety.

From the review of driver's behavior work, it can be concluded that research into understanding, analyzing, modeling and classifying driver's behavior is successful. It was also shown that classifying driver's behavior from vehicle trajectory data was attempted successfully. On the other hand, there is an evident lack of research on using feedforward and recurrent neural networks for classifying aggressive driver behavior. Therefore, this gives us a chance to test building aggressive driver classifier using these two different kinds of neural networks and compare them with the performance of SVMs.

It is also remarkable that no sufficient research has been carried out to test integrating driver behavior classification with traffic signal control. This provides us with an opportunity to incorporate driver behavior classification into traffic signal control in order to reduce delays.

CHAPTER 3 PROPOSED TRAFFIC SIGNAL CONTROL ARCHITECTURE

It is the aim of this thesis to design and test a new traffic signal control system for a two-approach single intersection which incorporates within it aggressive driver behavior classification in an attempt to reduce the intersection delays. In order to achieve this, it is critical to include two main subsystems in the designed architecture: aggressive driver behavior classification and traffic signal control. The choice of the methodologies to implement and integrate the two-subsystems is paramount and, accordingly, the proposed architecture has to be carefully designed to mitigate unnecessary or unpleasant results.

In this section, the proposed architecture is discussed and the different components are described. The evaluation measures used to evaluate the overall system are also defined.

3.1   Proposed Architecture

Integrating driver behavior classification within the framework of traffic signal control is challenging, as little work has been done in this area. Figure 6 shows the overall proposed architecture, which consists of three main modules: **vehicle data collection**, either through video cameras and tracking program or through vehicle detectors placed on roads, **driver behavior classification** and finally **fuzzy traffic signal control**. The main focus in this research is on the two modules of driver behavior classification and fuzzy traffic signal control. Each of these modules with their sub-architectures are described the following sections.
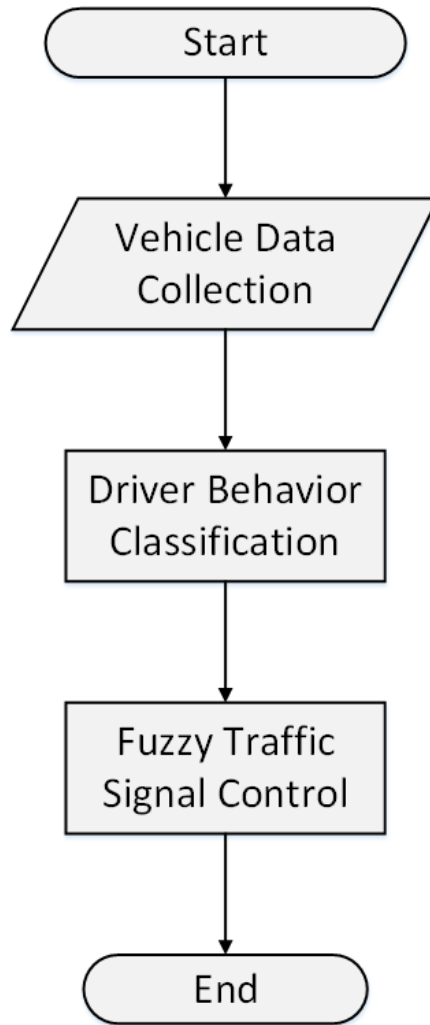
Figure 6 - Proposed Architecture

3.1.1 Vehicle Data Collection

For intelligent traffic signal control, status of the current traffic has to be collected and analyzed in order to assign traffic signals optimally. The standard approach of collecting vehicle data is through the use of inductive loop detectors/sensors buried under the road surface that detect vehicles passing over them. The problem of this method, however, is that it is intrusive and does not have the ability to capture significant information about the vehicle's behavior. Therefore, it could be accompanied with sensors that would capture the necessary information, such as speed, acceleration … etc. Another method considered by current traffic signal approaches consider computer vision techniques as an alternative to vehicle detectors. Vision-driven techniques are useful because video sensors can be easily installed, operated and

30

maintained [71]. More importantly, video sequences are a rich information source and can be analyzed to provide essential information regarding the state of the traffic, such as traffic object detection, tracking and recognition, as well as collision of vehicles [72]. Since this research considers driver behavior detection closely, video data input methodology would be a suitable method to provide more useful information for the remaining modules. However, due to the difficulties of collecting reliable data in suitable time period for the purpose of testing the feasibility of the research, publicly available vehicle data is sought.

Since the current research focuses on a single intersection with two approaches (North – South, and East – West) two video cameras should be installed, one in each approach, as shown in Figure 7.
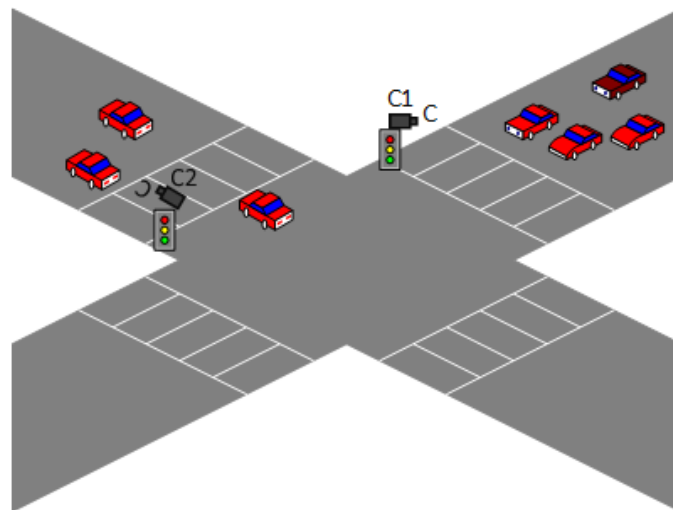
Figure 7 - Intersection Configuration

3.1.2 Driver Behavior Classification Module

The purpose of adding the driver behavior classification module is to incorporate driver's behavior analysis into traffic signal control. This is one of the critical modules in the proposed architecture and together with the traffic signal control module form the main focus of the current research. After reviewing the literature in the area, one can conclude that there is no

similar work to the one intended, and therefore experimentation needs to be carried out to reach to the best performance.

The proposed approach is to firstly classify aggressive driving behavior and then use classification output as an input to the traffic signal control module. In order to build the classifier, vehicle trajectory data containing aggressive behavior is essential. A major obstacle was to find publicly available dataset suitable for developing aggressive driver classification models. After extensive search, the 100-car naturalistic driving study [73] dataset was found to provide useful details for the purpose of this research. It is believed that this research is the first of its kind to be performed on this dataset. The following sections first start by describing the dataset and the labelling performed in order to prepare the data for use by the proposed classifier. Then the proposed architecture of the classifier is described.

Dataset Description

The 100-car study is considered the first instrumented-vehicle study performed to collect large-scale naturalistic driving data. In the study, the drivers were not given any special instructions and no experimenter was present. 100 vehicles were used in the study and each vehicle's data was collected over 12 to 13 months [73]. This resulted in 2,000,000 vehicle miles and 43,000 hours of driving data. The publicly available part of the dataset contains a compilation of detailed video analysis for 68 crashes and 760 near crashes. The data consists of multiple datasets and documents, of which only 3 were found useful for our study. The event narratives document contains narratives about detailed situation context for each crash and near crash. A time series data file is provided for each crash and near crash, and is a comma-delimited text file containing the sensor data collected during the event. Finally the event video reduced data contains details of the video reduced data for each event.

Data Labelling

The 100 car event narratives file provides a written description of each of the crash and near-crash events. The written description is a narrative about the environment and the subject, and is extracted from the event videos. This file is used in order to manually label aggressive and non-aggressive behaviors.

Firstly a list of aggressive behavior is constructed and according to which labelling is done. The list mainly focuses on the aggressive behaviors described by researchers [22], [74] and included the most frequent aggressive behaviors. In their study, Shinar and Compton [74] observed more than 2000 aggressive driving behaviors over a total of 72 hours at six different sites. They indicated that the most frequently occurring aggressive behavior is cutting across a single lane, followed by honking, and then cutting across multiple lanes and passing on the shoulders. According to the National Highway Traffic Safety Administration (NHTSA) aggressive driving includes "following too closely, changing lanes without caution or signal, running a red light, or improper passing" [75]. In their surveys, increased speed limits and tailgating were also identified as a form of aggressive driving. Therefore, the following non-exhaustive list of behaviors is used to label aggressive cases:

- Excessive speeding;
- Tailgating or following too closely;
- Cutting off;
- Running red light.

Other cases of aggressive driving are assessed according to the described situation. Figure 8 and Figure 9 show examples of records classified as aggressive and non-aggressive respectively.

Subject vehicle **pulls off of the road on the right side onto the shoulder**, then attempts to rejoin traffic by pulling back into the right lane. The first time the subject vehicle begins to cross over into the lane, she must pull back in order to avoid sideswiping the vehicle to her left in that lane (it appears that she thought that the vehicle in the lane would let her in, but it didn't).
Subject driver is on an undivided two lane road and approaching a flat bed truck that is stopped/parked in the lane ahead. It appears that the subject driver has time to slow to a stop behind the truck, but he **crosses the double line on the left to pass** instead. As the subject driver is passing the truck an SUV approaches head-on. Subject vehicle nearly hits the oncoming vehicle head-on.
Subject vehicle is in left lane when the driver realizes that he needs to take a right exit. **He suddenly changes into the lane** to the right, nearly hitting a lead vehicle in the rear, and immediately continues into the right exit lane,

Figure 8 - Examples of behaviors classified as aggressive

Subject driver is passing a vehicle in the right lane that has slowed down. That vehicle steers a little into the subject driver's lane. The subject driver decelerates quickly and steers a little to the left to avoid hitting that vehicle.
Subject driver is traveling in a middle lane and a car in the left lane tries to change into the middle lane almost sideswiping the subject's vehicle. Subject driver has to apply brakes to avoid hitting the other vehicle.
Subject driver is in the right lane and changing into the middle lane when a vehicle in the far left lane tries to change into the middle lane as well. Subject driver applies brakes and steers to the right to avoid being hit.

Figure 9 - Examples of behaviors classified as non-aggressive

Following the labelling of each event's behavior (crash/near crash) as aggressive or non-aggressive, the time series data is processed to extract the relevant fields. The time series data for each event is a comma-separated text file spanning at most 30s before the event and 10s after the event. This data was obtained in the study from the sensors attached to each vehicle. These sensors included, among others, forward and rearward radar, lateral and longitudinal accelerometers, GPS, and five channels of compressed digital video. Data was recorded every 0.1 second, therefore each second consisted of 10 rows of data. The variables collected cover time, gas pedal position, vehicle speed, yaw rate, heading GPS, lateral acceleration, longitudinal acceleration, types of lane markings, radar IDs, light intensity of the environment, brake signal state and turn signal state.

Proposed Aggressive Behavior Classifier Architecture

The main aim of this classifier is to test the feasibility of classifying aggressive and non-aggressive driving behavior from the time series data provided by the 100-car naturalistic driving study. Figure 10 shows an outline of the proposed architecture. The input to the system is the

event time series data obtained from the 100-car naturalistic driving study. Some preprocessing is firstly applied to the data to make it better suited for classification purposes.
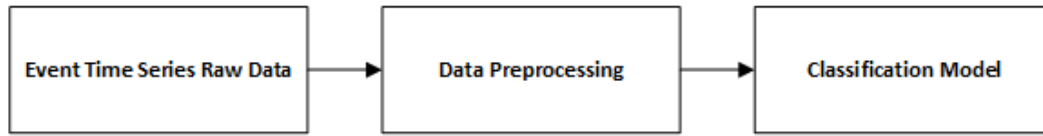
Figure 10 - Proposed Classifier Architecture Outline

Data preprocessing mainly comprises data aggregation, data reduction and normalization. **Aggregation** is achieved through time smoothing each second's 10 entries, thus producing a single entry per second rather than 10 separate entries. Data averaging aims to reduce the data size, as well as reduce noisy data produced from the sensors. Then **data reduction** is carried out by reducing the initial feature set. Some of the fields are meaningless to include in the classification, such as the type of lane markings (double line, single line, etc.) and radar station ID, therefore the irrelevant fields were removed. The feature set, therefore, started with the following 10 features: vehicle speed, yaw rate, heading GPS, lateral acceleration, longitudinal acceleration, left distance to lane markings, right distance to lane markings, light intensity, brake on/off, and turn signal state. The samples were also filtered to remove any incomplete data (having less than 30 seconds of data prior to the event). After labelling and filtering the samples, there were a total of 788 trajectory records, of which 582 were labelled non-dangerous and 206 were dangerous. Finally normalization was carried out to avoid dependence on the measuring unit, and to give each attribute an equal weight. Each attribute was scaled to fall in the range [-1, 1]. Equation 4 maps a value, x_i, of attribute A to x_i' in the range [-1, 1], where min_A and max_A are the minimum and maximum values of attribute A.

$$x_i' = \frac{2 * (x_i - min_A)}{max_A - min_A} - 1$$

Equation 4 - Normalization equation

Lastly, data is passed to the final stage of classification. In classification, data is randomly divided into training, validation and testing sets. The ratio used for classification was 70% training, 15% validation and 15% testing. Validation set is used while training the network/classifier on the training set, in order to test the accuracy of training so far and to prevent over fitting of the classifier to the training data. On the other hand, the testing set is used after training is over, in order to compare the performance of the different classifiers. Three different classification algorithms were tested: feedforward neural networks, long short-term memory recurrent neural networks and support vector machines.

3.1.3   Fuzzy Traffic Signal Control

Pappis and Mamdani's [13] traffic signal approach is the first traffic signal controller to use fuzzy logic. It is chosen in this research for implementation, in order to test the addition of the driver behavior classification module to it. The reason this approach is chosen and not one of the more recent fuzzy logic systems is that a totally new parameter is to be tested, and it is thought that to test it on the basic controller is an essential step before testing it on one of the more advanced approaches. Some details of the controller and the proposed modifications are described in this section.

Proposed Model

A model similar to Pappis and Mamdani's model is constructed and it follows the following assumptions:

1. Arrival of vehicles at the intersection is considered to be random; that is, arrival times are uniformly distributed.

2. Total lost time of 10s/cycle is assumed.

3. No turning traffic.

www.manaraa.com

Intersection Design and Vehicle Arrival Pattern

A simple intersection with two approaches is to be used: North – South and East – West, as shown in Figure 11. Each approach shall contain two detectors as shown in Figure 12, where detectors collect information about the vehicles and send them to the traffic controller. Vehicle arrival pattern at the intersection is to follow the pattern described by Pappis and Mamdani. According to Pappis and Mamdani, their model consisted of randomly arriving vehicles. They had 16 different configurations of mean arrival rate in the N – S and E – W traffic as shown in Table 2.
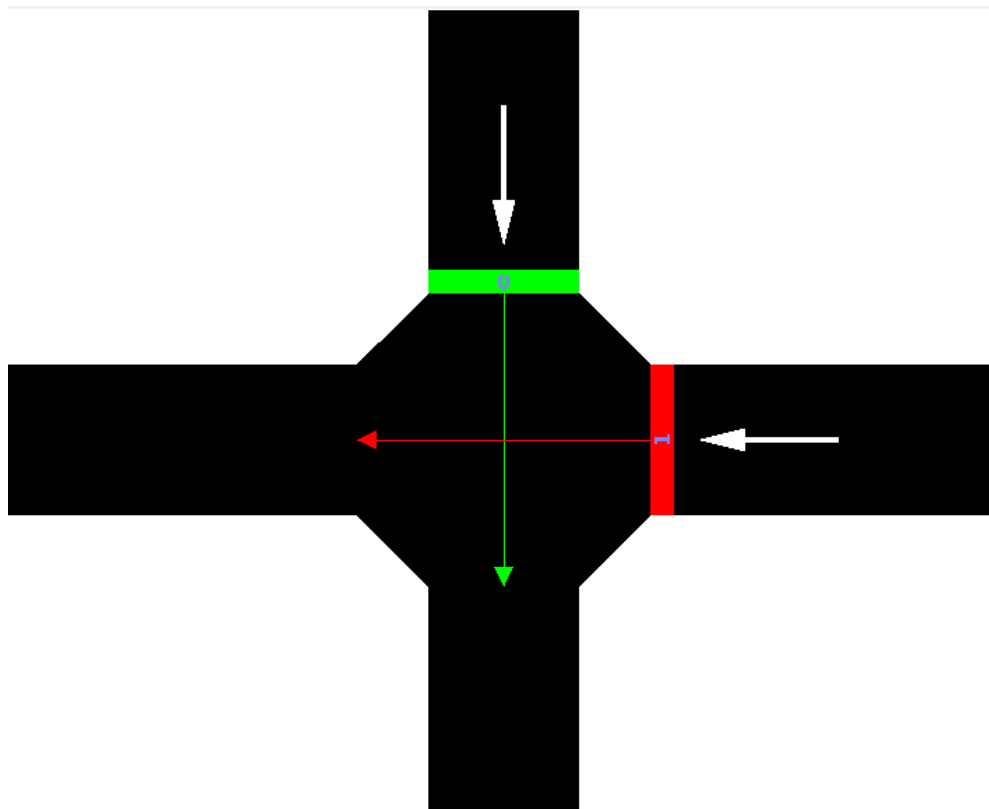
Figure 11 - Intersection design
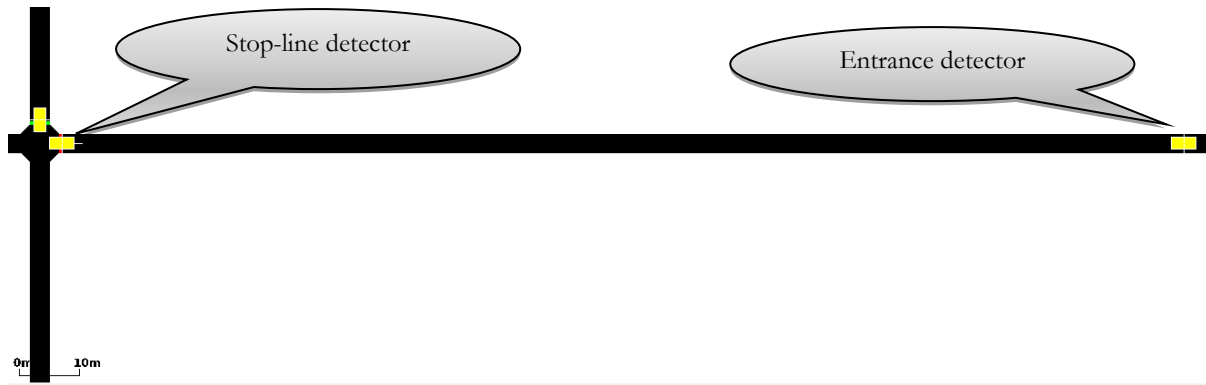
Figure 12 - Detectors' settings

Table 2 - Arrival rates

N – S Traffic Mean Arrival Rate (veh/hr)	E – W Traffic Mean Arrival Rate (veh/hr)
360	360
360	720
360	1080
360	1440
360	1800
360	2160
360	2520
720	720
720	1080
720	1440
720	1800
720	2160
1080	1080
1080	1440
1080	1800
1440	1440

Proposed Data Flow

Every second, data about the vehicles should be collected in order to use them for classification. This is achieved by keeping a **trackedVehicles** list, which contains the vehicles that are being tracked in the approach having right of way. Once a vehicle passes through the

38

entrance detector, its data is collected and stored in the list. The three variables collected are: vehicle speed, acceleration and brake signal status. If the data collected for any vehicle exceeds 5 seconds, then the earliest entry is deleted, keeping the most recent 5 seconds' data. The flowchart in Figure 13 shows what should typically happen every second before the controller intervention.
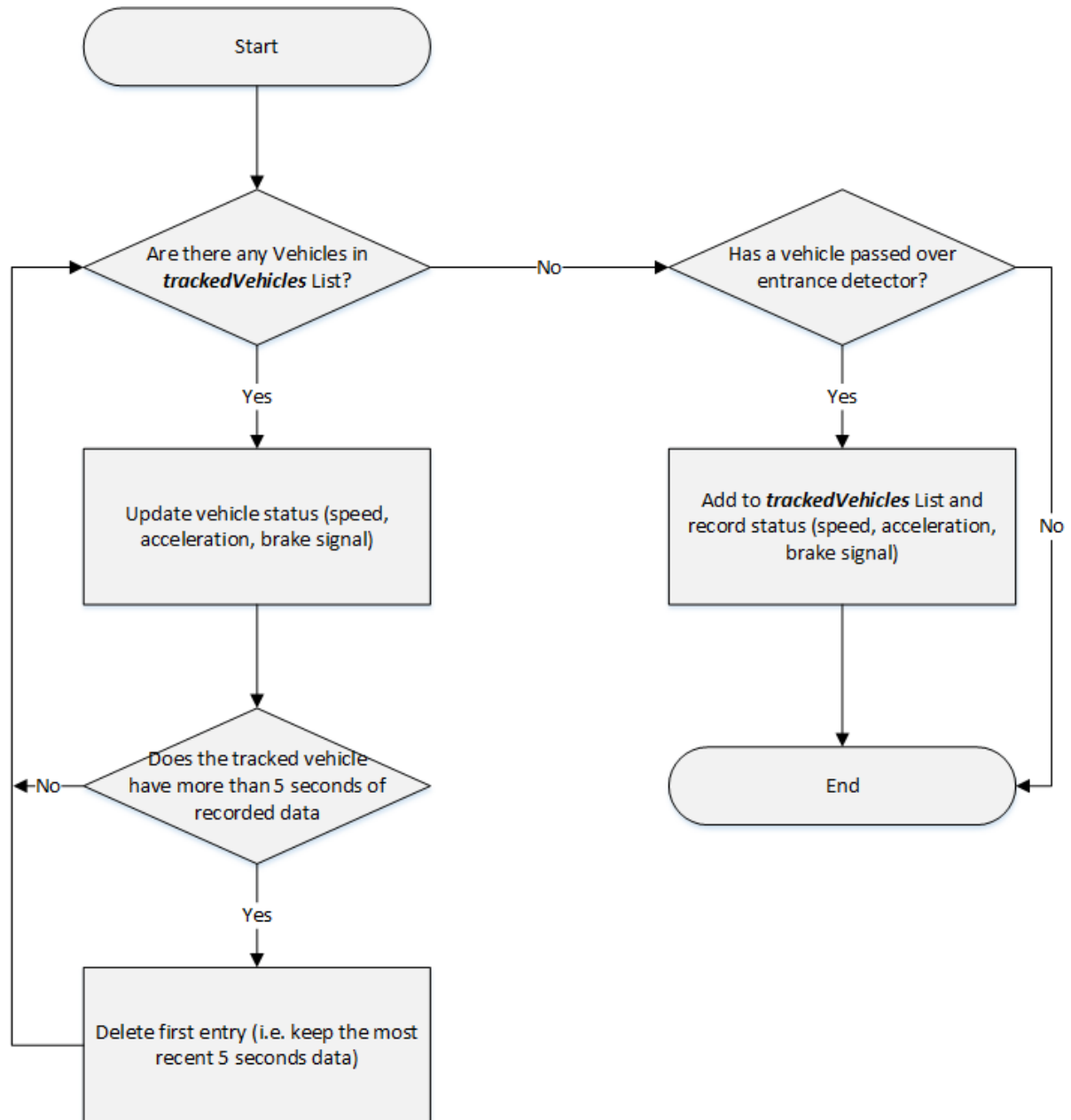
Figure 13 - Recording Vehicles' Statuses

Proposed Fuzzy Linguistic Variable

Mamdani's fuzzy logic controller originally uses two input arrays: **alpha** and **alpha_dash**. Alpha contains information about vehicles expected to cross the critical point in each of the next 10 seconds, while alpha_dash contains information about vehicles added to the queue in each of the next 10 seconds. In the proposed architecture, calculation of alpha and alpha_dash takes place just before the controller intervention, and is summarized in Figure 14.
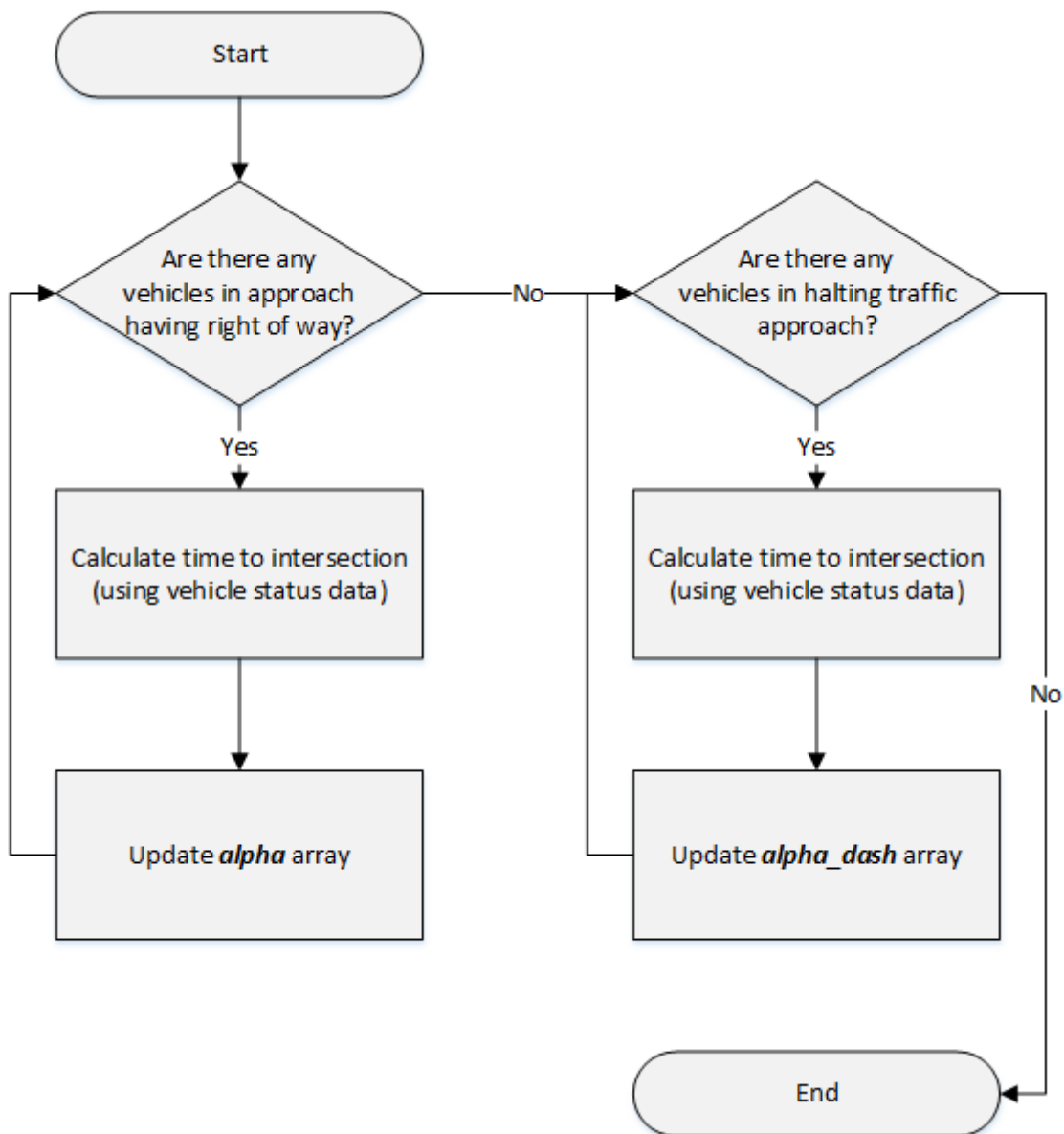
Figure 14 - Calculation of Alpha and Alpha_dash arrays

The additional linguistic variable that shall be added to the controller is **Aggressiveness**. This represents the ratio of aggressive drivers present in the arriving approach at a given point in time. Therefore, an additional controller input array (**alpha_aggressive**) is to be constructed. An entry in the **alpha_aggressive** array is calculated as shown in Figure 15. From this array, another array called (**beta_aggressive**) is constructed and is an accumulation of the values in **alpha_aggressive** array, as depicted in Figure 16.

```
1. For each entry (alpha[i]) in array alpha:
   a. If alpha[i] is 1:
         i. Use the aggressive driver behavior classifier and the values
            retrieved from simulation to determine if the driver is an
            aggressive driver.
        ii. If driver is aggressive:
               ▪ alpha_aggressive[i] = 1
       iii. else
               ▪ alpha_aggressive[i] = 0
```

Figure 15 - Calculation of alpha_aggressive array

```
1. For each entry (alpha_aggressive[i]) in array alpha_aggressive:
   a. If this is the first entry:
         ▪ num_aggressive = alpha_aggressive[i]
         ▪ beta_aggressive[i] = num_aggressive/beta[i]
   b. else:
         ▪ num_aggressive += alpha_aggressive[i]
         ▪ beta_aggressive[i] = num_aggressive/beta[i]
```

Figure 16 - Calculation of beta_aggressive array

A membership function similar to that used by Mamdani for the arrival variable is proposed for the aggressiveness variable and is shown in Table 3 and Figure 17. Also a more than operation is defined on the variable and is presented in Table 4 and Figure 18.

Table 3 - Membership Functions of Aggressiveness Variable

Fuzzy Sets	Ratio of Aggressive Drivers										
	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
not dangerous	1.0	0.5	0.2	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
slightly dangerous	0.5	1.0	0.5	0.2	0.1	0.0	0.0	0.0	0.0	0.0	0.0
dangerous	0.2	0.5	1.0	0.5	0.2	0.1	0.0	0.0	0.0	0.0	0.0
very dangerous	0.1	0.2	0.5	1.0	0.5	0.2	0.1	0.0	0.0	0.0	0.0
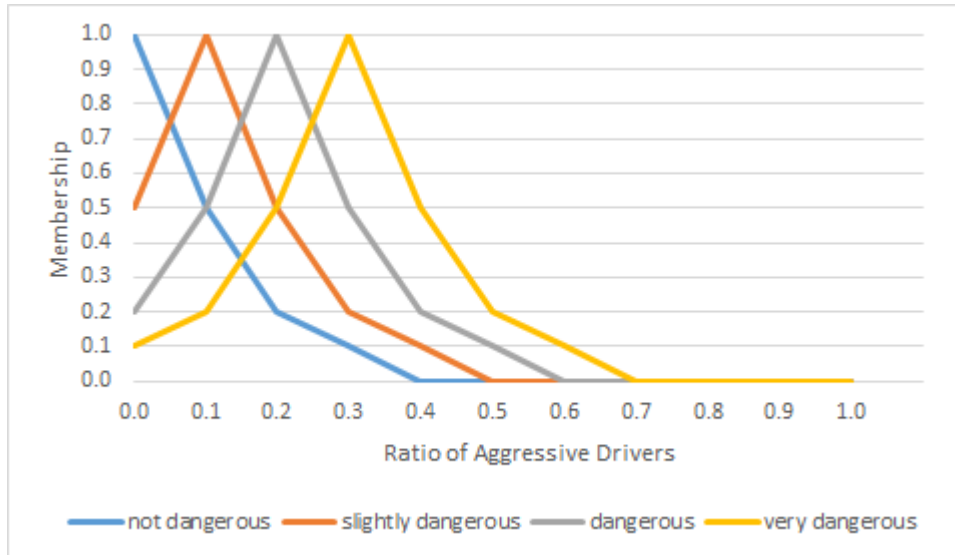
Figure 17 - Membership Functions of Aggressiveness Variable

Table 4 - "More than" operation on the Aggressiveness Variable

Fuzzy Sets	Ratio of Aggressive Drivers										
	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
mt(not_dangerous)	0.0	0.5	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0
mt(slightly_dangerous)	0.0	0.0	0.5	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0
mt(dangerous)	0.0	0.0	0.0	0.5	0.8	0.9	1.0	1.0	1.0	1.0	1.0
mt(very_dangerous)	0.0	0.0	0.0	0.0	0.5	0.8	0.9	1.0	1.0	1.0	1.0
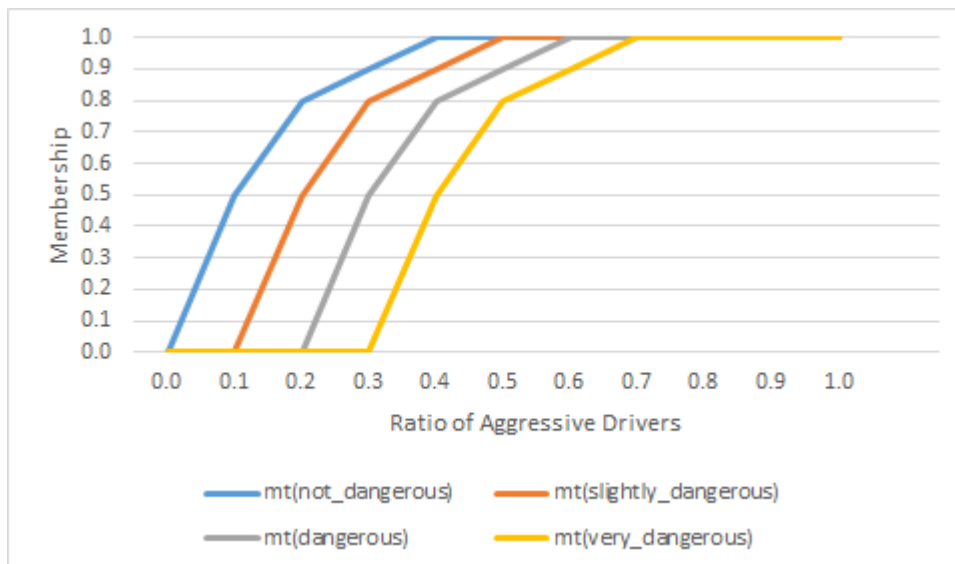
Figure 18 - "More than" operation on the Aggressiveness Variable

Proposed Rule Set

Mamdani's linguistic rule set was modified to accommodate for the added linguistic variable. The proposed 25 (5x5) rules are given in APPENDIX I – FUZZY LOGIC CONTROLLER RULE SET.

Traffic Controller Proposed Operation

The proposed operation of the controller is also similar to Pappis and Mamdani's controller, with the addition of the aggressiveness ratio input. It can be summarized as follows:

1. The intervention of the controller takes place **every 10s** during each phase's effective green period.

2. The first intervention takes place just after the **first 7s** of the period.

3. Decision taken:

   a. **Length of extension** of the effective green time for the phase having the right of way.

4. **Control input parameters**: three arrays corresponding to:

   a. Halted traffic (**alpha_dash** array),

   b. Traffic having the right of way (**alpha** array), and

   c. Ratio of aggressive drivers (**alpha_aggressive** array)

5. The fuzzy algorithm employed at each intervention for deciding the control action is represented by the union of five arrays (each array corresponds to a fuzzy rule, which is represented by a five-dimensional array).

6. Maximum of five interventions take place at: 7th, 17th, 27th, 37th, and 47th s. Thus the maximum possible effective green time is 57 s.

7. At each intervention the five rules are invoked (10 times).

8. At each intervention data is available for each of the next 10s.

9. Process of inferring the control action:

43

a. For each successive time unit (1s) for the next 10s, data concerning vehicles crossing the critical point, vehicles added to the queue, and the ratio of aggressive drivers in the approach having right of way are used as input to the algorithm array in use.

b. The extension will be selected which corresponds to the maximum degree of confidence.

3.1.4   Key Feature and Challenges

It might appear that each single module in the given architecture is simple and that various research has been done to efficiently design it. However, the key challenge in our approach is to incorporate driver behavior classification into traffic signal control of a single two-approach intersection in order to reduce delays. Incorporating this module results in the following challenges that have to be tackled by our proposed architecture:

1. Preparing the 100-car data for use by classification algorithms.
2. Designing an efficient classifier which classifies the aggressive driver behavior.
3. Using the output from the driver behavior classification module as an input parameter to the fuzzy logic control system.

3.2   Performance Evaluation Measures

Evaluation of the proposed architecture is important as it provides an indication to the effectiveness of our approach and allows comparison against the existing approaches. The two main contributing modules from are the driver behavior classification module and the signal control module. The following outcomes should be assessed:

1. Correctly classifying driver behavior

2. Reduced intersection delay

Each of these is described in the coming sections.

3.2.1 Evaluation of Driver Behavior Classification

In order to compare the different classification models experimented in this research, the Receiver Operating Characteristics (ROC) curve is used. According to [76] an ROC curve is a "technique for visualizing, organizing and selecting classifiers based on their performance". In the recent years, the use of ROC curves in machine learning community has increased, owing in part to the poor performance of the simple classification accuracy metric [76]. An ROC curve is plotted on a two-dimensional graph, with the true positive rate (TPR) plotted on the Y axis, and the false positive rate (FPR) plotted on the X axis. A true positive is a testing instance whose true class is positive, and is classified as positive as well. On the other hand, a false positive is a testing instance whose true class is negative, but is classified as positive. Equation 5 and Equation 6 show the formulae for calculating the TPR and FPR respectively. For the purpose of this research the positive class is the aggressive behavior, while the non-aggressive behavior is the negative class, with the probability of aggressiveness used as the threshold.

$$TPR = \frac{number\ of\ true\ positives}{total\ number\ of\ positive\ samples}$$

Equation 5 - True Positive Rate

$$FPR = \frac{number\ of\ false\ positives}{total\ number\ of\ negative\ samples}$$

Equation 6 - False Positive Rate

ROC curves act as a good visual tool comparing the different classifiers' performance, however, it would be easier if there is a single scalar number comparing performance. This scalar value can be extracted from the ROC and is the area under the curve (AUC). A classifier with a

45

higher AUC is expected to have a better average performance. Figure 19 shows an example of an ROC curve with the shaded area representing the area under the curve. AUC is always a value between 0.0 and 1.0.
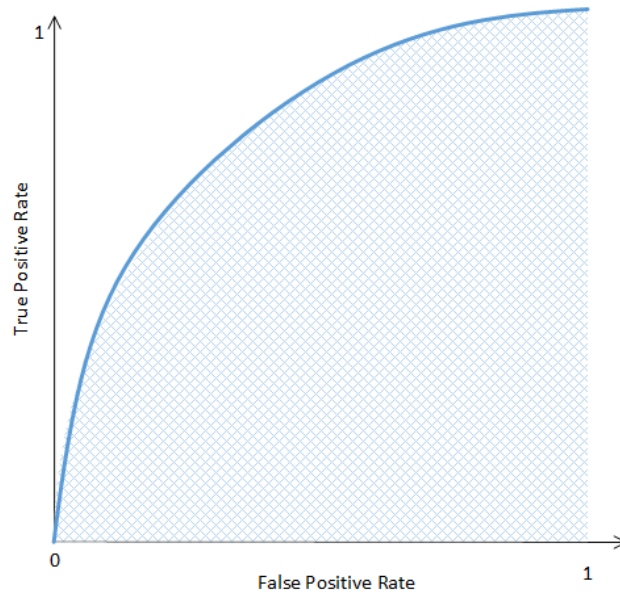
Figure 19 - An example of an ROC curve

3.2.2 Evaluation of Intersection Delay

According to [13], intersection delay is defined as the number of seconds a vehicle stayed inside the intersection. In order to compare delay from our approach with the other approaches, different configurations of traffic flow and drivers' behavior are used and delay calculated for each configuration. The results are then compared with the basic approach to traffic control.

In order to calculate the delay, two detectors are to be placed in each direction; one at the beginning of the intersection (around 11.5 seconds away from the traffic light) and another detector at the stop line close to the traffic junction as shown in Figure 12. Data collected from these two detectors would allow the calculation of the number of seconds spent inside the intersection for each vehicle.

46

Two types of files were generated from the simulator SUMO to calculate the delay:

1. **Traffic light switch state**: provides information about the states of the traffic light signal when it is changed.

2. **Detector's Output**: provides information about vehicles passing over the detector.

Figure 20 outlines a pseudo-code for calculating the intersection delay.

```
1. Open traffic light switch state file, and from it retrieve the starting
   phase (N-S or E-W) ~ file1
2. Open the four detectors' files ~ detect_in, detect_out (for each
   direction)
3. Loop until end of data:
     a. Initialize vehicleCount to 0, which represents the number of
        vehicles currently in the intersection
     b. Initialize totalDelayPerVehicle to 0, which represents summation of
        delay per vehicles across the cycles
     c. For each cycle:
          i. Retrieve cycle start time and end time of the cycle from
             file1.
         ii. Initialize delay to 0, which represents the total delay so
             far
        iii. Count number of vehicles which passed over in and out
             detectors during this cycle period (vehicleIn,VehicleOut)
         iv. Update delay as follows:
             1. delay = delay + vehicleCount − vehicleOut
             2. vehicleCount = VehicleCount + VehicleIn − VehicleOut
     d. At end of the cycle:
          i. delayPerVehicle = delay / numberOfVehicles
4. At end of all cycles:
     a. DelayPerVehiclePerCycle = totalDelayPerVehicle / numCycles
```

Figure 20 - Delay calculation pseudo-code

3.3 Summary

In order to achieve our objective of reducing intersection delays in a single two-phase intersection, integration of driver behavior classification is proposed. Data is firstly extracted from a single intersection with two approaches, and this data will be input to the driver

47

classification module together with the signal control module. Output from the driver classification module will also be input to the signal control module to give the final output of signal control. This section provided details of the proposed architecture.

48

CHAPTER 4 EXPERIMENTS AND RESULTS

In order to examine the feasibility and test the accuracy of the proposed approach, multiple experiments were conducted. Firstly a set of experiments were conducted for the aggressive driver behavior classifier module to test the performance of the different classification algorithms on the 100-car study data, and choose the best architecture for incorporation with the fuzzy logic controller. The second set of experiments targeted building the basic Mamdani's fuzzy logic traffic controller and comparing it with the static and actuated controllers. Finally, integration of the aggressive behavior classifier and fuzzy logic controller was attempted in the third set of experiments. Each of these experiments with their performance evaluation is discussed in the coming sections.

4.1   Experiment Set 1: Building an Aggressive Driver Behavior Classifier

This set of experiments aim to build a good aggressive driver classifier using the preprocessed 100-car study data. After preprocessing the 100-car study data to make it suitable for use by machine learning algorithms, as described in Section 3.1.2   Driver Behavior Classification Module, different algorithms were tested. These algorithms were: feedforward artificial neural networks (ANN), long short-term memory recurrent neural networks (LSTM RNN) and support vector machines (SVM). Experiments for each of these algorithms are described hereafter.

4.1.1   Artificial Neural Network Experiments

Objective

The objective of the ANN experiments is to test different data and network configurations, in order to conclude which combination of configurations yields the best performance.

Method

Several experiments were carried out testing different data configurations, in order to reach to the best classification performance. The different configurations tested the effects of 1) data averaging; 2) reducing the number of features and 3) using different time spans as shown in Table 5. In each of the experiments, multiple neural network architectures were tested and the ones giving best performance used for comparison. In most cases, the architecture giving the best performance had an input layer, two hidden layers, and an output layer. The transfer function used in the hidden layers was the hyperbolic tangent, while that used for the output layer was the softmax function. Training was done using the Levenberg-Marquardt training algorithm using a learning rate of 0.01, and using the mean square error as the performance function. If the network performance on the validation test did not improve for 6 epochs, training was then stopped. Experiments were carried out using Matlab Neural Network toolbox [77].

Table 5 - The different data configurations experimented

Exp#	Time span (sec)	Averaged/Non-averaged Data	Num Features per second	Vehicle Speed	Longitudinal Acceleration	Lateral Acceleration	Yaw Rate	GPS Heading	Brake signal on/off	Lane Distance Left	Lane Distance Right	Light Intensity	Turn Signal State
1	30	Non-averaged	10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2	30	Averaged	10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
3	30	Averaged	9	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓
4	30	Averaged	8	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓
5	30	Averaged	6	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓
6	30	Averaged	3	✓	✓	✓	✗	✗	✗	✗	✗	✗	✓
7	25	Averaged	10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
8	20	Averaged	10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
9	15	Averaged	10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
10	10	Averaged	10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
11	5	Averaged	10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Input to the neural network consisted of all the time series data combined together to form a single tuple for each of the 788 events/records. Taking the averaged data as an example, each event had 30 seconds of averaged recorded data. Each second's data consisted of 10 features: vehicle speed, yaw rate, heading GPS, lateral acceleration, longitudinal acceleration, left distance to lane markings, right distance to lane markings, light intensity, brake on/off, and turn signal state. This makes a total of 30x10 features, i.e. 300 features for each of the classified input data (event). Therefore the number of neurons in the input layer would be 300 in this case. Figure 21 shows the basic structure of the neural network used.
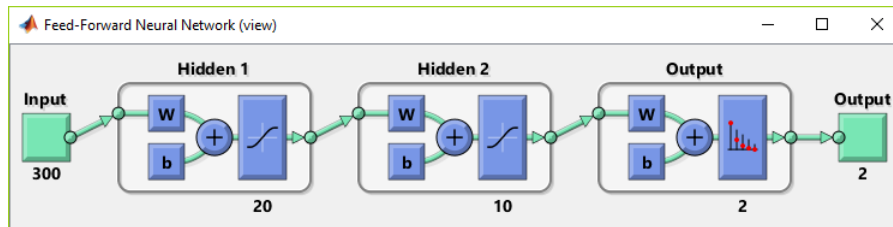
Figure 21 - **Structure of the basic feedforward NN used**

Exp1.1.1: NN Effect of Feature Averaging

Firstly, the effect of data averaging was tested by comparing the performance of averaged vs. non-averaged data. Figure 22 presents an ROC curve showing the effect of data averaging. The curve for the averaged data shows much better performance than the non-averaged data.

51
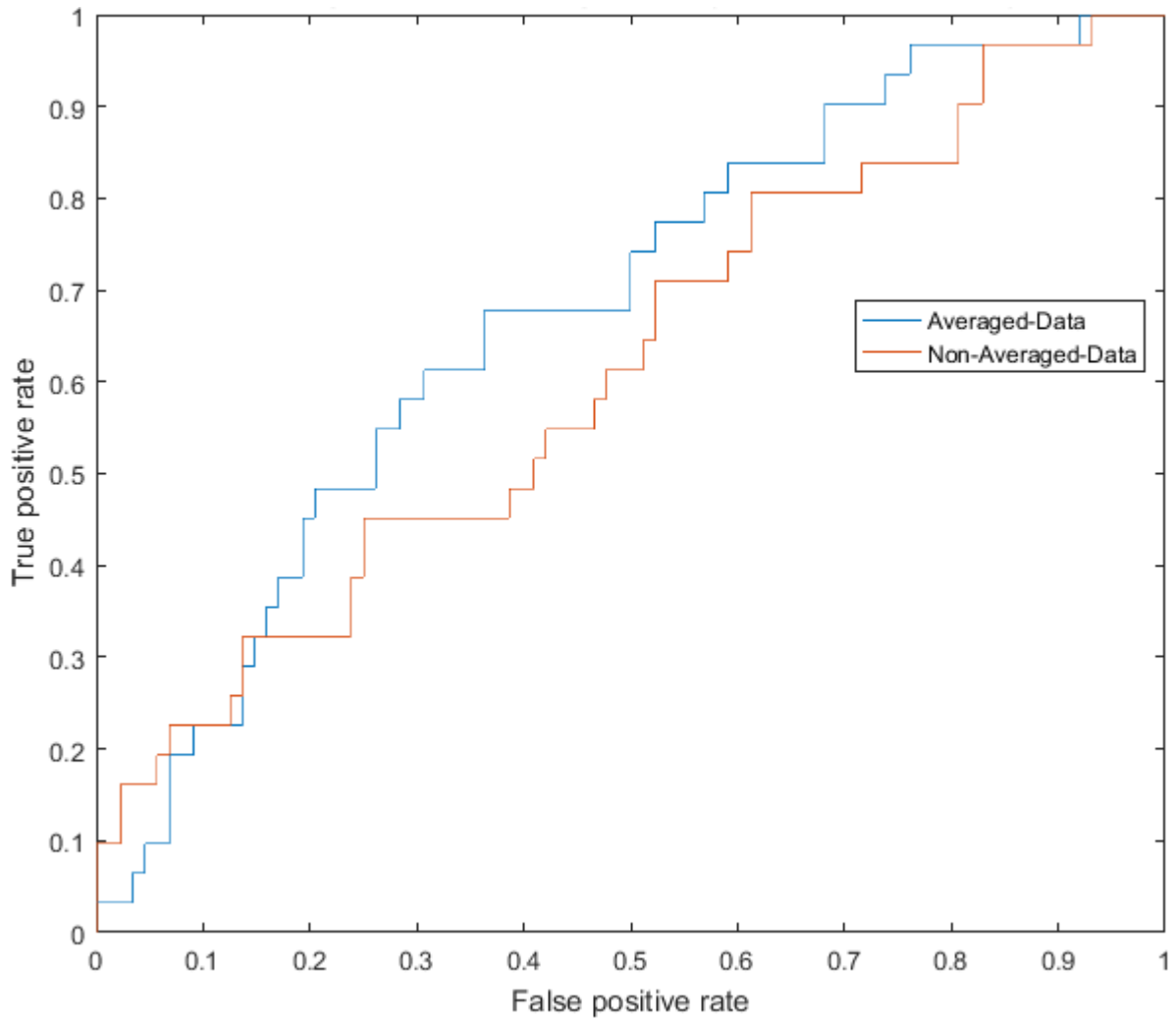
Figure 22 - NN Effect of data averaging

Exp1.1.2: NN Effect of Different Number of Features

Secondly, the effect of the number of features was examined. The weights of the neural network trained with the 10 features per second were analyzed to provide an indication about the importance of each feature. The least significant feature (the one with the lowest weight) was found to be the left distance to the lane, therefore it was removed and a new network was designed to test the remaining 9 features. Similarly different networks were designed for 8, 6 and 3 features. Figure 23 shows the effect of reducing the number of features per second. It can be observed that the best performing classifier is that trained on the whole 10 features. However, the 9 and 8 features per second networks show worse performance than the 6 and 3 features per

52

second. This indicates that performance is not necessarily associated with more number of features, but with a certain combination of features.
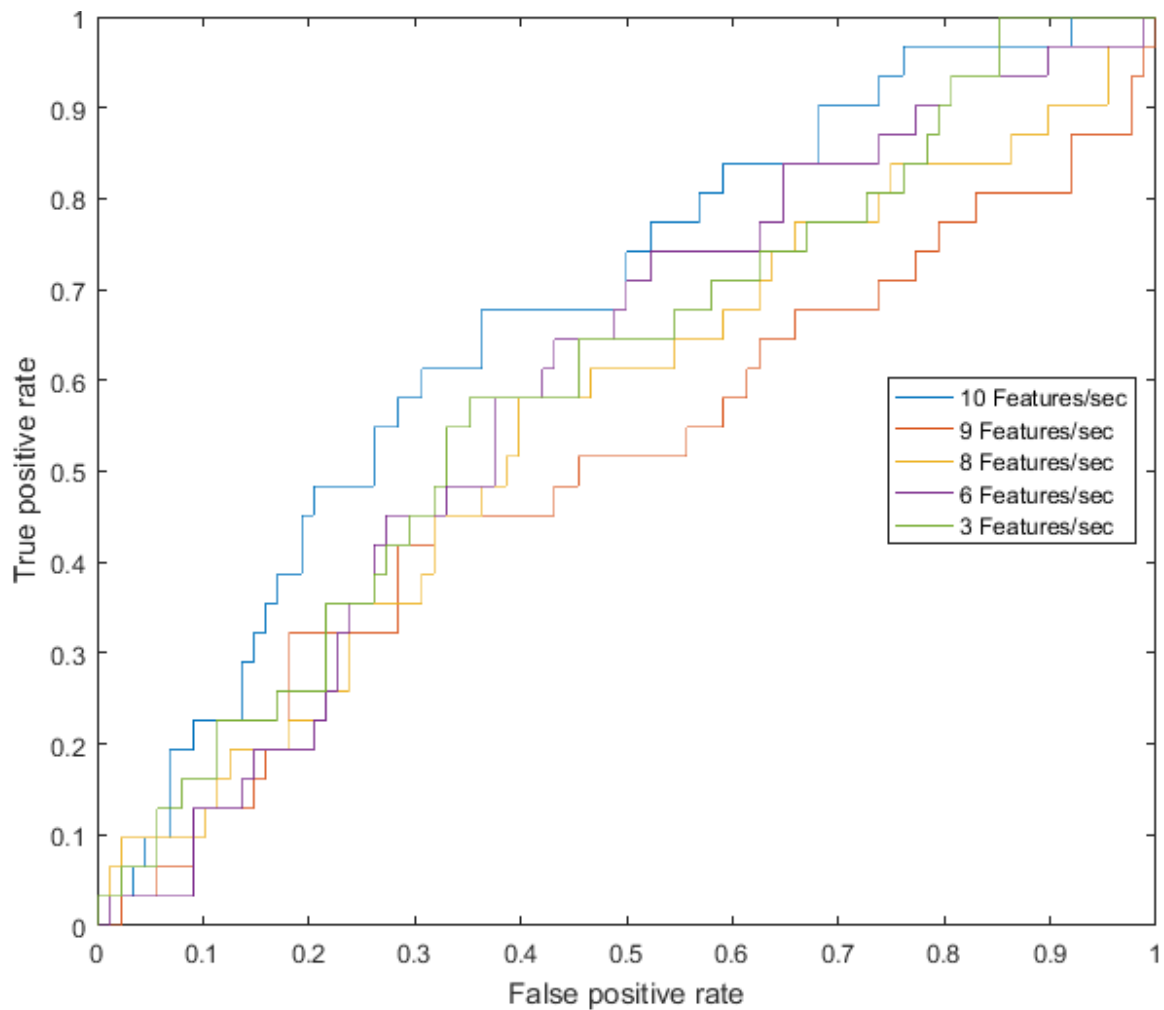
Figure 23 - NN Effect of using a subset of the features

Exp1.1.3: NN Effect of Different Timespans

Finally, the effect of time span was experimented. Time span is defined as the number of seconds before the crash/near crash event occurred. 5 different time spans, besides the 30s span, were investigated: 25, 20, 15, 10 and 5 seconds. From Figure 24, it is again evident that the best performing network is the one in which the time span was the whole 30 seconds. This is also confirmed when comparing the area under the curves for the different set of experiments, as shown in Figure 25.

53
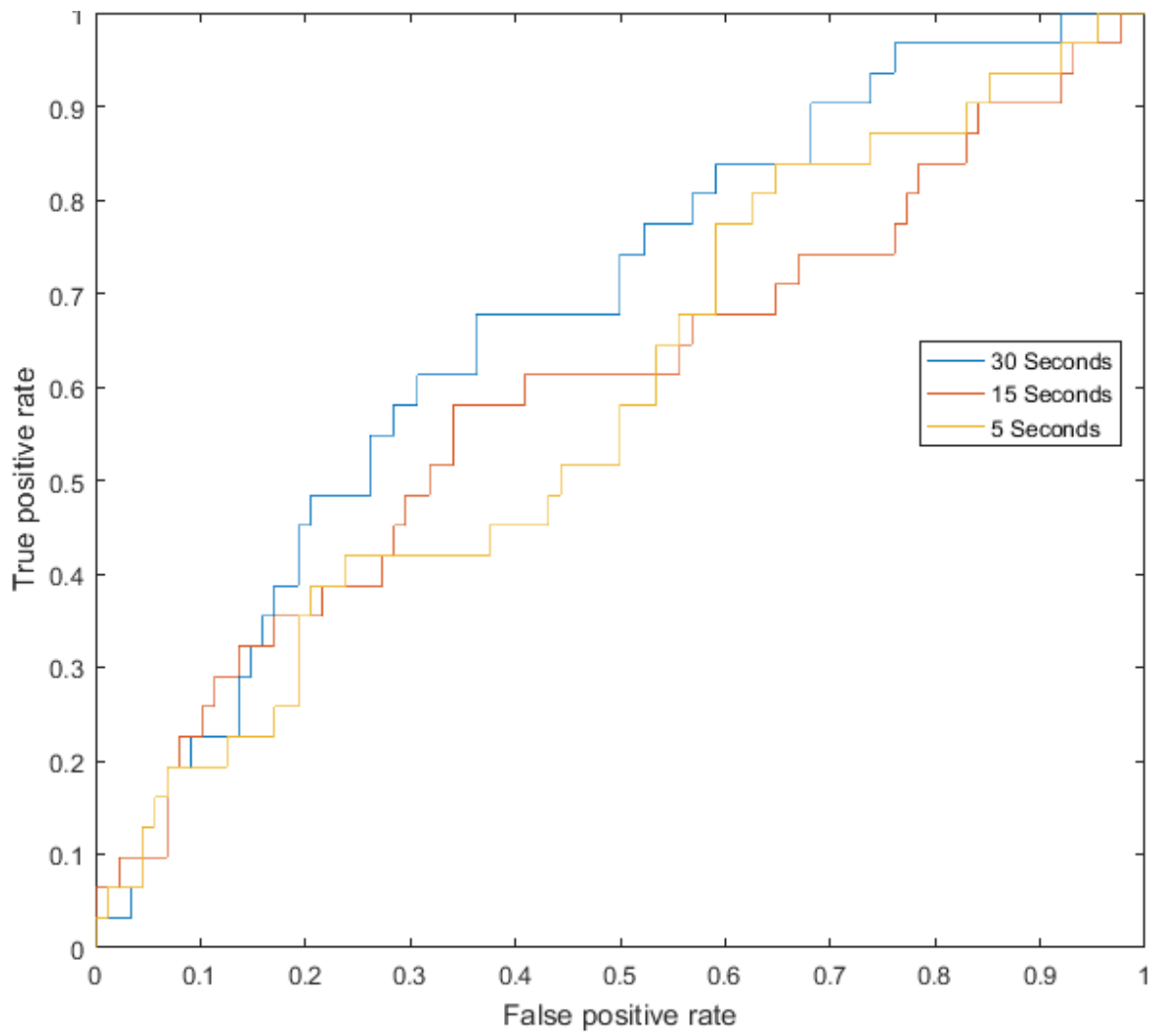
Figure 24 - NN Effect of reducing time span
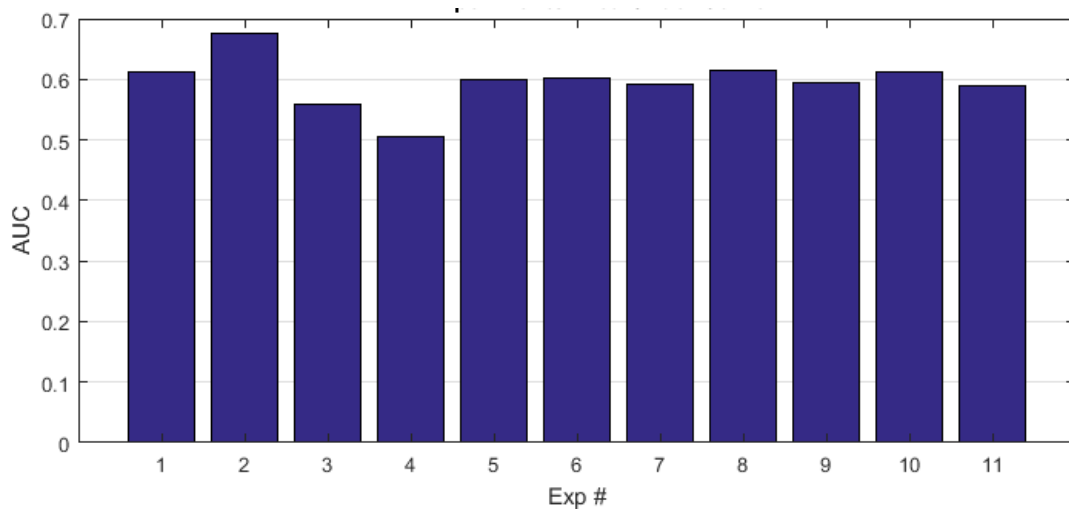
Figure 25 - AUCs for NNs

Discussion

Three different sets of data configurations were tested using feedforward artificial neural networks and their results presented. Firstly the effect of data averaging was tested and showed that averaging each second's 10 entries into a single entry yields better results than the non-averaged data. Moreover, the effect of selecting a subset of features from the original 10 features was experimented. Results show that using the whole set of 10 features gives the best results. Finally, the effect of time span over which data is recorded before the event was tested. The different timespans experimented were: 30, 25, 20, 15, 10 and 5 seconds. Again results revealed that the 30 seconds timespan yields the best results. Therefore, the best results under the artificial neural network is achieved when averaging the data, while using the whole feature set of 10 features per second and a timespan of 30 seconds.

4.1.2 Long Short-Term Memory Recurrent Neural Networks Experiments

Objective

It was interesting to study the classification performance of the LSTM networks on the 100-car study data, due to the ability of LSTMs to handle sequential data. In this experiment set, the effect of using LSTM on classification accuracy of the 100-car data is studied and reported.

Method

In this set of experiments, the same data configurations tested on the feedforward neural networks, were again tested using LSTM, except for the 5s time span experiment, because the least timespan accepted by the LSTM architecture used was 10 time steps. The architecture of the LSTM adapted for these experiments consisted of an input layer, an LSTM hidden layer followed by a softmax layer. The input to the LSTM network was a bit different from that of the feedforward network, in that it was given sequentially. Data for time step t is input separately to the network and passes through the LSTM and softmax layers, feeding its output to the next input layer, which accepts data for time step $t + 1$. The hyperbolic tangent transfer function was used in the LSTM layer, and gradient descent was used to learn the network weights. Cortexsys deep learning toolbox for Matlab was used [78].

Exp1.2.1: LSTM Effect of Feature Averaging

The effect of feature averaging over time using LSTM networks is presented in Figure 26. It is again evident that data averaging has great performance enhancement over the non-averaged data under the LSTM architecture.
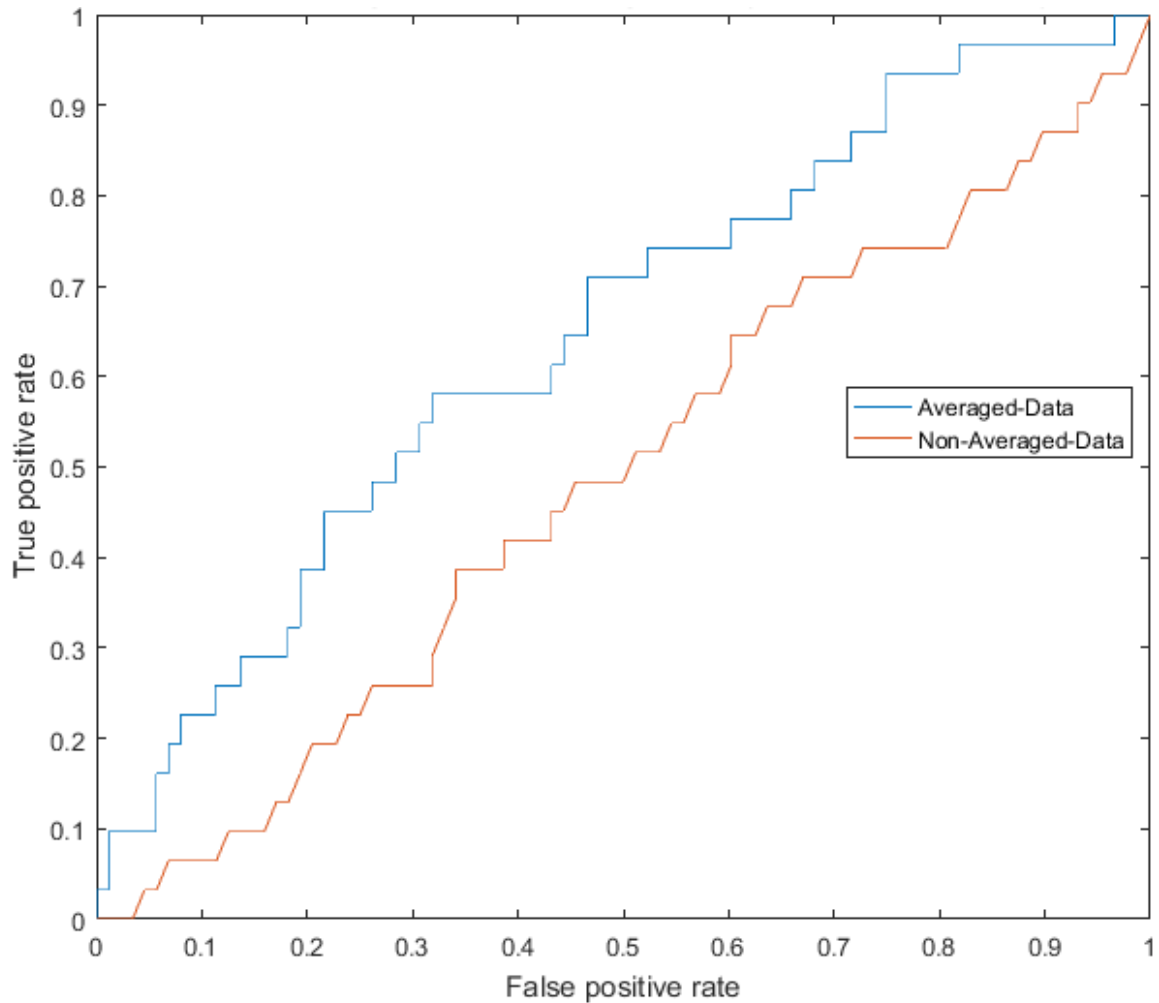
Figure 26 - LSTM Effect of data averaging

Exp1.2.2: LSTM Effect of Different Number of Features

Selecting a subset of features per second was experimented using the LSTM architecture. The results are shown in Figure 27. It is obvious that when the number of features used per second is 10 and 6, the results are very close to each other. On the other hand, as the number of features decreases, the AUC also decreases, indicating less performance of the networks. This points out towards the significance of the 6 features: vehicle speed, longitudinal acceleration, lateral acceleration yaw rate, GPS heading and brake signal; while the other 3 features (lane distance left, lane distance right, and light intensity) being less significant.

57
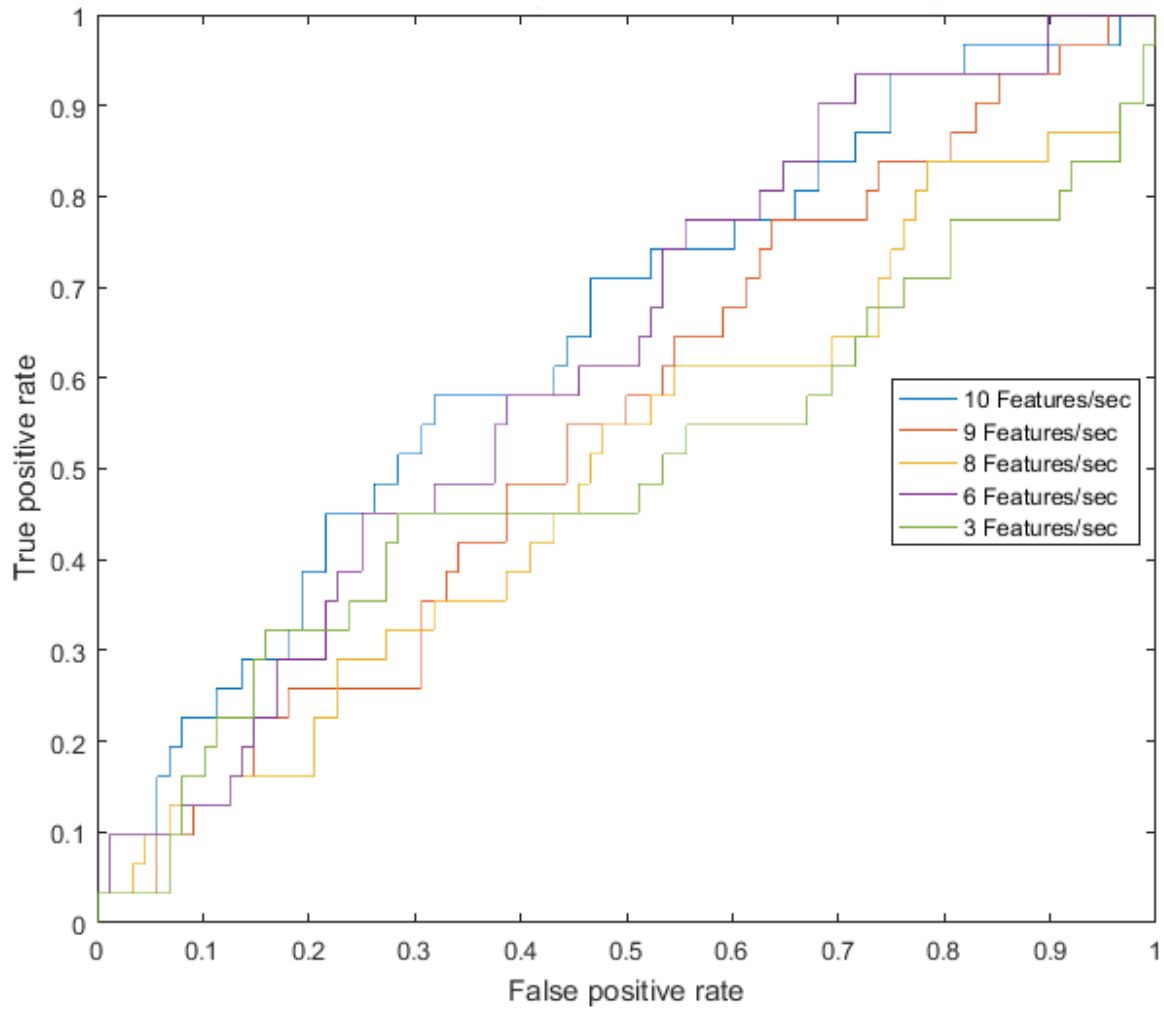
Figure 27 - LSTM Effect of using a subset of features

Exp1.2.3: LSTM Effect of Different Timespans

Figure 28 shows the ROC curves from testing the effect of different timespans on the performance of the LSTM network. It can be seen that timespans of 30, 25, 20 and 10 seconds gave very close performance, while the 15-seconds time span yielded the least performance.

58
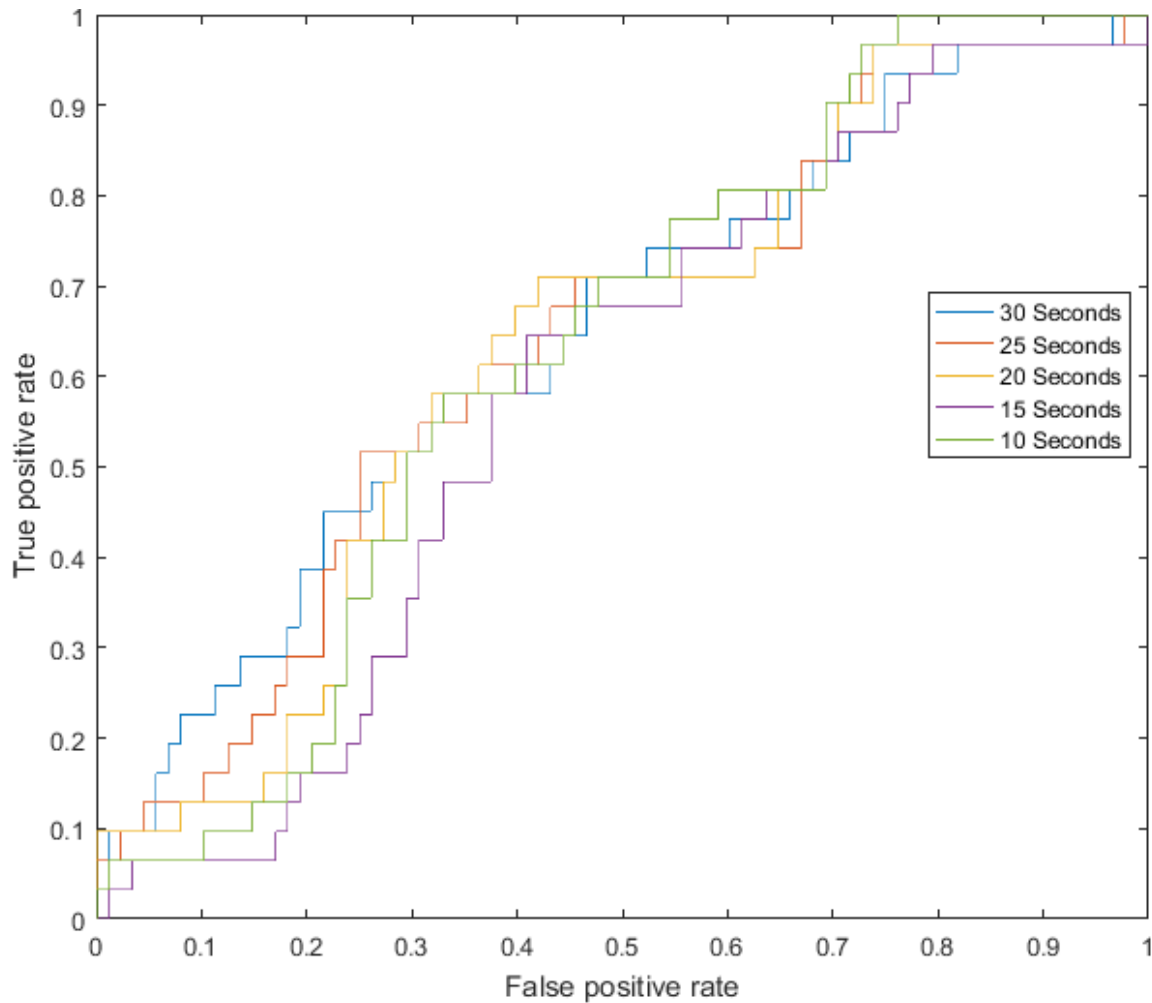
Figure 28 - LSTM Effect of different time spans

Discussion

Averaging the data and using the whole feature set yield better results when using the LSTM architecture. However when testing the effect of time span reduction, the effect was not that evident as in the feedforward network. Timespans of 30, 25, 20 and 10 seconds gave very close accuracies. This could be attributed to the ability of LSTMs to handle time-sequence data

59
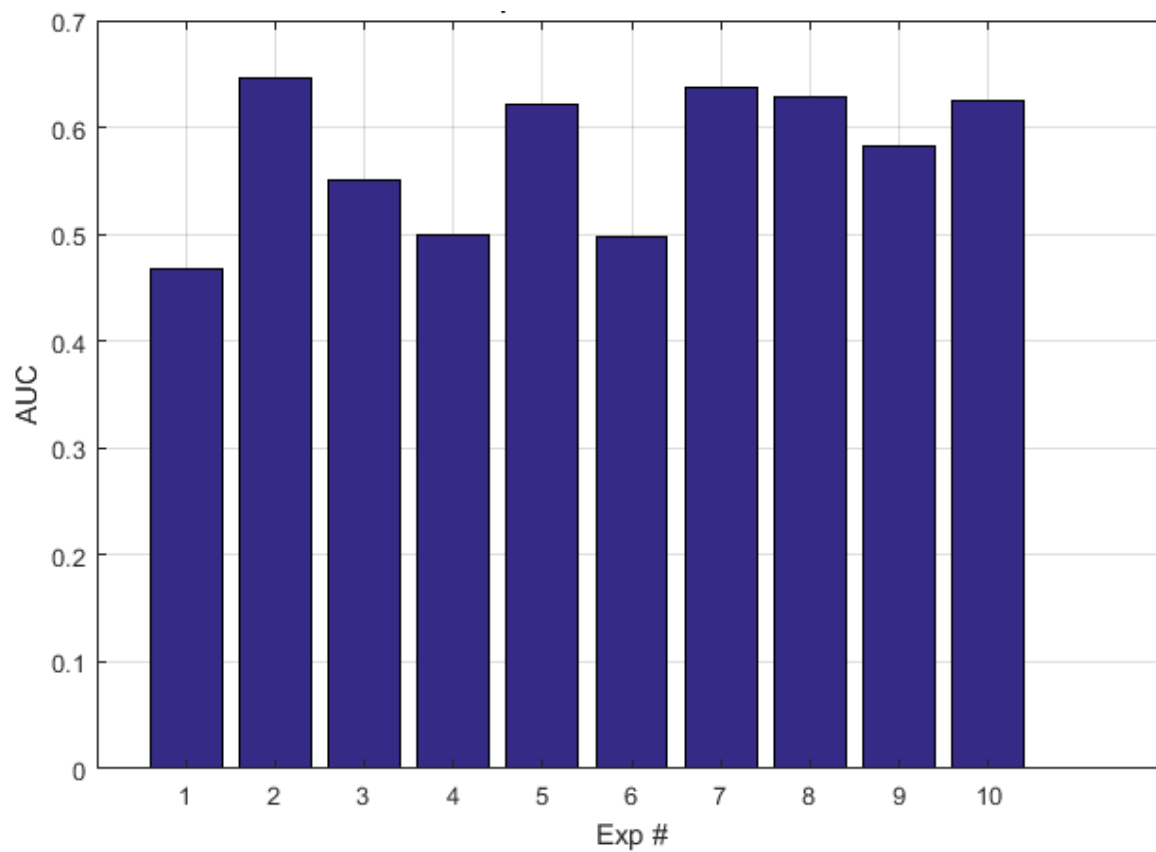
Figure 29 - AUCs for LSTM networks

60

4.1.3   Support Vector Machine Experiments

Objective

In addition to using the feedforward and LSTM neural networks for aggressive behavior classification, support vector machines were tested to compare the results against. Support vector machines are known to solve non-linear classification problems and have been previously used in the literature to classify driver behavior, therefore it would be interesting comparing the performance of neural networks against them.

Method

The same set of experiments listed in Table 5 were conducted using the statistics and machine learning toolbox of Matlab [79]. 75 % of the data was used in the training phase with 5-fold cross validation and 25% used for testing. Several kernels were experimented, and the ones giving the best results were used. This was usually either the linear kernel or the Gaussian one.

Exp1.3.1: SVM Effect of Feature Averaging

Results of feature averaging using SVM, presented in Figure 30, show that there is no noticeable performance enhancement when averaging the data.
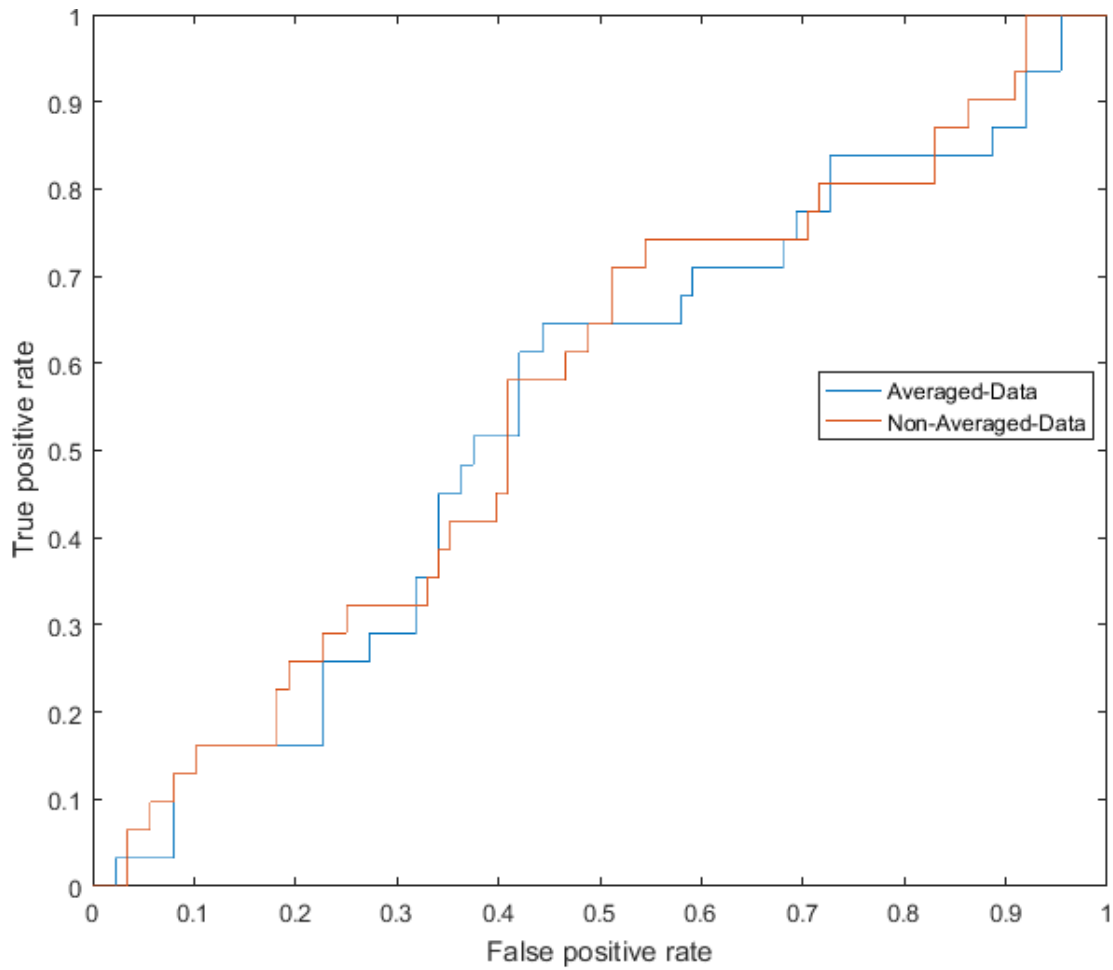
Figure 30 - SVM Effect of data averaging

Exp1.3.2: SVM Effect of Different Number of Features

Figure 31 shows how the number of features affect the SVM performance. It can be seen that using 10, 9 and 8 features per second gave very close results. This shows that selection of features has a different effect on SVM than their effect on NNs.
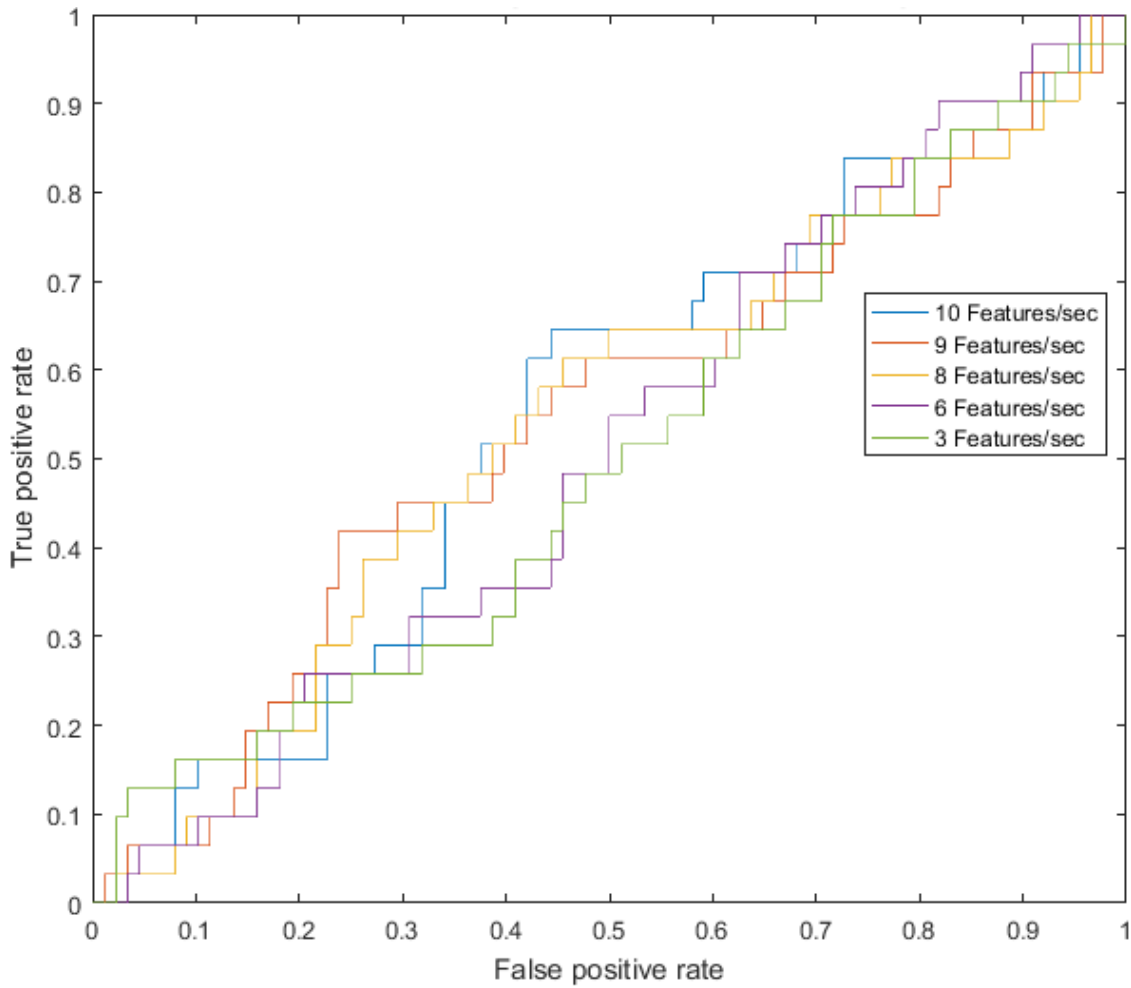
Figure 31 - SVM Effect of using a subset of features

Exp1.3.3: SVM Effect of Different Timespans

Finally, the outcome from experimenting different timespans is shown in Figure 32. The 25 seconds timespan is shown to give the best performance.
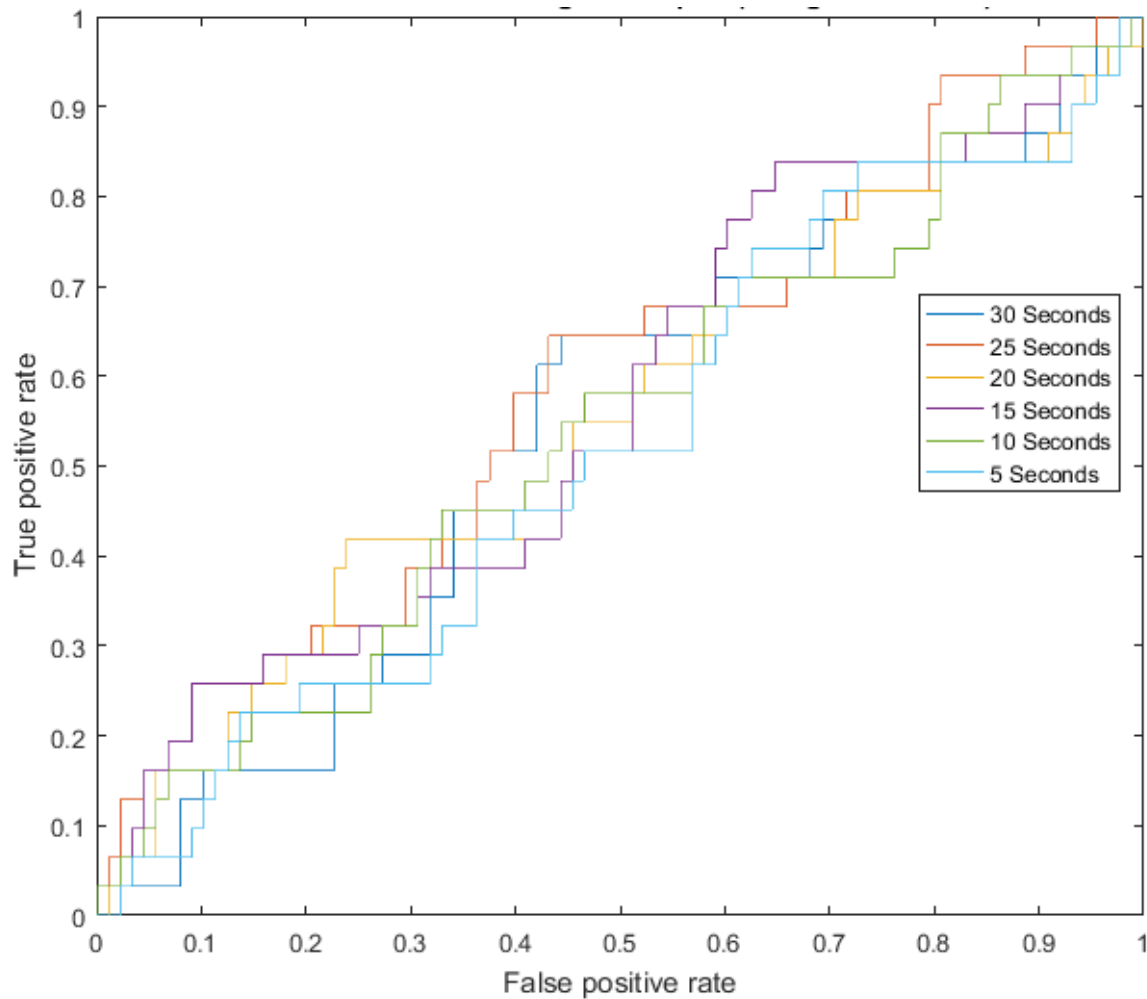
Figure 32 - SVM Effect of different time spans

Discussion

Results of experiments on the different data configurations using support vector machines were presented in this section. Data averaging did not provide a noticeable improvement over the non-averaged data. Moreover, using 10, 9 and 8 features per second gave very close results. Remarkably, the 25 seconds timespan using the 10 features per second on averaged data gave the best results. However, classification performance is still lower than that of the neural networks.
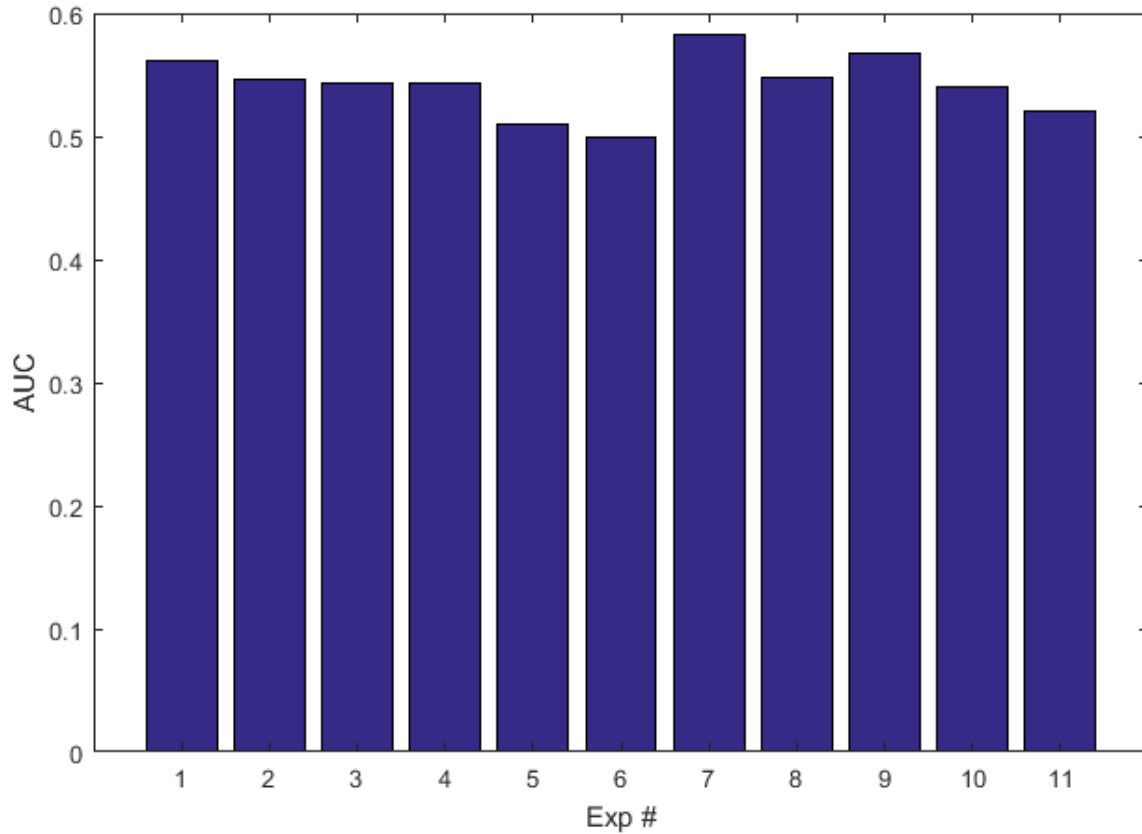
Figure 33 - AUCs for SVM networks

4.1.4 Conclusion

In this section three classes of driver behavior classification algorithms were experimented under different configurations of the 100-car study data. These were: feed-forward artificial neural networks, long short-term memory recurrent neural networks and support vector machines. These experiments are the first of their kind to be conducted on the 100-car study data and their objective was to provide a good classifier for classifying driving behavior into aggressive and non-aggressive. It was shown that the classifier with the best performance was the feed-forward artificial neural network when trained on the averaged data, with 10 features per second and using a timespan of 30 seconds.

4.2 Experiment Set 2: Implementing Mamdani's Traffic Signal Controller

The second set of experiments was conducted to implement, adjust and compare the performance of static, vehicle actuated and Mamdani's traffic signal controllers.

4.2.1 Static Controller

Objective

In this set of experiments, the main aim is to implement the static controller using SUMO and test some of its configuration.

Method

SUMO was used to simulate traffic flow and traffic signal control. In order to simulate traffic flow, description of the intersection and of traffic flow was given to SUMO. This was done by writing the descriptions in xml files in a predefined format

After the intersection was designed, Vehicles flowing according to the given arrival rates had been generated using a program written in C++, which generates a pseudo-random number at each time step and compares it with the arrival rate; according to which a vehicle arrival is decided. Vehicle data was generated for 7200 time steps (seconds). The data was stored in an xml file to be used by SUMO during simulation. 16 files have been generated according to the arrival rates in Table 2.

Controller Setup

A static controller is one in which the phase timings and phase order do not change according to the traffic conditions. In order to get similar results to those obtained by Pappis and Mamdani [13], optimal settings regarding the optimum cycle time and effective green time for each phase were used. These are given for the respective flow rates according to Equation 7. The resulting values are shown in Table 6.

$$C_0 = \frac{1.5L + 5}{1 - Y}$$

$$g_i = \frac{Y_i}{Y}(C_0 - L), i = 1, 2$$

Where:

C_0 optimum cycle time,

g_i effective green time, i = 1,2,

$Y_i = \frac{q_i}{s}$, where i = 1, 2,

q_i flow rate,

s saturation flow

$Y = Y_1 + Y_2$,

L total lost time per cycle (10s)

Table 6 - Optimal cycle time and green time for static controller

N – S Traffic Mean Arrival Rate (veh/hr)	E – W Traffic Mean Arrival Rate (veh/hr)	Optimal Cycle Time (sec)	g_1 (sec)	g_2 (sec)
360	360	25	8	8
360	720	28	6	12
360	1080	33	6	17
360	1440	40	6	24
360	1800	50	7	33
360	2160	67	8	49
360	2520	100	11	79
720	720	34	12	12
720	1080	40	12	18
720	1440	50	13	27
720	1800	66	16	40
720	2160	101	23	68
1080	1080	50	20	20
1080	1440	66	24	32
1080	1800	100	34	56
1440	1440	100	45	45

Exp2.1.1: Effect of Lost Time on Static Controller Delay

In order to reach to the best configurations of SUMO that would fit with Pappis and Mamdani's model [13], several experiments were carried out. Firstly, experiments were carried out to determine the best lost time per cycle. Pappis and Mamdani assumed that it was 10 seconds; however, from running the simulation, it was seen that the lost time might be greater than 10 seconds per cycle, and therefore one experiment was carried out with lost time 10 seconds per cycle and another one with lost time 12 seconds per cycle as shown in Table 7 and in Figure 34. It can be observed that the static controller with lost time of 10 seconds per cycle outperforms the one with 12 seconds per cycle, and this proved that our assumption of lost time 12 seconds was wrong.

Table 7 - Simulation Results

N – S Traffic Mean Arrival Rate (veh/hr)	E – W Traffic Mean Arrival Rate (veh/hr)	Delay from Static Controller (sec/veh)	
		Lost time = 10 sec Amber = 3 sec	Lost time = 12 sec Amber = 2 sec
360	360	11.98	18.63
360	720	19.74	13.71
360	1080	29.34	28.30
360	1440	30.44	30.44
360	1800	33.44	34.80
360	2160	41.44	43.07
360	2520	17.12	56.58
720	720	28.89	28.76
720	1080	32.45	29.69
720	1440	31.66	31.62
720	1800	37.30	39.82
720	2160	48.00	49.26
1080	1080	36.67	35.49
1080	1440	33.56	34.99
1080	1800	42.68	52.96
1440	1440	40.73	42.20
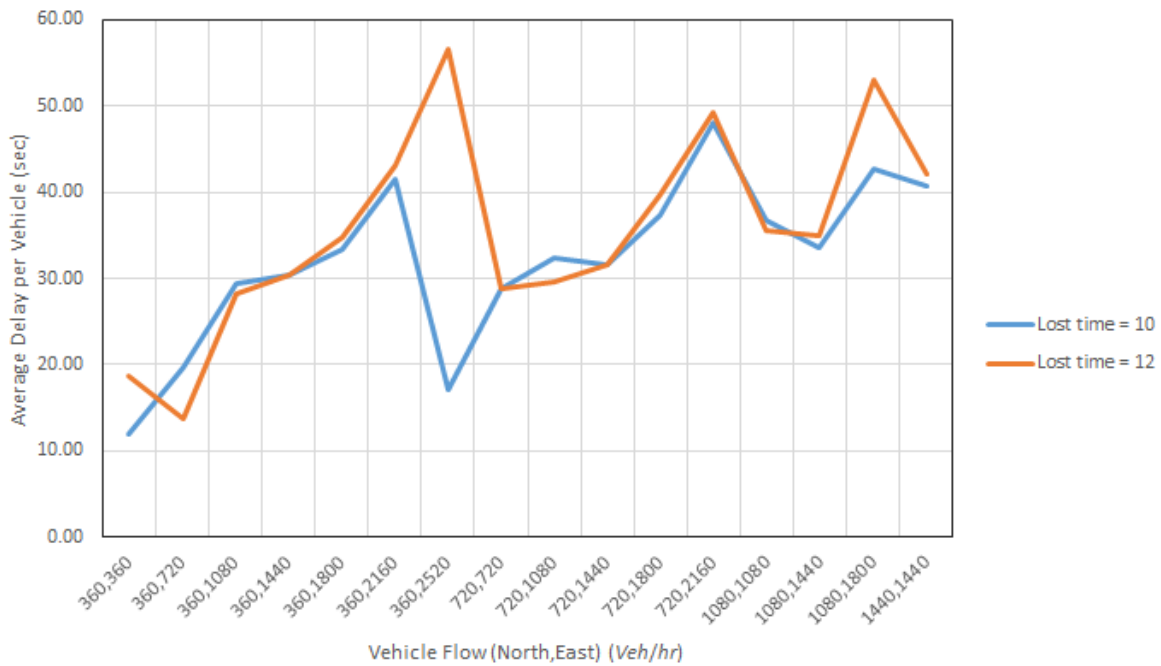
Figure 34 - Delay from Static Controller with Different Lost Time Settings

4.2.2 Vehicle Actuated Controller

Objective

In this experiment, the aim is to build a vehicle actuated controller using SUMO and compare its performance to the static controller. This is important in order to understand how the vehicle actuated controller acts under different flow volumes compared to the static controller.

Method

A vehicle actuated controller is similar to the static controller in construction, except that there are vehicle detectors placed at the stop line to detect the presence of vehicles. The green phase has a minimum green time and a maximum green time. If the traffic light is currently green and about to turn amber, the detection of vehicles at the stop line leads to the extension of the green phase, this continues until either no more vehicles are detected during a certain pre-defined period, or until the maximum green time is reached. In this experiment, the minimum

green that was used for each vehicle flow setting was the one calculated in the static actuated (Table 6), and the maximum green was set to 56 seconds, because this is the maximum green duration that was used in Mamdani's controller.

Exp2.2:.1 Vehicle-Actuated Controller vs Static Controller

The second experiment carried out was to test the vehicle actuated model. The results are plotted in Figure 35. It can be observed that for vehicle flows less than 360 veh/hour in N-S and 2520 veh/hour in E-W, both the static and vehicle actuated controllers had similar performance; however for traffic flows above these, the vehicle actuated controller performed better. This can be attributed to the fact that for higher traffic flows, static controller, leads to more delays and saturation on the halting traffic approach, and therefore it is not suitable for saturated intersections.

Table 8 - Delay from Static and Vehicle-actuated controllers

N – S Traffic Mean Arrival Rate (veh/hr)	E – W Traffic Mean Arrival Rate (veh/hr)	Average Delay from Static Controller (sec)	Average Delay from Vehicle-actuated controller
360	360	11.98	11.86
360	720	19.74	19.03
360	1080	29.34	30.39
360	1440	30.44	31.00
360	1800	33.44	33.70
360	2160	41.44	42.34
360	2520	17.12	16.83
720	720	28.89	18.20
720	1080	32.45	20.86
720	1440	31.66	21.46
720	1800	37.30	24.07
720	2160	48.00	28.93
1080	1080	36.67	23.40
1080	1440	33.56	23.97
1080	1800	42.68	29.80
1440	1440	40.73	42.87

Figure 35 - Delay from Static vs. Vehicle-Actuated Controller

4.2.3 Mamdani's Fuzzy Logic Controller

Objective

In this final experiment in the second experiment set, the aim is to build Mamdani's fuzzy logic controller and compare its performance against the static and vehicle-actuated controllers.

Method

Mamdani's controller was implemented using the python programming language interface provided by the **Tra**ffic **C**ontrol **I**nterface (TraCI). TraCI provides access to a running road traffic simulation from SUMO, where it allows the retrieval of simulated values from SUMO and the control of traffic light operations in the simulation.

71

Exp2.3.1: Fuzzy Logic Controller vs Static and Vehicle Actuated Controllers

Finally, Mamdani's fuzzy logic controller controller was simulated and the results tabulated and plotted against the static and vehicle-actuated controller as shown in Figure 36. It can be concluded that the fuzzy logic controller outperforms the static actuated controllers at all traffic flows. However when compared with the vehicle-actuated controller, the fuzzy logic controller results in less delays in all of the traffic flows, except when the flow is 1080 vehicles per hour in both directions, the delays from the two controllers are very close. This indicates that the vehicle-actuated controller performed good in this traffic flow as well as the fuzzy logic controller. It is worth pointing out that the results are not the same as those obtained by Mamdani in their paper [13], because of the simulation engine used (SUMO). The flow mechanics of the vehicles produced by the simulator are a bit more complicated than the model assumed by Mamdani, and this is the reason for the slight difference in results.
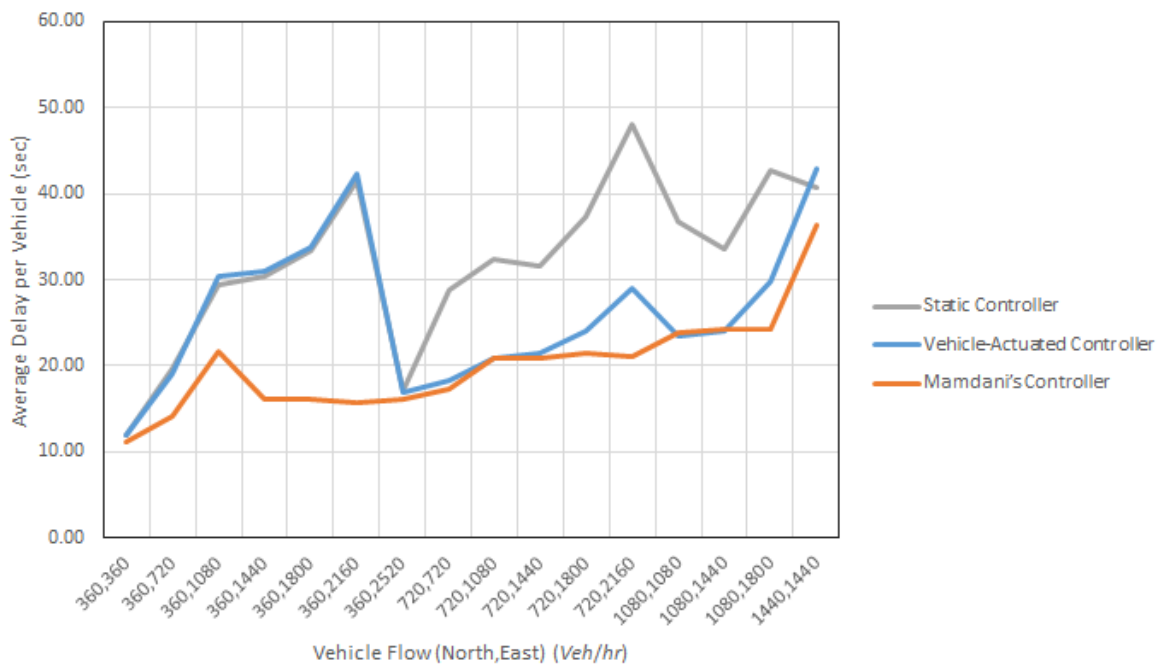
Figure 36 - Delay from Mamdani's Controller vs. Static and Vehicle-Actuated Controllers

72

4.2.4 Conclusion

This section presented the experiments conducted with three different traffic controllers for a two-phase single intersection: static controller, vehicle-actuated controller and Mamdani's fuzzy logic controller. When comparing the vehicle-actuated controller against the static controller, it was observed that vehicle flows less than 360 veh/hour in N-S and 2520 veh/hour in E-W, both the static and vehicle actuated controllers have equal delays; however for traffic flows above these, the vehicle-actuated controller performs better. This leads to the conclusion that under heavy traffic flow, static control leads to more saturation on the halting traffic approach, and therefore it is not suitable for saturated intersections. The fuzzy logic controller on the other hand produces less delays when compared to both the static and vehicle-actuated controllers on most traffic flow volumes, except for one case, where the vehicle-actuated controller had a slightly less delay, indicating the good performance of vehicle-actuation under this condition as well. Finally, it is worth noting that the difference between our results and the results presented by Mamdani is due to the more complicated vehicle dynamics of the simulator used in our experiments.

4.3 Experiment Set 3: Integrating Aggressive Driver Classifier into the Fuzzy Logic Controller

The final component in the proposed approach is integrating the aggressive driving behavior classifier into the fuzzy logic controller with the objective of reducing the intersection delays. For this to be correctly implemented, multiple experiments were conducted and are outlined in this section. Each of the following sections describes one of these three stages.

4.3.1 Accounting for Simulator Limitations

Firstly, simulator limitations had to be taken into consideration, because not all the features present in the 100-car study data could be retrieved from SUMO simulator, therefore another 3 sets of experiments were conducted for the three classification algorithms, using the five features which could be retrieved from SUMO.

Exp3.1.1: Best Classifier based on SUMO Feature Set

Objective

From the experiments conducted in Section 4.1   Experiment Set 1: Building an Aggressive Driver Behavior Classifier, it was concluded that the best aggressive behavior classifier is the feed-forward neural network using the averaged data and the whole 10 features per second, and a timespan of 30 seconds. Unfortunately, not all 10 features (vehicle speed, yaw rate, heading GPS, lateral acceleration, longitudinal acceleration, left distance to lane markings, right distance to lane markings, light intensity, brake on/off, and turn signal state) can be obtained using SUMO simulator, and therefore a similar set of experiments were again conducted to reach to the best classifier using features which SUMO can provide.

Method and Results

SUMO can provide 5 of the 10 features: vehicle speed, GPS heading, longitudinal acceleration, brake on/off signal and turn signal state. A total of 27 more experiments were conducted as outlined in Table 9. Comparison of the Area Under the Curve (AUC) of the 27 experiments shows that the best results was achieved in experiment 9 which used feed-forward neural network on averaged data, with 3 features per second (vehicle speed, longitudinal acceleration and brake on/off signal) and a timespan of 5 seconds. The configuration of this neural network is shown in Figure 38.
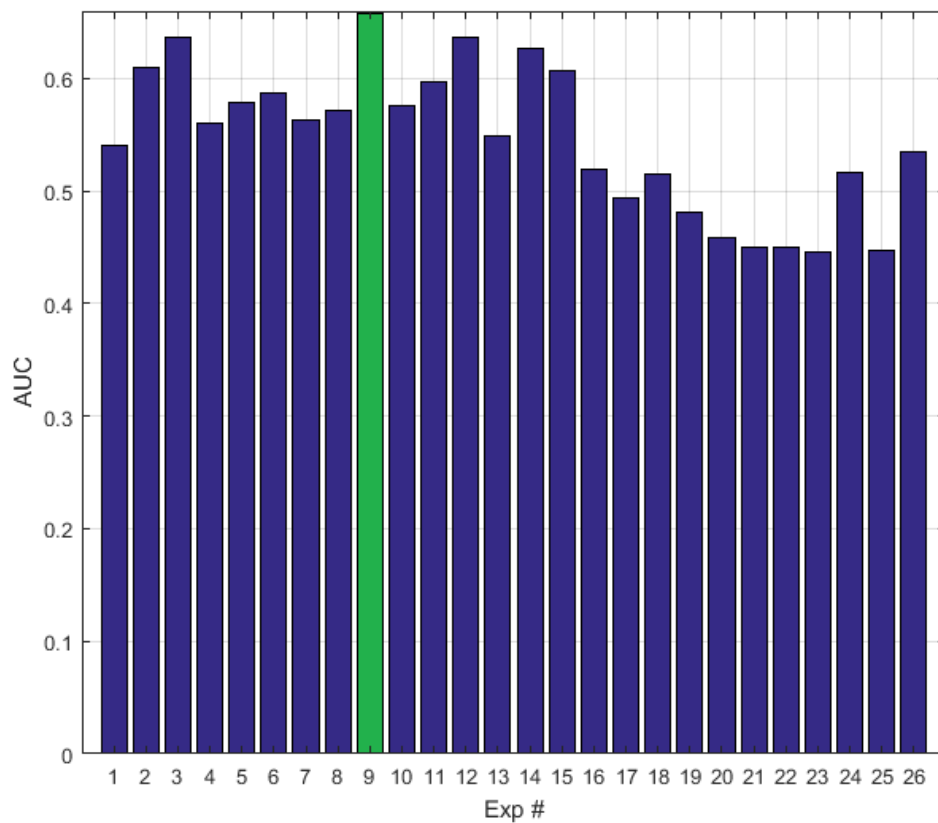
Figure 37 - AUCs for Classifier Experiments based on SUMO Feature Set

75

Table 9 - Classification Experiments using Simulator Feature Set

Exp#	Classifier	Time span (sec)	Averaged/Non-averaged Data	Num Features per second	Vehicle Speed	Brake signal on/off	Longitudinal Acceleration	Turn Signal State	GPS Heading
1	ANN	30	Averaged	5	✓	✓	✓	✓	✓
2	ANN	30	Averaged	4	✓	✓	✓	✓	✗
3	ANN	30	Averaged	3	✓	✓	✓	✗	✗
4	ANN	30	Averaged	2	✓	✓	✗	✗	✗
5	ANN	25	Averaged	3	✓	✓	✗	✗	✗
6	ANN	20	Averaged	3	✓	✓	✗	✗	✗
7	ANN	15	Averaged	3	✓	✓	✗	✗	✗
8	ANN	10	Averaged	3	✓	✓	✗	✗	✗
9	ANN	5	Averaged	3	✓	✓	✗	✗	✗
10	LSTM	30	Averaged	5	✓	✓	✓	✓	✓
11	LSTM	30	Averaged	4	✓	✓	✓	✓	✗
12	LSTM	30	Averaged	3	✓	✓	✓	✗	✗
13	LSTM	30	Averaged	2	✓	✓	✗	✗	✗
14	LSTM	25	Averaged	3	✓	✓	✗	✗	✗
15	LSTM	20	Averaged	3	✓	✓	✗	✗	✗
16	LSTM	15	Averaged	3	✓	✓	✗	✗	✗
17	LSTM	10	Averaged	3	✓	✓	✗	✗	✗
19	SVM	30	Averaged	5	✓	✓	✓	✓	✓
20	SVM	30	Averaged	4	✓	✓	✓	✓	✗
21	SVM	30	Averaged	3	✓	✓	✓	✗	✗
22	SVM	30	Averaged	2	✓	✓	✗	✗	✗
23	SVM	25	Averaged	3	✓	✓	✗	✗	✗
24	SVM	20	Averaged	3	✓	✓	✗	✗	✗
25	SVM	15	Averaged	3	✓	✓	✗	✗	✗
26	SVM	10	Averaged	3	✓	✓	✗	✗	✗
27	SVM	5	Averaged	3	✓	✓	✗	✗	✗
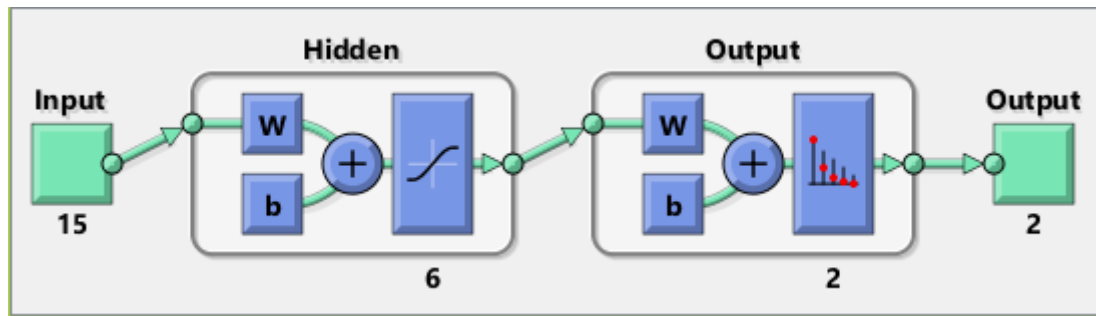
4.3.2 Aggressive Driving Behaviors and Delays

For the classifier and fuzzy logic integration to be meaningful, the simulation environment has to display some form of aggressive driving, and this has to be in line with the aggressive behaviors present in real life. In this section, an experiment is set up in order to define aggressive behaviors, implement them in the simulation environment and show the delay encountered as a result.

Exp3.2.1: Mapping Aggressive Behavior into the Simulated Environment

Objective

Some form of aggressive driving had to be exhibited by the vehicles produced in the simulation environment. The objective of this experiment is to determine how to simulate aggressive behavior using SUMO and test the effect of the mapped aggressive behavior on the intersection delay.

Methods and Results

In order to provide some kind of accurate mapping between real world aggressive driving and the simulated environment, definitions of aggressive driving and the factors affecting aggressive driving were reviewed. As mentioned in Section 2.2.4   Aggressive Behavior, Tasca defined aggressive driving as driving which is "deliberate, likely to increase the risk of collision and is motivated by impatience, annoyance, hostility and/or an attempt to save time" [22]. He

77

www.manaraa.com

mentioned that the factors increasing the probability of aggressive driving are: being young, male, driving anonymously in a traffic situation, being in an angry mood, having a belief that one has superior driving skills, and unexpected traffic congestions. Moreover, aggressive driving factors were reviewed by Wickens et al. [80] revealed that driving aggressivenes in males is greater than females, and younger drivers (age18 to 34) reported the highest occurrence of aggressive driving. Finally the oldest drivers reported the lowest rates of aggression. This led us to divide driver types into the following 12 types:

1. Young well-behaved male

2. Young well-behaved female

3. Young aggressive male

4. Young aggressive female

5. Middle-aged well-behaved male

6. Middle-aged well-behaved female

7. Middle-aged aggressive male

8. Middle-aged aggressive female

9. Elder well-behaved male

10. Elder well-behaved female

11. Elder aggressive male

12. Elder aggressive female

Table 10 shows the probability of well behaved and aggressive male and female drivers under each age category (young, middle aged and elder). According to the U.S. distribution of licensed drivers in 2014 [81], the ratio of male to female drivers is almost the same, however, the ration of young to middle aged to elder drivers is: 0.37:0.318:0.312, and therefore the probability distribution shown in Table 11 was calculated. This will help in generating the correct number of drivers under each driving category.

Table 10 - Probability of Aggressive Drivers in Each Age Category

Type	Probability		
	Young	Middle-Aged	Elder
well-behaved male	20%	30%	35%
well-behaved female	25%	35%	40%
aggressive male	30%	20%	15%
aggressive female	25%	15%	10%
Total	100%	100%	100%

Table 11 - Probability of each Driving Category

Type	Probability			
	Young	Middle-Aged	Elder	Total
well-behaved male	7.40%	9.54%	10.92%	27.86%
well-behaved female	9.25%	11.13%	12.48%	32.86%
aggressive male	11.10%	6.36%	4.68%	22.14%
aggressive female	9.25%	4.77%	3.12%	17.14%
Total	37.00%	31.80%	31.20%	100.00%

In order to map aggressive driving behavior in SUMO, several car-following variables were modified in each driving category to map its degree of aggressiveness. Aggressiveness as reviewed in Section 2.2.4 Aggressive Behavior is depicted by tailgating, higher speeds, following closely, … etc. The car-following model used in our SUMO simulations was developed by Krauß [82] and contains a set of parameters which could map the menionted aggressive behaviors. Table 12 shows the vehicle and car following parameters used to map the aggressive behaviors. Table 13 shows the values given to each of the parameters for each driving type.

Table 12 - Car-following parameters mapping aggressive driving behavior

Vehicle/Car-following Parameter	Description	Aggressive Driving Behavior Mapped
Acceleration (m/s^2)	Acceleration ability of the vehicle	Aggressiveness and responsiveness. The higher the acceleration, the more aggressive the driver.
Deceleration (m/s^2)	Decelration ability of the vehicle	Aggressiveness and responsiveness. The higher the decelration, the more aggressive the driver.
Sigma	Driver imperfection (between 0 and 1)	Imperfect drivers would cause more intersection delays, and therefore the lower the value (i.e.

79

minGap (m)	Empty space after the lead vehicle	Tailgating. The less the gap between the lead vehicle and the current vehicle, the more tailgating effect produced.
maxSpeed (m/s)	Maximum speed the vehicle can reach to	The higher the speed, the more aggressive the driving is.
speedDev	Deviation from vehicle speed	The higher the speed deviation the more aggressive the driving behavior is.
tau (s)	The driver's desired minimum time headway	Driver's reaction time. It is lower in more aggressive drivers.

(At top of table, continuation:) more imperfect), the more aggressive the driver.

Table 13 - Vehicle and car-following properties of each driver type

Type		acceleration $(\frac{m}{s^2})$	deceleration $(\frac{m}{s^2})$	sigma	minGap (m)	maxSpeed $(\frac{m}{s})$	speedDev	tau (s)
Young	well-behaved male	2.7	4.7	0.4	2.5	25	0.3	1
	well-behaved female	2.6	4.6	0.4	2.5	25	0.3	1
	aggressive male	3.2	5.5	0.3	0.9	35	0.1	1
	aggressive female	3	5.3	0.3	1	35	0.2	1
Middle-aged	well-behaved male	2.6	4.5	0.5	2.5	23	0.5	1.5
	well-behaved female	2.5	4.4	0.5	2.5	23	0.5	1.5
	aggressive male	3	5	0.4	1.1	28	0.3	1.3
	aggressive female	2.8	4.9	0.4	1.2	28	0.4	1.4
Elder	well-behaved male	2.5	4	0.6	2.5	23	0.6	1.9
	well-behaved female	2.4	3.9	0.6	2.5	23	0.6	2
	aggressive male	2.6	4.5	0.5	2.1	25	0.4	1.7
	aggressive female	2.5	4.3	0.5	2.2	25	0.5	1.8

With the aim of mapping this to SUMO, the 12 different vehicle types were defined in the each SUMO route's file, and the distribution of the different types used to calculate the number of vehicle from each type in each approach. The order of arrivals of the vehicles is then shuffled to make sure that at any point in time, the correct vehicle type distribution is mapped in the simulated intersection. Vehicles are then set to arrive following a uniform distribution as explained in Section 4.2.1   Static Controller.

Figure 39 shows the delay of the fuzzy logic controller when using the old and new vehicle types. It is observable that in 15 out of the 16 traffic flows, the delay encountered because of the new vehicle types is more than that encountered because of the old types, and this is attributed to the higher number of accidents and emergency stops of vehicles. This proves that aggressive driving behavior results in more intersection delays.
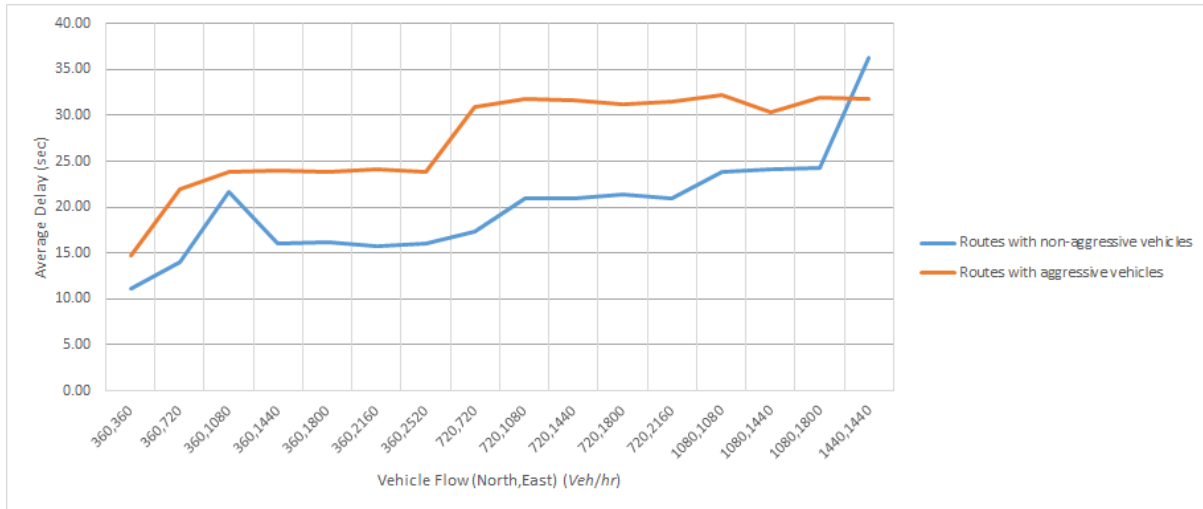
Figure 39 - Fuzzy Logic Controller Delay from Routes without and with aggressive vehicles

4.3.3 Aggressive Driving Behavior Classifier and Fuzzy Logic Controller Integration

After building a classifier which takes into account the simulator limitations, and mapping aggressive driving behavior into the simulation environment, the final stage of integrating the classifier and the fuzzy logic controller has to be implemented. In this section, details of its implementation and results are discussed. Also some experiments testing the effect of changing the membership function and the effect of the different aggressiveness levels on the controller performance are presented.

Exp3.3.1: Integration of Aggressive Driver Classifier and Fuzzy Logic Controller

Objective

This experiment aims to test the proposed approach to reducing intersection delays through incorporating aggressive driver behavior classification into fuzzy logic controller operation.

Methods and Results

The final rule set – given in APPENDIX I – FUZZY LOGIC CONTROLLER RULE SET - was derived after several adjustments to Mamdani's rule set. In other words, the trial and error method was followed to reach to the final rule set. Figure 40 shows the delay after integrating the aggressive driver behavior classifier within Mamdani's fuzzy logic controller. It can be observed that the new controller results in lower delays in most of the traffic flows, with the enhancements more significant at higher traffic flows. The enhancement in terms of lower delays can reach up to 30%.
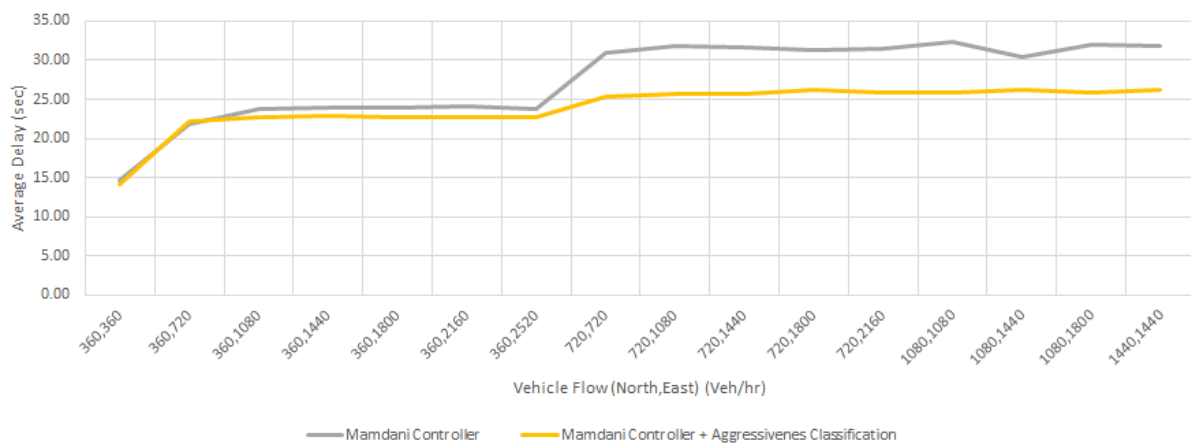
Figure 40 - Average Vehicle Delay from New Controller vs Mamdani's Controller

Exp3.3.2: Effect of Different Membership Functions on Controller's Performance

Objective

In this experiment, the objective is to test different membership functions and their effect on the updated controller's performance. This is important in order to make sure that the membership function chosen gives the best results.

Methods and Results

Three different ranges of membership function were tested as shown in Figure 41, Figure 42, and Figure 43. In these membership functions, the peaks of the fuzzy sets are being shifted

to the right. So taking the "not dangerous" fuzzy set as an example, shifting its membership function peak to the right, means that for lower aggressiveness ratios, it will have higher membership value. As the results in Figure 44 show, membership function 1 gave higher delays when compared against membership functions 2 and 3, which gave very near delay results. This indicates that higher value membership functions are more suitable for the proposed controller. The membership function which is chosen for the controller is the second one.

Figure 41 - Aggressiveness Fuzzy Membership 1

Figure 42 - Aggressiveness Fuzzy Membership 2

83
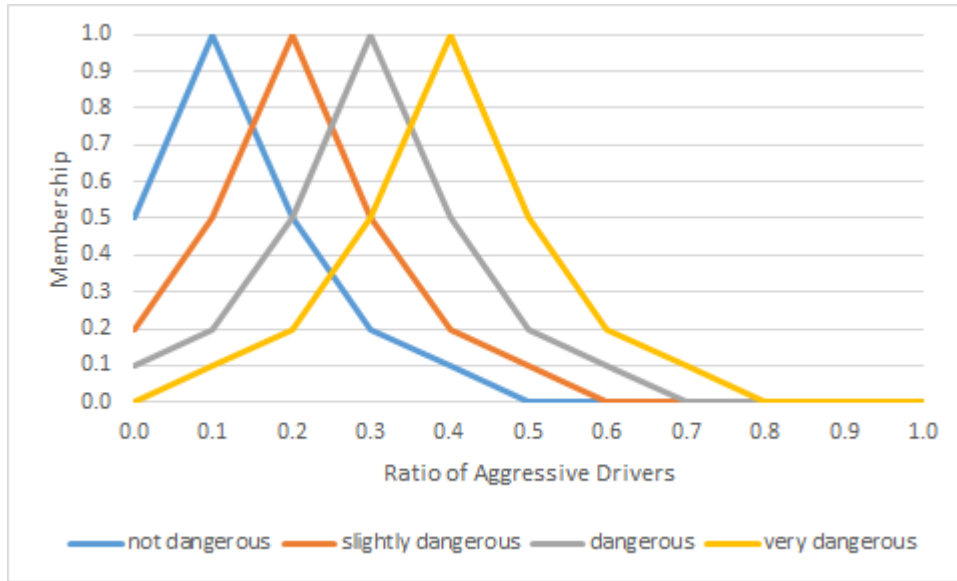
Figure 43 - Aggressiveness Fuzzy Membership 3
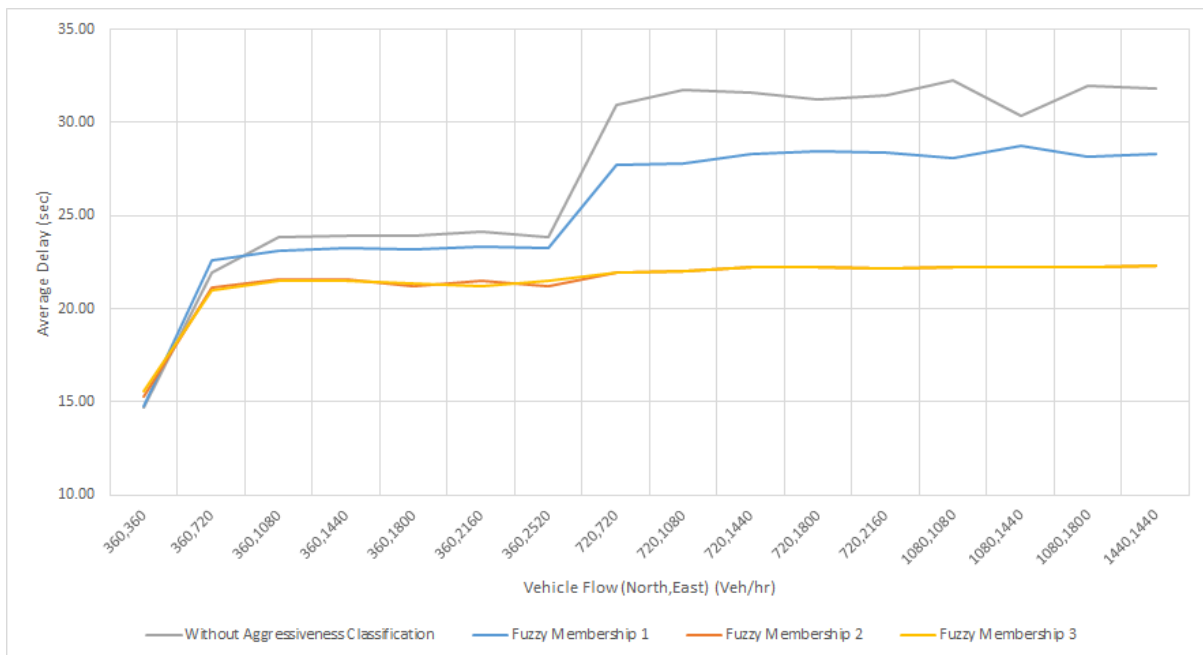
Figure 44 - Effect of Changing Aggressiveness Membership Function

84

Exp3.3.3: Effect of Aggressiveness Level on Controller's Performance

Objective

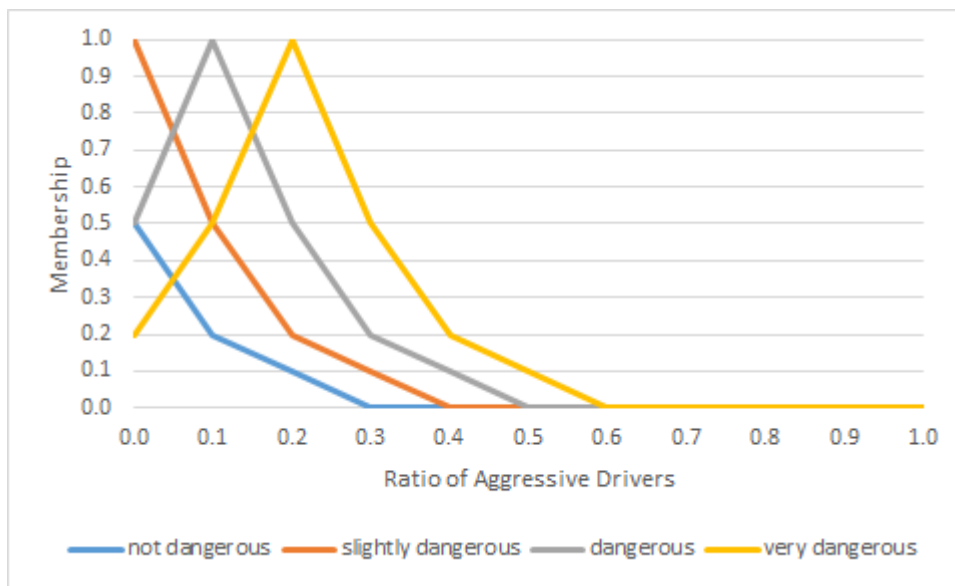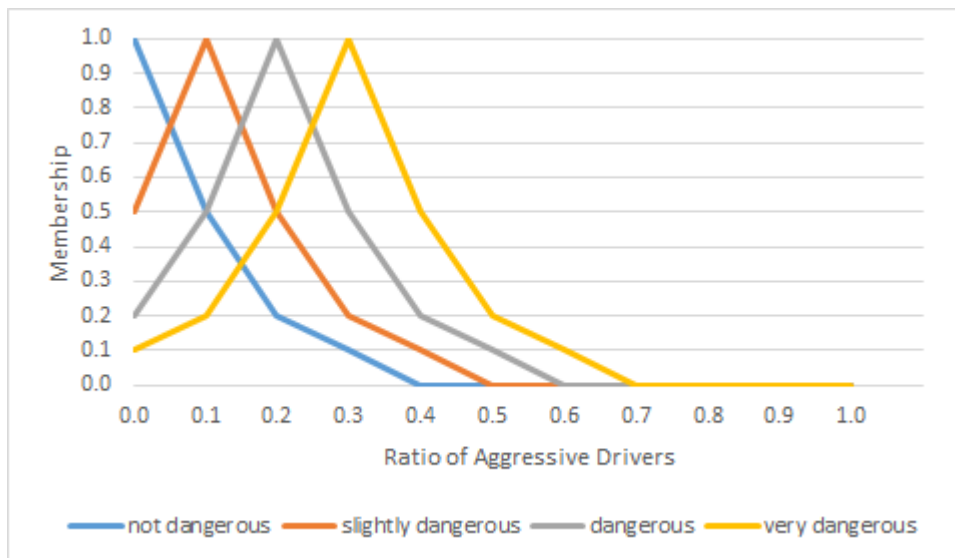Testing the controller's performance under different aggressiveness levels gives an indication about how effective would the simulator be in real life, where different scenarios occur.

Methods and Results

Three different aggressiveness levels were compared: low-aggressiveness, normal aggressiveness and high-aggressiveness situation. The ratio given to each of the vehicle categories described in section 4.3.2   Aggressive Driving Behaviors and Delays for each aggressiveness level is shown in Table 14, Table 15, and Table 16 and the results are shown in Figure 45. Vehicle flow rates below 360,1080 vehicles per hour show no specific trend for the enhancements under the different aggressiveness levels. Moreover, for flow rates between 360,1080 vehicles per hour and 360,2520 vehicles per hour, the controller had similar performance for the different aggressiveness levels. However for flow rates above 360,2520 vehicles per hour in North, East direction, the difference in performance is observable. For the low-aggressiveness situation, the percentage improvement of delay is least, while being almost the same for the normal-aggressiveness and high-aggressiveness situations. At higher vehicle flow rates, the effect of aggressiveness is usually higher and this explains why the results were more evident. The almost similar enhancements under the normal and high aggressiveness situations can be attributed to the simulator limitation of being unable to express very high aggressiveness levels.

Table 14 - Probability of Each Vehicle Category under Low-Aggressiveness Situation

Type	Probability			
	Young	Middle-Aged	Elder	Total
well-behaved male	16.65%	14.40%	13.95%	45.00%
well-behaved female	16.65%	14.40%	13.95%	45.00%
aggressive male	1.85%	1.60%	1.55%	5.00%
aggressive female	1.85%	1.60%	1.55%	5.00%
Total	37.00%	32.00%	31.00%	100.00%

Table 15 - Probability of Each Vehicle Category under Normal-Aggressiveness Situation

Type	Probability			
	Young	Middle-Aged	Elder	Total
well-behaved male	7.40%	9.54%	10.92%	27.86%
well-behaved female	9.25%	11.13%	12.48%	32.86%
aggressive male	11.10%	6.36%	4.68%	22.14%
aggressive female	9.25%	4.77%	3.12%	17.14%
Total	37.00%	31.80%	31.20%	100.00%

Table 16 - Probability of Each Vehicle Category under High-Aggressiveness Situation

Type	Probability			
	Young	Middle-Aged	Elder	Total
well-behaved male	1.85%	1.60%	1.55%	5.00%
well-behaved female	1.85%	1.60%	1.55%	5.00%
aggressive male	16.65%	14.40%	13.95%	45.00%
aggressive female	16.65%	14.40%	13.95%	45.00%
Total	37.00%	32.00%	31.00%	100.00%
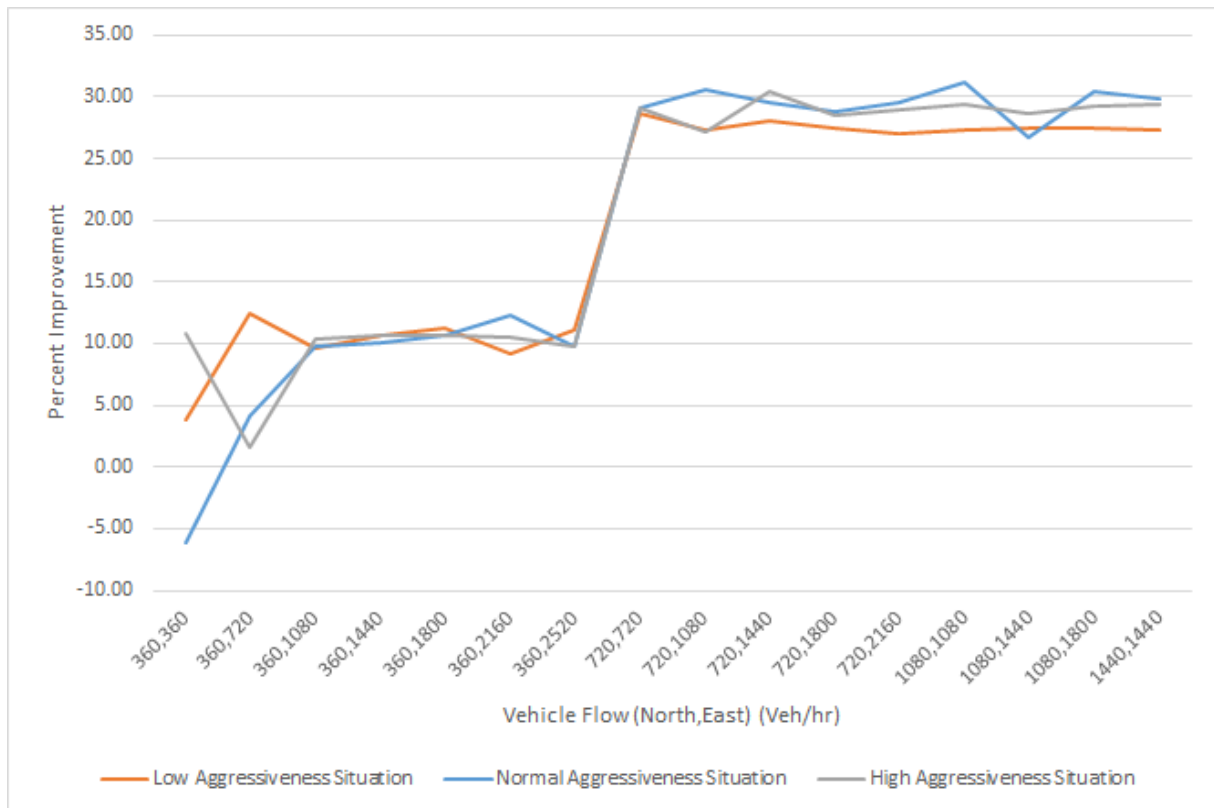
Figure 45 - Delay Percentage Improvement under Different Aggressiveness Levels

4.3.4   Conclusion

Details of integrating the aggressive driver behavior classifier within the fuzzy logic controller were discussed in this section. Firstly, a number of classification experiments were conducted to accommodate for the simulator limitation. The simulator used – SUMO – cannot provide all the 10 features used in the previous classifier experiments, and can only provide 5 of the features. Therefore, a set of experiments was conducted to test the accuracies of the different classification algorithms using the 5 features. The experiments has shown that the feed-forward neural network using 3 features (vehicle speed, longitudinal acceleration and brake signal) and 5 seconds as the timespan gave the best results. Secondly, a representation of the aggressive driver behavior based on real-life behaviors and statistics was suggested and simulated using SUMO. Comparing its delay with the previous simulated environment's delay showed that new vehicle

87

routes result in greater delay. Finally, the designed aggressive driver classifier (neural network) was incorporated within the built fuzzy logic controller and tested on the new vehicle routes. It was shown that this integration results in less delay in most of the traffic flow, with enhancements being more noteworthy under higher traffic flows. This leads to the conclusion that combining aggressive driver behavior classification with fuzzy traffic signal control provides evident improvement in terms of intersection delay.

CHAPTER 5 CONCLUSION AND FUTURE WORK

Traffic congestion is a considerable problem nowadays notably with the increase in the number of vehicles worldwide in general and particularly in Egypt. Increased traffic congestion leads to major side effects such as waste of resources (e.g. time and fuel), leading to decreased national economy, as well as fatal problems in terms of deadly road accidents. The elementary solution to solving traffic congestion is expanding the road capacity by adding more lanes to the existing roadways. This; however, is not feasible and not easily implemented due to the large costs and the huge time required to produce the changes. It is also important to change the aggressive driver behavior in order to reduce crashes and delays, but this is not always easy. Therefore, intelligent transportation systems originated to tackle the problem from another perspective. This research focused on improving the existing traffic signal control system in a two-phase single intersection by integrating driver behavior classification within the controller.

Literature on traffic signal control showed that fuzzy logic control systems form the basis of any traffic signal controller. The differences in research vary from the choice of the parameter set to use as an input to the different components of traffic that they control. The two most important input parameters are the number of vehicles in the arrival and the queuing approach. Advanced traffic signal controllers integrated different computational intelligence techniques with fuzzy logic to result in better optimization of the controller. This yielded lower delays and lower percentage stops. Work on driver behavior has also been reviewed and revealed that research from different areas were carried out to understand and model driver's behavior. Some work was also attempted in order to classify driver's behaviors into compliant drivers and violators. Furthermore, some researchers focused on the analysis of vehicle trajectory patterns to learn characteristics of normal vehicle trajectory. None of the previous research on behavior classification examined the use of neural networks to classify aggressive driving behavior. This encouraged us to use two different types of neural networks to classify aggressive driving

behavior and test their performance against support vector machines. Moreover, review of literature revealed the lack of significant research focusing on integrating driver behavior classification with traffic signal control. This motivated and provided a starting point to the current research.

The proposed architecture for integrating driver behavior classification into traffic signal control consisted of three modules: vehicle data collection, driver behavior classification and fuzzy traffic signal controller. Focus in the current research was on the modules of driver behavior classification and traffic signal controller. Data from the 100-car naturalistic driving study was used to build and test the different classification algorithms. This dataset consisted of a compilation of detailed video analysis for 68 crashes and 760 near crashes. The dataset was first prepared for use by the classifier, by manually labelling the 828 events and then preprocessing the data. The preprocessing operations performed were: data aggregation, data reduction and normalization. This resulted in a total of 788 trajectory records, of which 582 were labelled as non-dangerous and 206 were dangerous. Driver behavior classification's target is to classify if the current driver is an aggressive driver or not by using his trajectory pattern over time. Output from the classifier was then used as an input to the fuzzy traffic signal controller which is based on the model described by Pappis and Mamdani in [13], with modifications to allow for the integration to be successful. The reason for choosing this approach and not one of the advanced approaches is to test the applicability of integrating driver behavior classification into logic controllers in general. The proposed integration resulted in the following challenges which were tackled by the proposed architecture: 1) Preparing the 100-car data for use by classification algorithms, 2) designing an efficient classifier which classifies the aggressive driver behavior, and finally 3) using the output from the driver behavior classification module as an input parameter to the fuzzy logic control system.

Three major sets of experiments were performed in order to test the different components of the proposed architecture, and reach to a final configuration that yields better results. The first set of experiments conducted aimed to build a good aggressive driver classifier using the preprocessed 100-car study data. Three different classes of algorithms were tested, which are: feed-forward artificial neural networks (ANN), long short-term memory recurrent neural networks (LSTM RNN) and support vector machines (SVM). For each algorithm, different data configurations were experimented, resulting in a total of 32 experiments for the three machine learning algorithms. These experiments were the first of their kind to be conducted on the 100-car study data, and revealed that the best performing architecture is the feed-forward neural network when trained on the time averaged data, with 10 features per second and a timespan of 30 seconds.

In the second set of experiments, three traffic signal controllers, for a two-phase single intersection, were implemented using SUMO and their performance, in terms of intersection delay, compared. These were: static controller, vehicle-actuated controller and fuzzy logic controller. Vehicle-actuation when compared to the static controller resulted in lower intersection delays for high traffic flows, and is therefore more suitable in saturated intersections. On the other hand, the fuzzy logic controller was shown to outperform the vehicle-actuated controller on most traffic flow volumes.

A final set of experiments were conducted to test the integration of the classifier with the fuzzy logic controller. Initially, a number of classification experiments were conducted to accommodate for the simulator limitation of the features that it could provide. The classifier giving the best performance was a feed-forward artificial neural network trained on averaged data with three features per second (vehicle speed, longitudinal acceleration and brake signal) and a timespan of 5 seconds. Then, a representation of aggressive driving behavior was constructed based on real life behaviors and statistics. It was simulated using SUMO and showed an increase

91

in the intersection delay when compared with the simulated environment without aggressive behavior representation. The final experiment in the third set of experiments showed the integration of the aggressive driver classifier (neural network) with the built fuzzy logic controller. The new controller was tested on the vehicle routes showing aggressive behaviors and resulted in less delay than the basic Mamdani's controller. The improvement in terms of delay was more significant, however, at higher traffic flows. There are two major outcomes of this research. Firstly, an aggressive driver behavior classifier was successfully built utilizing the 100-car study data, and this is the first of its kind to be built on this data. Secondly, it was shown how the integration of aggressive driving classifier with the traffic controller proves our speculations put at the beginning of the research of reducing the intersection delays.

This study is an initial step towards utilizing driving behavior data in traffic control systems in order to reduce delays and congestions. Further work needs to be carried out to enhance the performance of both the classifier and controller. The current classifier performance is encouraging, but future work needs to concentrate on enhancing its quality by utilizing larger scale (and size of) intersection trajectory data. It is believed that building a classifier trained on intersection data rather than highway data would result in significant classifier enhancements. More LSTM architectures need to be examined, as it is assumed that more advanced LSTM architectures could result in better results than the ANN, because of the sequential nature of the data. One of the important future plans is to test the enhanced controller on real life data and behaviors instead of simulations, since this could not be achieved at this stage. It is also recommended that further research should be undertaken in the area of traffic signal controller. The effect of integrating the aggressive behavior classifier within other traffic signal controllers should also be experimented. Also larger scale networks have to be examined.

APPENDIX I – FUZZY LOGIC CONTROLLER RULE SET

Intervention: 7th second

```
if T = very short

and A = mt(none)

and Q = any

and AGG = mt(not_dangerous)

then E = very short

else

if T = short

and A = mt(a few)

and Q = lt(very small)

and AGG = mt(not_dangerous)

then E = short

else

if T = medium

and A = mt(few)

and Q = lt(very small)

and AGG = mt(slightly_dangerous)

then E = medium

else

if T = long

and A = mt(medium)

and Q = lt(very small)

and AGG = mt(slightly_dangerous)

then E = long

else

if T = very long
```

```
and A = mt(many)

and Q = lt(very small)

and AGG = mt(slightly_dangerous)

then E = very long.
```

Intervention: 17th second

```
if T = very short

and A = mt(none)

and Q = any

and AGG = mt(not_dangerous)

then E = very short

else

if T = short

and A = mt(a few)

and Q = lt(very small)

and AGG = mt(not_dangerous)

then E = short

else

if T = medium

and A = mt(few)

and Q = lt(very small)

and AGG = mt(slightly_dangerous)

then E = medium

else

if T = long

and A = mt(medium)

and Q = lt(very small)
```

```
and AGG = mt(slightly_dangerous)

then E = long

else

if T = very long

and A = mt(many)

and Q = lt(small)

and AGG = mt(dangerous)

then E = very long.
```

Intervention: 27th second

```
if T = very short

and A = mt(none)

and Q = any

and AGG = mt(not_dangerous)

then E = very short

else

if T = short

and A = mt(a few)

and Q = lt(very small)

and AGG = mt(slightly_dangerous)

then E = short

else

if T = medium

and A = mt(few)

and Q = lt(very small)

and AGG = mt(slightly_dangerous)

then E = medium
```

95

```
else
if T = long
and A = mt(medium)
and Q = lt(very small)
and AGG = mt(slightly_dangerous)
then E = long
else
if T = very long
and A = mt(many)
and Q = lt(small)
and AGG = mt(dangerous)
then E = very long.
```

Intervention: 37th second

```
if T = very short
and A = mt(none)
and Q= any
and AGG = mt(not_dangerous)
then E = very short
else
if T = short
and A = mt(a few)
and Q = lt(small plus)
and AGG = mt(slightly_dangerous)
then E = short
else
if T = medium
```

```
and A = mt(medium)

and Q = lt(small plus)

and AGG = mt(slightly_dangerous)

then E = medium

else

if T = long

and A = mt(many)

and Q= lt(medium)

and AGG = mt(dangerous)

then E = long

else

if T = very long

and A = mt(too many)

and Q = lt(long)

and AGG = mt(dangerous)

then E = very long.
```

Intervention: 47th second

```
if T = very short

and A = mt(none)

and Q = any

and AGG = mt(not_dangerous)

then E = very short

else

if T = short

and A = mt(a few)

and Q = lt(long)
```

```
and AGG = mt(slightly_dangerous)

then E = short

else

if T = medium

and A = mt(medium)

and Q = lt(long)

and AGG = mt(slightly_dangerous)

then E = medium

else

if T= long

and A = mt(too many)

and Q = /t(very long)

and AGG = mt(dangerous)

then E = long

else

if T = very long

and A = mt(too many)

and Q = lt(very long)

and AGG = mt(very_dangerous)

then E = very long.
```

REFERENCES

[1]   World Health Organization, "WHO | Number of registered vehicles," *WHO*, 2016. [Online].   Available: http://www.who.int/gho/road_safety/registered_vehicles/number_text/en/. [Accessed: 20-Nov-2016].

[2]   M. Papageorgiou, "Review of road traffic control strategies," *Proc. IEEE*, vol. 91, no. 12, pp. 2043–2067, 2003.

[3]   "Cairo Traffic is Much more than a Nuisance," 2012. [Online]. Available: http://www.worldbank.org/en/news/feature/2012/08/21/cairo-traffic-much-more-than-nuisance. [Accessed: 05-Jan-2014].

[4]   "The EU-Twinning Expertise for Enhancing Road Safety in Egypt."

[5]   World Health Organization, "Global Status Report on Road Safety 2015," 2015.

[6]   WHO, "Road Safety in ten Countries." World Health Organization, 2010.

[7]   World Health Organization, "WHO | Road traffic injuries," *WHO*, 2016. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs358/en/. [Accessed: 20-Nov-2016].

[8]   M. El Araby, "Urban growth and environmental degradation: The case of Cairo, Egypt," *Cities*, vol. 19, no. 6, pp. 389–400, 2002.

[9]   L. E. Y. Mimbela and L. Klein, "A Summary of Vehicle Detection and Surveillance Technologies used in Intelligent Transportation Systems," 2000.

[10]   "Signal Timing," *Federal Highway Administration*. [Online]. Available: http://ops.fhwa.dot.gov/publications/fhwahop08024/chapter1.htm. [Accessed: 01-Mar-2013].

[11]   S. M. Rahman and N. T. Ratrout, "Review of the Fuzzy Logic Based Approach in Traffic Signal Control: Prospects in Saudi Arabia," *J. Transp. Syst. Eng. Inf. Technol.*, vol. 9, no. 5, pp. 58–70, Oct. 2009.

[12]   D. Zhao, Y. Dai, and Z. Zhang, "Computational Intelligence in Urban Traffic Signal Control: A Survey," *Syst. Man, Cybern. Part C Appl. Rev.*, vol. 42, no. 4, pp. 485–494, 2012.

[13]   C. Pappis and E. Mamdani, "A Fuzzy Logic Controller for a Traffic Junction," *Syst. Man Cybern. IEEE Trans.*, vol. 7, no. 10, pp. 707–717, 1977.

[14]   J. M. Sussman, "ITS: A Short History and a Perspective on the Future," in *Perspectives on intelligent transportation systems (ITS)*, no. December 1996, Springer, 2005, pp. 115–125.

[15]   "Intelligent Transport Systems Education Network." [Online]. Available: http://www.its-edunet.org/typo3/index.php?id=21. [Accessed: 21-Jan-2014].

[16]   P. Koonce, L. Rodergerdts, and K. Lee, "Signal Timing Manual," *Fed. Highw. Adm.*, 2008.

[17]   P. P., "Cool facts about traffic signal systems," Oct. 2010.

[18]   United States Department of Transportation - Federal Highway Administration, "Traffic Signal Timing Manual: Chapter 1 - Office of Operations," 2013. [Online]. Available: http://ops.fhwa.dot.gov/publications/fhwahop08024/chapter1.htm#1.1.

[Accessed: 22-Jan-2014].

[19] M. Bellis, "Traffic lights - History of Traffic Lights." [Online]. Available: http://inventors.about.com/od/rstartinventions/a/History-Of-Roads_3.htm. [Accessed: 11-Mar-2013].

[20] D. Bullock and T. Urbanik, "Traffic Signal Systems addressing diverse technologies and complex usernNeeds," in *Transportation in the New Millinnium*, 2000.

[21] "Engineering Countermeasures to Reduce Red-Light Running (RLR)," 2004.

[22] L. Tasca, "A review of the literature on aggressive driving research," in *Ontario Advisory Group on Safe Driving Secretariat, Road User Safety Branch, Ontario Ministry of Transportation.*, 2000, pp. 1–25.

[23] A. Abraham, "Artificial neural networks," *Handb. Meas. Syst. Des.*, pp. 901–908, 2005.

[24] W. Duch and N. Jankowski, "Survey of neural transfer functions," *Neural Comput. Surv.*, vol. 2, pp. 163–212, 1999.

[25] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, "Action classification in soccer videos with long short-term memory recurrent neural networks," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6353 LNCS, no. PART 2, pp. 154–159, 2010.

[26] S. Hochreiter, S. Hochreiter, J. Schmidhuber, and J. Schmidhuber, "Long short-term memory.," *Neural Comput.*, vol. 9, no. 8, pp. 1735–80, 1997.

[27] LSA lab, "LSTM Networks for Sentiment Analysis," Nov-2016. [Online]. Available: http://deeplearning.net/tutorial/lstm.html. [Accessed: 21-Nov-2016].

[28] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A Training Algorithm for Optimal Margin Classiiers," *Proc. fifth Annu. Work. Comput. Learn. theory*, pp. 144–152, 1992.

[29] K. Shafiee, J. B. Lee, V. C. M. Leung, and G. Chow, "Modeling and simulation of vehicular networks," *Proc. first ACM Int. Symp. Des. Anal. Intell. Veh. networks Appl. - DIVANet '11*, p. 77, 2011.

[30] J. Barceló, "Models, Traffic Models, Simulation, and Traffic Simulation," in *Fundamentals of Traffic Simulation*, vol. 145, J. Barceló, Ed. New York, NY: Springer New York, 2010, pp. 1–62.

[31] C. S. Group, "Generic Mobility Simulation Framework (GMSF)." [Online]. Available: http://gmsf.sourceforge.net/. [Accessed: 28-Apr-2013].

[32] "VanetMobiSim - Newcom | Institut Eurecom | Politecnico di Torino." [Online]. Available: http://vanet.eurecom.fr/. [Accessed: 29-Apr-2013].

[33] "The smarter, faster way for model-based system development | VisSim." [Online]. Available: http://www.vissim.com/. [Accessed: 29-Apr-2013].

[34] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "SUMO - Simulation of Urban MObility: An Overview," in *Simul 2011, The Third International Conference on Advances in System Simulation*, 2011.

[35] "SUMO - Simulation of Urban MObility - TraCI." Institute of Transportation Systems at the German Aerospace Center.

[36] K. Tan, M. Khalid, and R. Yusof, "Intelligent traffic lights control by fuzzy logic," *Malaysian J. Comput. Sci.*, vol. 9, no. 2, pp. 29–35, 1996.

[37] I. Askerzade and M. Mahmood, "Control the Extension Time of Traffic Light in Single Junction by Using Fuzzy Logic," *Int. J. Electr. Comput. Sci.*, vol. 10, no. 2, pp. 52–59, 2010.

[38] R. Hoyer and U. Jumar, "Fuzzy control of traffic lights," in *Fuzzy Systems, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the Third IEE Conference on.*, 1994, pp. 1526–1531.

[39] G. Beauchamp-Baez, E. Rodriguez-Morales, and E. L. Muniz-Marrero, "A fuzzy logic based phase controller for traffic control," in *Fuzzy Systems, 1997., Proceedings of the Sixth IEEE International Conference on*, 1997, vol. 3, pp. 1533–1539.

[40] J. Niittymiiki and R. Nevala, "Fuzzy Adaptive Traffic Signal Control Principles and Results," no. C, pp. 2870–2875.

[41] J. Niitymaki and M. Pursula, "Signal control using fuzzy logic," *Fuzzy sets Syst.*, vol. 116, no. 1, pp. 11–22, 2000.

[42] J. Niittymaki, T. Kosonen, and R. Nevala, "Fuzzy traffic signal control in major arterials," *ITSC 2001. 2001 IEEE Intell. Transp. Syst. Proc. (Cat. No.01TH8585)*, pp. 346–351, 2001.

[43] J. Niittymaki and E. Turunen, "Traffic signal control on similarity logic reasoning," *Fuzzy sets Syst.*, vol. 133, no. 1, pp. 109–131, 2003.

[44] Y. S. Murat and E. Gedizlioglu, "A new approach for fuzzy traffic signal control," in *Proceedings of the 13th Mini-EURO Conference on Artificial Intelligence in Transportation Systems and Science, Bari, Italy*, 2002, pp. 174–180.

[45] Y. S. Murat and E. Gedizlioglu, "A fuzzy logic multi-phased signal control model for isolated junctions," *Transp. Res. Part C Emerg. Technol.*, vol. 13, no. 1, pp. 19–36, Feb. 2005.

[46] L. Zang, "Study on Intelligent Control Algorithm for Traffic Signals at Multi-phase Intersections," *J. Comput. Inf. Syst.*, vol. 8, no. 24, pp. 10477–10484, 2012.

[47] J. Kim, "A fuzzy logic control simulator for adaptive traffic management," in *Proceedings of 6th International Fuzzy Systems Conference*, 1997, vol. 3, pp. 1519–1524.

[48] A. N. H. Zaied and W. Al Othman, "Development of a fuzzy logic traffic system for isolated signalized intersections in the State of Kuwait," *Expert Syst. Appl.*, vol. 38, no. 8, pp. 9434–9441, Aug. 2011.

[49] W. Wei, Y. Zhang, and J. B. Mbede, "Traffic signal control using fuzzy logic and MOGA," *Syst. Man, Cybern. 2001 IEEE Int. Conf.*, vol. 2, pp. 1335–1340, 2001.

[50] X. Cheng and Z. Yang, "Intelligent Traffic Signal Control Approach Based on Fuzzy-Genetic Algorithm," *2008 Fifth Int. Conf. Fuzzy Syst. Knowl. Discov.*, vol. 3, no. 1, pp. 221–225, Oct. 2008.

[51] Y. Chiou and Y. Huang, "Stepwise genetic fuzzy logic signal control under mixed traffic conditions," *J. Adv. Transp.*, vol. 47, no. 1, pp. 43–60, 2013.

[52] W. Yang, L. Zhang, Z. He, and L. Zhuang, "Optimized two-stage fuzzy control for urban traffic signals at isolated intersection and Paramics simulation," in *2012 15th International IEEE Conference on Intelligent Transportation Systems*, 2012, pp. 391–396.

[53] L. Conglin, W. Wu, and T. Yuejin, "Traffic Variable Estimation and Traffic Signal

Control Based on Soft Computation," in *Intelligent Transportation Systems, 2004. Proceeding. The 7th International IEEE Conference on*, 2004, pp. 1045–1050.

[54] E. Bingham, "Reinforcement learning in neurofuzzy traffic signal control," *Eur. J. Oper. Res.*, vol. 131, no. 2, pp. 232–241, 2001.

[55] C. Y. D. Yang and W. G. Najm, "Examining driver behavior using data gathered from red light photo enforcement cameras," *J. Safety Res.*, vol. 38, pp. 311–321, 2007.

[56] R. A. Retting, A. F. Williams, and M. A. Greene, "Red-Light Running and Sensible Countermeasures: Summary of Research Findings," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 1640, no. 1, pp. 23–26, 1998.

[57] B. T. J. Gates, D. A. Noyce, L. Laracuente, and T. J. Gates, "Analysis of Dilemma Zone Driver Behavior at Signalized Intersections," in *Transportation Research Board 86th Annual Meeting*, 2007.

[58] Y. Liu, G. Chang, and J. Yu, "Investigating Critical Driver Behavioral Patterns during the Yellow Phase at Signalized Intersections." .

[59] D. Hurwitz, "Application of Driver Behavior and Comprehension to Dilemma Zone Definition and Evaluation," University of Massachusetts, 2009.

[60] R. A. Retting, J. F. Chapline, and A. F. Williams, "Changes in crash risk following re-timing of traffic signal change intervals," *Accid. Anal. Prev.*, vol. 34, pp. 215–220, 2002.

[61] J. A. Bonneson and K. H. Zimmerman, "Effect of Yellow Interval Timing on Red-Light Violation Frequency at Urban Intersections," *Transportation Research Board 83rd Annualt Meeting*, no. January 2004. 2003.

[62] N. Kuge, T. Yamamura, O. Shimoyama, and A. Liu, "A Driver Behavior Recognition Method Based on a Driver Model Framework," *SAE Trans.*, vol. 109, no. 6, pp. 469–476, 2000.

[63] J. C. McCall, D. P. Wipf, M. M. Trivedi, and B. D. Rao, "Lane Change Intent Analysis Using Robust Operators and Sparse Bayesian Learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 3, pp. 431–440, 2007.

[64] A. Pentland and A. Liu, "Modeling and Prediction of Human Behavior," *Neural Comput.*, vol. 11, no. 1, pp. 229–242, 1999.

[65] M. Liebner, M. Baumann, F. Klanner, and C. Stiller, "Driver Intent Inference at Urban Intersections using the Intelligent Driver Model," in *Intelligent Vehicles Symposium*, 2012, pp. 1162–1167.

[66] N. Oliver and A. P. Pentland, "Graphical Models for Driver Behavior Recognition in a Smart Car," in *IEEE Intelligent Vehicles Symposium, 2000*, 2000, no. Mi, pp. 7–12.

[67] G. S. Aoude, V. R. Desaraju, L. H. Stephens, and J. P. How, "Behavior Classification Algorithms at Intersections and Validation using Naturalistic Data," in *Intelligent Vehicles Symposium*, 2011, pp. 601–606.

[68] G. S. Aoude and J. P. How, "Using Support Vector Machines and Bayesian Filtering for Classifying Agent Intentions at Road Intersections." 2009.

[69] J. Owens and A. Hunter, "Application of the Self Organising Map to Trajectory Classification," in *Visual Surveillance, 2000. Proceedings. Third IEEE International Workshop on*, 2000, pp. 77–83.

[70] W. Hu, D. Xie, T. Tan, and S. Maybank, "Learning Activity Patterns Using Fuzzy

Self-Organizing Neural Network," *IEEE Trans. Syst. Man Cybern. Part B*, vol. 34, no. 3, pp. 1618–1626, Jun. 2004.

[71]   D. Koller, J. Weber, and J. Malik, *Robust Multiple Car Tracking with Occlusion Reasoning*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994.

[72]   J. Zhang, F. Wang, and K. Wang, "Data-driven intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1624–1639, Dec. 2011.

[73]   T. a. Dingus, S. G. Klauer, V. L. Neale, A. Petersen, S. E. Lee, J. Sudweeks, M. a. Perez, J. Hankey, D. Ramsey, S. Gupta, C. Bucher, Z. R. Doerzaph, J. Jermeland, and R. R. Knipling, "The 100-Car naturalistic driving study phase II – Results of the 100-Car field experiment," *Dot Hs 810 593*, no. April, p. No. HS-810 593, 2006.

[74]   D. Shinar and R. Compton, "Aggressive driving: An observational study of driver, vehicle, and situational variables," *Accid. Anal. Prev.*, vol. 36, no. 3, pp. 429–437, 2004.

[75]   A. T. McCartt, W. A. Leaf, T. L. Witkowski, and M. G. Solomon, "Evaluation of the Aggression Suppression Program, Milwaukee, Wisconsin," 2001.

[76]   T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006.

[77]   M. H. Beale, M. T. Hagan, and H. B. Demuth, *Neural Network Toolbox^{TM} User' s Guide*, no. June. MathWorks, 2015.

[78]   J. Cox, "Cortexsys 3.1 User Guide," 2016. [Online]. Available: https://www.google.com.sa/_/chrome/newtab?espv=2&ie=UTF-8. [Accessed: 22-Nov-2016].

[79]   Mathworks, *Statistics and Machine Learning Toolbox^{TM} User's Guide*. 2016.

[80]   C. M. Wickens, R. E. Mann, G. Stoduto, J. E. Butters, A. Ialomiteanu, and R. G. Smart, "Does gender moderate the relationship between driver aggression and its risk factors?," *Accid. Anal. Prev.*, vol. 45, pp. 10–18, 2012.

[81]   US Department of Transportation, "Distribution of Licensed Drivers - 2014," 2014. [Online]. Available: https://www.fhwa.dot.gov/policyinformation/statistics/2014/dl20.cfm. [Accessed: 29-Nov-2016].

[82]   S. Krauß, "Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics," 1998.

103