論文 / 著書情報
Article / Book Information

題目(和文)	
Title(English)	DEVELOPMENT OF REAL-TIME CRASH PREDICTION AND INTERVENTION MODEL FOR URBAN EXPRESSWAY WITH DYNAMIC BAYESIAN NETWORK AND DEEP REINFORCEMENT LEARNING
著者(和文)	ROY ANANYA
Author(English)	Ananya Roy
出典(和文)	学位:博士(学術), 学位授与機関:東京工業大学, 報告番号:甲第11335号, 授与年月日:2019年9月20日, 学位の種別:課程博士, 審査員:室町 泰徳,朝倉 康夫,屋井 鉄雄,松岡 昌志,福田 大輔
Citation(English)	Degree:Doctor (Academic), Conferring organization: Tokyo Institute of Technology, Report number:甲第11335号, Conferred date:2019/9/20, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

DOCTORAL DISSERTATION

DEVELOPMENT OF REAL-TIME CRASH PREDICTION AND INTERVENTION MODEL FOR URBAN EXPRESSWAY WITH DYNAMIC BAYESIAN NETWORK AND DEEP REINFORCEMENT LEARNING

ANANYA ROY

DEPARTMENT OF CIVIL AND ENVIRONMENTAL ENGINEERING

TOKYO INSTITUTE OF TECHNOLOGY

SEPTEMBER, 2019

TOKYO, JAPAN

DOCTORAL DISSERTATION

DEVELOPMENT OF REAL-TIME CRASH PREDICTION AND INTERVENTION MODEL FOR URBAN EXPRESSWAY WITH DYNAMIC BAYESIAN NETWORK AND DEEP REINFORCEMENT LEARNING

by

ANANYA ROY

Submitted to the Department of Civil and Environmental Engineering
In partial fulfilment of the requirement for the degree of
DOCTOR OF PHILOSOPHY
at
TOKYO INSTITUTE OF TECHNOLOGY
Tokyo, Japan

Dissertation committee:
Associate Professor Yasunori MUROMACHI (Chairperson)
Professor Yasuo ASAKURA
Professor Tetsuo YAI
Professor Masashi MATSUOKA
Associate Professor Daisuke FUKUDA

SEPTEMBER, 2019

ABSTRACT

This thesis is about improving safety of urban expressways. In order to achieve this goal, a real-time crash prediction model (RTCPM) and intervention is proposed. The idea of real-time crash prediction to identify the hazardous traffic condition leading to crash in advance to take measures to avoid the crash. RTCPMs are highly dependent on high quality and high density traffic and crash data. These data are collected from the detectors or sensors installed on road networks. For this thesis, route 3 Shibuya (11.9 km) and route 4 Shinjuku (13.5 km) - two radial routes of Tokyo metropolitan expressway were chosen because of the availability of traffic data collected round the clock, every minute from detectors which are on average 250 meters spaced.

Most RTCPMs studies are based on the detector data which are already installed and are fixed. For this reason, different approaches were taken by researchers to design the data collection process which effects the RTCPMs. As the construction of an RTCPM is computationally costly, it is not efficient if a model has to be built from scratch for every route and every detector layout. Additionally, instrumental failure of detectors are common. Hence rises the importance of a method to create uniformly distributed simulated detectors. A macroscopic model called cell transmission model (CTM) is adopted to serve this purpose. CTM takes traffic data from the existing detectors and divide it into 'cells' each of which consists of traffic flow data. Additionally, the CTM was modified to accommodate an intervention method called variable speed limit (VSL). The traffic data from CTM was compared to the existing detector data and it was found that the CTM can generate traffic flow data with an error of about 13%.

The performances and transferability of the CTMs were investigated by constructing RTCPMs. For RTCPMs, a machine learning method called Bayesian network (BN) and dynamic Bayesian network (DBN) was employed. Because, these methods are capable of adopting new traffic data and updating themselves. Moreover, BN has the property to handle missing data, which means, in case of instrumental failure, the model will still be able to predict crash likelihood with partially available data. These two are important properties to build a RTCPM. After validating the RTCPMs built with CTM generated data, it was found that the 11 out of 16 CTM-based RTCPMs (BN) outperformed the loop detector based RTCPMs, similar result was seen when RTCPMs were built with DBN.

VSL was chosen as an intervention method. Previously, in several studies VSL was applied to regulate traffic flow by limiting the allowable speeds on the road network. There are several situations when a lowering speeds could be effective, such as in situations of adverse weather, if an incident takes place at downstream, or congestion occurs in a specific segment of road etc. Different studies focused on objectives such as- reducing congestion, active traffic management, regaining bottleneck capacity on freeways etc. Few studies also focused on improving safety. Different strategies were selected by researchers to apply VSL. Still, there are some critical issues which have not been addressed such as- the optimum location of VSL control, duration of VSL control, transferability of the control strategy, effect of VSL control on crash risk. This thesis adopted deep learning methods called Q-learning (QL) and a deep reinforcement learning method called deep Q-learning (DQL). These methods getting popularity because of the model-free nature, which does not require to decide on a strategy for implementation, rather the model itself learns from its experience. To explain in brief, there is an 'agent' (the model) that observes different 'states' (traffic flow parameters, e,g, flow, speed, density etc.) and takes 'actions' (selects VSL values) in order to gain 'rewards' under a 'policy' (reducing crash risk). Hence, the model learns which VSL values are to be taken in which traffic condition to keep the crash risk under a threshold given. In this thesis, a crash risk threshold of 10 was chosen after investing RTCPM's performances and transferability in the earlier chapter. VSL was applied to a 0.90 km long segment of route 4 Shinjuku between Eifuku and Hatagaya with. By comparing both of the method with no speed control cases, it was found that the QL and DQL were able to improve the safety of the study location at the targeted segment where VSL was applied. In few cases, improvement at the upstream of the target location was also observed. By comparing the two method- QL and DQL, it was apparent that the DQL performed better than QL for the same study segment.

The necessity of establishing a universal RTCP and intervention method is undeniable, considering it is data intensive, computational labor dependent. Although, the idea of a model-free intervention method is still a new concept, it has a potential towards the universal crash prevention method. The goal of this thesis was to go one step forward to achieve the universality of RTCP and intervention method.

ACKNOWLEDGEMENT

My first thanks goes to my supervisor, Yasunori MUROMACHI, for his constant support, motivation, patience, and kindness during these past years. I highly regard his professional guidance, inspiration, encouragement and his enthusiasm for thinking out of the box when it comes to incorporating both new technology and classic modeling method to the problem in hand. It has not only been a great pleasure, but also a huge honor for me to work under the supervision of such a respected scholar. I look forward to continue learning from him and collaborating with him in the near future.

I am also profoundly thankful to Professor Tetsuo YAI, Professor Yasuo ASAKURA, associate professor Daisuke FUKUDA and Professor Masashi MATSUOKA for their mentoring and guidance inside and outside academia. Their experience and knowledge have always helped me to learn and improve my research. I'm indebted to Dr. Hossain Moinul for introducing me to my current laboratory and for his continuous guidance and effort to steer my way into the RTCP modeling.

It has been a memorable experience joining the talented research team of Transportation Study Unit (TSU) packed with promising young scientists. I would like to thank the former TSU members and students of Muromachi laboratory, especially to Dr. Seo Toru, Ando Takeru, Kojima Shumpei and Ryosuke Kobayashi for their valuable support and knowledge; and would like to express my gratitude to all my friends and TSU members for being there for me.

I would like to take this opportunity to thank Dr. Carlos F. Daganzo (professor, University of California, Berkeley), Dr. Yu (MARCO) Nie (professor, Northwestern University), Dr. Xiao Qin (Associate Professor, University of Wisconsin-Milwaukee), Dr. Alex Kurzhanskiy (PATH, University of California, Berkeley), Dr. Willem Himpe (Centre for Industrial Management / Traffic & Infrastructure, KU Leuven) for their valuable advice and guidance regarding the development of the CTM. Being a novice in the field of machine learning and AI, I learned and get to meet many enthusiasts from various backgrounds through the study group Machine Learning Tokyo (MLT) for which I am grateful.

Moreover, I would like to express my gratitude to Dr. Hiroshi Warita and the Tokyo Metropolitan Expressway Company Ltd. for supporting my research with necessary data and information.

None of this would have been possible without the financial support from the Ministry of Education, Culture, Sports, Science and Technology (MEXT).

This small effort of mine is dedicated to my family to bear with me throughout past five years.

CONTENTS

NOTATIONS AND ACRONYMS

AADT	Annual Average Daily Traffic
AE	Auto-Encoder
AI	Artificial Intelligence
ANN	Artificial Neural Networks
BN	Bayesian Network
BP	Backpropagation
CFL	Courant Friedrich and Lewy
CNN	Convolutional Neural Network
CRF	Conditional Random Fields
CSP	Constraint Satisfaction Problems
CTM	Cell-Transmission Model
DAG	Directed Acyclic Graph
DBN	Dynamic Bayesian Network/ Deep Belief Net
DDQN	Dueling Deep Q Network
DL	Deep Learning
DNN	Deep Neural Network
DQN	Deep Q Network
DRL	Deep Reinforcement Learning
FD	Fundamental Diagram
GAN	Generative Adversarial Network
GEH	Geoff E. Havers
GMDH	Group Method of Data Handling
GMM	Gaussian Mixture Models
GPU	Graphical Processing Unit
GRU	Gated Recurrent Units
ITS	Intelligent Transportation System
LSTM	Long Short-Term Memory

MaxEnt	Maximum Entropy
ML	Machine Learning
MLP	Multilayer Perceptron
NLP	Natural Language Processing
NN	Neural Network
RBM	Restricted Boltzmann Machines
RF	Random Forest
RL	Reinforcement Learning
RNN	Recurrent Neural Network
RTCPM	Real-Time Crash Prediction Model (S)
SGD	Stochastic Gradient Descent
SVM	Support Vector Machines
TMS	Truth-Maintenance-Systems
TPU	Tensor Processing Unit
VSL	Variable Speed Limit

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Background and context

1.1.1 Context

Road traffic crashes are one of the world's largest public health and injury prevention problems. According to WHO, approximately 1.3 million people die each year on the world's roads. Several crash prediction models have been built as a tool to augment road safety. A huge concern regarding the conventional models was their incapability to predict crashes that may be caused due to the suddenly developed unfavorable driving condition on a specific road section (Oh et al, 2001; Lee et al., 2003a,b). Oh et al. (2001) introduced a fourth component alongside road geometry and environment, vehicle and human; - the traffic dynamics. They suggested that crashes can occur even if the vehicle, environment and road geometry are favorable to safe driving. This happens due to sudden formation of hazardous traffic condition causing driving discomfort. Also, complex traffic dynamics instigates driving errors (Hossain and Muromachi, 2009). This contrived the opportunity to improve the shortcoming of the conventional crash prediction models that employ aggregated measures of traffic flow variables to identify hazardous locations. The findings lead to the necessity to predict the instantaneous crash risks for given road sections. Lee et al. (2003b) have compared this new idea with the already existing incident prediction models and implied that the proactive nature of the real-time crash prediction model contains much higher potential in improving road safety than the incident prediction models that minimizes the consequences of a crash by preventing secondary crashes.

With the advancement of Intelligent Transportation System (ITS) and development of advanced transportation information systems (ATIS), traffic data collection has become easier. Consequently, numerous Real-time Crash Prediction Models (RTCP) assessing crash or nearly crash-prone situations of highways and expressways were introduced. The concept of real time crash prediction is based on the hypothesis that the probability of a crash on a specific road section can be predicted for a very short time window using the instantaneous traffic flow data (Lee et al. 2002 and 2003a, b; Pande and Abdel-Aty, 2005; Hossain and Muromachi, 2010a,b).

The modern traffic data sensing technologies now present an array of options (inductive loop detectors, magnetic sensors, video image processors, microwave radar sensors, infrared sensors, laser radar sensors, etc.). Many of the modern sensor technologies have the ability to classify vehicles by their length and report vehicle presence, flow rate, occupancy, and speed for each class. The recent widespread availability of GPS technologies for in-vehicle navigation systems along with the advancements in the telecommunication sector and hand held devices such as smart phones have created the opportunity to collect traffic flow data directly from the vehicles which has, to a great extent, eliminated the needs of incrementing the expressways with sensors. When the data are available, they can be fed into the real-time crash prediction models and the output of these models can be utilized through the means to inform the road users about the prevailing road condition and/or providing driving guidance. The means can be posting warning message (Lee and Abdel-Aty, 2008) through variable message signs (VMS), variable speed limits, also known as VSL (Abdel-Aty et al., 2006b,c, 2008b;Lee and Abdel-Aty, 2008) and ramp metering (Lee et al., 2006b;Abdel-Aty et al., 2007).

1.1.2 Real-time crash prediction model

The task of real-time crash prediction is to detect the hazardous traffic condition formation. Recognizing hazardous traffic conditions from real-time traffic data is very important as it provides with knowledge about the crash mechanism, which in turn helps to introduce interventions for real-time traffic management.

Appropriate traffic variables needs to be chosen for RTCPM building, which can locate hazardous and non-hazardous traffic situations. The necessity of learning about hazardous situations and the availability of high density traffic data from the detectors have raise the importance of studying crash mechanisms for intervention application.

There are several limitations when it comes to use detector data. The most common is instrumental failure. For this, alternative detector spacing should also be available to ensure unhindered monitoring of the area of interest. This will also enable in identifying the optimum detector layout plan. In addition to detector data, sufficient amount of crash data must be available for RTCPMs. The model can only perform well when the quality of the data is ensured by removing all the faulty data.

Then, the next goal is to select a highly flexible modeling method should be chosen that can perform these activities without needing to re-build or re-calibrate itself from the scratch. The model should be able to update itself with the addition of new data, it should be able to incorporate new variables, and must be able to predict crash even if few data are missing

Most real-time crash prediction studies use volume/occupancy/speed parameters data extracted from sampled floating cars or road side detectors. It has been indicated that for different combinations of upstream and downstream traffic states, the crash involvement rates and crash risk ratios (ratio of crash cases and non-crash cases) are inconsistent. Considering that a crash can be induced by the disturbance of traffic flow before the crash occurs, time series traffic data consisting of several time intervals should be used to illustrate the dynamic process of traffic flow before crash occurrence. Thus, it is inessential to establish a single model that can address such time series data and the evolving process of traffic flow. Therefore, in this study, Bayesian network (BN) and dynamic Bayesian network (DBN) model of time sequence traffic data has been proposed to investigate the relationship between crash occurrence and dynamic traffic flow data with one-minute-interval of six months on one of the busiest Tokyo Metropolitan Expressway (March to August, 2014) kindly provided by the Tokyo Metropolitan Expressway Company Limited.

1.1.3 Macroscopic traffic model

The existing RTCPMs are developed depending on the detector layout that exists in the study area. Hence, variation in detector layout raises the issue of spatial transferability of the RTCPMs as altering the locations or installing new detectors to replicate the detector layout of the model is neither practical nor cost effective. Moreover, even for new roads, it may not always be plausible to follow a specific detector spacing. To address these issues, there has been an urge to devise a mechanism to simulate traffic sensor data of desired spacing from any detector layout that can be fed into an RTCPM. This study employed a macroscopic dynamic freeway traffic model called CTM, introduced by Daganzo (1994,1995), which is consistent with the hydrodynamic theory of traffic flow (Lighthill, 1955), to transform the traffic states obtained from non-uniform detector layouts into a pre-defined detector layout. The CTM was chosen for its analytical simplicity and ability to reproduce congestion propagation dynamics.

1.1.4 Real-time intervention

Several studies showed that it is possible to reduce the risks of road crashes by altering the traffic states with suitable interventions. For a real-time intervention, a threshold of crash likelihood can be a useful measure to formulate a proactive control strategy (Lee et al., 2004). Studies on real-time interventions followed either a traffic simulation program (Lee et al., 2004, 2006; Abdel-Aty et al., 2006a,b,2007a,b,2008, Yu and Abdel-Aty, 2014; Abdel-Aty and Wang, 2017) or a driving simulator (Lee and Abdel-Aty, 2008) to reproduce pre-crash traffic conditions and various countermeasures such as a variable-message sign (VMS) (Lee and Abdel-Aty, 2008) and a variable speed limit (VSL) (Lee et al., 2004; Abdel-Aty et al., 2006a,b,2007a,b,2008; Lee and Abdel-Aty, 2008; Yu and Abdel-Aty, 2014; Abdel-Aty and Wang, 2017). Similarly, a coordinated or uncoordinated ramp metering (Lee et al., 2006; Abdel-Aty et al., 2007a,b; Abdel-Aty and Gayah, 2010) helped control the crash-prone traffic conditions effectively. Park et al. (2018) used warnings in order to avoiding the secondary crashes.

Variable speed limit (VSL) triggered through RTCPMs were adopted in a number of studies which reduced the crash probability effectively (Lee et al., 2004; Li et al., 2016, 2017; Abdel-Aty et al., 2006a, 2007b, 2008). Lee et al. (2004) reduced the speed when a crash probability measured by an RTCPM crossed a predetermined threshold. Abdel-Aty et al. (2006a) was successful by changing the speed limit gradually every 10 min at 5 mph rate: an abrupt change in space; a reduction for upstream while an increase for downstream. Abdel-Aty et al. (2006b, 2007b, 2008) found that VSL was effective for medium to high speed regimes and had a limited impact for lower speeds. Later on, Abdel-Aty et al. (2008) suggested a homogeneous speed zone for VSL implementation. Lee and Abdel-Aty (2008) stated that VMS and VSL in tandem could reduce speed variations. Abdel-Aty et al. (2006b) found improved safety in the zone of VSL implementation but the high-crash potential was relocated at the downstream. Carlson et al. (2011) and Lu et al. (2011) applied VSL at the upstream of the bottleneck area to control the outflow of the VSL section. This way, the capacity drop at the bottleneck can be avoided, and the bottleneck capacity can be retrieved. A solution to the shifting of crash risk can be found in Yu and Abdel-Aty (2014) who proposed an optimization algorithm to minimize the overall crash risk for the total VSL corridor. Li et al. (2014) employed VSL close to the freeway recurrent bottlenecks to reduce rear-end collision risks where the control strategy included a controlling of start-up threshold, a

target speed limit (56.33 km/h), and a speed change rate (16.09 km/h every 30 s). Later on, Li et al. (2016) considered a start-up threshold of 20% as a control strategy to activate the VSL in a large-scale freeway segment and found that the crash risk was reduced by 22.62% and injury severity by 14.67%. Recently, Abdel-Aty and Wang (2017) applied VSL successfully to the congested weaving sections of an expressway for reduction of crash risk.

The aforementioned studies on the combination of RTCPM and VSL focused only on adjusting the speed limit in respect of control strategy, VSL control zone, time of control, and response time considering the pattern of hazardous traffic state. Accordingly, the objective was limited to bring the traffic back to normal in the best possible way using a predetermined set of VSL-based interventions administered during a specified time interval. Therefore, they lack the embedded intelligent agent capable of learning by itself to tackle non-recurrent complex traffic patterns. Reinforcement learning (RL), an artificial intelligence-based semi-supervised machine learning algorithm, can support VSL in this regard. In RL, an agent reacts with the environment through several trial and error to optimize the total reward by choosing a state-action pair for every time step (Watkins, 1989; Sutton and Barto, 1998; Hasselt, 2011). Thus, an RL agent exhibits ability for decision making in respect of proactive speed control (Li et al., 2017; Zhu and Ukkusuri, 2014, Davarynejad et al., 2011). The ability to take real-time proactive control decisions without the need for a model architecture makes RL appealing in ITS field (Rezaee et al., 2012; El-Tantawy et al., 2010; Abdulhai et al., 2003). A Q-learning-based multi-agent RL, which is the most commonly used RL algorithm, was used in studies to reduce crash risk by optimizing the ramp metering (Davarynejad et al., 2011; Rezaee et al., 2012), while, in another study (El-Tantawy et al., 2010), a multi-agent RL was used in conjunction with game theory to alleviate traffic gridlock. An R-Markov average reward technique (R-MART)-based RL was used to optimize VSL control for reducing travel time and vehicle emission (Zhu and Ukkusuri, 2014). Li et al. (2014, 2016) improved on their genetic algorithm-based VSL optimization strategy by using Q-learning-based RL at a freeway recurrent bottleneck (Li et al. 2017). Isele et al. (2018) learned policies and active sensing behaviors employing RL that exceeded the capabilities of the commonly used heuristic approaches in several categories such as task completion time, goal success rate, and ability to generalize the problem, for navigating occluded intersections with autonomous vehicles.

Although, Q-learning for VSL-controlled optimization performed pretty well compared to the traditional feedback-based VSL control, there are a few issues, such as adaptability of continuous traffic states, location of VSL control sections, and the reliability of the VSL models in real time in terms of crash risk reduction, which are yet to be resolved. In this study, Q-learning and deep Q-network (DQN) are adopted to design the self-learning intelligent agents for real-time proactive VSL control. The study commences by simulating the traffic state data from Shinjuku 4 routes of Tokyo Metropolitan Expressways Company Limited when no crash took place using the CTM- a microsimulation model. Then, Bayesian Network based four different RTCPMs were constructed considering different variable combinations. Afterwards, a deep reinforcement learning-based intelligent intervention is designed applying Q-learning and DQN for VSL control, which was programmed with Python based library called Keras which is integrated to the CTM. Eventually, the whole system utilizes the crash risk predicted with RTCPMs to evaluate real-time safety and compare the proactive safety management performances of QL and DQN-based VSL interventions with situations when 'no VSL control' was administered.

1.2 Definitions and terminologies

1.2.1 Definitions and terminologies: Traffic flow

The Highway Capacity Manual (2000) provides the following definitions of the basic traffic related terminologies. The symbols x and t represents distance (measured in the direction of traffic flow) and time respectively.

Speed $\boldsymbol{v}$ $(\boldsymbol{x}, \boldsymbol{t})$***:*** Speed is the rate of motion expressed as distance per unit time. Depending on how it is measured, it can be either space mean speed or time mean speed. The speed of a vehicle is defined as the distance it travels per unit of time. Most of the time, each vehicle on the roadway will have a speed that is somewhat different from those around it.

Space-mean speed: It is computed by dividing the length of a road by the average time it takes for vehicles to traverse it. It is the arithmetic mean of the speed of those vehicles occupying a given length of road at a given instant.

Time-mean speed: It is measured by taking average of the speeds of the vehicles observed passing a given point. Time-mean speed can be obtained directly from a measuring device, sensor or

detector. It is the arithmetic mean of the speed of vehicles passing a point during a given time interval.

Free-flow speed v (x,t): It is the average speed of the traffic measured under conditions of low volume, when vehicles can freely move at their desired speed.

Volume (x,t): Volume is simply the number of vehicles that pass a given point on the roadway in a specified period of time. By counting the number of vehicles that pass a point on the roadway during a 15-minute period, you can arrive at the 15-minute volume. Volume is commonly converted directly to flow (q), which is a more useful parameter.

Flow q (x,t): It is the total number of vehicles that pass by the point x, given a time interval t, divided by the length of the time interval. It is usually expressed as an hourly rate and is easily measured with road sensors. Flow is one of the most common traffic parameters. Flow is the rate at which vehicles pass a given point on the roadway, and is normally given in terms of vehicles per hour. The 15-minute volume can be converted to an hourly flow by multiplying the volume by four. If our 15-minute volume were 100 vehicles, we would report the flow as 400 vehicles per hour. For that 15-minute interval of time, the vehicles were crossing our designated point at a rate of 400 vehicles/hour.

Density k (x,t): It is the number of vehicles occupying a length of road about point x at time instant t. This measurement is difficult as it requires observation of a stretch of road. Instead, it is often approximated from measurements of flow and speed as:

$$k\,(x,t) = \frac{q(x,t)}{v(x,t)} \qquad (2.4)$$

Density refers to the number of vehicles present on a given length of roadway. Normally, density is reported in terms of vehicles per mile or vehicles per kilometer. High densities indicate that individual vehicles are very close together, while low densities imply greater distances between vehicles.

Headway (t): Headway is a measure of the temporal space between two vehicles. Specifically, the headway is the time that elapses between the arrival of the leading vehicle and the following vehicle at the designated test point. Headway between two vehicles can be measured by starting a

chronograph when the front bumper of the first vehicle crosses the selected point, and subsequently recording the time that the second vehicle's front bumper crosses over the designated point. Headway is usually reported in units of seconds.

Spacing $(\boldsymbol{x})$***:*** Spacing is the physical distance, usually reported in feet or meters, between the front bumper of the leading vehicle and the front bumper of the following vehicle. Spacing complements headway, as it describes the same space in another way. Spacing is the product of speed and headway.

Gap $(\boldsymbol{x})$***:*** Gap is very similar to headway, except that it is a measure of the time that elapses between the departure of the first vehicle and the arrival of the second at the designated test point. Gap is a measure of the time between the rear bumper of the first vehicle and the front bumper of the second vehicle, where headway focuses on front-to-front times. Gap is usually reported in units of seconds.

Clearance $(\boldsymbol{x})$***:*** Clearance is similar to spacing, except that the clearance is the distance between the rear bumper of the leading vehicle and the front bumper of the following vehicle. The clearance is equivalent to the spacing minus the length of the leading vehicle. Clearance, like spacing, is usually reported in units of feet or meters.

Demand: It is the number of vehicles that desire to use a given facility during a specified time period.

Capacity: It's the maximal hourly rate at which vehicles reasonably can be expected to traverse a point or a uniform section of a lane or a roadway during given time period under prevailing roadway, traffic and control conditions.

Bottleneck: It is defined as any road element where demand exceeds capacity. Freeway bottlenecks sometimes appear near heavy on-ramps, where a localized increase in demand is combined with a decrease in capacity due to lane changing.

Jam density $\boldsymbol{k_j}$ $(\boldsymbol{x}, \boldsymbol{t})$: the density when speed and flow are zero.

Shockwaves: Shockwaves occur as a result of differences in flow and density which occur when there are constrictions in traffic flow. These constrictions are called bottlenecks. The speed of growth of the ensuing queue is the shockwave, and is the difference in flow divided by the difference in density.

Hazardous state: In this thesis, hazardous state refers to the traffic states 3 minutes before a crash took place.

Normal state: In this thesis, normal state refers to the traffic states on a normal or no-crash day, 3 minutes before corresponding to the hazardous state.

1.2.2 Definitions and terminologies: Cell transmission model

Cell: It is the subdivision of a road network which is generated by CTM and from where traffic flow parameters can be generated.

Links: Inside a network, each cell is connected with links with one another.

Nodes: There are three kinds of nodes- simple, diverge and merge. The cells, which has a diverging location, it is called a diverge node, if it has a merging location, it's called a merge node, all other nodes are called simple nodes.

1.2.3 Definitions and terminologies: Real time crash prediction

Real-time: It describes computing systems that are able to deal with and use new information immediately and therefore influence or direct the actions of the objects supplying that information (Cambridge Advanced Learner's Dictionary, 2018).

Intelligent*:* A model or agent can be called intelligent if it is able to acquire knowledge from the surroundings or environment given and can come up with reasoning and is able to formulate solutions accordingly.

Road safety management system: a system that is capable of monitoring and evaluating the road safety condition in real-time by instantaneously sensing the traffic flow data, introducing appropriate intervention upon detecting sign of hazardous traffic formation, varying its evaluations

and/or interventions in response to varying situations and past experience, and updating itself through its inherent capability to adapt to new situations (Hossain, dissertation 2011).

1.2.4 Definitions and terminologies: Deep reinforcement learning

Supervised learning: In the field of ML, there are three types of tasks that can be done- supervised learning, unsupervised learning and reinforcement learning. Supervised learning is learning from a training set of labeled examples provided by a knowledgeable external supervisor.

Unsupervised learning: Unsupervised learning is typically about finding structure hidden in collections of unlabeled data. Reinforcement learning (RL) is the task of learning how agents ought to take sequences of actions in an environment in order to maximize cumulative rewards.

Reinforcement learning: whereas RL is different from unsupervised learning because it tries to maximize a reward signal instead of trying to find hidden structure. Reinforcement learning algorithms study the behavior of subjects in an environment and learn to optimize that behavior.

Deep Learning: DL is a branch of machine learning that uses what's called "supervised" learning – where the neural network is trained using labeled data – or "unsupervised" learning – where the network uses unlabeled data and looks for recurring patterns.

Deep reinforcement learning: Deep reinforcement learning is the combination of reinforcement learning (RL) and deep learning.

Agent: An agent is a program that observes its environment and collects data, then performs some actions following a policy in order to maximize reward.

Environment: An environment is where an agent interacts via states and actions.

Reward: This is the self-evaluation method of an agent about the quality of the action it takes following a policy.

1.3 Objective and scope

1.3.1 Objective

The main objective of this thesis is to affirm traffic safety on urban expressway. As discussed in the previous section, traffic safety analysis has been a crucial issue in the field of transportation engineering for a long time. Over the last few decades several methods have been employed to achieve a universal way to prevent crash and maintain uninterrupted traffic flow, especially in the urban context where traffic flow is huge.

To achieve this objective, two major tasks needs to be done- predict the crash and prevent it from happening. This research's focus is to build a real-time crash prediction model in order to receive the information of a possible occurrence of crash or, the crash likelihood before it takes place so that we can take preventive measures.

The concept of real-time crash prediction is yet considered as 'emerging'. Several researchers has proposed different methodologies to build an efficient RTCPM by introducing different modifications and incorporating various features to improve the model. Whilst, it is necessary to explore different arenas for improvements, it is also important to have a common ground or universality for an RTCPM in order to ensure transferability and applicability of the model. This is because of the obvious fact that the traffic modelling requires handling of tremendous amount of data and long time for model training and development. Due to the huge computational cost, it is inefficient to construct RTCPMs from scratch every time, individually for every road networks all over the world. The efficient way to build an RTCPM is to add such features which would increase its universality, so that in future, a model built for one road network could be used for predicting crash of another one. Hence, a modified version of a classic macroscopic traffic model based on kinematic wave theory called the cell transmission model (CTM) is adopted in this research. The idea is to apply CTM to divide the study area into homogeneous small sections called 'cell'. These cells would act as virtual detectors encompassing traffic flow parameters from the existing detectors. The advantages of uniformly distributed detectors are two fold- (a) the ability of collecting and controlling traffic data at preferred sections of study area, and (b) creating a scope of transferability in terms of geometric design of a road network.

The second major task is to prevent the predicted crash. Amongst different methods, the variable speed limit (VSL) is adopted here as an intervention. But, VSL control has few limitations such as deciding when, where and what amount of speed needs to be decreased/ increased in order to prevent crash while maintaining uninterrupted traffic flow. Instead of deciding these matters by ourselves, which could be confusing at times because of versatility of intervention policies by different transportation experts, an intelligent deep reinforcement learning (DRL) agent is given the task of deciding upon the policies. DRL is gaining popularity recently due to its model-free features, which enables the DRL agent to learn from the environment through interaction by taking actions, then learning whether the decision of taking those actions aere actually good or not by receiving rewards over time. Because of its model-free property, we do not have to feed the agent with labeled data to train it like we usually do in case of classification or regression, or other machine learning problems. The agent can observe the states in an environment e.g. traffic data, take an action e.g. select a VSL value, then get a positive or negative reward for taking that action, e.g. positive if safety improves, negative otherwise- by doing it over and over, and the agent learns which actions are the best ones in which states. This is the basic concept of DRL which is utilized in this thesis for deciding VSL-based intervention modeling.

Hence, the objectives of this research is to address the answers of the following questions-

- How to predict crash risk in real-time?
- In what way the universality of a RTCPM can be improved?
- How to decide on a policy for an intervention to prevent crash?

1.3.2 Scope

The scope of this research is exclusively limited to the urban expressways. The entire research is conducted with the traffic flow data i.e. flow (veh/min), speed (km/h) and occupancy (%) collected during march, 2014 to August, 2014 from the two major radial routes- route 3 Shibuya and route 4 Shinjuku of Tokyo metropolitan expressway company ltd. The route 3 Shibuya is a 11.9 km long two-lane radial route stretching between Takaido and Gaien consists of (43+41) freeway detectors and 6 on-ramp and 5 off-ramp detectors in the inbound and outbound directions. Route 4 Shinjuku is 13.5 km long radial route located between Yoga and Takagicho harboring (50+40) freeway, 10

on-ramp and 11 off-ramp active detectors altogether including both lanes and both directions. These are the detectors which were found to be in excellent working conditions throughout the day and collect traffic flow data 24x7 which was later aggregated to one minute. Data was extracted during March 2014 to August, 2014. This high density data makes these routes suitable candidates for RTCPM and intervention construction. Moreover, the route encompasses about 300 crash cases recorded from reported data. Additionally, both of the routes has long freeway segment which was required for the CTM model in order to avoid merging and diverging conditions. The detectors are spaced about 250 meters on average. The high density data with accuracy and fairly equidistant detector layout and availability of long basic freeway segment etc. makes this study location appropriate for this research.

1.4 Contributions

The contributions of this thesis are situated on two main components of the traffic safety- RTCPM and intervention, and on the synergy that emerges when they are considered together.

1.4.1 Real-time crash prediction

A machine learning algorithm namely Bayesian network and dynamic Bayesian network was adopted for RTCPM construction with the traffic flow parameters extracted from the loop detectors. The models were built and validated with various combinations of information parameters to investigate the robustness of the models and to ensure the adaptability property of the BN algorithm. The most influential traffic parameters in terms of crash prediction was also identified which later on, were used to build the intervention model.

1.4.2 Macroscopic traffic simulation

The first order kinematic wave theory based macroscopic traffic model called the cell transmission model (CTM) was used to generate uniformly distributed simulated detectors to achieve the second objective mentioned earlier. The CTM was then modified to enable it for incorporating variable speed limit (VSL). The reason for modifying the original CTM was to use it for traffic data simulation when VSL is activated. The original CTM model is driven by a fixed fundamental diagram (FD) introduced by Daganzo (1994, 1995), but the modified CTM has the flexibility to

change its FD depending on the speed condition of the study location and the on the intervention model.

1.4.3 Real-time intervention

The final contribution of this thesis is the real-time intervention for the crash predicted earlier. This is done by introducing a deep reinforcement learning (DRL) based model-free intervention method where an intelligent 'agent' will observe 'states' its 'environment' and will take 'actions' in order to receive 'rewards'. The agent's goal is to maximize the rewards, and to do that, it undergoes numerous episodes of observations to learn the appropriate actions according to several states. This state-action relationship results in policy optimization. In this study, the agent observes traffic data, and looking at the crash risks associated with the traffic data observed at a specific time, takes an action of changing the speed limit and gets a positive or negative reward depending on the improvement in safety of the target location.

1.5 Outline

The outline of thesis is described below. A schematic diagram of the flow of the thesis is shown in **Figure 1.1**.

The thesis is divided into six chapters, chapter 1 being the introduction, where a general idea of the real-time crash prediction (RTCP) and intervention methods are discussed. Then the objectives, scope and contribution of the thesis is defined.

Chapter 2 is the literature review discussing the evolution and current condition of macroscopic models, RTCPMs and VSL control methods are explained along with the scope of improvements.

Chapter 3 is on the study area and construction of CTM-based macro simulation and validation. The necessity of uniformly distributed simulated detectors and how to generate those are discussed here.

Chapter 4 is about the RTCPMs constructed with BN and DBN method. Although, later on, only the BN-based RTCPMs were adopted for intervention, a comparison between the performance of BN and DBN models in terms of predicting crash likelihood is shown.

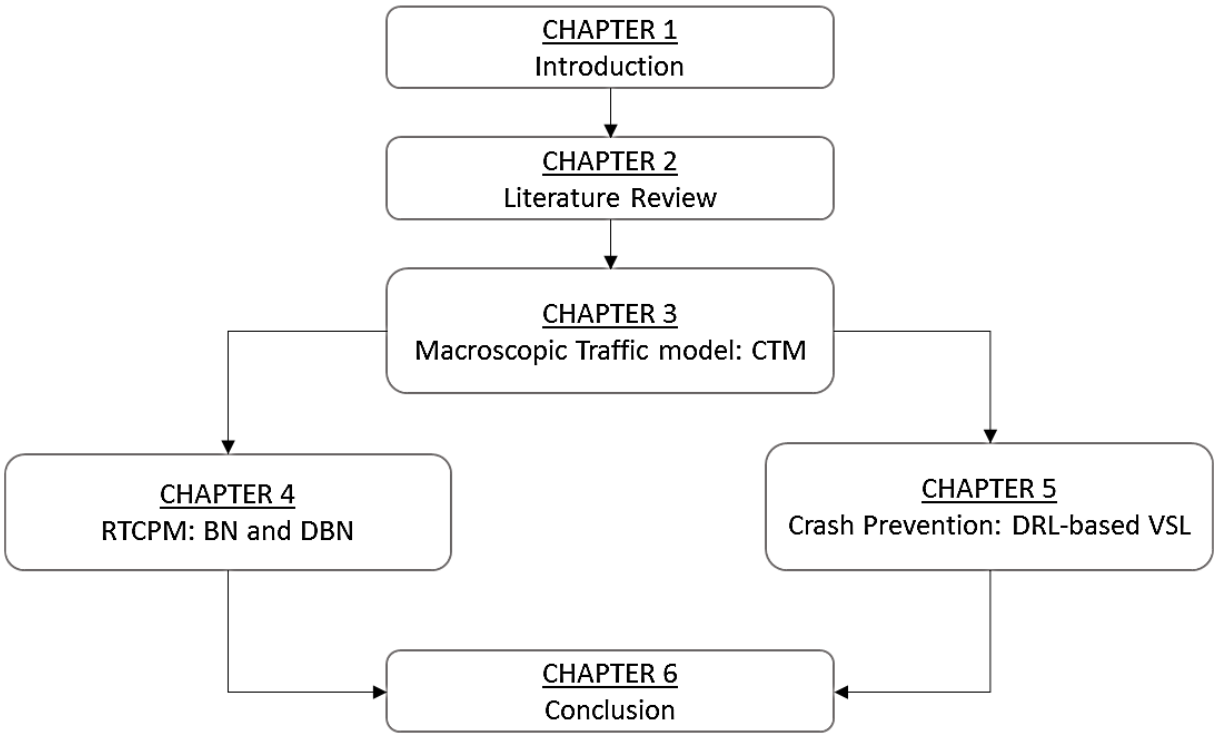


Figure 1. 1 Schematic diagram of the flow of the thesis.

Chapter 5 is the deep reinforcement learning- based real-time intervention is explained. A brief overview of te evolution of deep learning (DL) and reinforcement learning (RL) is also discussed as these are comparatively new in the field of traffic management.

Finally, chapter 6 concludes the findings and discusses the results in terms of achieving the objectives along with future scope of work.

1.5 Chapter Conclusion

Road safety management has always been an essential topic in the field of transportation engineering. Several attempts have been made to 'perfect' the crash prediction models. RTCPMs are one of those attempts which utilizes the advancement of ITS and active traffic management systems. The main issue in improving RTCPMs was found to be the adaptability and transferability of the models, so that the model could update itself when new traffic data is received without requiring to build from scratch. Another issue addressed here is the inconsistency of the inter-

detector spacing throughout the route, which requires to take different data collection and modeling methods for different routes. A uniformly distributed detector layout could solve this issue while addressing the situations of instrumental failure or inaccessible road geometry. And, last but not the least, the intervention which is intelligent enough to learn and decide by itself how to control the allowable speed limit when a crash prone situation occurs is discussed. It is an emerging technology, yet with high potential.

1.6 Chapter References

Abdel-Aty M. and Pande A. (2005) Identifying crash propensity using specific traffic speed conditions, Journal of Safety Research, Vol. 36 No.1, pp. 97-108. Abdel-Aty, M. Uddin, N. And Pande, A. (2005). Split Models for Predicting Multivehicle Crashes During High-Speed and Low-Speed Operating Conditions on Freeways. Transportation Research Record: Journal of the Transportation Research Board, No. 1908, TRB, National Research Council, Washington, D. C., pp. 51-58.

Abdel-Aty M., Gayah V., 2010. Real-time crash risk reduction on freeways using coordinated and uncoordinated ramp metering approaches. Transportation Engineering 136, 410-423.

Abdel-Aty, M. and Pande, A. (2006). ATMS Implementation System for Identifying Traffic Conditions Leading to Potential Crashes. IEEE Transactions on Intelligent Transportation Systems, Vol. 7, No. 1. pp. pp. 78-91. Abdel-Aty, M., J. Dilmore, and L. Hsia. (2006a) Applying Variable Speed Limits and the Potential for Crash Migration. In Transportation Research Record: Journal of the Transportation Research Board, No. 1953, Transportation Research Board of National Academics, Washington, D.C.,. 2006, pp. 21-30.

Abdel-Aty, M., and Wang, L., 2017. Implementation of variable speed limits to improve safety of congested expressway weaving segments in microsimulation, Transportation Research Procedia 27, 577-584.

Abdel-Aty, M., Cunningham, R.J., Gayah, V.V., Hsia, L., 2008. Dynamic variable speed limit strategies for real-time crash risk reduction on freeways. In: Transportation Research Record:

Journal of the Transportation Research Board, No. 2078, Transportation Research Board of the National Academies, Washington, D.C. pp. 108-116.

Abdel-Aty, M., Dilmore, J. and Dhindsa, A. (2006b). Evaluation of Variable Speed Limits for Realtime Freeway Safety Improvement. Journal of Accident Analysis and Prevention. Vol. 38. No. 2 pp. 335-345.

Abdel-Aty, M., Dilmore, M.J., Gayah, V.V., 2007a. Considering various ALINEA ramp metering strategies for crash risk mitigation on freeways under congested regime. Journal of Transportation Research, Part C 15(2), 113-134.

Abdel-Aty, M., Dilmore, M.J., Hsia, L., 2006b. Applying variable speed limits and the potential for Crash Migration. In: Transportation Research Record: Journal of the Transportation Research Board, No. 1953, Transportation Research Board of the National Academies, Washington, D.C., pp. 21-30.

Abdel-Aty, M., Pande, A., Das, A. and Knibbe, W. J. (2008a). Assessing Safety on Dutch Freeways with Data from Infrastructural-based Intelligent Transportation Systems. Transportation Research Record: Journal of the Transportation Research Board, No. 2083, TRB, National Research Council, Washington, D. C., pp. 153-161.

Abdel-Aty, M., Pande, A., Lee, C., 2007b, Gayah, V., Santos, C. D., 2007b. Crash risk assessment using intelligent transportation systems data and real-time intervention strategies to improve safety on freeways. Journal of Intelligent Transportation Systems 11(3), 107-120.

Abdel-Aty, M., Pemmanaboina, R. And Hsia, L. (2006c). Assessing Crash Occurrence on Urban Freeways by Applying a System of Interrelated Equations. Transportation Research Record: Journal of the Transportation Research Board, No. 1953, TRB, National Research Council, Washington, D. 47 C., pp. 1-9.

Abdulhai, B., Pringle, R., Karakoulas, G.J., 2003. Reinforcement learning for true adaptive traffic signal control. Journal of Transportation Engineering, ASCE 129, 278-285. DOI: 10.1061/(ASCE)0733-947X(2003)129:3(278).

Carlson, R.C., Papamichail, I., Papageogiou, M., 2011. Local feedback based mainstream traffic flow control on motorways using variable speed limits. IEEE Trans. Intell. Transp. Syst., 12(4), 1261-1276.

Daganzo, C. F. Fundamentals of transportation and traffic operations. Oxford: Pergamon, 1997.

Daganzo, C. F. The Cell Transmission Model, Part II: Network Traffic. Transportation Research Part B, Vol. 29, 1995, pp. 79–93.

Daganzo, C. F. The Cell Transmission Model: A Dynamic Representation of Highway Traffic Consistent With the Hydrodynamic Theory. Transportation Research Part B, Vol. 28, 1994, pp. 269–287.

Davarynejad, M., Hegyi, A., Vrancken, J., van den Berg, J., 2011. Motorway Ramp-Metering Control with Queuing Consideration using Q-learning, Proc. 14th IEEE Int. Conf. Intell. Transp. Syst., Washington, DC, USA, pp. 1652-1658.

El-Tantawy S., Abdulhai, B., 2010. Towards multi-agent reinforcement learning for integrated network of optimal traffic controllers. Transp. Letters 2(2), 89-110.

Hasselt, H.V., Guez, A., Silver, D., 2016. Google DeepMind. Deep Reinforcement Learning with Double Q-Learning, Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16), pp. 2094-2100.

Hasselt, H.V., Insights in Reinforcement Learning. PhD thesis, 2011. Utrecht University, the Netherlands.

Hossain, M., Abdel-Aty, M., Quddus, M. A., Muromachi, Y. and Nafis, S. S. Real-time crash prediction models: state-of-the-art, design pathways and ubiquitous requirements, In the Journal of Accident Analysis and Prevention, Vol. 124, pp. 66-84.

Hossain, M., Muromachi, Y., 2012. A Bayesian network based framework for real-time crash prediction on the basic freeway segments of urban expressways. Accident Analysis and Prevention. 45, 373-381. http://doi.org/10.1016/j.aap.2011.08.004.

Hossain, M., Muromachi, Y., 2013. Real-time crash prediction model for the ramp vicinities of urban expressway. IATSS Research 37(1), 68-79. http://doi.org/10.1016/j.iatssr.2013.05.001.

Hossain, M., Muromachi, Y., 2013a. Understanding crash mechanism on urban expressways using high-resolution traffic data. Accid. Anal. Prev. 57, 17–29.

Isele, D., Rahimi, R., Cosgum, A., Subramanian, K., Fujimura, K., 2018. Navigating occluded intersections with autonomous vehicles using deep reinforcement learning. https://arxiv.org/pdf/1705.01196.pdf

Jensen, F.V., Nielsen, T.D., 2007. Bayesian networks and decision graphs. Springer, NY.

Katrakazas, C., Quddus, M.A., Chen, W.H., 2017. A simulation study of predicting conflict-prone traffic conditions in real-time. Presented at the Transportation Research Board 96th Annual Meeting, Washington D.C., USA, 8-12 January 8-12 2017.

Kojima, S., Muromachi, Y., 2017. Study on Safety Measures using a Real-time Traffic Accident Prediction Model on Urban Expressway, Proceedings of 37th Traffic Engineering Research Presentation, Japan Society of Traffic Engineers, pp. 235-238.

Krizhevsky, A., Sutskever, I., Hinton, G., 2012. ImageNet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. 25, 1106-1114.

Lee, C., Abdel-Aty, M., 2008. Testing effects of warning messages and variable speed limits on driver behavior using driving simulator. In: Transportation Research Record: Journal of the Transportation Research Board, No. 2069, Transportation Research Board of the National Academies, Washington, D.C., pp. 55-64.

Lee, C., Hellinga, B., and Ozbay, K., 2006. Quantifying effects of ramp metering on freeway safety. Journal of Accident Analysis and Prevention 38(2), 279-288.

Lee, C., Hellinga, B., Saccomanno, F., 2003. Real-time crash prediction model for the application to crash prevention in freeway traffic. Transportation Research Record: Journal of the Transportation Research Board, 1840: 67–77.

Lee, C., Hellinga, B., Saccomanno, F., 2004. Evaluation of variable speed limits to improve traffic safety. Transport. Res. Part C: Emerg. Technol 14(3), 213-228.

Li, Z., Liu, P., Wei Wang, Chengcheng Xu, 2014. Development of a control strategy of variable speed limits to reduce rear-end collision risks near freeway recurrent bottlenecks, J. Central South Univ., 21(6), 2526-2538.

Li, Z., Liu, P., Xu, C., Duan, H., and Wang, W., 2016. Optimal mainline variable speed limit control to improve safety on large-scale freeway segments. Computer-Aided Civil and Infrastructure Engineering 31, 366-380.

Li, Z., Liu, P., Xu, C., Duan, H., and Wang, W., 2017. Reinforcement learning-based variable speed limit control strategy to reduce traffic congestion at freeway recurrent bottlenecks. IEEE Transactions on Intelligent Transportation Systems 18(11).

Lighthill, M., and G. Whitham. On kinematic waves II. A theory of traffic flow on long crowded roads. Proceedings Royal Society of London, Part A, Vol. 229, No. 1178, 1955. http://dx.doi.org/10.1098/rspa/1955.0089.

Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D., 2016. Continuous Control with Deep Reinforcement Learning. Google DeepMind, conference paper at ICLR, London, UK, 2016.

Lin, L., Wang, Q., Sadek, A.W., 2015. A novel variable selection method based on frequent pattern tree for real-time traffic accident risk prediction. Transportation Research Part C: Emerging Technologies 55, 444-459. http://doi.org/10.1016/j.trc.2015.03.015.

Liu, M., Chen, Y., 2017. Predicting real-time crash risk for urban expressways in China. Mathematical Problems in Engineering, Article ID: 6263726.

Lu, X.Y., Kim, Z., Cao, M., Varaiya, P. and Horowitz, R., 2010. Deliver a Set of Tools for Resolving Bad Inductive Loops and Correcting Bad Data. California PATH Research Report.

Lu, X.Y., Varaiya, P., Horowitz, R., Su, D., and Shladover, S., 2011. Novel freeway traffic control with variable speed limit and coordinated ramp metering. Transp. Res. Rec., 2229, 55-65.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M., 2013. Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D., 2015. Human-level control through deep reinforcement learning. Nature 518, 529-533.

Oh, C., Oh, J., Ritchie, S., Chang, M., 2001. Real-time estimation of freeway accident likelihood. In: Presented at the 80th Annual Meeting of the Transportation Research Board, Washington, D.C.

Pande, A., Abdel-Aty, M., 2005. A Freeway Safety Strategy for Advanced Proactive Traffic Management. Journal of Intelligent Transportation Systems, Vol. 9, No. 3, pp. 145–158.

Park, H., Haghani, A., Samuel, A., Knodler, M.A., 2018. Real-time prediction and avoidance of secondary crashes under unexpected traffic congestion. Accident Analysis and Prevention 112, 39-49.

PTV-AG, VISSIM 5.10 User Manual. 2008.

Rezaee, K., Abdulhai, B., Abdelgawad, H., 2012. Application of Reinforcement Learning with Continuous State Space to Ramp Metering in Real-World Conditions. Proc. 15th IEEE Int. Conf. Intell. Transp. Syst. (ITSC), Sep 2012, pp. 1590-1595.

Roy, A., Hossain, M., and Muromachi, Y., 2018a. Enhancing the prediction performance of real-time crash prediction model: A Cell Transmission-Dynamic Bayesian Network approach, 97th Annual Meeting of Transportation Research Board, Washington D.C.

Roy, A., Hossain, M., and Muromachi, Y., 2018b. Development of Robust Real-Time Crash Prediction Models Using Bayesian Networks, Asian Transportation Studies, Volume 5, Issue 2, 349-361.

Sun, J., Sun, J., 2015. A Dynamic Bayesian Network model for real-time crash prediction using traffic speed conditions data. Transportation Research Part C: Emerging Technologies 54, 176-186. http://doi.org/10.1016/j.trc.2015.03.006.

Sutton, R.,S., and Barto, A.,G. 1998. Introduction to Reinforcement Learning. MIT Press.

Thrun, S., Shwartz, A., Lee, C.B., 1993. Issues in Using Function Approximation for Reinforcement Learning. Proceedings of the 1993 Connectionist Models Summer School, Lawrence Erlbaum Publisher, Hillsdale, NJ.

Wang, Z., Schaul, T., Hessel, M., Hasselt, H.V., Lanctot, M., Freitas, N., 2016. Google DeepMind London UK. Dueling Network Architectures for Deep Reinforcement Learning, Proceedings of the 33rd International Conference on Machine Learning, JMLR: W & CP 48, New York, NY, USA.

Watkins, C. J. C. H., 1989. Learning from delayed rewards. PhD thesis, University of Cambridge, England.

Wu, Y., Abdel-Aty M., 2018. Developing an algorithm to assess the rear-end collision risk under fog conditions using real-time data, Transportation Research C 87, 11-25.

Yang, K., Wang, X., Quddus, M., Yu, R., 2018. Deep learning for real-time crash prediction on urban expressways, 97th Annual Meeting of Transportation Research Board, Washington D.C.

Yu, R., and Abdel-Aty, M., 2014. An optimal variable speed limits system to ameliorate traffic safety risk, Journal of Transportation Research: Part C 46, 235-246. DOI: 10.1016/j.trc.2014.05.016.

Yuan, J.H., Abdel-Aty, M., Wang, L., Lee, J.Y., Wang, X.S., Yu, R.J., 2018. Real-Time Crash Risk Analysis of Urban Arterials Incorporating Bluetooth, Weather, and Adaptive Signal Control Data. Accepted for presentation at 97th Annual Meeting of the Transportation Research Board, TRB No. 18-00590, Washington DC, January 2018.

Zhu, F., Ukkusuri, S.V., 2014. Accounting for dynamic speed limit control in a stochastic traffic environment: A reinforcement learning approach. Transportation Research Part C 2014, 41, 30-47.

CHAPTER 2

LITERATURE REVIEW

2.1 Macroscopic traffic flow model

The first beginnings for traffic flow descriptions on a highway are derived from observations by Greenshields, firstly shown to the public exactly 75 years ago (Proc., 13th Annual Meeting of the Highway Research Board, Dec. 1933). He carried out tests to measure traffic flow, traffic density and speed using photographic measurement methods for the first time. Greenshields postulated a linear relationship between speed and traffic density, as shown in **Figure 2.1** using the relation: flow = density * speed.

The linear speed–density relation converts into a parabolic relation between speed and traffic flow. Increasingly even the term "flow" was not known 75 years ago and Greenshields called that term "density-vehicles per hour" or density of the second kind. In this model some traffic flow characteristics are expressed well. It shows a maximal traffic flow with the related optimal traffic density. In the q-v-diagram exists two regimes, meaning it is possible to have two speeds at the same traffic flow. By this the traffic flow is classified in a stable and an unstable regime. Greenshields linear relation would be called a univariate model, because both regimes are calculated with the same formula. Early studies at traffic capacity of motorways had two different approaches. On the one side speed–traffic density relations were analyzed. Here a constant (free) speed was implied.

$$q = v_f * k \tag{2.1}$$

On the other side distance phenomenon at high traffic density were analyzed and as easiest approach a constant reaction time tr was implied, which brings you to the gross headway

$$l = l_0 + v * t_r \ \ with, k = 1/l \tag{2.2}$$

$$q = -\ l_0/t_r * (k - k_{max}) \ \ with, k_{max} = l/l_0 \tag{2.3}$$

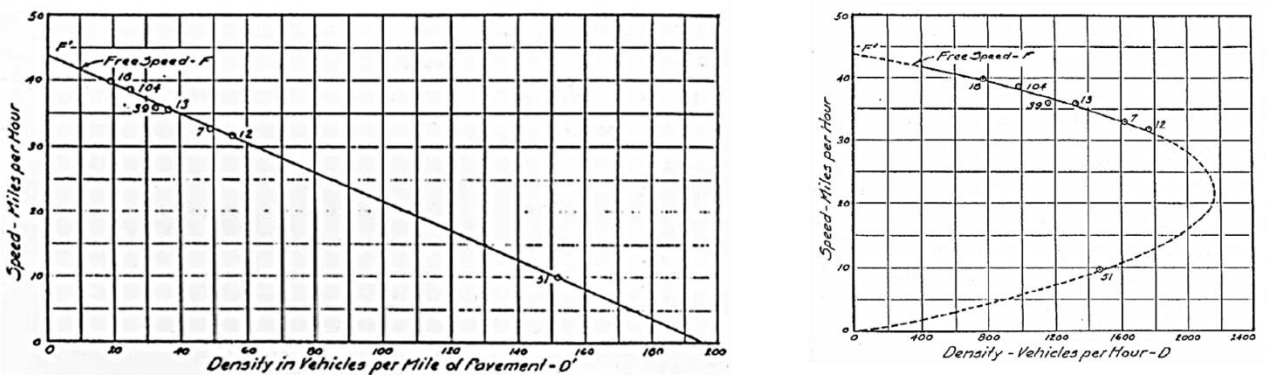


Figure 2. 1 Speed–density relation V (Greenshields 1934) and The first Fundamental Diagram as q-v diagram (Kuˈhne,R.D.,2011)

Also a linear relationship between q and k, but with a negative congestion broadening speed—l0/tr as proportionality constant. Summarizing both regimes creates a triangle function as traffic flow–traffic density relation. Lighthill and Whitham as well as Richards preached this triangle function as flow–density–curve and the use of the Kinematic wave theory on road traffic as instrument to combine both fields and to explain the dispersion of shock waves as reverted going congestion front (LWR theory). Also the q-k-relation established by Lighthill and Whitham has a parabolic curve progression and it's a one field model, too. The maximum stands for the expected road capacity of a motorway section. The insights of Greenshields inspired the development of two- and multiple-regime models in the aftermath.

2.1.1 Microscopic model and macroscopic model

There exists two fundamentally different approaches for traffic modelling. The microscopic approach seeks to reproduce the behavior of an individual vehicle, as its driver responds to its environment by adjusting its speed and lane. Microscopic models typically involve variables such as vehicle positions, speed and headway. On the other hand, the macroscopic approach ignores the dynamics of individual vehicles and instead attempts to replicate the aggregate response of a large number of cars. These models represent traffic as a compressible fluid in terms of flow, speed, density. Traffic engineering has benefitted immensely from macroscopic models. They are widely used in the design of freeway facilities, and they are present nearly in all model-based ramp metering designs. Because of its emphasis on quick and quantitative assessment, the tools for operational planning and procedures are based on macroscopic models that are easier to assemble,

calibrate and automate compared to their microscopic counterparts. Carrying out micro-simulations for all plausible actions is not practical in every situation.

2.1.2 First order kinematic wave theory for macroscopic model: The Lighthill-Witham-Richards (LWR) (1956)

Many approaches to the macroscopic, analytical descriptions of traffic on a single homogenous road have been developed over the years. Its theory and impact on DNL and dynamic traffic assignment (DTA) literature are summarized next, to quickly converge to the elements that are important for the development of the link model for our network model. The fundamentals of kinematic wave theory for traffic have been described by LWR-theory (Lighthill and Whitham, 1955) and (Richards, 1956) more than half a century ago. It is composed of a partial differential equation (PDE) that represents the conservation of vehicles, and a fundamental relation (**Figure 2.2**) between instantaneous flow and local density called the fundamental diagram (FD). The LWR-PDE is described in function of the density of vehicles along a road. Classical first order finite difference approaches (Godunov, 1959) for numerically solving PDEs are widely used in practice, e.g. CTM of (Daganzo, 1994) or (Lebacque, 1996). These methods operate on a computation grid that discretizes space in cells and time in slices to provide an approximate solution to the PDE. The solutions have interesting physical properties like storage of vehicles (in terms of vehicle densities) along a link, the formation of shock waves and the relations between traffic states along characteristic speeds such as observed on real roads. However, this method is also computationally demanding requiring a link to be cut into smaller cells and updating times to be limited by the Courant Friedrich and Lewy (CFL) conditions (Courant et al., 1967). These CFL-conditions state that the time discretization should not be longer than the shortest possible travel time of traffic over a cell. With short cells and high speeds, this time step is in realistic networks typically in the order of a few seconds.

This thesis is built on the seminal work of Newell on simplified theory of kinematic waves (Newell, 1993a, 1993b, 1993c). He simplified the FD to a triangular shape that proves to be efficient for numerical schemes in realistic networks (Yperman et al., 2005), and more importantly, he used cumulative vehicle number (CVN) functions to represent traffic. This intermediate abstraction (Moskowitz, 1965) by CVN functions (referred to as $(x,)$ at location x and time instant t) has also been of great importance for building exact solutions. It was formalized by (Claudel and Bayen,

2010a; Daganzo, 2005a; Friesz et al., 2013; Laval and Leclercq, 2013) using theory of variations and Hamilton Jacobi (HJ) PDEs for general FD. Solutions to HJ-PDEs are described in function of CVN from which travel times and densities are easily derived (Szeto and Lo, 2005). The numerical evaluation of the CVNs involves the convex Legendre-Frechel transformation of the FD and the use of Lax-Hopf formulas to translate the initial values and boundary conditions into a traffic state at any location or time within the homogenous link (Mazaré et al., 2011). This Lax-Hopf formula can also be used to construct a grid that leads to exact and efficient solutions using dynamic programming techniques while assuming a simplified triangular FD (Daganzo, 2005b). Such grids become increasingly difficult to define in large networks (Raadsen et al., 2014) so a simpler rectangular grid is assumed, with a triangular FD defined for each link. The grid is evaluated by an iterative procedure such that for each result only locally (at each grid point) the Lax-Hopf formula holds. This leads to faster algorithms, with substantially less grid points to be stored by the algorithm, while still computing a good approximate solution that is sufficient for practical applications.

2.1.3 Triangular fundamental diagram

The existence of a fundamental relation between traffic states has been empirically proven and many approximations have been proposed in literature (Greenberg, 1959; Greenshields et al., 1935; Underwood, 1961). Usually this relation is illustrated by plotting the instantaneous flow (veh/h) in function of the local density (veh/km). An example of observed aggregated traffic data (density and flow) on a highway lane and an approximate triangular shaped fundamental relation is shown in **Figure 2.2**. Free-flow states are represented by diamond markers and triangular markers represent congested traffic states. The hypocritical and hypercritical branch of the triangular function can be found by simple regression techniques on both sets of points. The slope of both linear approximations represent the characteristic wave speeds, in this case the free flow speed (γ) in uncongested traffic and the spillback speed of perturbations in congestion ($-\omega$). Also the maximum storage capacity or jam density of a link (j) and the maximum throughput (c) at the critical density (k_c) are defined on the graph.

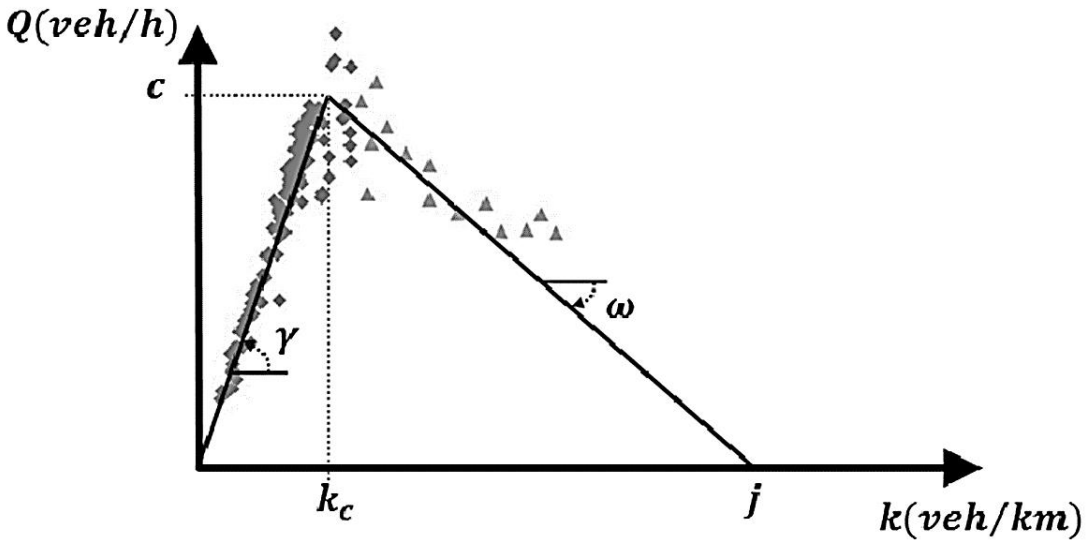


Figure 2. 2 Speed–density relation V (Greenshields 1934) and The first Fundamental Diagram as q-v diagram (Kuˈhne,R.D.,2011)

he triangular shape is often used because it leads to efficient numerical solution schemes (CTM, LTM). The Lax-Hopf equations (see discussion in the next section) are reduced significantly in this case because only two characteristic kinematic wave speeds (one forward and one backward) govern the prevailing traffic states (Claudel and Bayen, 2010b; Daganzo, 2005b; Han et al., 2012b; Jin, 2015). In principle the iterative procedure that is proposed in the next sections can be adopted to handle any concave FD.

2.1.4 An overview of existing macroscopic models

Real-time crash prediction model (RTCPM) is based on the hypothesis that the probability of a crash on a specific road section can be predicted for a very short time window using the instantaneous traffic flow data (Lee et al, 2003; Pande et al, 2005). Current RTCPMs are mostly based on traffic flow data from dual loop detectors, placed at fixed locations beneath the road surface where the inter-detector spacing varied substantially with an average of around 0.80 km and had high standard deviation (Abdel-Aty et al, 2012, Hossain and Muromachi, 2011; Shi et al, 2015). A study (Abdel-Aty et al, 2012) showed a range of standard deviation between 0.88 and 3.60 km for an AVI system. Another study (Shi et al, 2015) reported a minimum spacing of 0.16 km to a maximum of 5.90 km with a standard deviation ranging between 0.16 and 1.56 km. Hong and Fukuda, 2012 used cell transmission model (CTM) with ensemble Kalman filter (EnKF) in estimating the impact of various detector location configurations on estimation of travel speed and

concluded that sensors located at large distances from each other without location optimization lead to an overestimation of travel speed, whereas sensor numbers can be reduced if their locations are optimal to achieve a better estimate of travel speed. Many studies attempted to minimize the number of detectors (Bianco et al, 2001; 2006; 2007) whereas Morrison and Martonosi, 2014 examined whether there are sufficient conditions for optimal solutions for the detector locations. This non-uniformity in existing detector layouts along with high variability in detector spacing among expressways/freeways raise questions about the universality as well as transferability of such models. The doubt gets further bolstered by the studies (Abdel-Aty et al, 2012; Hossain and Muromachi, 2010) where the authors examined the performances of RTCPMs built with traffic data collected from six different combinations of detector layouts and found different results for each combination. Therefore, there is sufficient evidence postulating that variation in detector layout has an impact on the states of traffic flow variables.

The existing RTCPMs are developed depending on the detector layout that exists in the study area. Hence, variation in detector layout raises the issue of spatial transferability of the RTCPMs as altering the locations or installing new detectors to replicate the detector layout of the model is neither practical nor cost effective. Moreover, even for new roads, it may not always be plausible to follow a specific detector spacing. To address these issues, there has been an urge to devise a mechanism to simulate traffic sensor data of desired spacing from any detector layout that can be fed into an RTCPM.

Crashes occurring on freeways/expressways are considered to relate closely to preceding traffic states occurring before the crash, which are time-varying (Mihajlovic et al, 2001). Thus, it is essential to establish a single model that can address such time series data and the evolving process of traffic flow to predict crash risk in real time. Substantial effort has been put into improving the RTCPMs by employing sophisticated modeling methods such as artificial neural networks, Bayesian structural equations, Bayesian networks (BNs), dynamic Bayesian networks (DBNs), Bayesian classifier, support vector machines, genetic programming etc.(Hossain and Muromachi, 2001, Sun et al, 2015, 2016; Lee et al, 2006). Among these, the modeling architecture of a dynamic Bayesian network (DBN) conforms to this requirement (Mihajlovic et al, 2001; Sun et al, 2015; Roy et al, 2016).

This study employed a macroscopic dynamic freeway traffic model called CTM, introduced by Daganzo (1994, 1995), which is consistent with the hydrodynamic theory of traffic flow (Lighthill et al, 1955), to transform the traffic states obtained from non-uniform detector layouts into a pre-defined detector layout. The CTM was chosen for its analytical simplicity and ability to reproduce congestion propagation dynamics. Then, RTCPMs were built by simulating uniformly spaced detector data from fixed detector layouts and using BD and DBNs as the modeling method to distinguish between crash prone and normal traffic conditions.

2.1.5 Daganzo's (1996) kinematic wave theory based on the triangular fundamental diagram

The CTM proposed by Daganzo (1994, 1995) is based on the relationship between traffic flow (q) and density (k). If the relationship can be expressed in a triangular form as in **Figure 2.2,** then the Lighthill, Whitham, and Richards (LWR) equations for a single highway link can be approximated by a set of difference equations where current conditions (the state of the system) are updated with every clock interval as:

$$q = \min\{V_F k, q_{max}, w\,(k_j - k)\}, \qquad \text{for } 0 \le k \le k_j \tag{2.5}$$

Where V_F is the free flow speed,

q_{max} is the maximum flow (or capacity),

w is the back wave or wave speed, and

k_j is the jam density.

In the CTM, a road segment is divided into several homogeneous cells, i, whose length is equal to the free flow speed times one clock interval. The state of the system at instant t is then given by the number of vehicles contained in each cell, n_i (t). The following parameters are defined for each cell:

$N_i(t)$ = maximum number of vehicles that can be present in cell i at time t;

$Q_i(t) =$ maximum number of vehicles that can flow into cell i when the clock advances from t to $t + 1$.

The flow evolution in each cell can be expressed as:

$$n_i(t + 1) = n_i(t) + y_i(t) - y_{i+1}(t) \quad (2.6)$$

where $y_i(t)$ is the inflow to cell *i* in the time interval (*t*, *t* + 1), given by

$$y_i\ (t) = \min\ \{n_{i-1}(t),\ Q_i(t),\ \mathrm{d}[N_i(t) - n_i(t)]\} \quad (2.7)$$

where $d = w/v$

2.1.6.1 Velocity based cell transmission model

As mentioned earlier, the primary parameters of a CTM are defined by the left limb, the right limb and the apex of the triangle of the FD. The left limb decides the sending flow from an upstream cell to the downstream cell and the right limb decides how much flow it can receive depending on its current capacity. Generally, the FD for a link or segment remains the same throughout the simulation which is one of the positive sides of using a macroscopic traffic model. However, the model has to be in such a way so that the FD does not remain fixed throughout the simulation. The left and right limbs, or the sending and receiving functions needs to be adjusted to accommodate the VSL values (Mihn et al, 2015). So, the sending and receiving functions shown of the traditional CTM will be changed to the followings-

$$s_i(t) = \min\{V_{SL}(t).\, k_i(t), Q_{VSL}\}\ ; \quad \text{Where, } V_{SL}(t) \in [\ a_t\ ] \quad (2.8)$$

$$r_i(t) = \min\left\{\omega_i.\left(k_{i,jam} - k_i(t)\right), Q_{VSL}\right\} \quad (2.9)$$

Where, $s_i(t)$ is the sending flow at time t, $r_i(t)$ is the receiving flow, a_t is the action set which will be decided by the RL agent.

The flow in a cell i can be defined by the following receiving function:

$$q_i(t) = \min\{s_{i-1}(t), r_i(t)\} \quad (2.10)$$

The density evolution can be determined by the following equation:

$$k_i(t+1) = k_i(t) + \Delta T / L_i (q_{i-1}(t) - q_i(t)) \quad (2.11)$$

Here, ΔT is the simulation time step, which is equal to the time with which a vehicle passes a cell at free-flow speed. L_i is the cell length of cell i. The speed within each cell can be determined according to the current density, $k_i(t)$ and speed limit, V_{SL}:

$$v_i(t) = \begin{cases} \min\{V_{SL}(t-1), V_F\} & if\ k_i(t) \leq k_{VSL} \\ \left(k_{i,jam}(t) - k_i(t).\frac{w_i}{k_i(t)}\right) & if\ k_i(t) > k_{VSL} \end{cases} \quad (2.12)$$

Where, d_{VSL} is the density associated with the flow Q_{VSL} (Figure 2.3.) under the speed limit V_{SL}.

To summarize, in order to build the modified CTM model, the basic parameters of an FD which are free-flow speed, back wave speed, jam density and the maximum flow will be estimated from the traffic flow data from the loop detectors and these parameters will remain fixed. Then, similar as the traditional CTM, the simulation time step will be decided to get the cell length according to the formula cell length = free-flow speed times time step. This is to ensure that the vehicles don't cross multiple cells at a time step. So, after calculating the free-flow speed and deciding a suitable cell length, the time step is calculated. Next, unlike the traditional FD, the modified FD (Figure 2.3.) whose left limb can be moved lower depending on the VSL value chosen, will be constructed with a sending and receiving function as shown in the equation 2.8 and 2.9 . The sending function will choose either the maximum flow under current VSL value, Q_{VSL} or the actual flow under current VSL value ($V_{SL}(t).k_i(t)$) whichever is the minimum. At the same time, the receiving function will follow equation 2.8, and receive the minimum of the maximum flow under current VSL value, Q_{VSL} and the back-wave speed calculated from the jam density and current density, $\omega_i.\left(k_{i,jam} - k_i(t)\right)$. According to Daganzo's original CTM theory, the sending flow of upstream cell $(i-1)$ is the receiving flow of downstream cell i. Hence, the flow entering cell i will follow the minimum of the sending flow of cell $(i-1)$ and receiving flow of cell i, according to equation 2.10. As for the speed evolution in a cell, it will depend on, if the current density falls to the left or right hand side of the density under current VSL value. If it falls on the left region,

the speed will be dominated by either the V_F or VSL value; on the other hand, if it falls under the right-hand side, the speed will be decided by the jam density and the back-wave speed.

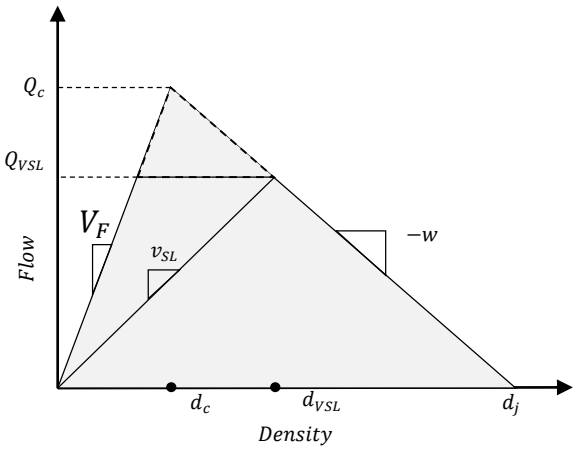


Figure 2. 3 The modified CTM with VSL control

2.2 Real-time crash prediction models

The concept of real-time proactive road safety management in general comprises of detection of hazardous traffic formation (which is known as real-time crash prediction) and subsequent introduction of interventions to reduce the chance of a crash occurrence in real-time. Sometimes, the data that have been collected for model construction are also utilized for knowledge discovery. This holds paramount importance as knowing the crash mechanism can help in building more accurate models as well as designing effective countermeasures. In other words, the RTCP model should be able to answer – 'how can an implementable real-time crash prediction model be built to ensure timely detection of hazardous traffic formation?'.

In order to choose the appropriate variables for model building, it is important to know which variables have higher capacity to distinguish between a hazardous and normal traffic condition. This has created the necessity to perform an in-depth study on crash mechanism using high resolution real-time traffic sensor data that will provide insight into crash phenomena and help to identify the most significant variables for model building.

It is quite common for a detector to go out of order due to natural wear and tear. For this, alternative detector spacing should also be available to ensure unhindered monitoring of the area of interest. This can be achieved by carefully choosing a study area that is densely packed with detectors having relatively uniform inter-detector spacing. This will enable in identifying the optimum

detector layout plan. It should also be noted that the study area must have a large enough crash sample to give the models acceptability.

A highly flexible modeling method should be chosen that can perform these activities without needing to re-build or re-calibrate itself from the scratch: i) update itself in short time interval and with partially available data, ii) add new variables if and when needed, iii) make predictions with partially missing data, and iv) possess capability to handle highly correlated data.

Crashes occurring on freeways/expressways are considered to relate closely to previous traffic conditions occurred before the crash, which are time-varying. Meanwhile, most studies use volume/occupancy/speed parameters data extracted from sampled floating cars or road side detectors. It has been indicated that for different combinations of upstream and downstream traffic states, the crash involvement rates and crash risk ratios (ratio of crash cases and non-crash cases) are inconsistent. Considering that a crash can be induced by the disturbance of traffic flow before the crash occurs, time series traffic data consisting of several time intervals should be used to illustrate the dynamic process of traffic flow before crash occurrence. Thus, it inessential to establish a single model that can address such time series data and the evolving process of traffic flow. Therefore, in this study, a dynamic Bayesian network (DBN) model of time sequence traffic data has been proposed to investigate the relationship between crash occurrence and dynamic traffic flow data with one-minute-interval of six months on one of the busiest Tokyo Metropolitan Expressway (March to August, 2014) provided by the Tokyo Metropolitan Expressway Company Limited.

2.2.1 A review of existing RTCPMs

Since last few decades, there had been several studies targeting the improvement of RTCPMs. Different researchers have tried to construct RTCPMs in different ways depending on the goals, study locations, base and derivative of traffic parameters, road geometry, weather condition, type of crash, type of vehicles etc. A brief review is given below.

Oh et al. (2001) developed the first real-time crash prediction model where he separated traffic dynamics into two categories – disruptive and normal, and assessed the likelihood of future traffic flow data falling into either of these two categories. Normal condition was specifically defined as

a 5 minute period occurring at 30 minutes prior to the crash and the disrupted condition was defined as the 5 minute time period just before the crash. They used 52 crash data and corresponding traffic data from loop detectors and applied nonparametric Bayesian approach (Oh et al. 2005a) to identify the real-time crash likelihood. In the later study (Oh et al. 2005b) they applied Probabilistic Neural Network (PNN). They employed t-test on the mean and deviation of three variables – occupancy, flow and speed, to identify the crash indicator. However, there was no suggestion explaining if they have tested the data for normality as t-test is applicable only with the assumption that the data follow normal distribution. Some later studies (e.g., Luo et al., 2006) found that the standard deviation of speed on freeways does not follow normal distribution. Another study (Ulfarssonet al., 2005) indicated that mean and standard deviation of speed are correlated. Oh et al., in their first two studies (2001, 2005a) identified standard deviation of speed to be the most significant variable. But in a later study (Oh et al., 2005b) they selected standard deviation of speed as well as the average occupancy to be the predictors. Later they evaluated their newly built model by randomly selecting 30 crash data from their sample and testing their outcome and repeating the process for 30 times. They used two threshold values and the prediction success was respectively 38.2% and 44.9%.

The motivation for Golob et al. (2002) to develop a real-time crash prediction model was to understand the complex relationship between traffic flow and traffic crashes rather than to develop a proactive highway safety system. The study by Lee et al. (2002) was the first of its kind to point out the potential of real-time crash prediction to be applied as a proactive road safety management system, i.e., anticipate future crashes and apply counter measures to prevent it from taking place. They introduced a new term called 'crash precursors', which was defined as traffic conditions that exist before the occurrence of a crash. Their second study (Lee et al., 2003) basically reduced the number of assumptions they made in the first study to make it more acceptable. In their latest model they selected speed variations along a lane, traffic queue and traffic density at given road geometry, weather condition and time of the day as predictors and applied aggregated first order log-linear model to predict crash. The study used 234 crash data. Data for the normal traffic condition traffic were collected by acquiring the traffic data of 2 week days with clear weather condition when no crash happened. The developed model was not validated with another dataset and the prediction success was represented with the overall model fit, statistical significance of the coefficients and the consistency of the coefficients with the order of levels of crash precursors.

The study by Abdel-Aty and Pande (2004) and Abdel-Aty and Abdalla (2004) are acceptable due to relatively larger sample size, meticulous considerations for modeling, choice of predictors and acceptable model validation method. In the first study (Abdel-Aty and Pande, 2004), they used a sample size of 148 crashes, of which, 100 were used to generate the model and the rest 48 were used for validation. They used the concept of logistic regression and odds ratio to develop a new index called 'Hazard ratio', which essentially represents the factor with which the risk of observing a crash in the vicinity of the 'station of the crash' will increase with unit increase in the corresponding risk factor (here, the predictors of crash). Lastly, they used Probabilistic Neural Network (PNN) to distinguish between crash and non crash situation. They found the coefficient of variation in the speed obtained from the station near the crash and two stations immediately preceding in the upstream direction prior to crash to be the most suitable predictors. Although their study produced by far the best results to predict crashes, the overall classification, i.e., for both crash and non-crash situations together, was poor (62%). In a later study (Abdel-Aty and Abdalla, 2004) they used Generalized Estimating Equation method where they included road geometry as variables as well. The study found that high variability in speed for a period of 15 minutes for a specific location increases the likelihood of crash and also, low variability in volume over 15 minutes at a given location increases the crash likelihood in the downstream.

Alongside these successful studies, there are two more mentionable studies that could not find any relationship between crashes and their prior traffic conditions. The first one was conducted by Kockelmanet al. (2004) where they used variation of speed as well as average speed both along the lane and for different defined sections to predict future crashes. They used several conventional statistical distributions to calculate the likelihood of crash and concluded that speed or its measures of dispersion were not correlated according to their data. They further used a very low data aggregation (30 seconds) but came down to the same conclusion. One of the latest additions in real-time crash prediction has been the study by Luo and Garber (2006) which conducted a comprehensive critical review of the major past studies concluding that the previous studies are weak in one or more of the three areas - unrealistic data requirements, inconsiderate to the interaction among variables and selecting predictors without validating them with pattern recognition techniques. Therefore, Luo and Garber (2006) invested substantial amount of labor in identifying crash patterns with three different methods – K means clustering, Naive Bayes method and Discriminant Analysis. They also studied the joint effects of two or more traffic variables to

identify traffic patterns leading to crash. They considered daytime and nighttime crashes separately, too. However, the outcome of the study was inconclusive.

Road crash is a complex phenomenon and it is yet to be known how to classify a traffic condition to be hazardous or normal by observing the traffic flow variables. Another major concern with these models is the possibility of calibration in future as it can be expected that driving behavior varies depending on demography and each model associated with specific detectors needs to be calibrated separately (Kuchangi, 2006). This may even require addition or truncation of some demography related variables. An actionable model must use variables which are universal (e.g., traffic flow) to most of the types of detectors and the model must have capability to update itself with time as, quite understandably, the initial model will be built with very few samples. So far the concentration in previous studies has been on how to improve the prediction success and for that, the researchers used complex forms of traffic flow variables and different statistical models (mainly). In this study, we attempt to address these issues by developing a real-time crash prediction model with readily available variables and use a real-time risk assessment method – the Bayesian Network. The idea of developing models for predicting crash in real-time for expressways from high resolution sensor data is now more than a decade old.

2.2.2 Shortcomings of the existing models

In spite of several improvements which have been made for an efficient RTCPM, there are still some lacking which needs to be addressed.

- Location of detector

The performance of RTCPM largely depends on the quality of data which are collected from detectors. Hence, it is important to consider type, spacing and layout of detectors. Most studies uses traffic data from fixed loop detectors (Abdel-Aty et al., 2004, 2005; Abdel-Aty and Pande, 2005) which are installed on the road network usually without any preliminary analysis about which would be the optimum location to install those. Roshandel et al. (2015) pointed this out as their location is fixed on the road and their distance from crash locations cannot be controlled. Several studies have been conducted in order to find the optimum detector location for RTCPM. For example, Hossain and Muromachi (2011) showed that if a detector pair is 750 meters apart

with the upstream detector more closely spaced to the location of crash occurrence, the model can predict with about 65% accuracy. The issue of optimum detector location is still being studied.

- Variable Space

The major traffic variable used for RTCPMs are flow, speed and occupancy, and/ or their derivatives (Lee et al., 2003b, 2006a; Pande and Abdel-Aty, 2006b). Several other studies have adopted weather conditions, visibility and road geometry as variable to construct RTCPMs too (Pande and Abdel-Aty, 2006a; Lee et al., 2003b), while other researchers also considered congestion index as a variable congestion index (Dias et al., 2009). Some studies employed engineering judgment to choose the variables, others applied statistical methods (testing the significance by developing logistic regression models with one variable at a time) as a solution to the problem (Abdel-Aty et al., 2004,2005; Abdel-Aty and Pande, 2005; Pande and Abdel-Aty, 2005, 2006a, 2007; Zheng et al., 2010). Choosing an appropriate variable space is important because the inappropriate variables might result in a huge variable space within a small data set. The variable space might also suffer from the correlation problems as traffic variables are closely related. Moreover, in situations, where a variable is not available, another variables should be able to capture the traffic situation properly. So, choosing variables which are influential to RTCPMs are important.

- Modeling Method

With the advancement of ITS and availability of high density traffic data, numerous modeling methods have been adopted over the decades. These methods can broadly divided into two categories- statistical (Abdel-Aty et al., 2004, 2005; Abdel-Aty and Pande, 2005; Pande and Abdel-Aty, 2007; Lee et al.,2006a; Zheng et al., 2010) and machine learning based such as neural networks (Abdel-Aty et al., 2008a; Abdel-Aty and Pande, 2005; Pande and Abdel-Aty,2006a; Oh et al., 2005a,b), fuzzy logic (Oh et al., 2006) and classification trees (Pande and Abdel-Aty, 2006b) etc. As mentioned earlier, traffic variables are highly related, which could be a problem while modeling with statistical approaches like regression analysis due to co-linearity problem. Machine learning can overcome this problem. Although, ML-based modeling requires a vast amount of database and huge computational time.

2.3 Decision support system for Intervention: A comparative analysis of existing traffic intervention methods

In the field of intelligent transportation system (ITS) real-time crash prediction and interventions have been one of the major concerns. With the increase in application of advance ITS technologies it is now possible to get access to the high quality real-time traffic data which calls for proactive traffic safety measure to reduce the crash risks on a road network. To introduce a proactive safety measure, it is important to understand the crash mechanism under different traffic conditions to identify the contributors to crash occurrence. Given that the traffic parameters change over time, the crash likelihood must be predicted in real-time. Numerous real-time crash prediction models (RTCPMs) have been developed over the years to predict crash risks given the real-time traffic data. A number of cutting edge modelling methods, such as, various kinds of Neural Networks (Liu and Chen, 2017; Park et al., 2018), Support Vector Machine (Katrakazas et al., 2017), Bayesian Network (Hossain and Muromachi, 2013), Dynamic Bayesian Network (Roy et al., 2018), Deep Neural Network (Yang et al., 2018), etc.; they are to some extent transferrable (Park et al., 2018); and their accuracy in predicting crash is commendable (Yang et al., 2018 – 96%, Wu et al., 2018 – 87%) with low false alarm (Yang et al., 2018 – 10%, Wang et al., 2017a – 2.7%).

For real-time intervention, a certain threshold of real-time crash likelihood can be used as a measurement to formulate a proactive control strategy (Lee et al., 2004). Several studies have concluded that abrupt change in speed plays a vital role for crash occurrence, such as increase in speed limit can cause higher fatal crashes (Kloeden et al., (2001); Ossiander et al., 2002). Generally, the low speed limit reduces the dissemination of speeds and hence reduces the possibility of vehicle collisions. Additionally, lower speed limits decrease the chance of high crash. Variable speed limit (VSL) control has been proven as a well-established and innovative technique to reduce crash risks in mainline freeways (Lee et al., 2004; Li et al., 2016, 2017; Abdel-Aty et al., 2006, 2007, 2008). The core idea of VSL is- it is an ITS device which is triggered in order to improve the traffic operation by changing the allowable speed limit of a road segment once a certain threshold of control factor is exceeded. Previous studies have suggested a number of strategies to apply VSL control. For example, Lee et al. (2004) considered four start-up thresholds of crash potential (0, 2.5, 12, 18) depending on the road geometry, while a set of durations of intervention (2, 5, 10, 15 min) were chosen and the magnitude of VSL was selected from- a set of seven fixed speed limits,

or, the average speed of downstream detector, or, average of up and downstream detectors of the current location (transition speed). Their study showed that the overall crash potential and travel time (in case of threshold 12) was reduced by 5-17% and 0.4% respectively when the transition speed was selected as the magnitude of the VSL control. In another study (Li et al., 2016) considered a start-up threshold of 0.2 as a control strategy to activate the VSL in a large scale freeway segment and found that the crash risk was reduced by 22.62% and injury severity by 14.67%. In a later study (Li et al., 2014), the authors employed VSL close to freeway recurrent bottlenecks to reduce rear-end collision risks where the control strategy included start-up threshold (25%), target speed limit (35mile/h) and speed change rate (10mile/h/30sec). Few studies (Carlson et al., 2011; Lu et al., 2011) applied VSL at the upstream of the bottleneck area in order to control the outflow of the VSL section. This way the capacity drop at the bottleneck can be avoided and the bottleneck capacity can be retrieved. However, by reducing the capacity at the VSL section, a congestion might form at the bottleneck, propagating further upstream which in turn resulting in increased risks of rear-end collisions in the upstream area. Hence, the VSL control strategy that reduces delay, might increase rear-end crash risks at the VSL section of a freeway recurrent bottleneck. Abdel-Aty et al. (2008) used VSL strategies to reduce rear-end and lane-changing crash risks in a network with huge speed differences between up and downstream. They concluded that VSL is effective in reducing crash risk during uncongested condition only. From these studies it's evident that the strategy of VSL application in freeways to reduce crash risk is yet to be well established.

Besides setting the VSL strategies, the optimization of the strategy is also an eminent issue while designing the real-time intervention. A popular optimization method is the automated control strategies by PARAMICS (Lee et al., 20046, Abdel-Aty et al., 2005, 2006, 2008) which has been vastly used due to its scalability and proven application on freeways. It has been employed to optimize VSL strategies of split models for multi-vehicle crashes during both congested and uncongested conditions (Lee et al., 200, Abdel-Aty et al., 2005, 2006), for rear-end and lane changing crash risk reduction in homogenous speed zones while inducing 60, 80, 90% loading of traffic volumes to the model (Abdel-Aty et al., 2008). In contrast to the traditional optimization method, Abdulhai et al. (2003) introduced an artificial intelligence based semi-supervised machine learning algorithm called reinforcement learning (RL) to find the optimal control of traffic signal in a heavily congested condition. In RL, an agent reacts with the environment through several trial

and error to optimize the total reward by choosing a state-action pair for every time step (Watkins, 1989; Sutton and Barto, 1998; Hasselt, 2011). RL is gaining popularity in traffic control problems since it does not require a model structure per se which reduces its dependency on the previous information of the system to be controlled enabling it to tackle non-recurrent complex traffic patterns. A Q-learning based multi-agent RL, which is the most commonly used RL algorithm, was used in a study for motorway ramp-metering control with queuing consideration (Davarynejad et al., 2011) and traffic control performance evaluation (Rezaee et al., 2012). Another study (El-Tantawy et al., 2010) a multi-agent RL was used in conjunction with the game theory to alleviate traffic gridlock. An R-Markov average reward technique (R-MART) based RL was used to optimize VSL control for reducing travel time and vehicle emission (Zu et al., 2014). A study by Li et al. (2014, 2016) applied genetic algorithm for VSL optimization. However, in later study Li et al. (2017) used Q-learning based RL at a freeway recurrent bottleneck to optimize VSL control. The RL based solution was applied to a metro station in Shanghai and it proved its efficacy in minimizing safety risk and simultaneously alleviating passenger congestion at certain stations. Unsignalized intersections are considered as one of the most crash prone locations on a road network and therefore, training autonomous vehicles to drive at such locations is a difficult challenge. Isele et al. (2018) learned policies and active sensing behaviors employing RL that exceeded the capabilities of commonly-used heuristic approaches in several categories, such as, task completion time, goal success rate and ability to generalize the problem.

In spite of the promising success records of application of Q-learning for VSL control optimization compared to traditional feedback based VSL control method, there are few issues such as adaptability of non-discretized (continuous) traffic states, location of VSL control sections and the reliability of the VSL models in terms of crash risk reduction in real-time is still under vigilance. Another new RL method namely dueling DQN (Mihn et al., 2013, 2015; Hasselt et al., 2016) which can accommodate continuous traffic states and can more quickly identify the correct action during policy evaluation as redundant or similar actions are added to the learning problem. has been used in Hence, to address these issues and inspired by the recent success of Q-learning and dueling DQN in designing intelligent agents to solve transportation problems in real-time, two reinforcement learning based intelligent frameworks for real-time proactive variable speed limit control are being proposed in this manuscript.

2.4 Chapter References

Abbess, C., Jarett, D. and Wright, C.C. (1981). Accidents at Blackspots: Estimating the Effectiveness of Remedial Treatment, With Special Reference to the Regression-to-Mean Effect. Traffic Engineering and Control, Vol. 22, No. 10, pp. 535-542.

Abdel-Aty M. and Abdalla F. (2004). Linking Roadway Geometrics and Real-Time Traffic Characteristics to Model Daytime Freeway Crashes Using Generalized Extreme Equations for Correlated Data. Journal of the Transportation Research Board, No. 1897, pp. 106-115.

Abdel-Aty M. and Pande A. (2005) Identifying crash propensity using specific traffic speed conditions, Journal of Safety Research, Vol. 36 No.1, pp. 97-108. Abdel-Aty, M. Uddin, N. And Pande, A. (2005). Split Models for Predicting Multivehicle Crashes During High-Speed and Low-Speed Operating Conditions on Freeways. Transportation Research Record: Journal of the Transportation Research Board, No. 1908, TRB, National Research Council, Washington, D. C., pp. 51-58.

Abdel-Aty M., Gayah V., 2010. Real-time crash risk reduction on freeways using coordinated and uncoordinated ramp metering approaches. Transportation Engineering 136, 410-423.

Abdel-Aty, M. and Pande, A. (2006). ATMS Implementation System for Identifying Traffic Conditions Leading to Potential Crashes. IEEE Transactions on Intelligent Transportation Systems, Vol. 7, No. 1. pp. pp. 78-91. Abdel-Aty, M., J. Dilmore, and L. Hsia. (2006a) Applying Variable Speed Limits and the Potential for Crash Migration. In Transportation Research Record: Journal of the Transportation Research Board, No. 1953, Transportation Research Board of National Academics, Washington, D.C.,. 2006, pp. 21-30.

Abdel-Aty, M., and Wang, L., 2017. Implementation of variable speed limits to improve safety of congested expressway weaving segments in microsimulation, Transportation Research Procedia 27, 577-584.

Abdel-Aty, M., Cunningham, R.J., Gayah, V.V., Hsia, L., 2008. Dynamic variable speed limit strategies for real-time crash risk reduction on freeways. In: Transportation Research Record:

Journal of the Transportation Research Board, No. 2078, Transportation Research Board of the National Academies, Washington, D.C. pp. 108-116.

Abdel-Aty, M., Dilmore, J. and Dhindsa, A. (2006b). Evaluation of Variable Speed Limits for Realtime Freeway Safety Improvement. Journal of Accident Analysis and Prevention. Vol. 38. No. 2 pp. 335-345.

Abdel-Aty, M., Dilmore, M.J., Gayah, V.V., 2007a. Considering various ALINEA ramp metering strategies for crash risk mitigation on freeways under congested regime. Journal of Transportation Research, Part C 15(2), 113-134.

Abdel-Aty, M., Dilmore, M.J., Hsia, L., 2006b. Applying variable speed limits and the potential for Crash Migration. In: Transportation Research Record: Journal of the Transportation Research Board, No. 1953, Transportation Research Board of the National Academies, Washington, D.C., pp. 21-30.

Abdel-Aty, M., H. M. Hassan, M. Ahmed, and A. S. Al-Ghamdi. Real-time Prediction of Visibility Related Crashes. Transportation Research Part C, Vol. 24, 2012, pp. 288–298.

Abdel-Aty, M., Pande, A., Das, A. and Knibbe, W. J. (2008a). Assessing Safety on Dutch Freeways with Data from Infrastructural-based Intelligent Transportation Systems. Transportation Research Record: Journal of the Transportation Research Board, No. 2083, TRB, National Research Council, Washington, D. C., pp. 153-161.

Abdel-Aty, M., Pande, A., Lee, C., 2007b, Gayah, V., Santos, C. D., 2007b. Crash risk assessment using intelligent transportation systems data and real-time intervention strategies to improve safety on freeways. Journal of Intelligent Transportation Systems 11(3), 107-120.

Abdel-Aty, M., Pemmanaboina, R. And Hsia, L. (2006c). Assessing Crash Occurrence on Urban Freeways by Applying a System of Interrelated Equations. Transportation Research Record: Journal of the Transportation Research Board, No. 1953, TRB, National Research Council, Washington, D. 47 C., pp. 1-9.

Abdel-Aty, M., Uddin, N., Pande, A., Abdalla, M. F. and Hsia, L. (2004) Predicting Freeway Crashes from Loop Detector Data by Matched Case-control Logistic Regression. Transportation

Research Record: Journal of the Transportation Research Board, No. 1897, TRB, National Research Council, Washington, D. C., pp. 88-95. 46

Abdulhai, B., Pringle, R., Karakoulas, G.J., 2003. Reinforcement learning for true adaptive traffic signal control. Journal of Transportation Engineering, ASCE 129, 278-285. DOI: 10.1061/(ASCE)0733-947X(2003)129:3(278).

Ahmed, M., Abdel-Aty, M., Yu, R., 2012. A Bayesian updating approach for real-time safety evaluation using AVI data. Transp. Res. Rec. 2280, 60–67.

Balke, K. et al. (2005). Dynamic Traffic Flow Modeling For Incident Detection and Short-Term Congestion Prediction. Year 1 Progress Report Prepared For the Texas Department of Transportation Research and Technology Implementation Office, Austin, Texas, USA. Bliss, T. (2004). Implementing the Recommendations of the World Report on Road Traffic Injury Prevention, Transport Note No. TN – 1, World Bank, Washington DC, USA.

Bianco, L., G. Confessore, and M. Gentili. Combinatorial Aspects of the Sensor Location Problem. Annals of Operations Research, Vol. 144, No. 1, 2006, pp. 201–234.

Bianco, L., G. Confessore, and P. Reverberi. A Network Based Model for Traffic Sensor Location with Implications on O/D matrix Estimates. Transportation Science, Vol. 35, No. 1, 2001, pp. 50–60.

Bianco, L., R. Cerulli, and M. Gentili. New Resolution Approaches for the Sensor Location Problem. Presented at Tristan VI Symposium, Phuket Island, Thailand, 2007.

Bliss, T. and Breen J. (2009). Country Guidelines for the Conduct of Road Safety Management Capacity Reviews and the Specification of Lead Agency Reforms, Investment Strategies and Safe System Projects. The World Bank Global Road Safety Facility, Washington DC, USA.

Bossche, F. V., Wets, G., and Brijs, T. (2006). Predicting Crashes using Calendar Data. 85th TRB Annual Meeting, Washington DC., USA. Brodsky, H., Hakkert, A.S., (1988). Risk of a Road Accident in Rainy Weather. Journal of Accident Analysis and Prevention. Vol. 20, No. 2, pp. 161–176.

Caliendo, C., Guida, M. and Parisi, A. (2007). A Crash-prediction Model for Multilane Roads. Accident Analysis and Prevention, Vol. 39. pp. 657-670. Cambridge Advanced Learner's Dictionary. (2010).

Carlson, R.C., Papamichail, I., Papageogiou, M., 2011. Local feedback based mainstream traffic flow control on motorways using variable speed limits. IEEE Trans. Intell. Transp. Syst., 12(4), 1261-1276.

Chow, A. H. F., G. Gomes, A. A. Kurzhanskiy, and P. Varaiya. AURORA RNM – A Macroscopic Simulation Tool for Arterial Traffic Modeling and Control. PATH Technical Note, Institute of Transportation Studies, University of California, Berkeley. 2009.

Chow, A. H. F., G. Gomes, A. A. Kurzhanskiy, and P. Varaiya. AURORA RNM – A Macroscopic Simulation Tool for Arterial Traffic Modeling and Control. PATH Technical Note, Institute of Transportation Studies, University of California, Berkeley. 2009.

Coifman, B. Using Dual Loop Speed Traps to Identify Detector Errors, TRR, 2014.

Daganzo, C. F. The Cell Transmission Model, Part II: Network Traffic. Transportation Research Part B, Vol. 29, 1995, pp. 79–93.

Daganzo, C. F. The Cell Transmission Model: A Dynamic Representation of Highway Traffic Consistent With the Hydrodynamic Theory. Transportation Research Part B, Vol. 28, 1994, pp. 269–287.

Davarynejad, M., Hegyi, A., Vrancken, J., van den Berg, J., 2011. Motorway Ramp-Metering Control with Queuing Consideration using Q-learning, Proc. 14th IEEE Int. Conf. Intell. Transp. Syst., Washington, DC, USA, pp. 1652-1658.

Delen, D., Sharda, R., Besson, M., (2006). Identifying Significant Predictors of Injury Severity in Traffic Accidents using a Series of Artificial Neural Networks. Journal of Accident Analysis and Prevention. Vol. 38, 434–444.

Dervisoglu, G., G. Gomes, J. Kwon, R. Horowitz, and P. Varaiya. Automatic Calibration of the Fundamental Diagram and Empirical Observations on Capacity. Presented at the Transportation Research Board 88th Annual Meeting, 2009.

El-Tantawy S., Abdulhai, B., 2010. Towards multi-agent reinforcement learning for integrated network of optimal traffic controllers. Transp. Letters 2(2), 89-110.

Garber, N. and Luo, L. (2006). Identification of Traffic Patterns Leading to Crashes. Research Report Prepared For the Center of ITS Implementation Research. U.S. DOT University Transportation Center, Virginia, USA. Research Report No. UVACTS-15-0-101.

Garber, N. J. and Wu, L. (2001). Stochastic Models Relating Crash Probabilistic with Geometric and Corresponding Traffic Characteristics Data. Project Report Prepared For the National ITS Implementation Research Center, USA. Report No. UVACTS-15-5-54.

Golob, T. and Recker, W. (2001). Relationships Among Urban Freeway Accidents, Traffic Flow, Weather and Lighting Conditions. California PATH Working Paper UCB-ITS-PWP-2001-19, Institute of Transportation Studies. Berkeley, University of California.

Golob, T. F., Recker, W. W., and Alvarez, V. M. (2004). Freeway Safety as a Function of Traffic Flow. Journal of Accident Analysis and Prevention, Vol. 36, No. 6, 933-946.

Golob, T., Recker, W., Pavlis, Y., 2008. Probabilistic models of freeway safety performance using traffic flow data as predictors. Saf. Sci. 46 (9), 1306–1333.

Greenberg, H. A Mathematical Analysis of Traffic Flow. Tunnel Traffic Capacity Study, Port of New York Authority, New York, 1958.

Greenberg, H. An Analysis of Traffic Flow. In Operations Research, Vol.7, No.1, 1959.

Greenshields, B. D., and F. M. Weids. Statistics with Applications to Highway Traffic Analyses. Eno Foundation for Highway Traffic Control, Saugatuck, Conn., 1952.

Hasselt, H.V., Guez, A., Silver, D., 2016. Google DeepMind. Deep Reinforcement Learning with Double Q-Learning, Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16), pp. 2094-2100.

Hasselt, H.V., Insights in Reinforcement Learning. PhD thesis, 2011. Utrecht University, the Netherlands.

Hellinga, B. and Samimi, A. (2007). Safety Evaluations Using a Real-Time Crash Potential Model: Sensitivity to Model Calibration. In: Proceedings of the ITE Canadian District Annual Conference, May, 2007, Toronto, Canada.

Heydecker, B.G., J. Wu (2001) Identification of Sites for Road Accident Remedial Work by Bayesian Statistical Methods: An Example of Uncertain Inference. Advances in Engineering Software, Vol. 32, pp. 859-869.

Hong, Z., and D. Fukuda. Effects of Traffic Sensor Location on Traffic State Estimation. 15th Meeting of the EURO Working Group on Transportation, Procedia - Social and Behavioral Sciences, Vol. 54, 2012, pp. 1186–1196.

Hossain, M., Abdel-Aty, M., Quddus, M. A., Muromachi, Y. and Nafis, S. S. Real-time crash prediction models: state-of-the-art, design pathways and ubiquitous requirements, In the Journal of Accident Analysis and Prevention, Vol. 124, pp. 66-84.

Hossain, M., Abdel-Aty, M., Quddus, M. A., Muromachi, Y. and Nafis, S. S., 2019, Real-time crash prediction models: state-of-the-art, design pathways and ubiquitous requirements, In the Journal of Accident Analysis and Prevention, Vol. 124, pp. 66-84.

Hossain, M., and Y. Muromachi. A Bayesian Network Based Framework for Real-time Crash Prediction on the Basic Freeway Segments of Urban Expressways. Accident Analysis and Prevention, Vol. 45, 2011, pp. 373–381

Hossain, M., and Y. Muromachi. A Real-time Crash Prediction Model for the Ramp Vicinities of Urban Expressway. IATSS Research, Vol. 37, No. 1, 2013, pp. 68–79.

Hossain, M., and Y. Muromachi. Optimum Detector Spacing for Real-Time Monitoring of Hazardous Locations on Urban Expressways. Japanese Society of Civil Engineers, Vol. 27, No. 5, 2010, pp. 1045–1054.

Hossain, M., Muromachi, Y., 2012. A Bayesian network based framework for real-time crash prediction on the basic freeway segments of urban expressways. Accident Analysis and Prevention. 45, 373-381. http://doi.org/10.1016/j.aap.2011.08.004.

Hossain, M., Muromachi, Y., 2013a. Understanding crash mechanism on urban expressways using high-resolution traffic data. Accid. Anal. Prev. 57, 17–29.

Hu, J., Sun, J., Zhao, L., 2014. Some Flow Features at Urban Expressway On-ramp Bottlenecks in Shanghai. In: Presented at the 93rd Annual Meeting of the Transportation Research Board, Washington, D.C.

Hu, J., Zhang, L., Liang, W., 2012. Opportunistic predictive maintenance for complex multi-component systems based on DBN-HAZOP model. Process Saf. Environ. Prot. 90 (5), 376–388.

Isele, D., Rahimi, R., Cosgum, A., Subramanian, K., Fujimura, K., 2018. Navigating occluded intersections with autonomous vehicles using deep reinforcement learning. https://arxiv.org/pdf/1705.01196.pdf

J. Sun, J. Sun, 2015. Dynamic Bayesian network model for real-time crash prediction using traffic speed conditions data.Transportation Research Part C 54 (2015) 176–186.

Jensen, F.V., Nielsen, T.D., 2007. Bayesian networks and decision graphs. Springer, NY.

Jovanis, P. P., and Chang, H. L. (1986) Modeling the Relationship of Accidents to Miles Traveled. In Transportation Research Record: Journal of the Transportation Research Board, No. 1068, Transportation Research Board of the National Academics, Washington, D.C., pp. 42-51.

Katrakazas, C., Quddus, M.A., Chen, W.H., 2017. A simulation study of predicting conflict-prone traffic conditions in real-time. Presented at the Transportation Research Board 96th Annual Meeting, Washington D.C., USA, 8-12 January 8-12 2017.

Kjaerulff, U. B., and Madsen, A. L. (2007). Bayesian Networks and Influence Diagrams: A Guide to Construction and Analysis. Springer, New York, USA.

Kojima, S., Muromachi, Y., 2017. Study on Safety Measures using a Real-time Traffic Accident Prediction Model on Urban Expressway, Proceedings of 37th Traffic Engineering Research Presentation, Japan Society of Traffic Engineers, pp. 235-238.

Kown, J., Varayiya, P. California's bottlenecks, 2005, submitted ot the GoCalifornia Expert Review Panel.

Krizhevsky, A., Sutskever, I., Hinton, G., 2012. ImageNet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. 25, 1106-1114.

Kurzhanskiy, A. A., and P. Varaiya. Active Traffic Management on Road Networks: A Macroscopic Approach. Philosophical Transactions of the Royal Society A, Vol. 368, 2010, pp. 4607–4626.

Kurzhanskiy, A. A., and P. Varaiya. Active Traffic Management on Road Networks: A Macroscopic Approach. Philosophical Transactions of the Royal Society A, Vol. 368, 2010, pp. 4607–4626.

Lee, C., Abdel-Aty, M., 2008. Testing effects of warning messages and variable speed limits on driver behavior using driving simulator. In: Transportation Research Record: Journal of the Transportation Research Board, No. 2069, Transportation Research Board of the National Academies, Washington, D.C., pp. 55-64.

Lee, C., B. Hellinga, and F. Saccomanno. Real-time Crash Prediction Model for the Application to Crash Prevention in Freeway Traffic. Transportation Research Record: Journal of the Transportation Research Board, No. 1840, 2003, pp. 67–77.

Lee, C., Hellinga, B., and Ozbay, K., 2006. Quantifying effects of ramp metering on freeway safety. Journal of Accident Analysis and Prevention 38(2), 279-288.

Lee, C., Hellinga, B., Saccomanno, F., 2004. Evaluation of variable speed limits to improve traffic safety. Transport. Res. Part C: Emerg. Technol 14(3), 213-228.

Lee, C., M. Abdel-Aty, and L. Hsia. Potential Real-time Indicators of Sideswipe Crashes on Freeways. Transportation Research Record: Journal of the Transportation Research Board, No. 1953, 2006, pp. 41–49.

Lee, C., Saccomanno, F., Hellinga, B., 2003. Real-time crash prediction model for the application to crash prevention in freeway traffic. Transp. Res. Rec. 1840, 67–77.

Li, Z., Liu, P., Wei Wang, Chengcheng Xu, 2014. Development of a control strategy of variable speed limits to reduce rear-end collision risks near freeway recurrent bottlenecks, J. Central South Univ., 21(6), 2526-2538.

Li, Z., Liu, P., Xu, C., Duan, H., and Wang, W., 2016. Optimal mainline variable speed limit control to improve safety on large-scale freeway segments. Computer-Aided Civil and Infrastructure Engineering 31, 366-380.

Li, Z., Liu, P., Xu, C., Duan, H., and Wang, W., 2017. Reinforcement learning-based variable speed limit control strategy to reduce traffic congestion at freeway recurrent bottlenecks. IEEE Transactions on Intelligent Transportation Systems 18(11).

Li, Z., Liu, P., Xu, C., Duan, H., and Wang, W., 2017. Reinforcement learning-based variable speed limit control strategy to reduce traffic congestion at freeway recurrent bottlenecks. IEEE Transactions on Intelligent Transportation Systems 18(11).

Lighthill, M., and G. Whitham. On kinematic waves II. A theory of traffic flow on long crowded roads. Proceedings Royal Society of London, Part A, Vol. 229, No. 1178, 1955. http://dx.doi.org/10.1098/rspa/1955.0089.

Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D., 2016. Continuous Control with Deep Reinforcement Learning. Google DeepMind, conference paper at ICLR, London, UK, 2016.

Lin, L., Wang, Q., Sadek, A.W., 2015. A novel variable selection method based on frequent pattern tree for real-time traffic accident risk prediction. Transportation Research Part C: Emerging Technologies 55, 444-459. http://doi.org/10.1016/j.trc.2015.03.015.

Liu, M., Chen, Y., 2017. Predicting real-time crash risk for urban expressways in China. Mathematical Problems in Engineering, Article ID: 6263726.

Lu, X., Kim, Z., Cao, M., Varaiya, P., Horowitz, R. Deliver a Set of Tools for Resolving Bad Inductive Loops and Correcting Bad Data, California PATH Research Report, 2010.

Lu, X.Y., Kim, Z., Cao, M., Varaiya, P. and Horowitz, R., 2010. Deliver a Set of Tools for Resolving Bad Inductive Loops and Correcting Bad Data. California PATH Research Report.

Lu, X.Y., Varaiya, P., Horowitz, R., Su, D., and Shladover, S., 2011. Novel freeway traffic control with variable speed limit and coordinated ramp metering. Transp. Res. Rec., 2229, 55-65.

Mihajlovic, V., and M. Petkovic. Dynamic Bayesian Network: A State of Art. Doctoral Dissertation, University of Twente, the Netherlands, 2001.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M., 2013. Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D., 2015. Human-level control through deep reinforcement learning. Nature 518, 529-533.

Morrison, D., and S. Martonosi. Characteristics of Optimal Solutions to the Sensor Location Problem. Annals of Operations Research, Vol. 226, No. 1, 2014, pp. 463–478. https://doi.org/10.1007/s10479-014-1638-y.

Muñoz, L., Sun, X., Sun, D., Horowitz, R. and Alvarez, L. Traffic Density Estimation with Cell Transmission Model, Proceedings of the American Control Conference Denver, Colorado.2003.

Murphy, K., 2002. Doctoral dissertation, Dynamic BayesianNetworks: Representation, Inference and Learning.

Oh, C., Oh, J., Ritchie, S., Chang, M., 2001. Real-time estimation of freeway accident likelihood. In: Presented at the 80th Annual Meeting of the Transportation Research Board, Washington, D.C.

Pande, A., Abdel-Aty, M., 2005. A Freeway Safety Strategy for Advanced Proactive Traffic Management. Journal of Intelligent Transportation Systems, Vol. 9, No. 3, pp. 145–158.

Park, H., and A. Haghani. Real-time Prediction of Secondary Incident Occurrences using Vehicle Probe Data. Transportation Research Part C, Vol. 70, 2015, pp. 69–85.

Park, H., Haghani, A., Samuel, A., Knodler, M.A., 2018. Real-time prediction and avoidance of secondary crashes under unexpected traffic congestion. Accident Analysis and Prevention 112, 39-49.

PTV-AG, VISSIM 5.10 User Manual. 2008.

Rezaee, K., Abdulhai, B., Abdelgawad, H., 2012. Application of Reinforcement Learning with Continuous State Space to Ramp Metering in Real-World Conditions. Proc. 15th IEEE Int. Conf. Intell. Transp. Syst. (ITSC), Sep 2012, pp. 1590-1595.

Roy, A., Hossain, M., and Muromachi, Y., 2018a. Enhancing the prediction performance of real-time crash prediction model: A Cell Transmission-Dynamic Bayesian Network approach, 97th Annual Meeting of Transportation Research Board, Washington D.C.

Roy, A., Hossain, M., and Muromachi, Y., 2018b. Development of Robust Real-Time Crash Prediction Models Using Bayesian Networks, Asian Transportation Studies, Volume 5, Issue 2, 349-361.

Roy, A., R. Kobayashi, M. Hossain, and Y. Muromachi. Real-time Crash Prediction Model for Urban Expressway Using Dynamic Bayesian Network. Journal of Japan Society of Civil Engineers, Series D3, Vol. 72, No. 5, 2016, pp. 1331–1338.

Shi, Q., and M. Abdel-Aty. Big Data Applications in Real-time Traffic Operation and Safety Monitoring and Improvement on Urban Expressways. Transportation Research Part C, Vol. 58, 2015, pp. 380–394.

Sun, J. and J. Sun. Dynamic Bayesian Network Model for Real-time Crash Prediction Using Traffic Speed Conditions Data, Transportation Research Part C, Vol. 54, 2015, pp. 176–186.

Sun, J., and J. Sun. Real-time Crash Prediction on Urban Expressways: Identification of Key Variables and a Hybrid Support Vector Machine Model. IET Intelligent Transport Systems, Vol. 10, No. 5, 2016, pp. 331–337.

Sun, J., Sun, J., 2015. A Dynamic Bayesian Network model for real-time crash prediction using traffic speed conditions data. Transportation Research Part C: Emerging Technologies 54, 176-186. http://doi.org/10.1016/j.trc.2015.03.006.

Sutton, R.,S., and Barto, A.,G. 1998. Introduction to Reinforcement Learning. MIT Press.

Thrun, S., Shwartz, A., Lee, C.B., 1993. Issues in Using Function Approximation for Reinforcement Learning. Proceedings of the 1993 Connectionist Models Summer School, Lawrence Erlbaum Publisher, Hillsdale, NJ.

Wang, Z., Schaul, T., Hessel, M., Hasselt, H.V., Lanctot, M., Freitas, N., 2016. Google DeepMind London UK. Dueling Network Architectures for Deep Reinforcement Learning, Proceedings of the 33rd International Conference on Machine Learning, JMLR: W & CP 48, New York, NY, USA.

Watkins, C. J. C. H., 1989. Learning from delayed rewards. PhD thesis, University of Cambridge, England.

Wu, Y., Abdel-Aty M., 2018. Developing an algorithm to assess the rear-end collision risk under fog conditions using real-time data, Transportation Research C 87, 11-25.

Xu, C., Wang, W., Liu, P., 2013. A genetic programming model for real-time crash prediction on freeways. IEEE Trans. Intell. Transp. Syst. 14 (2), 574–586.

Xu, C., Wang, W., Liu, P., Guo, R., Li, Z., 2014a. Using the Bayesian updating approach to improve the spatial and temporal transferability of real-time crash risk prediction models. Transport. Res. Part C: Emerg. Technol. 38, 167–176.

Xu, C., Wang, W., Liu, P., Zhang, F., 2014b. Development of a real-time crash risk prediction model incorporating the various crash mechanisms across different traffic states. Traffic Injury Prevent. 16 (1), 28–35.

Yang, H., Bartin, B., Ozbay, K., 2014. Mining the characteristics of secondary crashes on highways. J. Transport. Eng. 140 (4), 04013024.

Yang, K., Wang, X., Quddus, M., Yu, R., 2018. Deep learning for real-time crash prediction on urban expressways, 97th Annual Meeting of Transportation Research Board, Washington D.C.

Yeo, H., Jang, K., Skabardonis, A., Kang, S., 2013. Impact of traffic states on freeway crash involvement rates. Accid. Anal. Prev. 50, 713–723.

Yu, R., and Abdel-Aty, M., 2014. An optimal variable speed limits system to ameliorate traffic safety risk, Journal of Transportation Research: Part C 46, 235-246. DOI: 10.1016/j.trc.2014.05.016.

Yuan, J.H., Abdel-Aty, M., Wang, L., Lee, J.Y., Wang, X.S., Yu, R.J., 2018. Real-Time Crash Risk Analysis of Urban Arterials Incorporating Bluetooth, Weather, and Adaptive Signal Control Data. Accepted for presentation at 97th Annual Meeting of the Transportation Research Board, TRB No. 18-00590, Washington DC, January 2018.

Zhang, K., Taylor, M.A.P., 2006. Effective arterial road incident detection: a Bayesian network based algorithm. Transport. Res. Part C: Emerg. Technol. 14 (6), 403–417.

Zheng, Z., Ahn, S., Monsere, C.M., 2010. Impact of traffic oscillations on freeway crash occurrences. Accid. Anal. Prev. 42 (2), 626–636.

Zhu, F., Ukkusuri, S.V., 2014. Accounting for dynamic speed limit control in a stochastic traffic environment: A reinforcement learning approach. Transportation Research Part C 2014, 41, 30-47.

CHAPTER 3

STUDY AREA, DATA COLLECTION AND DEVELOPMENT OF MACROSCOPIC TRAFFIC MODEL: THE CELL TRANSMISSION MODEL (CTM)

3.1 Introduction

The concept of a real-time crash prediction model (RTCPM) building is based on the hypothesis that the probability of a crash on a specific road section can be predicted for a very short time window using the instantaneous traffic flow data ((Lee et al, 2003; Pande et al, 2005). Most existing literature has made use of traffic flow data from dual loop detectors, placed at fixed locations beneath the road surface where the inter-detector spacing varied substantially with an average of around 0.80 km and had high standard deviation (Abdel-Aty et al, 2012, Hossain and Muromachi, 2011; Shi et al, 2015). A study (Abdel-Aty et al, 2012) showed a range of standard deviation between 0.88 and 3.60 km for an AVI system. Another study (Shi et al, 2015) reported a minimum spacing of 0.16 km to a maximum of 5.90 km with a standard deviation ranging between 0.16 and 1.56 km. Hong and Fuduka, 2012 used cell transmission model (CTM) with ensemble Kalman filter (EnKF) in estimating the impact of various detector location configurations on estimation of travel speed and concluded that sensors located at large distances from each other without location optimization lead to an overestimation of travel speed, whereas sensor numbers can be reduced if their locations are optimal to achieve a better estimate of travel speed. Many studies attempted to minimize the number of detectors (Bianco et al, 2001; 2006; 2007) whereas Morrison and Martonosi, 2004 examined whether there are sufficient conditions for optimal solutions for the detector locations. This non-uniformity in existing detector layouts along with high variability in detector spacing among expressways/freeways raise questions about the universality as well as transferability of such models. The doubt gets further bolstered by the studies (Abdel-Aty et al, 2012; Hossain and Muromachi, 2010) where the authors examined the performances of RTCPMs built with traffic data collected from six different combinations of detector layouts and found different results for each combination. Therefore, there is sufficient evidence postulating that variation in detector layout has an impact on the states of traffic flow variables.

The existing RTCPMs are developed depending on the detector layout that exists in the study area. Hence, variation in detector layout raises the issue of spatial transferability of the RTCPMs as altering the locations or installing new detectors to replicate the detector layout of the model is neither practical nor cost effective. Moreover, even for new roads, it may not always be plausible to follow a specific detector spacing. To address these issues, there has been an urge to devise a mechanism to simulate traffic sensor data of desired spacing from any detector layout that can be fed into an RTCPM. This study employed a macroscopic dynamic freeway traffic model called CTM, introduced by Daganzo (1994, 1995), which is consistent with the hydrodynamic theory of traffic flow (Lighthill et al, 1955), to transform the traffic states obtained from non-uniform detector layouts into a pre-defined detector layout. The CTM was chosen for its analytical simplicity and ability to reproduce congestion propagation dynamics.

Crashes occurring on freeways/expressways are considered to relate closely to preceding traffic states occurring before the crash, which are time-varying (Mihajlovic et al, 2001). Thus, it is essential to establish a single model that can address such time series data and the evolving process of traffic flow to predict crash risk in real time. Substantial effort has been put into improving the RTCPMs by employing sophisticated modeling methods such as artificial neural networks, Bayesian structural equations, Bayesian networks (BNs), dynamic Bayesian networks (DBNs), Bayesian classifier, support vector machines, genetic programming etc. (Hossain and Muromachi, 2001, Sun et al, 2015, 2016; Lee et al, 2006). Among these, the modeling architecture of a dynamic Bayesian network (DBN) conforms to this requirement (Mihajlovic et al, 2001; Sun et al, 2015; Roy et al, 2016). Therefore, this study develops RTCPMs by simulating uniformly spaced detector data from fixed detector layouts and uses DBNs as the modeling method to distinguish between crash prone and normal traffic conditions. Finally, the model performance is compared with the models constructed with fixed detector layouts and for which BNs are employed as the modeling method.

3.2 study area and data collection

Tokyo Metropolitan Expressway's 11.9 km long route 3 Shibuya and 13.5 km long route 4 Shinjuku are the two routes of this study which sustains substantial number of crashes throughout the year. The routes have two lanes in each direction with 43 and 41 detectors in the inbound and outbound direction respectively for route 3; and 50 and 44 detectors in the inbound and outbound direction respectively for route 4 (roughly 250 m apart) as shown in **Table 3.1**. **Figure 3.1** shows both routes and the distribution of active detectors which were found to be valid and functioning without errors in the inbound direction (route 4) and outbound direction (route 3). The fixed loop detectors collect flow, average speed, and occupancy for each minute round the clock.

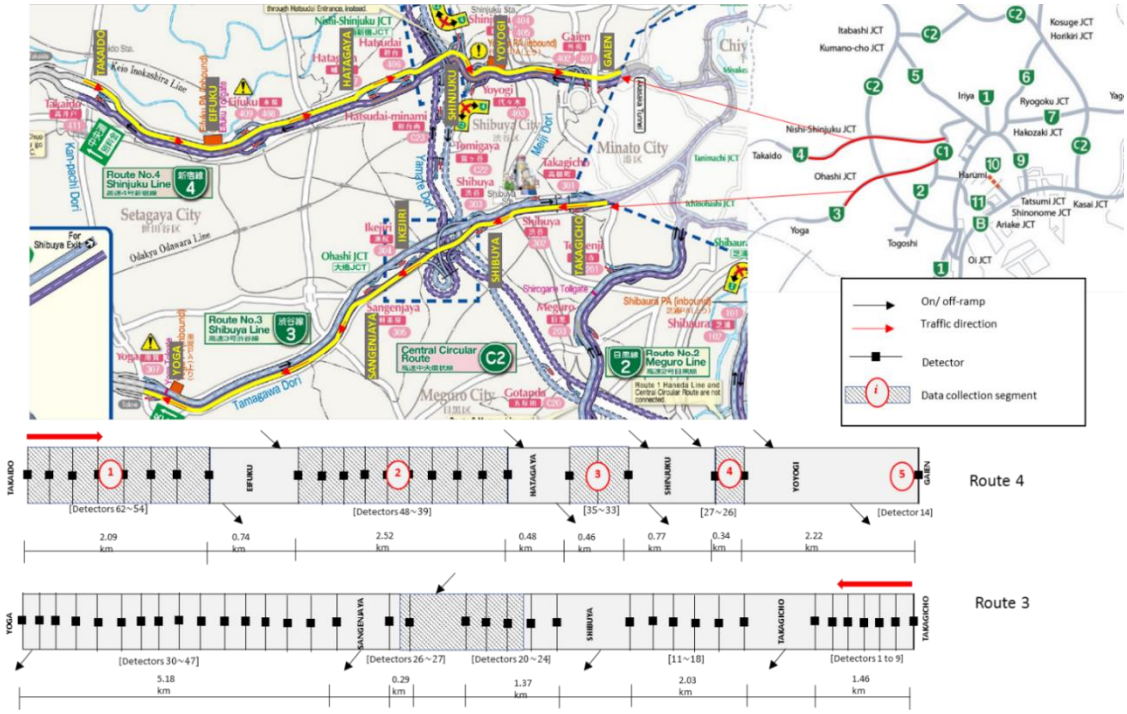


Figure 3. 1 Tokyo Metropolitan Expressway Route 4 (Shinjuku) and Route 3 (Shibuya)

Source: Tokyo Metropolitan Expressway Co. Ltd. Website

Table 3. 1 Tokyo Metropolitan Expressway Route 4 (Shinjuku) and Route 3 (Shibuya) number of detectors

	Route 4 detectors			Route 3 detectors		
	BFS	On-ramp	Off-ramp	BFS	On-ramp	Off-ramp
Inbound	50	7	3	43	2	4
Outbound	44	3	7	41	4	1

In order to construct RTCPMs, both crash data and corresponding high-resolution detector data are needed. Therefore, the study collected detector data and the matching crash data from Route 3 (Shibuya) and Route 4 (Shinjuku) of the Tokyo Metropolitan Expressway for a period from March 2014 to August 2014. The pre-crash traffic data were then extracted for 101 crash cases for which complete matching detector data were found. The one-minute pre-crash data was collected, three minutes prior to the occurrence of the crash, and the corresponding normal (or no-crash) data was collected for the same time frame, but from the days when no crashes took place. The data comprising the time and location of the crash helped when matching the crash and detector databases. The fixed loop detectors collected the cumulative vehicle count (veh/min), average speed (km/h), and average occupancy (%) aggregated for every minute, round the clock.

In order to build the RTCPMs, about 101 crash cases and 1732 no-crash cases were considered for model building and validation. While validating, to distinguish between crash and no-crash cases, different thresholds of probability of crash prediction was tested. In several studies by Abtel-Aty et al. (2005), Sun J. (2015), Hellinga et al. (2003a) crash and corresponding normal cases were selected in the similar manner as is in this study which is a well-established method of sampling in RTCP modeling. In addition, unlike several statistical methods, machine learning methods such as BN and DBN has adaptation property which makes the models incorporate new real-time traffic data into the model while fading away the old traffic data which is one of the important features of these models.

The issue inappropriate mixture of traffic conditions relating to crash and non-crash cases is referred as data imbalance (Kui et al, 2018). Several studies tried to address it by approximately balancing the data which caused biased outcomes. Deep neural network (DNN) was used for predicting crash-prone traffic conditions (Kui et al., 2018) where they found that DNN can predict 63-65% of the crashes with 5% false alarm rate. Furthermore, it also addressed the class- balancing issue of the training data and concluded that the prediction performance degrades with the increasing size of balanced data as huge amount of data gets eliminated during class balancing of data and deep learning methods like DNNs requires big data.

3.3 Theoretical background of CTM

The workflow followed in this study is illustrated by **Figure3.2**. According to the CTM (Daganzo, 1995; Lee et al, 2006), if the relationship between traffic flow (q) and density (k) can be expressed in a triangular form as in **Figure 3.3**, then the Lighthill, Whitham, and Richards (LWR) equations for a single highway link can be approximated by a set of difference equations where current conditions (the state of the system) are updated with every clock interval as:

$$q = \min\{V_F k, q_{max}, w\,(k_j - k)\}, \qquad \text{for } 0 \le k \le k_j \tag{3.1}$$

Where V_F is the free flow speed,

q_{max} is the maximum flow (or capacity),

w is the back wave or wave speed, and

k_j is the jam density.

In the CTM, a road segment is divided into several homogeneous cells, i whose length is equal to the free flow speed times one clock interval. The state of the system at instant t is then given by the number of vehicles contained in each cell, $n_i(t)$. The following parameters are defined for each cell:

$N_i(t)$ = maximum number of vehicles that can be present in cell i at time t;

$Q_i(t) =$ Maximum number of vehicles that can flow into cell i when the clock advances from t to $t + 1$.

If cells are numbered consecutively starting with the upstream end of the road from i = 1 to *I*, the recursive relationship of the CTM can be expressed as:

$$n_i(t + 1) = n_i(t) + y_i(t) - y_{i+1}(t) \tag{3.2}$$

Where $y_i(t)$ is the inflow to cell *i* in the time interval (t, t + 1), given by

$$y_i\,(t) = \min\,\{n_{i-1}(t),\, Q_i(t),\, \mathrm{d}\,[N_i(t) - n_i(t)]\} \tag{3.3}$$

Where $d = w/v$

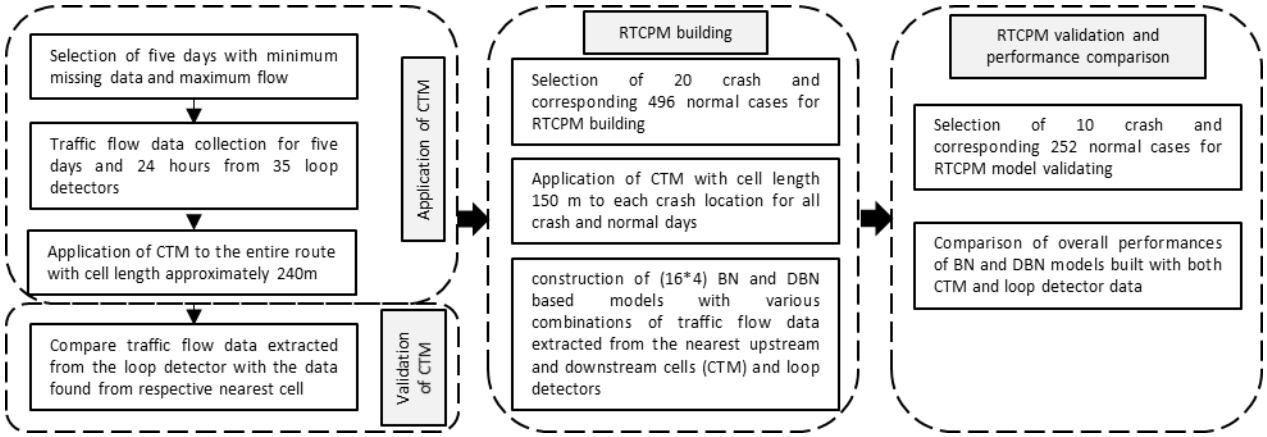


Figure 3. 2 Workflow diagram

3.4 Development and validation of CTM

3.4.1 Experimental setup and model development

The objective of this thesis was to observe the crash risk situation with and without VSL control. In order to do that, a simulated traffic model was essential. Previously, microscopic simulation model has been constructed using loop detector data. Hence, to investigate an alternative way to investigate and compare the crash risk situation with and without VSL control, in this thesis, the CTM was adopted to simulate the traffic data using the high density loop detector data.

Moreover, in order to resolve the issue of non-uniform detector spacing, CTM was employed. If CTM transforms the inter-detector spacing into a homogeneous distribution then it can be used anywhere in the world with any detector spacing with necessary modification. This uniformity in detector layout also provides with the opportunity to investigate the optimal location problems for traffic data collection and VSL control application.

The fundamental diagrams (FD) with VSL control have been chosen in various ways in the previous studies. For example, a study (Frejo, J. R. D., 2018) combined first order (Hegyi and Hoogendoorn, 2010) and second order (Carlson et al., 2010, 2011) macroscopic model to find a better FD in order to ensure safety and resolution of traffic congestion. In their proposed model, the compliance affects the VSL-induced critical density and capacity. So, the model changes with the change in compliances. The other parameters such as k_c and Q_c can be adjusted accordingly.

Their study found that the Kc increases with decrease of VSL, Q_c decreased with reduction of VSL, For low densities, compliance can be very low and When densities are increased, the VSL-induced speeds start to decrease substantially. The proposed model can accommodate both of the models, though has some limitations.

Another study by Papageorgiou et al., 2008 found that the parameters of FD do not necessarily remain constant, in fact, the parameters may change considerably from day to day without any obvious reasons. They found that the application of VSLs decrease the slope of the flow–occupancy diagram at under critical conditions, and shift the critical occupancy to higher values, and enable higher flows at the same occupancy values in overcritical conditions.

Nunzio et al., 2014, took an attempt to generate a variable length model (VLM) with the influence of both traffic light and VSL control to analyze steady-state behavior of the system. In this particular model, Q_max of the road segment remains fixed, and the max allowed speed (VSL), back-wave (w), jam density (k_j) etc. are uniquely defined after the nominal critical density (k_c). The objective functions included traveling time reduction, infrastructure utilization and energy consumption reduction. The outcome showed that if congestion doesn't spill back or disappear, the system is stable and multiple equilibrium points can be reached via VSL. Location of operation control and desired traffic conditions were identified but did not meet all of the objective functions.

The 'traditional CTM' or the general CTM proposed by Daganzo, C.F. (1994) is with the shape of a trapezoid where the back-wave speed is less than the free-flow speed and the flow in a cell is decided by the equations (3.1 to 3.3). The physical meaning could be vehicle moving at a free-flow speed till the first critical density, then face the increasing density and the move toward a congested situation then will move with a congested speed.

In the modified CTM, the maximum flow, jam-density, back-wave speed are assumed to be fixed for the day on April, 8, 2014. Only the free-flow speed can be changed according to the VSL control value decided by the RL-agent (value selected from the action set). This is to keep the VSL-based CTM model simple and more flexible to accommodate VSL values at times of crash risk ≥ 10. If all the parameters (back-wave, capacity flow, jam density etc.) kept susceptible to change, it becomes difficult to execute VSL control along with RL-based intervention as RL-based simulation requires several time consuming simulations. Hence, in order to allow the CTM model

incorporate the VSL property into it, the three parameters of FD is kept fixed for the study segment for the day.

The general procedure for networks involves two steps for each clock interval.

(i) Determine the flow on each link with the equivalent of **Equation (3.3)**.

(ii) Update the cell occupancies by transferring the flows of step (i) from the beginning cell to the end cell of each link.

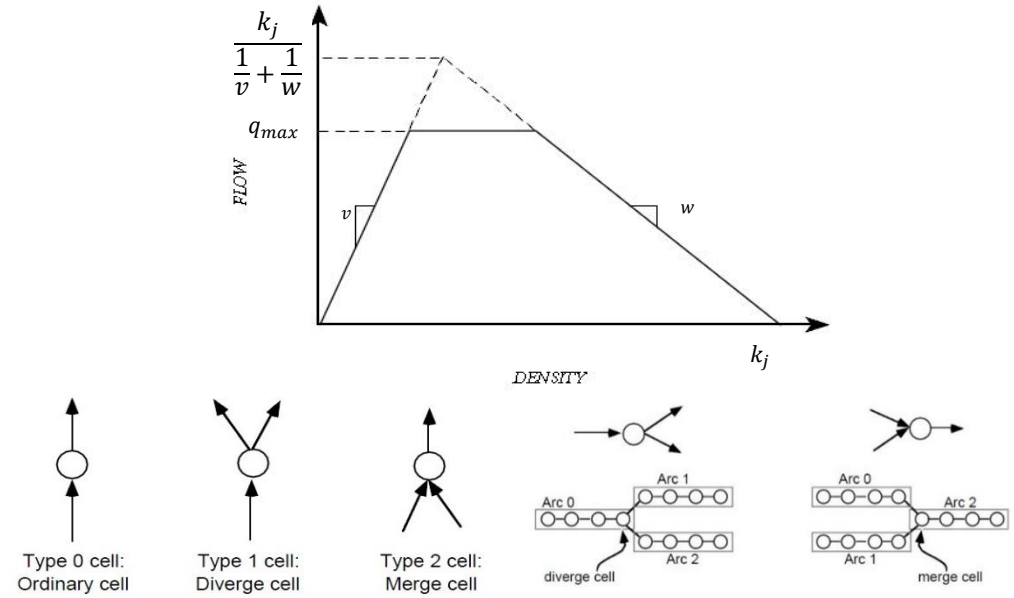


Figure 3. 3 Fundamental diagram and nodes (simple, diverge and merge) of CTM

In CTM, a road segment is described by nodes and directed arcs where each arc possesses some physical data that includes its length and the parameters defining a flow–density relation for a steady state of traffic of the type shown in **Figure 3.3**. There are three kinds of cells: (i) ordinary, (ii) diverge, and (iii) merge cell as shown.

For this study, five days (April 1, August 21, June 26, March 26, and May 2, 2014) were selected as on these days most of the detectors in Shibuya Route 3 line inbound direction were in sound condition (fewer missing data) and there was enough traffic flow to ensure the fundamental diagram (FD) to fulfill the conditions to form the CTM. The FD was generated in the following way.

1. For each sound detector on the freeway, the flow and density data were extracted.

2. The maximum value of the flow was considered as the capacity, q_{max}.

3. According to Dervisoglu et al. (2009) and Kurzhanskiy and Varaiya (2009, 2010), the density–flow pair provides a good fit for free flow speed,v. The least-squares method was used to estimate the free-flow speed, v.

4. The critical density was found using $k_c = \frac{q_{max}}{V_F}$.

5. The constrained least-squares method was employed to determine the back-wave speed, w.

Afterwards, CTM was applied to the entire Shibuya Route 3 (inbound direction) for 24 hours on the aforementioned five days using traffic flow data for each minute (total of 1440 minutes a day). Since the cell length is the product of free flow speed and time step, to keep the cell length approximately uniform, the time step was chosen accordingly. To build the CTM model, the AuroraRNM (Kurzhanskiy, 2009) simulator was used and it has an option that enables the user to choose different time steps for different links. The traffic flow data generated using CTM was validated using the traffic data extracted from the fixed loop detectors.

3.4.2 Model validation

To employ the CTM to the entire route, a FD of traffic was generated using a flow–density graph. **Figure 3.4 (a) and (b)** show examples of good and poor datasets, respectively.

The values of FD parameters ranged as follows: capacity (q_{max}) 1800–3900 vehicles/h; free flow speed (v) 66.92–139.2 km/h; critical density (k_c) 22.6–31.52 vehicles/km; jam density (k_j) 75–130 vehicles/km; and back wave (w) 33.6–40.3 km/h. The cell length was kept at 240 m, the average inter-detector spacing for this study route, to generate uniformly distributed detectors.

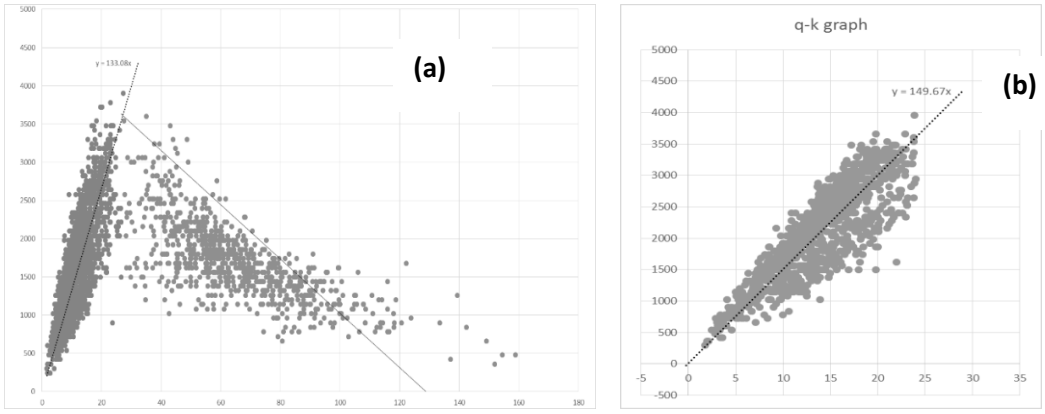


Figure 3. 4 Estimating the fundamental diagram: (a) good data; and (b) poor data

Validation of the CTM was done by comparing traffic flow parameters of the cells located closest to each of the 35 detectors. For example, on June 26, 2014 between detectors 03-01-24 and 03-01-27, the free flow speed was found to be 139.2 km/h with the distance between these two detectors being 1.14 km. To maintain the cell length to be approximately 240 m, the average inter-detector spacing for this route, a time interval of 6 seconds was chosen (cell length = free flow speed times time interval). Three cells in the link between detectors 03-01-24 and 03-01-27 were generated by the CTM **(Figure 3.5)**. Between 07:00 and 09:00, the flow in this segment was uninterrupted and, therefore, a plot of the flow, speed, and occupancy data during this time stretch was generated to allow a comparison between the CTM and loop detector data. For comparative purpose, cell 2 which is closest to detector 03-01-26 was selected. It can be observed from **Figure 3.5** that the simulated value follows the measured value well when it comes to flow and occupancy data; however, the speed data differ from the original data to some extent. Such a comparison was conducted for all 35 detectors with their nearest cells for 24 h for every minute of 5 days as suggested by previous study (Muñoz et al, 2003). The mean percentage error (MPE) defined by **Equation3.4** was calculated to validate the data, where N = total number of data for a day = number of detectors × time (min) = 35 × 1440 (**Table 3.2**).

$$\frac{1}{N}\sum_{1}^{N}\left|\frac{Value_{Original-Value_{CTM}}}{Value_{original}}\right| \qquad (3.4)$$

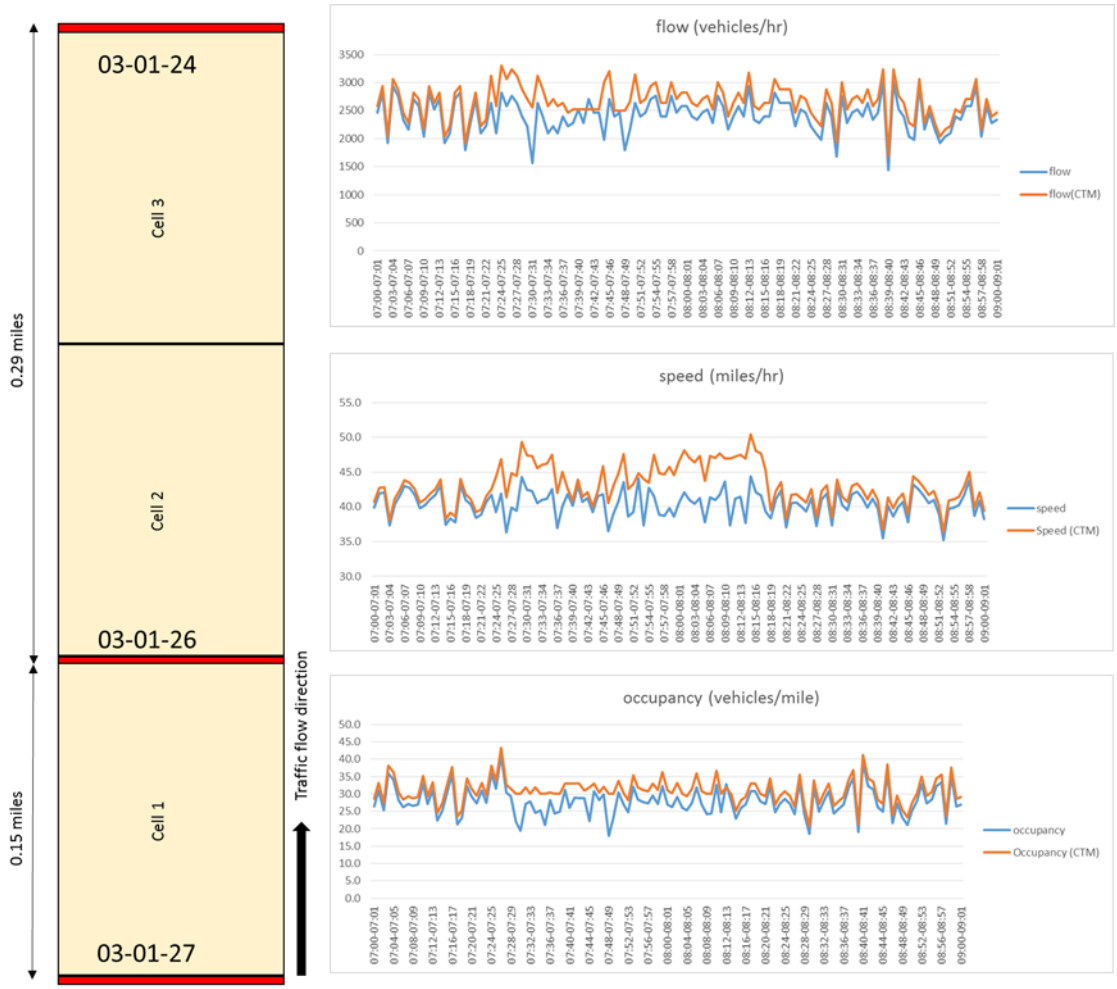


Figure 3. 5 Measured and simulated flow, speed, and occupancy between detector 03-01-27 and 03-01-24 on June 26, 2014 (07:00–09:00)

Table 3. 2 Mean percentage error for flow, speed, and occupancy estimates

	Mean percentage error (%)		
Date	Flow	Speed	Occupancy
APRIL 1	0.177	0.403	0.080
AUGUST 21	0.040	0.180	0.057
JUNE 26	0.090	0.314	0.108
MARCH 26	0.084	0.111	0.064
MAY 2	0.025	0.142	0.071

3.5 Development and validation of modified CTM

In the previous sections, the construction and validity of a CTM was performed using traffic data of route 3 Shibuya. Later on, in this thesis, a variable speed limit is used as intervention, for which

traffic data of route 4 Shinjuku is employed. The intervention is integrated with the CTM model with the help of python program. Hence, in this section the construction of the CTM for route 4 Shinjuku will be discussed.

As mentioned earlier, the primary parameters of a CTM are defined by the left limb, the right limb and the apex of the triangle of the FD. The left limb decides the sending flow from an upstream cell to the downstream cell and the right limb decides how much flow it can receive depending on its current capacity. The theoretical background of the basic CTM has been discussed in the earlier section. Now, to incorporate the VSL, as a measure for reducing crash risk, the CTM needs to be modified.

Generally, the FD for a link or segment remains the same throughout the simulation which is one of the positive sides of using a macroscopic traffic model. In this thesis, the VSL control optimization and policy making will be conducted by a reinforcement learning method, hence, the modified CTM model does not have to be burdened with solving the optimality problem for crash risk reduction. However, the model has to be in such a way so that the FD does not remain fixed throughout the simulation. The left and right limbs, or the sending and receiving functions needs to be adjusted to accommodate the VSL values. So, the sending and receiving functions of the traditional CTM will be changed to the followings-

$$s_i(t) = \min\{V_{SL}(t).\, k_i(t), Q_{VSL}\}\,; \quad \text{Where, } V_{SL}(t) \in [\, a_t \,] \tag{3.5}$$

$$r_i(t) = \min\left\{\omega_i.\left(k_{i,jam} - k_i(t)\right), Q_{VSL}\right\} \tag{3.6}$$

Where, $s_i(t)$ is the sending flow at time t, $r_i(t)$ is the receiving flow, a_t is the action set which will be decided by the reinforcement learning (RL) agent. For example, $a_t = \{\pm 20, \pm 0,\ V_F\}$ km/h, which means the current speed could be reduced or increased by 20km/h, or the free flow speed for the segment will be chosen, depending on the optimization target- reducing crash risk at a target cell. V_F is the free-flow speed, $V_{SL}(t)$is the speed limit chosen at time k, $k_i(t)$is the density, $Q_{VSL}$is the maximum flow under current speed limit $V_{SL}(t)$, ω_i is the back wave speed and $k_{i,jam}$is the jam density.

The flow in a cell i can be defined by the following receiving function:

$$q_i(t) = \min\{s_{i-1}(t), r_i(t)\} \quad (3.7)$$

The density evolution can be determined by the following equation:

$$k_i(t+1) = k_i(t) + \Delta T / L_i (q_{i-1}(t) - q_i(t)) \quad (3.8)$$

Here, ΔT is the simulation time step, which is equal to the time with which a vehicle passes a cell at free-flow speed. L_i is the cell length of cell i. The speed within each cell can be determined according to the current density, $k_i(t)$ and speed limit, V_{SL}:

$$v_i(t) = \begin{cases} \min\{V_{SL}(t-1), V_F\} & if\ k_i(t) \leq k_{VSL} \\ \left(k_{i,jam}(t) - k_i(t).\frac{w_i}{k_i(t)}\right) & if\ k_i(t) > k_{VSL} \end{cases} \quad (3.9)$$

Where, d_{VSL} is the density associated with the flow Q_{VSL} (**Figure 3.6**.) under the speed limit, V_{SL}.

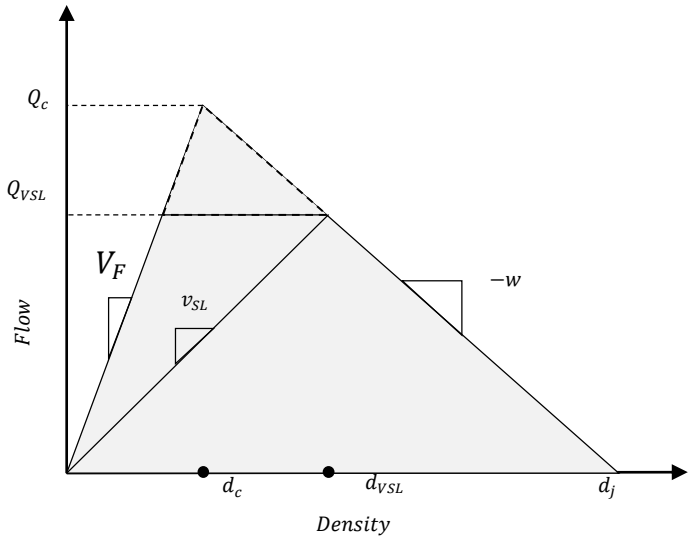


Figure 3. 6 The modified CTM with VSL control

To summarize, in order to build the modified CTM model, the basic parameters of an FD which are free-flow speed, back wave speed, jam density and the maximum flow will be estimated from the traffic flow data from the loop detectors and these parameters will remain fixed. Then, similar as the traditional CTM, the simulation time step will be decided to get the cell length according to the formula cell length = free-flow speed times time step. This is to ensure that the vehicles don't cross multiple cells at a time step. So, after calculating the free-flow speed and deciding a suitable

cell length, the time step is calculated. Next, unlike the traditional FD, the modified FD (**Figure 3.6**.) whose left limb can be moved lower depending on the VSL value chosen, will be constructed with a sending and receiving function as shown in the **Equation 3.5** and **3.6**. The sending function will choose either the maximum flow under current VSL value, Q_{VSL} or the actual flow under current VSL value ($V_{SL}(t).k_i(t)$) whichever is the minimum. Not to mention, the VSL values will be chosen by the RL agent that can choose a value ranging from [10km/h to 110 km/h,V_F, no-control] etc. At the same time, the receiving function will follow **Equation 3.6** and choose the minimum of the maximum flow under current VSL value, Q_{VSL} and the back-wave speed calculated from the jam density and current density$\omega_i.\left(k_{i,jam}-k_i(t)\right)$. According to Daganzo's original CTM theory, the sending flow of upstream cell ($i-1$) is the receiving flow of downstream cell, i. Hence, the flow entering cell i will follow the minimum of the sending flow of cell ($i-1$) and receiving flow of cell,i, according to **Equation 3.7**. As for the speed evolution in a cell, it will depend on the if the current density fall to the left or right hand side of the density under current VSL value. If it falls on the left region, the speed will be dominated by either the V_F, or VSL value; on the other hand, if it falls under the right-hand side, the speed will be decided by the jam density and the back-wave speed.

In this thesis a modified CTM is considered to incorporate VSL control properties into it. The CTM is based on FD where the three major parameters- capacity flow, free flow speed and jam density needs to be defined. The left limb of the triangular FD represents free-flow condition and the right limb represents congested condition. In order to incorporate VSL control, the FD of the modified CTM has flexibility to change its free-flow limb, whereas the right limb remained fixed assuming the back-wave speed will not be controlled by this modified CTM. However, the change of the traffic condition with the change of the FD for different speed limits can be demonstrated briefly as follows-

The sending and receiving function of the modified CTM is expressed as:

$s_i(t) = \min\{V_{SL}(t).k_i(t), Q_{VSL}\}$;

Where,$V_{SL}(t) \in [\, a_t = 10, 20 \ldots., no\ control, V_F]$

In case, when the there is no congestion, the traffic is in free-flow condition. Hence, all the flow values will be on the left limb. Let's assume a traffic state a on the left limb and a speed limit V_{SL} was applied. If $q_a < Q_{VSL}$, then after applying the V_{SL}, the traffic state a will move to a^{new} position which means higher density with the same flow, q_a.

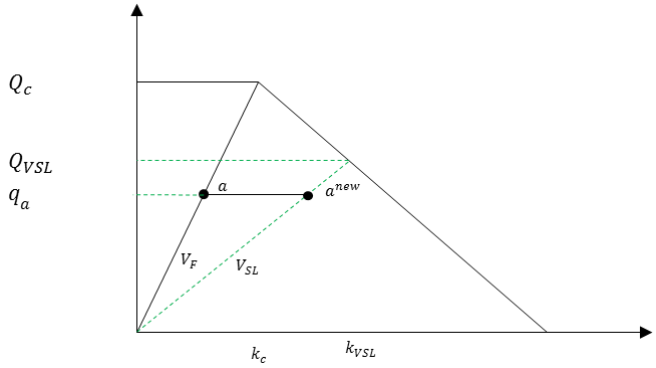


Figure 3. 7 modified CTM when $q_a < Q_{VSL}$,

In case, when the there is no congestion, the traffic is in free-flow condition. Hence, all the flow values will be on the left limb. Let's assume a traffic state a on the left limb and a speed limit V_{SL} was applied. If $q_a > Q_{VSL}$, then after applying theV_{SL}, the traffic state a will move to a^{new} position which means higher density with the lower flow, Q_{VSL}.

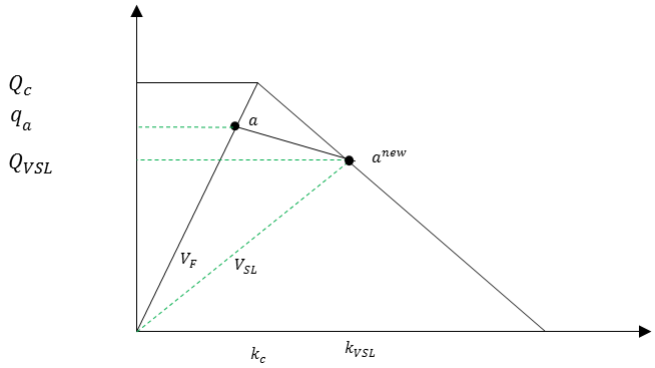


Figure 3. 8 modified CTM when $q_a > Q_{VSL}$

Algorithm for modified CTM

Algorithm-1: modified CTM

Initialize CTM

Define CTM parameters using FD: $q_{max}, V_F, k_j, w, k_c, \Delta T, L_i$

Generate traffic flow parameters in cells using equation 6 and 7

At time=t

If Crash risk > threshold, activate VSL

[crash risk by RTCPM]

[VSL by RL]

Select Q_{VSL} *for chosen* V_{SL}.

Get Speed $v_i(t)$

$$v_i(t) = \begin{cases} \min\{V_{SL}(t-1), V_F\} & if\ k_i(t) \leq k_{VSL} \\ \left(k_{i,jam}(t) - k_i(t).\frac{w_i}{k_i(t)}\right) & if\ k_i(t) > k_{VSL} \end{cases}$$

Calculate cell flow, $q_i(t)$:

$$s_{i-1}(t) \begin{cases} V_{SL}.k_i(t) & if\ \left(V_{SL}.k_i(t)\right) < Q_{VSL} \\ Q_{VSL} & ,otherwise \end{cases}$$

$$r_i(t) \begin{cases} \boldsymbol{\omega_i}.\left(\boldsymbol{k_{i,jam}} - \boldsymbol{k_i(t)}\right) & if\ \left(V_{SL}.k_i(t)\right) < Q_{VSL} \\ Q_{VSL} & ,otherwise \end{cases}$$

Get flow $q_i(\boldsymbol{t}) = \min\{s_{i-1}(\boldsymbol{t}), r_i(\boldsymbol{t})\}$

Update density,

$$\boldsymbol{k_i(t+1) = k_i(t) + \Delta T / L_i(q_{i-1}(t) - q_i(t))}$$

Repeat for t= t+1

For this thesis, traffic data of inbound direction on April 8th, 2014 is chosen for data collection and simulation as on this day, no crashes occurred and the traffic was steady. There are two segments of route 4: segment 1 (2.09 km) and 2 (2.52 km) which encompasses the longest expressway segments. So, these two segments will be used for intervention application in chapter 5. First of all, to construct the CTM for route 4, the FD needs to be defined. FD was generated using traffic data collected during 12:00-12:01 to 17:59-18:00 time from segment 1 and 2 **(Figure 3.1**), because, during this time the traffic was steady whilst with the presence of back wave. Both

free-flow and back wave is necessary to construct the FD to generate left and right hand side of the triangle. In **Figure 3.9**, the space-time diagram of speed data, speed is color coded from green (higher speed) to red (lower speed). It shows that the traffic was more or less steady in both segments 1 (detector 62 to 54) and 2 (detector 48 to 39) except for the segment 2 during 16:00-16:15 when there was a back-wave generated probably due to the sudden rush hour. After considering the availability of steady flow, the FD was constructed as shown in **Figure 3.8**.

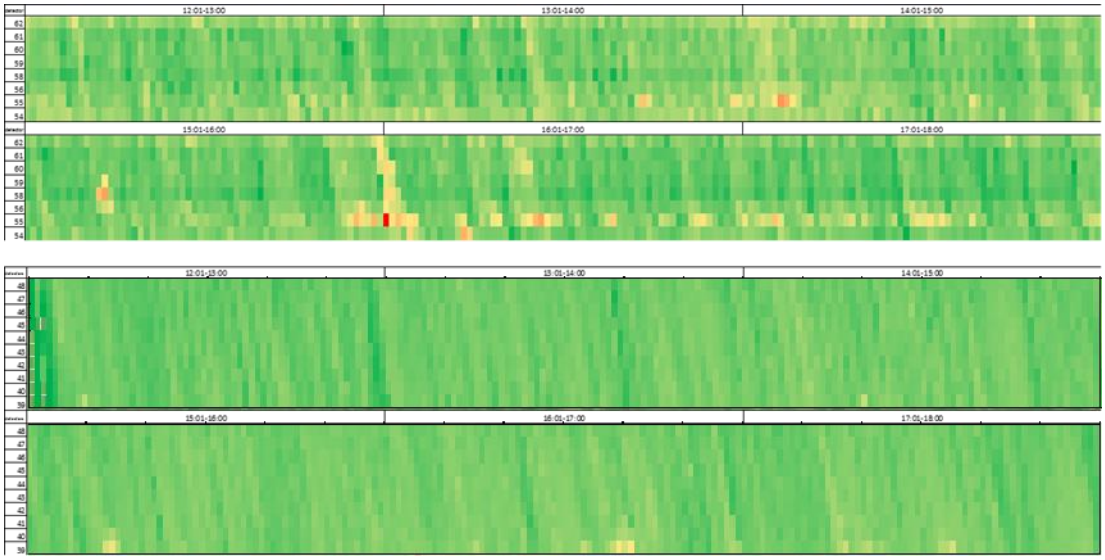
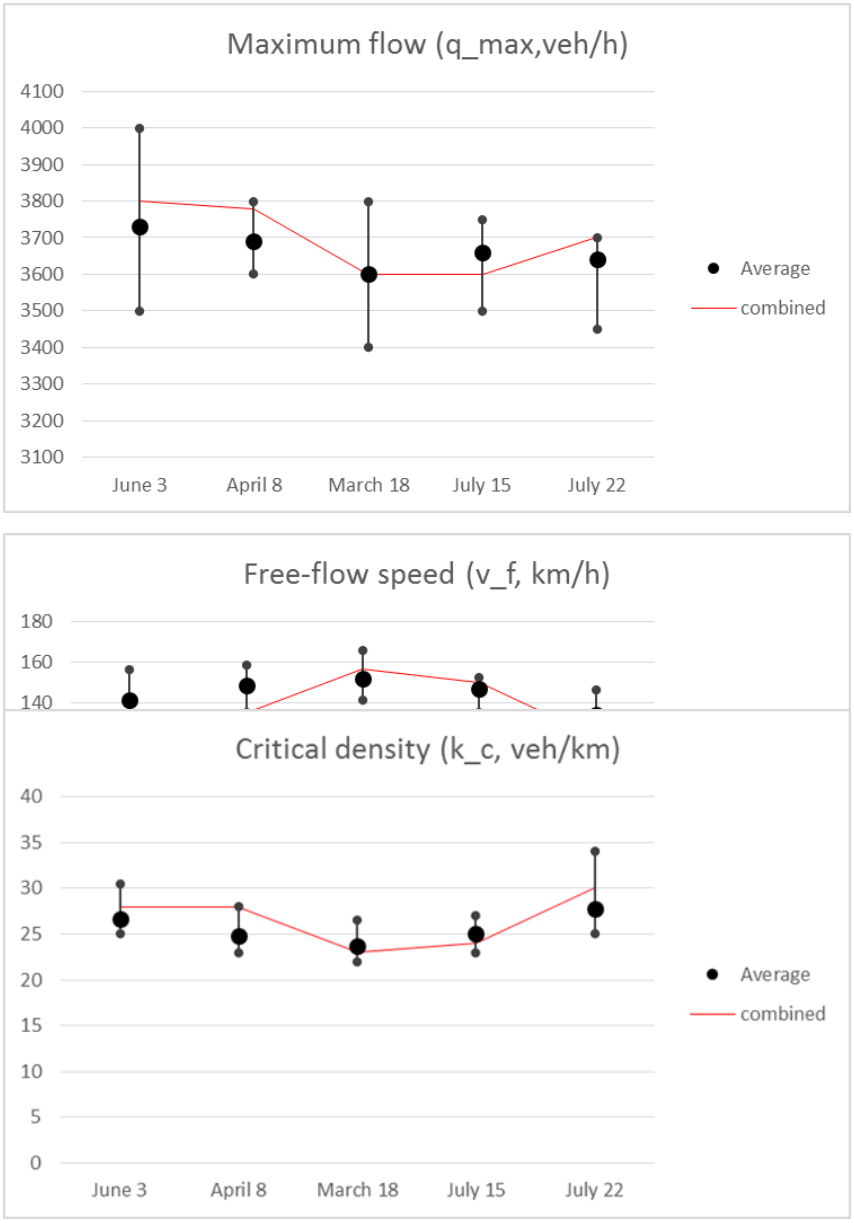


Figure 3. 9 Space-time diagram of speed of route 4 segments 1 (detector 62 to 54) and 2 (detector 48 to 39) during 12:00-12:01 to 17:59-18:00

In order to investigate the variation of FDs among each detectors in a segment, and to compare the day-to-day variations, the min, max, average plots of the parameters of FDs are generated (**Figure 3.1 and Appendix I**). The parameters are tabulated in the following tables as well and the individual FDs are enclosed in the **Appendix I**. From the plot, it was found that the parameters change from detector to detector and from day to day. This plot is from five Tuesdays in the study segment. The inter-detector variations: maximum flow varied between 4000 to 3400 veh/h, free-flow speed 166 to 109 km/h, critical density 34 to 22veh/km, jam density 180 to 109 veh/km, congested speed 24 to 43 km/h. The day-to-day variations were also considerable (**Figure 3.10**) maximum flow varied between 4000 to 3700 veh/h, minimum flow varied between 3400 to 3600 veh/h; maximum free-flow speed varied between 166 to 146 km/h, minimum free-flow speed varied between 141 to 109 km/h; maximum critical density varies between 34 to 27veh/km, minimum critical density 22 to 25veh/km ; maximum jam density varied between 180 to 150 veh/km, minimum jam density varied between 125 to 109 veh/km; maximum congested speed

varied between 43 to 35 km/h, minimum congested speed varied between 24 to 27 km/h. The value of free-flow speed and the back-wave speed of some fundamental diagrams (FDs) were higher than the usual range. For example, the free-flow speed was found to be 154km/h on March 18 near detector number 40 (Table 3 in Annex I) and back-wave speed was found to be 43km/h on June 3 and April 8 at detector 48 (Table 1, 2 in Annex I). The reason for this could be unusual speed of some vehicles and also a sudden change in the traffic flow due to sudden release of congestion, which later on contributed in the formation of a steady flow at those particular detector regions. However, the average values of the parameters more or less were similar to the FD built with the aggregated data of all the detectors in the segment as shown in the **Figure 3.10-**



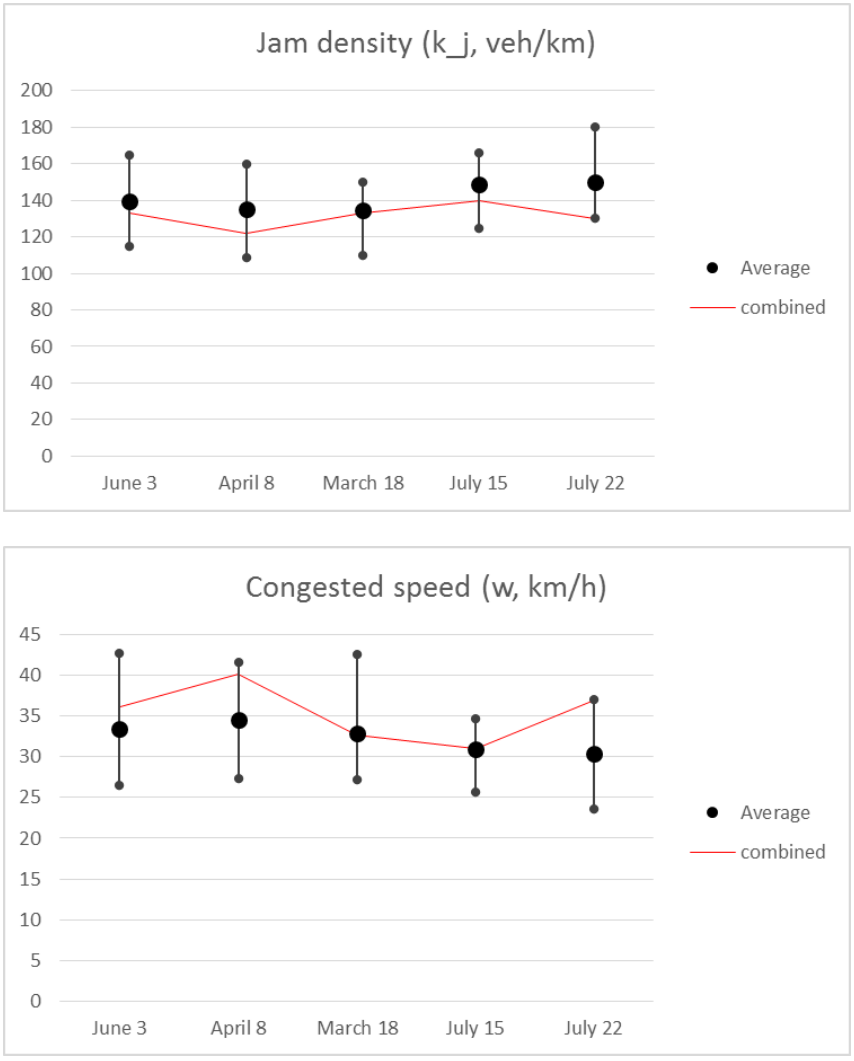
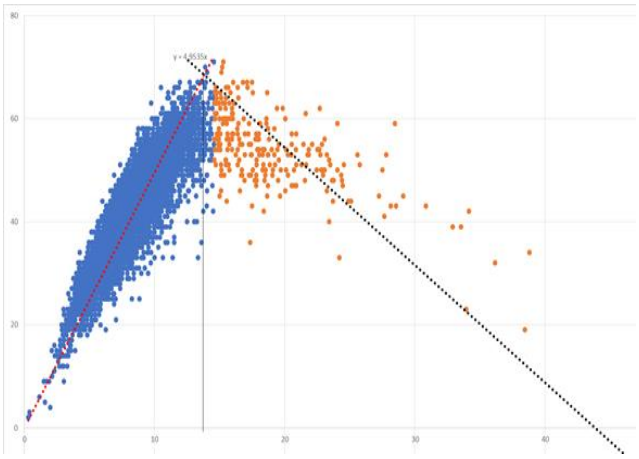


Figure 3. 10 Variations of FD parameters of each detectors in the study segment and on different days

For this thesis, the FDs were generated for 5 freeway segments of route 4, and FDs were investigated days of the week basis. For this thesis the aggregated (all detectors in a freeway segments) FD is constructed for the study segments based on days of the week.



Capacity (q_{max}), veh/hr	3780
free flow speed (v), km/h	135
Jam density (k_j), veh/km	120
back wave (w), km/h	41
Critical density, (k_c), veh/km	28

Figure 3. 11 Fundamental diagram (FD) for route 4 Shinjuku

3.5.1 Model construction and validation

The values of FD parameters are as follows: capacity (q_{max}) 3780 vehicles/h; free flow speed (V_F) 135 km/h; critical density (k_c) 28 vehicles/km; jam density (k_j) 120 vehicles/km; and back wave (w) 41 km/h. The cell length was kept at 240 m, the average inter-detector spacing for this study route, to generate uniformly distributed detectors. Additionally, the Q_{VSL} values were calculated for different V_{SL} values using the FD i.e.$V_{SL,}Q_{VSL}$ = {(10,750), (20,1350), (30,1830), (40,2220), (50,2520), (60,2880), (70,3060), (80,3240), (90,3420), (100,3600), (110,3720)}.

Validation of the CTM was done by comparing traffic flow parameters of the cells located closest to original detectors. To maintain the cell length to be approximately 150 m, the average inter-detector spacing for this route, a time interval of 4 seconds was chosen (cell length = free flow speed times time interval). As shown in **Figure 3.12**, the cells generated through the CTM are validated by comparing traffic data from the nearest original detector. For example, cell 1 and 2 comparing with detector data from detector number 62 and so on. The mean percentage error (MPE) defined by **Equation 3.4** was calculated to validate the data, where N = total number of data for six hours = number of detectors × time (min) x variables = 18 × 360 x 3= 19440. After cleaning the data, 18348 data were found. MPE is found to be 0.0825, 0.1667, and 0.0449 for flow, speed and occupancy respectively and an average MPE of about 10% was calculated.

1	2	3	4	5	6	7	8	9	10	11	12	13	14
62		61		60		59		58		56		55		54
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
48		47		46		45		44		43		42		41		40		39

Figure 3. 12 CTM simulated detector arrangement alongside original detector layout

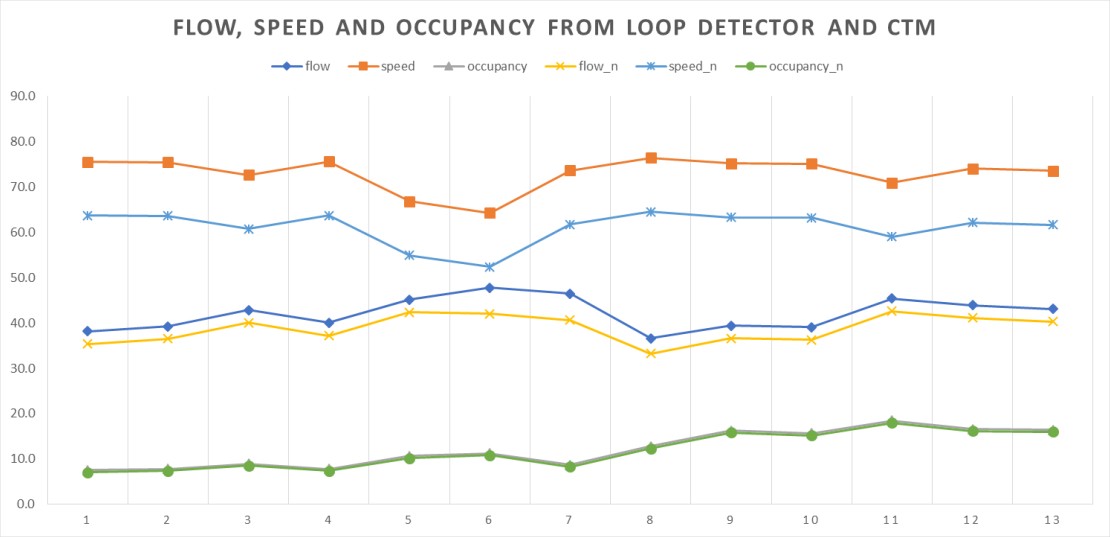


Figure 3. 83 Flow, speed and occupancy from loop detector and modified CTMs

The Courant–Friedrichs–Lewy or CFL condition is a condition for the stability of unstable numerical methods that model convection or wave phenomena. CFL condition states- the full numerical domain of dependence must contain the physical domain of dependence (Laney, 1998). It means, the distance that any information travels during a time-step within the cell must be lower than the distance between cell elements. The Courant number is a dimensionless quantity and can be stated as follows:

$$C = a\,\frac{\Delta t}{\Delta x} \tag{3.10}$$

Where $a$the magnitude of the velocity (length/time) is, Δt is the time step (time), Δx is the length interval (length). For any explicit simple linear convection problem, the Courant number must be equal or smaller than 1 (Courant, R.; Lewy, H.; Friedrichs, 1928).

The Courant number was observed for the speed values with given cell length (150 m) and time step (4 seconds). If the values are less than or equal to 1, the condition is considered met. From the Courant number for six hours speed data, it can be seen that the value is less than 1 and thus meets the CFL condition. The values are shown in the **Appendix I.**

3.6 Chapter conclusion

In this chapter, a method of dividing a route uniformly into same length cells using a macroscopic traffic model called the cell transmission model (Daganzo, 1994, 1995) was generated with the loop detector data. The idea behind this is to create a uniform layout of simulate detectors. This will give us more flexibility over controlling the data collection locations. Many times, due to instrumental fault or geographical difficulties, traffic data cannot be collected from every location a transport researcher would require. This macroscopic model could be of a solution to that issue. Moreover, over the years accident researchers have used different data collection methods and layouts, to create crash prediction models. This CTM-based model would provide a universal layout generation technique for future.

A traditional CTM model was constructed with route 3 detector data and the model was validated which was able to produce traffic data with an average MPE of 13%. This model will be used in chapter 4 for constructing RTCPMs. Furthermore, as the goal of this thesis is to propose an intervention along with crash prediction, a modified CTM is also introduced. This CTM incorporates the property of accommodating a flexible FD, which is used to incorporate VSL into the model. Generally, there is an optimization problem which is addressed while using VSL as an intervention for crash prevention. In this thesis, that will be conducted with a reinforcement learning (RL) agent in chapter 5. Hence, this modified CTM does not require to address the optimization into it. The modified CTM was constructed with the route 4 Shinjuku data for April 8, 2014 and was validated with the loop detector data which has showed an average MPE of about 10%. In chapter 5, it is checked how the model performs in conjunction with the RL agent.

3.7 Chapter References

Abdel-Aty, M., H. M. Hassan, M. Ahmed, and A. S. Al-Ghamdi. Real-time Prediction of Visibility Related Crashes. Transportation Research Part C, Vol. 24, 2012, pp. 288–298.

Bianco, L., G. Confessore, and M. Gentili. Combinatorial Aspects of the Sensor Location Problem. Annals of Operations Research, Vol. 144, No. 1, 2006, pp. 201–234.

Bianco, L., G. Confessore, and P. Reverberi. A Network Based Model for Traffic Sensor Location with Implications on O/D matrix Estimates. Transportation Science, Vol. 35, No. 1, 2001, pp. 50–60.

Bianco, L., R. Cerulli, and M. Gentili. New Resolution Approaches for the Sensor Location Problem. Presented at Tristan VI Symposium, Phuket Island, Thailand, 2007.

Chow, A. H. F., G. Gomes, A. A. Kurzhanskiy, and P. Varaiya. AURORA RNM – A Macroscopic Simulation Tool for Arterial Traffic Modeling and Control. PATH Technical Note, Institute of Transportation Studies, University of California, Berkeley. 2009.

Coifman, B. Using Dual Loop Speed Traps to Identify Detector Errors, TRR, 2014.

Daganzo, C. F. Fundamentals of transportation and traffic operations. Oxford: Pergamon, 1997.

Daganzo, C. F. The Cell Transmission Model, Part II: Network Traffic. Transportation Research Part B, Vol. 29, 1995, pp. 79–93.

Daganzo, C. F. The Cell Transmission Model: A Dynamic Representation of Highway Traffic Consistent With the Hydrodynamic Theory. Transportation Research Part B, Vol. 28, 1994, pp. 269–287.

Dervisoglu, G., G. Gomes, J. Kwon, R. Horowitz, and P. Varaiya. Automatic Calibration of the Fundamental Diagram and Empirical Observations on Capacity. Presented at the Transportation Research Board 88th Annual Meeting, 2009.

Hong, Z., and D. Fukuda. Effects of Traffic Sensor Location on Traffic State Estimation. 15th Meeting of the EURO Working Group on Transportation, Procedia - Social and Behavioral Sciences, Vol. 54, 2012, pp. 1186–1196.

Hossain, M., and Y. Muromachi. A Bayesian Network Based Framework for Real-time Crash Prediction on the Basic Freeway Segments of Urban Expressways. Accident Analysis and Prevention, Vol. 45, 2011, pp. 373–381

Hossain, M., and Y. Muromachi. A Real-time Crash Prediction Model for the Ramp Vicinities of Urban Expressway. IATSS Research, Vol. 37, No. 1, 2013, pp. 68–79.

Hossain, M., and Y. Muromachi. Optimum Detector Spacing for Real-Time Monitoring of Hazardous Locations on Urban Expressways. Japanese Society of Civil Engineers, Vol. 27, No. 5, 2010, pp. 1045–1054.

Kurzhanskiy, A. A., and P. Varaiya. Active Traffic Management on Road Networks: A Macroscopic Approach. Philosophical Transactions of the Royal Society A, Vol. 368, 2010, pp. 4607–4626.

Lee, C., B. Hellinga, and F. Saccomanno. Real-time Crash Prediction Model for the Application to Crash Prevention in Freeway Traffic. Transportation Research Record: Journal of the Transportation Research Board, No. 1840, 2003, pp. 67–77.

Lee, C., M. Abdel-Aty, and L. Hsia. Potential Real-time Indicators of Sideswipe Crashes on Freeways. Transportation Research Record: Journal of the Transportation Research Board, No. 1953, 2006, pp. 41–49.

Lighthill, M., and G. Whitham. On kinematic waves II. A theory of traffic flow on long crowded roads. Proceedings Royal Society of London, Part A, Vol. 229, No. 1178, 1955. http://dx.doi.org/10.1098/rspa/1955.0089.

Lu, X., Kim, Z., Cao, M., Varaiya, P., Horowitz, R. Deliver a Set of Tools for Resolving Bad Inductive Loops and Correcting Bad Data, California PATH Research Report, 2010.

Mihajlovic, V., and M. Petkovic. Dynamic Bayesian Network: A State of Art. Doctoral Dissertation, University of Twente, the Netherlands, 2001.

Morrison, D., and S. Martonosi. Characteristics of Optimal Solutions to the Sensor Location Problem. Annals of Operations Research, Vol. 226, No. 1, 2014, pp. 463–478. https://doi.org/10.1007/s10479-014-1638-y.

Muñoz, L., Sun, X., Sun, D., Horowitz, R. and Alvarez, L. Traffic Density Estimation with Cell Transmission Model, Proceedings of the American Control Conference Denver, Colorado.2003.

Pande, A., and M. Abdel-Aty. A Freeway Safety Strategy for Advanced Proactive Traffic Management. Journal of Intelligent Transportation Systems, Vol. 9, No. 3, 2005, pp. 145–158.

Park, H., and A. Haghani. Real-time Prediction of Secondary Incident Occurrences using Vehicle Probe Data. Transportation Research Part C, Vol. 70, 2015, pp. 69–85.

Roy, A., R. Kobayashi, M. Hossain, and Y. Muromachi. Real-time Crash Prediction Model for Urban Expressway Using Dynamic Bayesian Network. Journal of Japan Society of Civil Engineers, Series D3, Vol. 72, No. 5, 2016, pp. 1331–1338.

Shi, Q., and M. Abdel-Aty. Big Data Applications in Real-time Traffic Operation and Safety Monitoring and Improvement on Urban Expressways. Transportation Research Part C, Vol. 58, 2015, pp. 380–394.

Sun, J. and J. Sun. Dynamic Bayesian Network Model for Real-time Crash Prediction Using Traffic Speed Conditions Data, Transportation Research Part C, Vol. 54, 2015, pp. 176–186.

Sun, J., and J. Sun. Real-time Crash Prediction on Urban Expressways: Identification of Key Variables and a Hybrid Support Vector Machine Model. IET Intelligent Transport Systems, Vol. 10, No. 5, 2016, pp. 331–337.

CHAPTER 4

REAL-TIME CRASH PREDICTION MODEL (RTCPM)

4.1 Introduction

Predicting crash likelihood in real-time is a relatively much newer concept as compared to the conventional crash prediction models. Even the inspiration to build RCPMs in some way was derived from the willingness to improve the existing crash prediction models. However, this thesis takes a stand in suggesting that these two are very different types of models and has their own non-overlapping utility in road safety. For this, the distinctions between real-time and non- real-time crash prediction models were analyzed based on three broad categories – contextual issues, policy and practice related issues, methodological issues. The contextual issues are pertaining to the notion behind conducting the study, i.e., purpose and objectives. Policy and practice related issues refer to how the outcome of the models could be used in practice as well as to formulate policies. The methodological issues discuss how the model building processes differ. Due to their high interdependency, the contextual and policy and practice related issues are merged together in the following discussion.

4.2 Theoretical background

4.2.1 Bayesian Network

Bayesian Belief Net, also known as Bayesian Network, is a probabilistic graphical modeling method where we represent a system with a graph and a joint probability distribution compacted with the notion of conditional independence. Later, we can use this model of system to understand the dynamics within the system and also to predict the state of variables in lights of the evidence on any one or more variables. **Figure 4.1(a)** presents a simple BBN involving five variables. Here, each variable is represented with a node and the influence of one variable on others is demonstrated with directed edges (may or may not represent causality). We would like to mention here that these graphs are acyclic in nature and are called acyclic directed graph (DAG). The nodes from which edges generate are parents to the child nodes where the directed edge ends. In case of prediction modeling, the variable which we are predicting is called the "outcome variable" and the rest are called "decision variables". Now, as per **Figure 4.1 (a)**, A is a parent to B, B is a parent to D and

C is a parent to B, D and E. Nodes D and E are parents to none. Now, regarding the joint probability distribution, if a BBN has a universe of variable *U = {A1, …., An}* then using the chain rule of probability, its joint probability distribution can be presented as:

$$\mathrm{P(U)} = \prod_{i=1}^{n} P(A_i | A_1 \dots A_{n-1}) \quad (4.1)$$

Now, BBN suggests that assuming conditional independence of the variables, Eq. 1 can be compacted as:

$$\mathrm{P(U)} = \prod_{i=1}^{n} P(A_i | Pa(A_i)) \quad (4.2)$$

Where $Pa(A_i)$ is the set of parents of A_i. This substantially reduces the size of the conditional probability tables (CPTs) of different nodes. Now, if we have evidence $e_1 \dots \dots . e_m$ on m number of variables out of n (n > m), then we can re-write **Equation 4.2** as:

$$\mathrm{P(U, e)} = \prod_{i=1}^{n} P(A_i | Pa(A_i)) \prod_{j=1}^{m} e_j \quad (4.3)$$

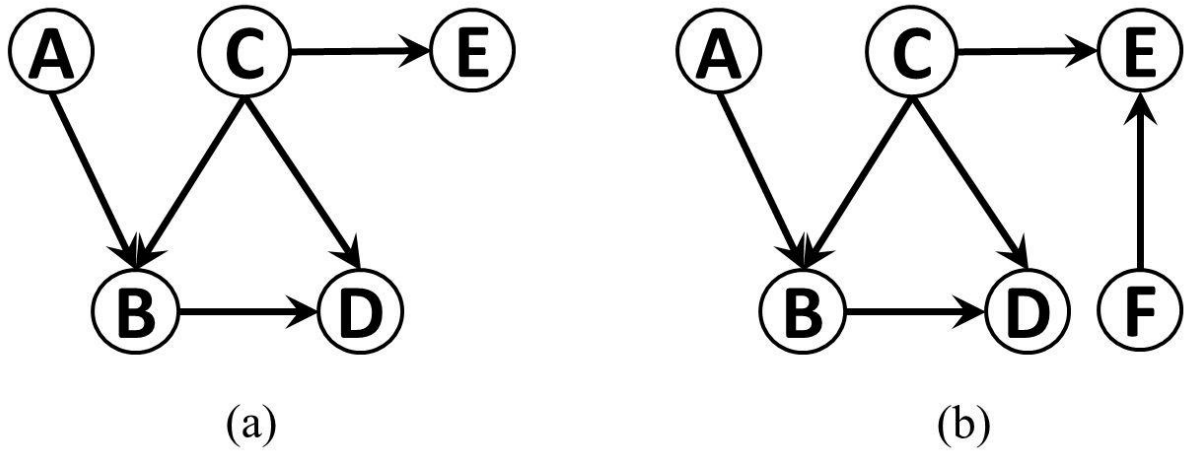


Figure 4. 1 An example Bayesian Belief Net

At this point, the probability of any variable A from the universal variable space U can be calculated by marginalizing *P(U,e)* as follows:

$$P(A|e) = \frac{\sum_{U/e} P(U,e)}{P(e)} = \frac{\sum_{U/A} P(U,e)}{P(A,e)} \quad (4.4)$$

Hence, our BBN in **Figure 4.1** can be written as:

$$P(A,B,C,D,E) = P(A)P(B|A,C)P(C)P(D|B,C)P(E|C) \quad (4.5)$$

Now, assume that a new variable F gets introduced as presented with **Figure 4.1(b).** Then the model equation can be updated as:

$$P(A,B,C,D,E,F) = P(A)P(B|A,C)P(C)P(D|B,C)P(E|C,F)P(F) \quad (4.6)$$

We can observe here that the addition of a new variable could be accommodated only by partially updating the existing model keeping most of it almost unchanged.

4.2.2 Structural Learning: NPC-Algorithm

When the interaction among variables within a problem domain is not known, we can employ structural learning algorithms to determine the direction of the edges in a BBN. In order to understand NPC-Algorithm it is important to understand its predecessor PC-Algorithm. Both the algorithms explore the data to come up with a set of conditional dependence and independence relationship through statistical tests and then based on that they decide the direction of the edges. In PC-Algorithm, at first, the conditional independence of all pairs of variables in the BBN is evaluated through statistical tests. Then an undirected graph is drawn by adding connections between those pairs of variables for which no conditional independences were found. Next, the pairs of links which meet at a node (also known as colliders) are identified in such a way that the BBN remains a DAG. Afterwards, using the colliders and positively identified conditional independences, directions of undirected edges are established. Lastly, for those links which still remains undirected, their directions are randomly assigned ensuring that the BBN remains a DAG. PC-algorithm performs well when there is no limitation in sample size. NPC-Algorithm overcomes this weakness of the PC-algorithm by adding an extra constrain of existence of a path:

"the necessary path condition for the absence of an edge says that in order for two variables X and Y to be independent (in a DAG faithful data set) conditional on a minimal set SXY, there must exist a path between X and every Z $\in$SXY (not crossing Y) and between Y and every Z $\in$SXY (not crossing X), For those edges where decisive directions cannot be found (also known as ambiguous zone), the researchers can use their expert opinion to choose those direction.

4.2.3 Batch Learning (EM-Algorithm) and Sequential Learning (Adaptation Algorithm)

The task of EM-Algorithm in BBN is to determine the conditional probability tables (CPTs) for nodes based on prior probabilities and availability of new N number of records. The algorithm has two steps – calculating the expected sufficient statistics and then maximizing its likelihood. To elaborate more, if no probability is assigned to a variable for which we are estimating the parameters, a uniform distribution is assumed. Then, with presence of a batch of data, the new parameter is estimated in such way that first, the expected sufficient statistics under that parameter is calculated and then the log-likelihood of that parameter under the expected sufficient statistics is maximized. This is an iterative process and it stops when one of these two criteria are satisfied –i) the maximum number of iteration specified by the user has exceeded, or, ii) the relative log-likelihood between two successive iterations is smaller than the preset minimum difference value. Here, it is important to mention that the EM-algorithm does not need data on each of the variables to update the model. The adaptation algorithm (Lauritzen, 1995) is similar to the EM-algorithm with the exception that here the evidence from each record is propagated throughout the network and the parameters for each of the variables are updated accordingly.

4.3 Dynamic Bayesian Network

One of the evolving areas that would certainly occupy computer scientists in the next decade is concerned with building software, which will be able to make conclusions based on information gathered from various sources. Interesting path to the solution of this problem would be to simulate reasoning process of humans, based on their ability to sense the environment in multiple ways and to integrate this sensed information in one global picture of the environment.

Human beings as well as the other animals integrate observations received from multiple senses to comprehend the environment and to take proper actions. Probability theory, thus, with its inherent notions of uncertainty and confidence had found widespread popularity in the multisensory fusion community. Various researchers had proposed many different probabilistic models for this purpose.

In this thesis, focus will be given in BNs and their extensions that try to unify temporal dimension with uncertainty. It starts with the simple concepts and then introduce static Bayesian networks, as well as basics of dynamic Bayesian networks as powerful tools for representing such uncertain events. Different levels in creating DBNs are presented.

Most of the events that we meet in our everyday life are not detected based on a particular point in time, but they can be described through a multiple state of observations that yield a judgment of one complete final event. Statisticians have developed numerous methods for reasoning about temporal relationships among different entities in the world. This field is generally known as time-series analysis. According to (Adhikari et al, 2013) time-series is a sample realization of a stochastic process consisting of a set of observations made sequentially over time.

Time is also an important dimension in the field of AI and reasoning. However, BNs do not provide direct mechanism for representing temporal dependencies. In attempting to add temporal dimension into the BN model various approaches has been suggested. Frequent names used to describe this new dimension in BN models are "temporal" and "dynamic". However, the difference between these models and their denomination cannot uniquely point to one typical model. Sterritt et al. tried to distinguish these categories in the manner that would be described below.

Dynamic Bayesian Networks (DBN) should be a name of a model that describes a system that is dynamically changing or evolving over time. This model will enable users to monitor and update the system as time proceeds and even predict further behavior of the system. In such models word dynamic is connected with a "motive force". Changing the nature of the static BN to model "motive forces" can then be thought of as adapting it to dynamic model. Although every system that changes its state involve time, authors differentiate between the two terms dynamic and temporal. In temporal models explicitly model time as continuous permanent category as opposed to other changes in the system such as the change in state or a system. Hence, temporal models would be a sub-class of dynamic. If every time slice of a temporal model corresponds to one particular state of a system, and if the movement between the slices reflects a change in state instead or time, in most cases that model is classified as a dynamic model.

According to the same authors, considering time representation, temporal approaches could be classified into two main categories, namely those models, which represent time as points

(instances) or as time intervals. However time intervals can be thought of as a set of consecutive time points. Therefore, time-point representation seems to be more appropriate and more expressive.

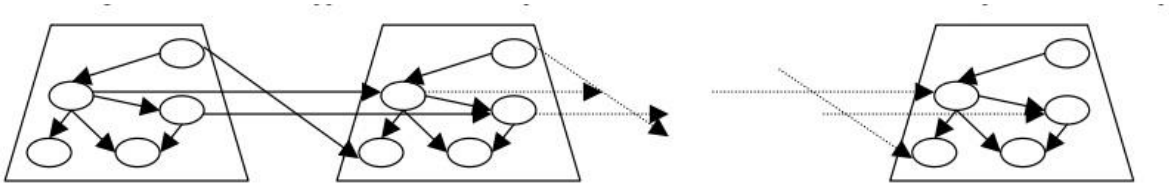


Figure 4. 2 Diagram showing approach where time slice is used to present a snapshot of the evolving temporal process

Figure 4.2 represents an approach where time slice is used to represent snapshot of the evolving temporal process. It can be said that the belief network consists of a sequence of sub-models each representing the system at a particular point or interval in time (time slice). These time slices are interconnected by temporal relations, which are represented by the arcs joining particular variables from two consecutive time slices.

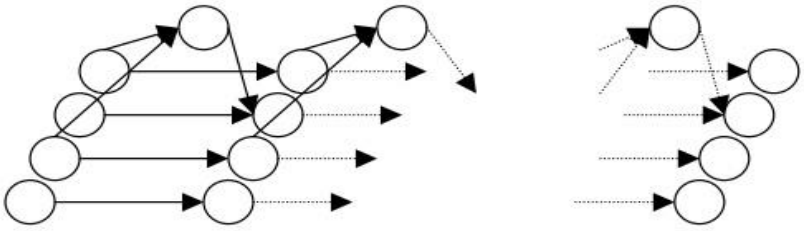


Figure 4. 3 Temporal model with duplicated time slices over time

Figure 4.3 represents another model where the network is composed of identical sub-models duplicated over each time slice. This means that it has the same temporal structure as previous model. However, links between state variables within a time slice are here disallowed.

Dynamic Bayesian Networks are usually defined as special case or singly connected Bayesian Networks specifically aimed at time series modeling like stated in previous section. All the nodes, edges and probabilities that form static interpretation of a system is identical to BN variables here can be denoted as the state of a DBN, because they include a temporal dimension. The states or any system described as a DBN satisfy the Markovian condition that is defined as follows: The state of a system at time t depends only on its immediate past. i.e. its state at time t-1. Also, this

property is frequently considered as a definition of First order Markov property: the future is independent of the past given present.

Now, as we can see, the BN model is expanded, we can allow not only connections within time slice known as intra-slice connections, but also the one between time slices. These temporal connections incorporate condition probabilities between variables from different time slices. The transition matrix that represents these time dependencies is often called a Conditional Probability Table (CPT), since it represents the CPD in tabular form, Intra-slice CPDs can also be represented by CPTs, i.e. in tabular form.

The states of a dynamic model do not need to be directly observable. They may influence some other variables that we can directly measure or calculate. Also, the state or some system needs not to be a unique, simple state. It may be regarded as a complex structure of interacting states. Each state in a dynamic model at one time instance may depend on one or more states at the previous time instance or/and on some states in the same time instance. It was shown that complex structures like this could also be represented as DBNs. So, generally, in DBN states of a system at time, t may depend on systems states at time t-1 and possibly on current states of some other nodes in the fragment of DBN structure that represents variables at time t.

We can describe DBN saying that it consists of probability distribution function on the sequence of T hidden-state variables $X = \{X_0, \ldots, X_{T-1}\}$ and the sequence of T observable variables $Y = \{Y_0 \ldots, X_{T-1}\}$, where T is the time boundary for the given event we are investigating. This can be expressed by the following term:

$$\Pr(X, Y) = \prod_{t-1}^{T-1} \Pr(x_t \mid x_{t-1}) \prod_{t-0}^{T-1} \Pr(y_t \mid x_t) \,.\, \Pr(x_0) \tag{4.7}$$

In order to completely specify a DBN we need to define three sets of parameters:

• State transition pdfs $Pr(X_T \mid X_{T-1})$ that specifies time dependencies between the states

• Observation pdfs $Pr(Y_T|X_T)$ that specifies dependencies of observation nodes regarding to other nodes at time t and

• Initial state distribution $Pr(X_0)$ that brings initial probability distribution in the beginning of the process.

First two parameters had to be determined for all states in all time slices t= 1....T. There is a possibility that conditional pdfs can depend on time instance, that is to be time-varying ($Pr(X_T|X_{T-1}) = Pr(X_T|X_{T-1}, t)$), or time invariant. Time invariant conditional pdfs can be parametric ($Pr(X_T|X_{T-1}) = Pr(X_T|X_{T-1}, \theta)$), or nonparametric, when they are described using probability tables (CPTs). Depending on the type of the state space or hidden and observable variables, a DBN can be discrete, continuous, or combination of these two.

Similarly, as we propose in static BNs, in DBNs we may be interested in the following tasks:

1. Inference: estimate the pdf of unknown states given some known observations, and initial probability distribution.

2. Decoding: find the best-fitting probability values for sequence or hidden states that have generated the known sequence or observations.

3. Learning: given a number of sequences of observations, estimate parameters of a DBN such that they best fit to the observed data, and make the best model for the system.

4. Pruning: distinguishing which nodes are semantically important for inference in DBN structure, and which are not, and removing them from the network.

Within every time slice it is possible to identify a subset of the nodes that describes current state of the world (environment), denoted as $W_1(t) \ldots . W_q(t)$, which represents either the entire world state, or the part we want to inspect. We call these the designated world nodes. These nodes are chosen in the way that if their states are known, then nodes V(t_k). where $t_k < t$, are no longer relevant to the overall goal of the inference. If for every i in [1 ... ... q] there is an 's' such that (Pr($W_t(t) = s_i$) = 1 , then the general pruning action is as follows: (I) delete all nodes V(t_k) where $t_k < t$, and (2) explicitly incorporate knowledge that W, (t) = s,. We explicitly

incorporate the information that $W_i(t) = s_i$ is known to be in state s_i , by deleting all states except for state s_i, and in that way, reducing the state space. Also, if a node V(t) has no successors and if $\Pr(V(t) = s) = 0$, then we can delete state 's'.

In a special case, when the node V(T) has no predecessors, its state is known to be s_i and its other states $s_j \neq s_i$, are deleted, the conditional probability tables of its successor nodes must be updated to reflect this. Now, there may exist states that are impossible. All such states must then be deleted, and the pruning procedure performed recursively on successors.

These are just some basic pruning procedures, and they are not sufficient for controlling the inference complexity. There are also some marginal pruning procedures. For example, if there is no initialization information, no pruning will be performed. Even with initialization information, if a DBN models sensors failure there is always a small chance that the data is incorrect.

The basis of making a pruning decision is the tradeoff between the savings on execution of the inference versus the likelihood of making an error.

4.3.1 Methodology

In this study, the hidden state variables are the crash likelihood and the observation variables are the traffic flow variables (i.e. flow, speed, occupancy, difference between upstream and downstream flow, speed, occupancy, etc.). Thus, the state transition could be denoted as Pr (crash|previous crash).

Traffic data was collected from the nearest upstream and downstream loop detectors of the crash location. In the case of CTM-based RTCPMs, for each crash case, one the segment of the road was selected where both upstream and downstream detectors were present and traffic flow data was extracted from the closest upstream and downstream cells from the crash location. RTCPMs are highly dependent on the information variables chosen. In previous studies (Mihajlovic et al, 2001; Sun et al, 2015; Roy et al, 2016) BN and DBN models were generated using various combinations of traffic flow variables, i.e. flow, speed, occupancy, etc., along with their descriptive statistics. Existing literature also suggests that the overall performance of DBN models built with a combination of traffic parameters are better than the models with only one traffic parameter (Lee et al, 2006; Roy et al, 2016). From these studies, it was established that the information variables

such as downstream speed, occupancy, difference of upstream and downstream speed occupancy, and upstream flow are suitable predictors to develop BN and DBN models. In this study, 16 different combinations of 6 base variables and 3 relative variables were produced (**Table 4.1**) to construct BN and DBN models to compare their performances in terms of sensitivity, specificity etc. to identify the best possible combination of the variables.

Table 4. 1 List of BN and DBN Models

Description of RTCPMs (both BN and DBN)	
Model 1	Downstream flow, speed, occupancy, difference of upstream and downstream flow
Model 2	Upstream flow, speed, occupancy, difference of upstream and downstream flow
Model 3	Downstream flow, speed, occupancy, difference of upstream and downstream occupancy
Model 4	Upstream flow, speed, occupancy, difference of upstream and downstream occupancy
Model 5	Downstream flow, speed, occupancy, difference of upstream and downstream speed
Model 6	Upstream flow, speed, occupancy, difference of upstream and downstream speed
Model 7	Difference of upstream and downstream flow, speed, occupancy, downstream occupancy
Model 8	Difference of upstream and downstream flow, speed, occupancy, upstream occupancy
Model 9	Difference of upstream and downstream flow, speed, occupancy, downstream flow
Model 10	Difference of upstream and downstream flow, speed, occupancy, upstream flow
Model 11	Difference of upstream and downstream flow, speed, occupancy, downstream speed
Model 12	Downstream flow, speed, occupancy
Model 13	Upstream flow, speed, occupancy
Model 14	Downstream speed, occupancy, difference of upstream and downstream flow
Model 15	Upstream speed, occupancy, difference of upstream and downstream flow
Model 16	Difference of upstream and downstream flow, speed, occupancy, upstream speed

Models 1, 3, and 5 consist of all downstream variable and one relative variable assuming that the downstream traffic condition is more reliable to demonstrate a crash prone situation. Models 2, 4, and 6 include the upstream variables along with a relative variable that deals with the changes in traffic upstream of the incident. Models 7–11 basically account for all the relative traffic parameters intending to detect the crash prone situation from only the longitudinal changes of traffic. Models 12 and 13 consist of only the base variables whereas models 14–16 are developed with different combinations of base and relative traffic variables to investigate the performance of models in the case of missing predictors.

In this manuscript, 30 crash cases and corresponding 748 normal cases were extracted to construct BN and DBN based RTCPMs. The objective of such models is to maintain a high detection rate for the detection of crash prone traffic conditions but at the same time maintain low false alarm. In order to construct the model, first, the raw data was screened for faulty detectors and the accurate crash time was estimated from the reported crash time by investigating the upstream and downstream detector's traffic data for each crash. Since the traffic data is recorded every minute, the variation in traffic flow variables were evident and it was possible to locate the actual time of each crash. Then, for crash cases, data was collected for 1 minute just before the incident occurred. For the corresponding normal data, traffic flow data was collected from the same detectors and cells during the same time as crash cases, but for those days when no incident took place. For example, if a crash had occurred on Wednesday March 3 at 15:00, then traffic flow data on March 3 at 14:59 was counted as crash data and traffic flow data from all other Wednesdays at 14:59 was considered normal data. In addition, for a DBN model with three time slices, traffic data was extracted at 14:59, 14:58, and 14:57 timestamps as 1, 2, and 3 minutes before crash, respectively. Traffic data was collected during a six-month period. Thus, for a single crash case there are approximately (5 months × 4 days + 1 month × 3 days = 23 days × 33 crash cases =) 759 days of normal data was found; of which 748 were selected after removing erroneous data. This data was divided into training (20 crash and corresponding 496 normal) and test (10 crash and corresponding 252 normal) data for RTCPM building and validation purposes, respectively. Several studies conducted by Abdel-Aty et al. (2012), Sun and Sun (2016), and Lee et al. (2003) used both crash and normal traffic data to construct RTCPMs in a similar manner. A total of 16 × 2 × 2 = 64 BN- and DBN-based RTCPMs were generated by employing existing loop detectors and the simulated CTM data. Performances based on accuracy of crash detection and false alarm rate were calculated using validation data.

4.3.2 ANALYSIS, RESULTS, AND DISCUSSION

In chapter 3, the procedure of constructing CTM and validation is explained in details. In this chapter the RTCPMs with CTM and loop detector data will be explained. To employ the CTM to the entire route, a FD of traffic was generated using a flow–density graph. If a detector with poor data was found, the next closest detector data was used for analysis.

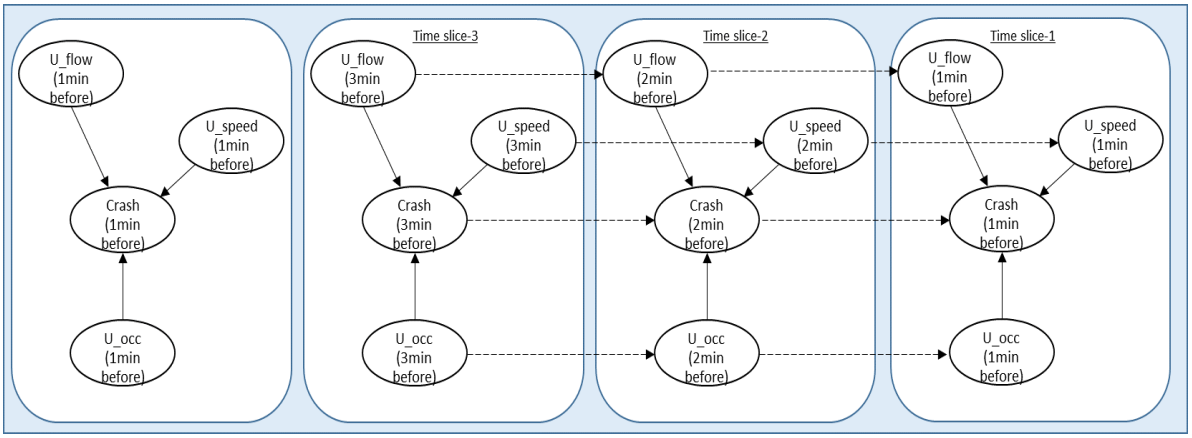


Figure 4. 4 BN model structure (left) and DBN model structure with three time slices (right)

The values of FD parameters ranged as follows: capacity (q_{max}) 1800–3900 vehicles/h; free flow speed (v) 66.92–139.2 km/h; critical density (k_c) 22.6–31.52 vehicles/km; jam density (k_j) 75–130 vehicles/km; and back wave (w) 33.6–40.3 km/h. The cell length was kept at 240 m, the average inter-detector spacing for this study route, to generate uniformly distributed detectors. Validation of the CTM was done by comparing traffic flow parameters of the cells located closest to each of the 35 detectors.

To generate traffic flow data for crash prediction analysis. While generating BN and DBN models, CTMs were reproduced separately for each road segment corresponding to a crash for each day, thus the FDs for separate detectors were different based on their locations and days of data collection. A BN and corresponding DBN model structure of model 13 is shown in **Figure 4.4.**

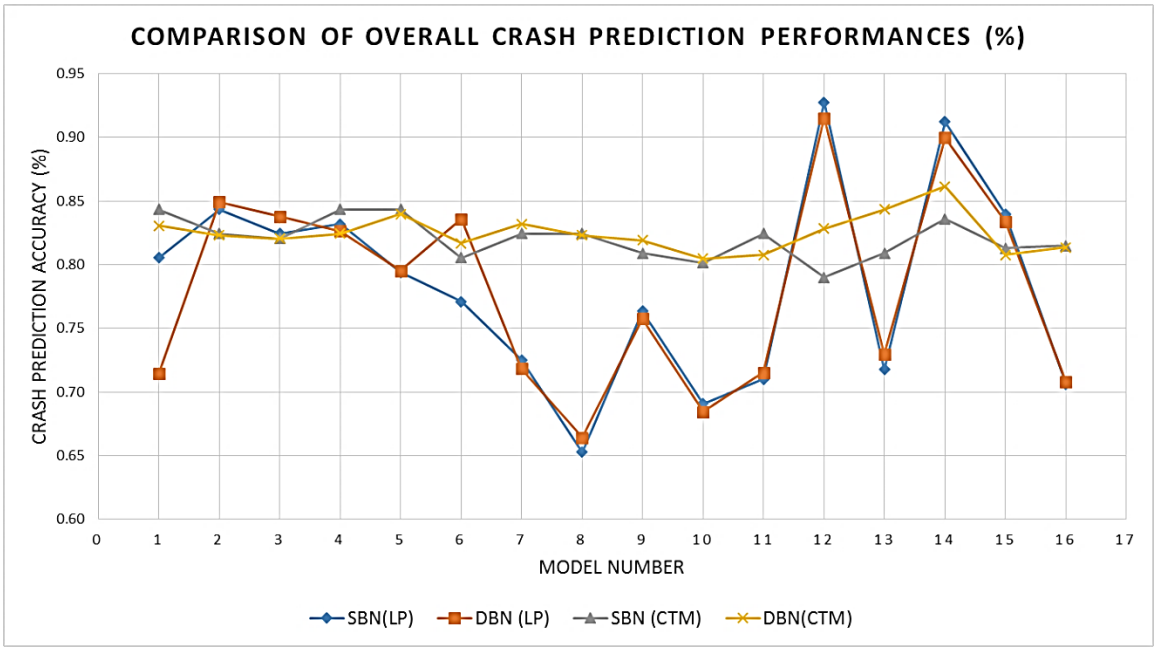


Figure 4. 5 Comparison of overall crash prediction performances (%).FIGURE 4.5 Comparison of overall crash prediction performances (%).

Separate test data was used to validate all (16 × 4) BN and DBN models built with both loop detectors and the CTM data for different threshold values. Crash prediction accuracy and sensitivity–specificity were calculated for each model (**Table 4.2). Figure 4.5** shows the comparison of overall accuracy of prediction of crash likelihood of DBN- and BN-based models built with the loop detectors and the CTM data. All the models can predict with an overall accuracy over 65%. In the case of loop detectors, both BN- and DBN-based models perform well and 10 BN models showed better prediction accuracy than the DBN models. Similar results are seen in the case of models with the CTM data. Since the cells are shorter in length (150 m) than the average loop detector spacing, this result shows that the cells can generate traffic flow data that can be used for RTCPMs. In addition, in situations of faulty detectors, the CTM can be used to produce the traffic data from the vicinity of that incident.

Table 4. 2 Performance Evaluation of BN- and DBN-based Real-time Crash Prediction Models with Loop Detectors and CTM Data

Modeling method		BN				DBN			
Data Type	Model Number	Sensitivity (%)	Specificity (%)	Accuracy (%)	False alarm (%)	Sensitivity (%)	Specificity (%)	Accuracy (%)	False alarm (%)
Loop Detector Data (LP)	1	0.50	0.82	0.81	0.18	0.30	0.73	0.71	0.27
	2	0.60	0.85	0.84	0.15	0.20	0.88	0.85	0.12
	3	0.20	0.85	0.82	0.15	0.40	0.86	0.84	0.14
	4	0.40	0.85	0.83	0.15	0.40	0.84	0.83	0.16
	5	0.20	0.82	0.79	0.18	0.20	0.82	0.80	0.18
	6	0.50	0.78	0.77	0.22	0.40	0.85	0.84	0.15
	7	0.60	0.73	0.73	0.27	0.20	0.74	0.72	0.26
	8	0.70	0.65	0.65	0.35	0.20	0.68	0.66	0.32
	9	0.60	0.77	0.76	0.23	0.40	0.77	0.76	0.23
	10	0.70	0.69	0.69	0.31	0.30	0.70	0.68	0.30
	11	0.60	0.71	0.71	0.29	0.20	0.74	0.72	0.26
	12	0.10	0.96	0.93	0.04	0.20	0.94	0.92	0.06
	13	0.20	0.74	0.72	0.26	0.10	0.76	0.73	0.24
	14	0.60	0.92	0.91	0.08	0.20	0.93	0.90	0.07
	15	0.70	0.85	0.84	0.15	0.30	0.86	0.83	0.14
	16	0.70	0.71	0.71	0.29	0.20	0.73	0.71	0.27
Cell transmission model data (CTM)	1	0.70	0.85	0.84	0.15	0.60	0.84	0.83	0.16
	2	0.60	0.83	0.82	0.17	0.60	0.83	0.82	0.17
	3	0.80	0.82	0.82	0.18	0.70	0.83	0.82	0.17
	4	0.80	0.85	0.84	0.15	0.60	0.83	0.82	0.17
	5	0.80	0.85	0.84	0.15	0.80	0.84	0.84	0.16
	6	0.60	0.81	0.81	0.19	0.60	0.83	0.82	0.17
	7	0.80	0.83	0.82	0.17	0.80	0.83	0.83	0.17
	8	0.80	0.83	0.82	0.17	0.80	0.82	0.82	0.18
	9	0.70	0.81	0.81	0.19	0.80	0.82	0.82	0.18
	10	0.40	0.82	0.80	0.18	0.60	0.81	0.80	0.19
	11	0.80	0.83	0.82	0.17	0.80	0.81	0.81	0.19
	12	0.20	0.81	0.79	0.19	1.00	0.82	0.83	0.18
	13	0.60	0.82	0.81	0.18	0.60	0.85	0.84	0.15
	14	0.80	0.84	0.84	0.16	1.00	0.86	0.86	0.14
	15	0.80	0.81	0.81	0.19	0.60	0.82	0.81	0.18
	16	0.44	0.84	0.81	0.16	0.44	0.84	0.81	0.16

From the overall accuracy of prediction of crash likelihood, loop-detector-based model 12 has the highest performance of 93% (BN) and 92% (DBN) with false alarm rates of 4% and 6%, respectively. In the case of CTM-based models, the overall accuracy of prediction of crash likelihood of model 14 was 84% (BN) and 86% (DBN) and with false alarm rates was 16% and 14%, respectively. The information variables of models 12 and 14 are: downstream flow, speed, occupancy, difference of upstream and downstream flow. In previous studies (Pande and Abdel-Aty, 2005; Morison et al, 2013,Roy et al, 2016; Hossain and Muromachi, 2013), downstream flow and speed and the relative flow value were found to play a significant role in RTCPM. Model 12 consists of downstream flow, speed, and occupancy, which implies that the change in traffic situation at the downstream of a crash location can detect an impending hazardous situation. On the other hand, information variables of model 14 explain hazardous situations with not only downstream speed and occupancy, but also with the sudden change of traffic flow. Compared with loop detectors, 11 out of 16 BN models using the CTM data showed on average 8% more overall accuracy of prediction of crash likelihood, whereas 9 out of 16 DBN models showed on average 10% more overall accuracy.

4.3.3 Transferability of the models

Four RTCPMs are chosen which includes downstream speed, downstream occupancy and difference of up and downstream flow as the hyperparameters because, from chapter 3, these parameters were found to be the most influential ones for predicting crash. Data was collected from Route 4 (the Shinjuku Expressway) to construct the RTCPM, and separate validation data from both Route 4 and Route 3 (the Shibuya Expressway) were employed to check the transferability of the model. the existing practice of real-time intervention design in the literature is to adopt one highly accepted RTCPM construction methodology and focus on the intervention planning and design, and the RTCPM employed for this thesis follows the literature (Hossain and Muromachi, 2012; Roy et al., 2018 [a, b]). In these studies, it was evident that BN-based RTCPMs possess properties of adaptability and can withstand data that are missing due to infrastructural failure. The adaptability of the models enables them to be transferable. In this study, the transferability was tested with data from Routes 3 and 4. As shown in **Table 4.3** and **Figures 4.7** and **4.8**, the overall performance and the predictive accuracy of the of four models built with route 4 data after validating separately with data from Routes 3 and 4. **Figures 4.7** and **4.8** show the

overall accuracy of crash likelihood predictions of the four models corresponding to four thresholds. The RTCPMs were built with route 4 data and validated with route 4 and 3 data respectively. In the figures route 4 means model validated with route 4 data, route 3 means model validated with route 3 data. It can be observed that the models are able to predict crash likelihood from the data for both routes with decent accuracy.

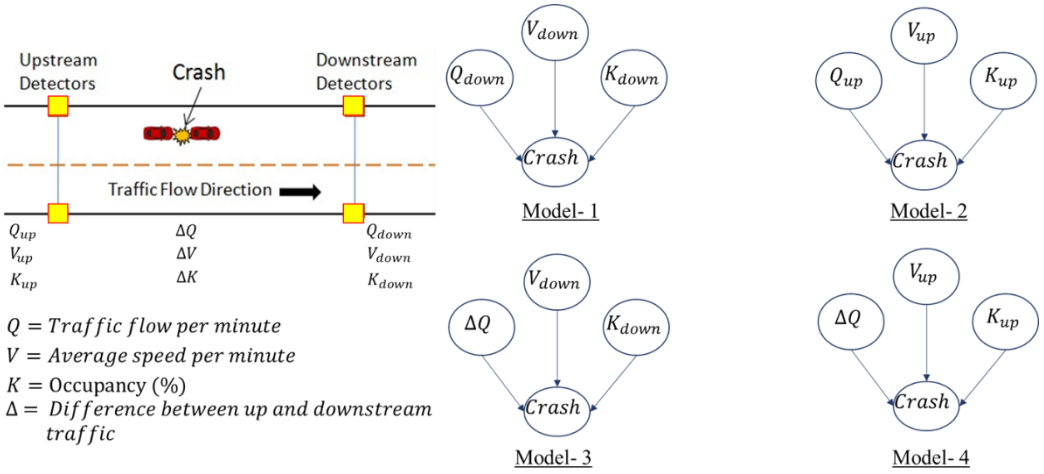


Figure 4. 6 Four BN-based RTCPMs for rash risk calculation

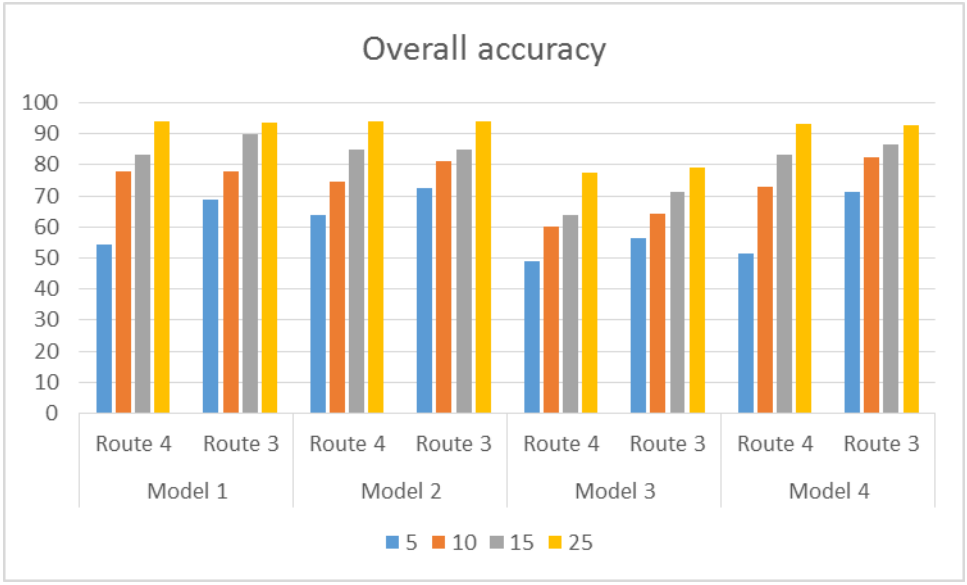


Figure 4. 7 Overall accuracy (%) of the four RTCPMs with respect to four thresholds (%).

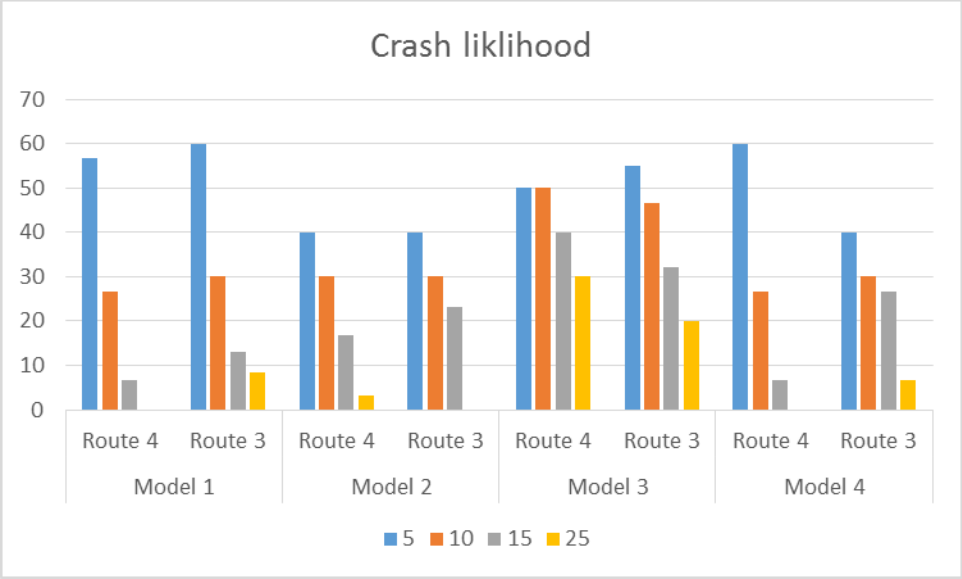


Figure 4. 8 Crash likelihood (%) of the four RTCPMs with respect to four thresholds (%).

Table 4.3 summarizes the performances of the four models in terms of overall accuracy and crash detection accuracy (crash likelihood) evaluated with respect to four threshold values – 5%, 10%, 15%, and 25% respectively.

The performance of the RTCPMs can be expressed in terms of prediction accuracy of crash likelihood and overall accuracy. These indices are explained below.

Table 4. 3 Confusion matrix

		Predicted	
Actual		Crash	No crash
	Crash	TP	FN
	No crash	FP	TN

In the confusion matrix,

TP= crashes predicted as crashes (T)

FP= normal cases predicted as crashes (F)

FN= crashes predicted as normal cases (F)

TN= normal cases predicted as normal cases (T)

The prediction accuracy of crash likelihood, or also known as true positive, or sensitivity is estimated in the following way-

$$\text{probability of crash likelihood (\%)} = \left(\frac{TP}{TP+FN}\right) \quad (4.8)$$

Which is the ratio of number of crashes predicted correctly and the actual total number of crashes.

On the other hand, the overall accuracy is defined as follows-

$$\text{Overall accuracy (\%)} = \left(\frac{TP+TN}{TP+FN+FP+TN}\right) \quad (4.9)$$

Which is the ratio of total number of crash and normal cases predicted correctly and the total number of crash and normal cases.

The confusion matrix for model-3 for threshold 10 is given below-

Table 4. 4 Confusion matrix for model 3

Threshold=10		Predicted	
Actual		Crash	No crash
	Crash	14	16
	No crash	103	439

The accuracy of crash likelihood decreases with the increase of threshold as the model tends to give more false negative values, which means, with the increase of threshold, the model categorizes actual crashes as normal conditions. Whereas, in case of overall accuracy, the models performance increases with increase of threshold. Because, overall accuracy represents the model's capacity to identify both crash and normal cases correctly. In this case, the model is able to identify more normal cases correctly with the increase of threshold, which means it gives lower false alarm. The accuracy of crash likelihood varied because the accuracies were estimated for threshold of 5, 10, 15 and 25 and the distribution of probability of crash likelihood among the crash cases were more concentrated near the value of 10% probability as that's approximately the average crash prediction probability of the model.

There are few other indices in addition to overall accuracy and true positive rate or, sensitivity, to indicate model's performance, such as misclassification rate: which explains overall, how often the model predicts wrong; false positive rate: it refers to when it's actually no crash, how often

does it predict yes; prevalence: it means how often does the crash condition actually occur in the sample, etc. Another way to show model's performance is by using an ROC curve.

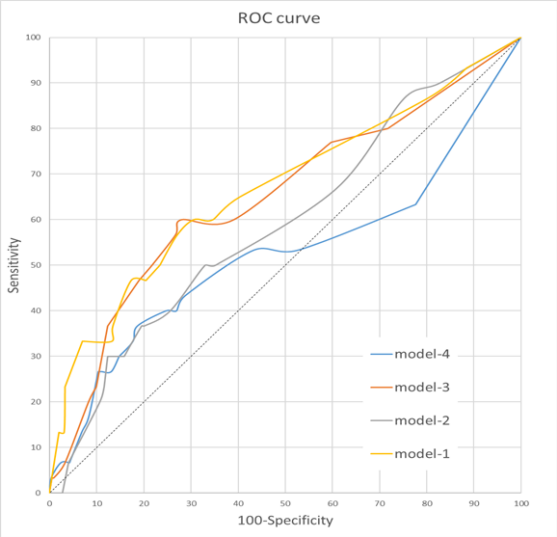


Figure 4. 9 ROC curve for model 1, 2 ,3 ,4 at different threhsolds.

A receiver operating characteristic curve (ROC) is shown below. The ROC curve is a sensitivity versus (1-specificity) curve. It demonstrates how much model is capable of distinguishing between correct and the wrong prediction by the model with respective to different threshold values. The higher is the area under the ROC curve, the better is the performance of the model. From the **Figure 4.9**, it can be seen that the area under ROC curve for model 3 and 1 are larger than model 2 and 4.

In the practical scenario, experts can set the threshold values based on their experience considering their priorities, available countermeasures to calm down the traffic condition and costs associated with not predicting a future crash and cost of false alarms due to misclassifying a normal traffic condition. They can even set different threshold value at different time of day.

It is clear that a lower threshold value will help the model predict most of the crashes at the cost of misclassifying normal situations causing a low overall accuracy and vice versa. All the models have exhibited overall accuracy over 49% with the highest being 94% (Model 2) which means that Model 2 is able to identify 94% of the crash and no-crash conditions correctly at a threshold of 25. Model 1 and 4 can classify most number of crashes accurately (60%) for a threshold of 5, which means that the model is able to identify 60% of the crashes correctly with the risk $\geq 5\%$.

Additionally, it can also be observed that the crash detection rate (%) decreases vastly with the increase in threshold in the case of Models 1, 2, and 4, whereas the value remained fairly consistent for Model 3. The observations from Figure 4.7 and 4.8 are-

- From the overall accuracy, it can be seen that the overall accuracy has increased with increase in thresholds for all four models.

- However, the accuracy of predicting crash likelihood decreased with increase in threshold for all four models, especially model -1, 2 and 4.

- Model-3 on the other hand showed a consistency in prediction accuracy and the value was kept over 20% for all thresholds.

- In case of transferability, even with validated by route 3 data, all four models showed overall accuracy at least over 49%. And in case of threshold 5, the overall accuracy was higher than model validated with route 4 for all the models, whereas at threshold 25, it was almost the same. For threshold 10 and 15, it was either higher or the same. From the prediction accuracy of crash likelihood, at threshold 5, model-1 and 3 performed better when validated with rote 3 data. At threshold 10, model-1 and 4 performed better; at threshold 15, model-1, 2 and 4 performed better; at threshold 25, model- 1 and 4 performed better. Hence, model- 1 was able to adopt the transferability the most. However, model-1's prediction accuracy is lower than other three models.

Hence, to summarize, model- 3 showed consistency in prediction accuracy of crash likelihood at all thresholds and both validation. From **Figure 4.8**, for model- 3, although the highest accuracy was found at threshold 5 and 10 (when validated with route 4), the overall accuracy of model-3 at threshold 5 (49%) was less compared to threshold 10 (60%). So, in the later chapters of this thesis, model-3 will be used as the RTCPM and crash risk threshold of 10% will be the threshold for intervention decision making.

From previous studies by Lee et al. (2003), Pande et al. (2005), Lu et al. (2010) and Roy et al. (2018) it was evident that an impending hazardous traffic condition can be detected with higher accuracy by those RTCPMs using as information variables the downstream flow, speed,

occupancy, and difference in traffic flows upstream and downstream of the crash location. These information variables proved to be more reliable in identifying hazardous situations. Model 3 has the information variables of downstream flow, speed, and relative flow, and showed a consistent performance in predicting crash likelihood compared to the other three models. Therefore, Model 3 is used in the later part of the study for evaluating the efficacy of RL-based VSL application.

Table 4. 4 Crash prediction accuracy for the RTCPMs

Crash Risk Threshold		Validation Routes	5	10	15	25
Model 1	Overall Accuracy (%)	Route 4	54.20	77.97	83.22	94.06
		Route 3	68.90	77.80	89.90	93.50
	Crash Likelihood (%)	Route 4	56.67	26.67	6.67	0.00
		Route 3	**60.00**	30.00	13.00	8.50
Model 2	Overall Accuracy (%)	Route 4	63.81	74.48	84.97	**94.23**
		Route 3	72.60	81.30	85.00	94.10
	Crash Likelihood (%)	Route 4	40.00	30.00	16.67	03.33
		Route 3	40.00	30.00	23.30	0.00
Model 3	Overall Accuracy (%)	Route 4	49.13	60.14	63.99	77.45
		Route 3	56.30	64.20	71.50	79.20
	Crash Likelihood (%)	Route 4	**50.00**	**50.00**	**40.00**	**30.00**
		Route 3	**55.00**	**46.70**	**32.00**	**20.00**
Model 4	Overall Accuracy (%)	Route 4	51.40	73.08	83.39	93.36
		Route 3	71.30	82.30	86.40	92.80
	Crash Likelihood (%)	Route 4	**60.00**	26.67	6.67	00.00
		Route 3	40.00	30.00	26.70	6.70

4.3.4 The CTM and unsteady traffic state

In case of the CTM, the traffic condition in a cell is assumed to be steady. Hence, it might seem like that the model can only operate at steady state conditions of traffic. However, the CTM does not assume steady traffic condition among the cells, i.e. inter-cell traffic can still be unsteady. In this thesis, the RTCPMs were constructed with traffic flow parameters such as downstream flow, speed and difference of up and downstream density. Hence, the difference of traffic parameters between the adjacent cells were used as information variables of the RTCPMs which can capture if the traffic condition between adjacent cells were unsteady depending on the threshold given. Hence, a unsteady traffic situation can be captured from the inter-cell traffic difference of a CTM.

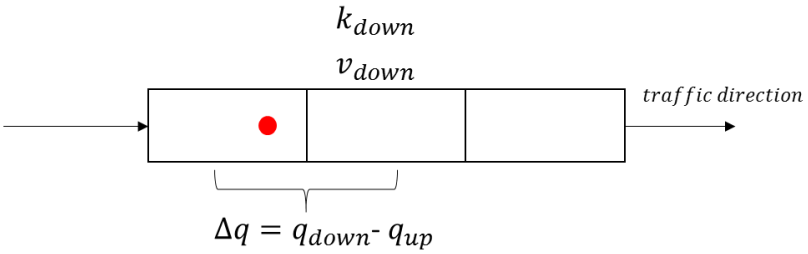


Figure 4 .9 Traffic parameters collection from crash location

In order to construct RTCPMs, in this thesis, the crash data was collected from 3 minutes before the crash occurred, and normal data was collected for similar duration from a day when no crash occurred. The traffic conditions before the crash took place is considered as the hazardous traffic condition; similarly, the traffic state during the no crash data is considered normal condition.

The RTCPMs were trained with the combinations of these data, and were validated with the hazardous and normal conditions data separately to check the model's prediction accuracy at different thresholds.

For example, if the threshold is set to 5, the model will predict all the traffic conditions which led to probability of crash likelihood of 5 or more. In the confusion matrix, it is shown that at threshold 5, the model identifies 1 crash cases correctly, but also predicts 188 normal cases as crash, which is false alarm (34% false alarm rate). Because, it was decided by setting the threshold that above it, all the traffic conditions will be considered as hazardous.

crash	normal
0.04	0.04
0.17	0.03
0.02	0.04
0.03	0.11
0	0.12
0.19	0
0.04	0.12
0.19	0.02
0.19	0.12
0	0.03
0.02	0.11
0.09	0
0.13	0.14
0.12	0.12
0.04	0.05
0.04	0.04
0.17	0.03
0.31	0.07
0.12	0.11
0.04	0.2
0.31	0.2
0.07	0.12
0.06	0.12
0.03	0.14
0.12	0.12
0.31	0.11
0.31	0.03
0.03	0.03
0.17	0
0.07	0.03
	0
	0.03
	0
	0.03
	0.11
	0.11
	0.03

threshold	crash	normal	total crash cases
0.05	18	12	30
0.1	14	16	30
0.15	10	20	30
0.2	4	26	30
0.25	4	26	30

threshold	crash	normal	total normal cases
0.05	188	354	542
0.1	111	431	542
0.15	38	504	542
0.2	17	525	542
0.25	11	531	542

threshold= 5		Predicted	
		crash	no crash
Actual	crash	18	12
	no crash	188	354
Overall Accuracy（%）		65.03	
crash likelihood（%）		60.00	

Figure 4 .10 Example of number of crashes over threshold = 5

According to the definition of hazardous and normal state, all the crashes took place in hazardous state. Hence, the total number of crashes (101 crashes) are considered happened in hazardous state. However, the number of crashes are not only defined from the raw traffic data, but also from the RTCPM output. If the probability of crash is found more than equal to the threshold set, the model outputs all those cases as the harardous states and vice-versa.

4.4 Chapter Conclusion

In this study it has been attempted to establish a method for using a uniformly and densely distributed simulated detector layout to create a framework for developing a real-time crash prediction model which is transferrable over space. A simple method of the CTM was employed in the study route to generate the simulated detector data. It was found that the CTM-based method

could reproduce values of the traffic flow variables with an average error of 13% where the speed data showed higher MPE as compared to flow and occupancy data. One of the reasons is that the basic FD does not allow us to control speed resulting in simulated speed values unresponsive of the traffic situation. However, with a flexible FD the speed can be controlled which could generate speed data with greater accuracy. Moreover, recent studies identified (Coifman, 2014; Lu, 2010) speed data to be the most vulnerable data considering the detector type, traffic state, and the quality of other traffic variables.

The performance of BN- and DBN-based RTCPMs built with traffic data from both fixed detectors and CTM has been also investigated. A series of models with different information variables suggested the most influential variables to be downstream flow, speed, and occupancy, difference of upstream and downstream flow. Although, the results showed that the loop detector based RTCPMs performed slightly better than the CTM based models, it is too early to draw statistically significant conclusions. In any case, it is certain that these preliminary results indicate that CTM is able to generate reliable traffic parameters to overcome the transferability and facilitate a solution to the missing data problem of the future universal RTCPMs. One area of future work will be to the investigation of the optimum location of the simulated detectors or cells by experimenting with different cell lengths and different locations (50, 100, and 150 m upstream and downstream). Moreover, other than freeway stretches, the model in future can be upgraded to incorporate ramps rather than considering only the basic freeway segments. The RTCPM is highly dependent on the quantity as well as the quality of the traffic flow data. Therefore, it is recommended that more case studies and different time-steps should be incorporated to ensure an efficient model that can be implemented in real-time.

After investigating the transferability of the four RTCPMs, model- 3 showed consistency in prediction accuracy of crash likelihood at all thresholds and both validation. For model- 3, although the highest accuracy was found at threshold 5 and 10 (when validated with route 4), the overall accuracy of model-3 at threshold 5 (49%) was less compared to threshold 10 (60%). So, in the later chapters of this thesis, model-3 will be used as the RTCPM and crash risk threshold of 10% will be the threshold for intervention decision making.

4.5 Chapter References

R. Adhikari, R. Agrawal, An Introductory Study on Time Series Modeling and Forecasting LAP LAMBERT Academic Publishing, Saarbrucken, Germany (2013).

Lee, C., B. Hellinga, and F. Saccomanno. Real-time Crash Prediction Model for the Application to Crash Prevention in Freeway Traffic. Transportation Research Record: Journal of the Transportation Research Board, No. 1840, 2003, pp. 67–77.

Pande, A., and M. Abdel-Aty. A Freeway Safety Strategy for Advanced Proactive Traffic Management. Journal of Intelligent Transportation Systems, Vol. 9, No. 3, 2005, pp. 145–158.

Abdel-Aty, M., H. M. Hassan, M. Ahmed, and A. S. Al-Ghamdi. Real-time Prediction of Visibility Related Crashes. Transportation Research Part C, Vol. 24, 2012, pp. 288–298.

Hossain, M., and Y. Muromachi. A Bayesian Network Based Framework for Real-time Crash Prediction on the Basic Freeway Segments of Urban Expressways. Accident Analysis and Prevention, Vol. 45, 2011, pp. 373–381

Shi, Q., and M. Abdel-Aty. Big Data Applications in Real-time Traffic Operation and Safety Monitoring and Improvement on Urban Expressways. Transportation Research Part C, Vol. 58, 2015, pp. 380–394.

Hong, Z., and D. Fukuda. Effects of Traffic Sensor Location on Traffic State Estimation. 15th Meeting of the EURO Working Group on Transportation, Procedia - Social and Behavioral Sciences, Vol. 54, 2012, pp. 1186–1196.

Bianco, L., G. Confessore, and P. Reverberi. A Network Based Model for Traffic Sensor Location with Implications on O/D matrix Estimates. Transportation Science, Vol. 35, No. 1, 2001, pp. 50–60.

Bianco, L., G. Confessore, and M. Gentili. Combinatorial Aspects of the Sensor Location Problem. Annals of Operations Research, Vol. 144, No. 1, 2006, pp. 201–234.

Bianco, L., R. Cerulli, and M. Gentili. New Resolution Approaches for the Sensor Location Problem. Presented at Tristan VI Symposium, Phuket Island, Thailand, 2007.

Morrison, D., and S. Martonosi. Characteristics of Optimal Solutions to the Sensor Location Problem. Annals of Operations Research, Vol. 226, No. 1, 2014, pp. 463–478. https://doi.org/10.1007/s10479-014-1638-y.

Hossain, M., and Y. Muromachi. Optimum Detector Spacing for Real-Time Monitoring of Hazardous Locations on Urban Expressways. Japanese Society of Civil Engineers, Vol. 27, No. 5, 2010, pp. 1045–1054.

Daganzo, C. F. Fundamentals of transportation and traffic operations. Oxford: Pergamon, 1997.

Daganzo, C. F. The Cell Transmission Model, Part II: Network Traffic. Transportation Research Part B, Vol. 29, 1995, pp. 79–93.

Daganzo, C. F. The Cell Transmission Model: A Dynamic Representation of Highway Traffic Consistent With the Hydrodynamic Theory. Transportation Research Part B, Vol. 28, 1994, pp. 269–287.

Lighthill, M., and G. Whitham. On kinematic waves II. A theory of traffic flow on long crowded roads. Proceedings Royal Society of London, Part A, Vol. 229, No. 1178, 1955. http://dx.doi.org/10.1098/rspa/1955.0089.

Mihajlovic, V., and M. Petkovic. Dynamic Bayesian Network: A State of Art. Doctoral Dissertation, University of Twente, the Netherlands, 2001.

Sun, J. and J. Sun. Dynamic Bayesian Network Model for Real-time Crash Prediction Using Traffic Speed Conditions Data, Transportation Research Part C, Vol. 54, 2015, pp. 176–186.

Lee, C., M. Abdel-Aty, and L. Hsia. Potential Real-time Indicators of Sideswipe Crashes on Freeways. Transportation Research Record: Journal of the Transportation Research Board, No. 1953, 2006, pp. 41–49.

Sun, J., and J. Sun. Real-time Crash Prediction on Urban Expressways: Identification of Key Variables and a Hybrid Support Vector Machine Model. IET Intelligent Transport Systems, Vol. 10, No. 5, 2016, pp. 331–337.

Park, H., and A. Haghani. Real-time Prediction of Secondary Incident Occurrences using Vehicle Probe Data. Transportation Research Part C, Vol. 70, 2015, pp. 69–85.

Roy, A., R. Kobayashi, M. Hossain, and Y. Muromachi. Real-time Crash Prediction Model for Urban Expressway Using Dynamic Bayesian Network. Journal of Japan Society of Civil Engineers, Series D3, Vol. 72, No. 5, 2016, pp. 1331–1338.

Dervisoglu, G., G. Gomes, J. Kwon, R. Horowitz, and P. Varaiya. Automatic Calibration of the Fundamental Diagram and Empirical Observations on Capacity. Presented at the Transportation Research Board 88th Annual Meeting, 2009.

Kurzhanskiy, A. A., and P. Varaiya. Active Traffic Management on Road Networks: A Macroscopic Approach. Philosophical Transactions of the Royal Society A, Vol. 368, 2010, pp. 4607–4626.

Chow, A. H. F., G. Gomes, A. A. Kurzhanskiy, and P. Varaiya. AURORA RNM – A Macroscopic Simulation Tool for Arterial Traffic Modeling and Control. PATH Technical Note, Institute of Transportation Studies, University of California, Berkeley. 2009.

Muñoz, L., Sun, X., Sun, D., Horowitz, R. and Alvarez, L. Traffic Density Estimation with Cell Transmission Model, Proceedings of the American Control Conference Denver, Colorado.2003.

Coifman, B. Using Dual Loop Speed Traps to Identify Detector Errors, TRR, 2014.

Lu, X., Kim, Z., Cao, M., Varaiya, P., Horowitz, R. Deliver a Set of Tools for Resolving Bad Inductive Loops and Correcting Bad Data, California PATH Research Report, 2010.

Hossain, M., and Y. Muromachi. A Real-time Crash Prediction Model for the Ramp Vicinities of Urban Expressway. IATSS Research, Vol. 37, No. 1, 2013, pp. 68–79.

S. L. Lauritzen (1995). The EM algorithm for graphical association models with missing data. Computational Statistics & Data Analysis, 19:191-201.

CHAPTER 5

REAL-TIME INTERVENTION MODEL: DEEP REINFORCEMENT LEARNING-BASED VARIABLE SPEED LIMIT

5.1 Introduction

Deep learning (DL) has flourished with enormous success in last few decades in a variety of application domains. This new field of machine learning has been growing rapidly since 2012 and has been applied to most traditional application domains like computer vision, image processing, speech recognition etc., as well as some new areas that present more opportunities. Different methods have been proposed based on different categories of learning, including supervised, semi-supervised, and un-supervised learning. Several advances have occurred in the area of Deep Learning (DL), starting with the Deep Neural Network (DNN) (Rosenblatt, 1957). The survey goes on to cover Convolutional Neural Network (CNN) (LeCun, 1989), Recurrent Neural Network (RNN) (Schmidhuber et al, 1997), including Long Short-Term Memory (LSTM) (Schmidhuber et al, 1997) and Gated Recurrent Units (GRU), Auto-Encoder (AE) (Liou et al, 2014), Deep Belief Network (DBN) (Hinton, 2009), Generative Adversarial Network (GAN) (GoodFelow, 2014), and AlexNet (Krizhevsky, 2014).

DL is a subset of machine learning that uses a cascade of many layers of nonlinear processing units for feature extraction and transformation (Deng and Yu, 2016; Bengio et al, 2013, 2015; Schmidhuber, 2015). Each successive layer uses the output from the previous layer as input. The algorithms may be supervised or unsupervised and applications include pattern analysis (unsupervised) and classification (supervised). DL can be represented in many ways such as a vector of intensity values per pixel, or in a more abstract way as a set of edges, regions of particular shape, etc. Some representations are better than others at simplifying the learning task (e.g., face recognition or facial expression recognition (Glauner, 2015). One of the promises of deep learning is replacing handcrafted features with efficient algorithms for unsupervised (where input data are not labeled) or semi-supervised (when part of input dataset is labeled) feature learning and hierarchical feature extraction (Song et al, 2013). In some articles, DL has been described as a universal learning approach that is able to solve almost all kinds of problems in different application domains. In other words, DL is not task specific (Bengio, 2009***)***.

5.2 Variable speed limit and deep reinforcement learning

Advancement in intelligent transportation systems (ITS) has made it possible to obtain high-quality real-time traffic data that calls for a proactive traffic safety management system to reduce crash risks on a road network. However, this also demands assessing crash risk with changing traffic conditions reliably. In response to that, numerous real-time crash prediction models (RTCPMs) have been developed over the years to predict crash risks based on the real-time traffic data. A number of cutting-edge modeling methods, such as, various types of Neural Networks (Liu and Chen, 2017; Park et al., 2018), Support Vector Machine (Katrakazas et al., 2017), Bayesian Network (Hossain and Muromachi, 2013), Dynamic Bayesian Network (Roy et al., 2018a), Deep Neural Network (Yang et al., 2018), etc. have been employed successfully. The accuracy of the recent RTCPMs in predicting crash risk is observed to be commendable (crash risks: Yang et al., 2018 – 96%, Wu et al., 2018 – 87%; low false alarms: Yang et al., 2018 – 10%, Wang et al., 2017a – 2.7%). Some of these models are even transferrable to a limited extent (Park et al., 2018).

A proactive safety management can systematically reduce the risks of road crashes by altering the traffic states with suitable measures. For a real-time intervention, a threshold of crash likelihood can be a useful measure to formulate a proactive control strategy (Lee et al., 2004). Studies on real-time interventions followed either a traffic simulation program (Lee et al., 2004, 2006; Abdel-Aty et al., 2006a,b,2007a,b,2008, Yu and Abdel-Aty, 2014; Abdel-Aty and Wang, 2017) or a driving simulator (Lee and Abdel-Aty, 2008) to reproduce pre-crash traffic conditions and various countermeasures such as a variable-message sign (VMS) (Lee and Abdel-Aty, 2008) and a variable speed limit (VSL) (Lee et al., 2004; Abdel-Aty et al., 2006a,b,2007a,b,2008; Lee and Abdel-Aty, 2008; Yu and Abdel-Aty, 2014; Abdel-Aty and Wang, 2017). Similarly, a coordinated or uncoordinated ramp metering (Lee et al., 2006; Abdel-Aty et al., 2007a,b; Abdel-Aty and Gayah, 2010) helped control the crash-prone traffic conditions effectively. Park et al. (2018) used warnings such as ‘watch out’ or ‘pay attention’ to enable the drivers more vigilant thereby avoiding the secondary crashes.

A number of studies reduced the crash probability effectively by implementing VSL triggered through RTCPMs (Lee et al., 2004; Li et al., 2016, 2017; Abdel-Aty et al., 2006a, 2007b, 2008). Lee et al. (2004) reduced the speed when a crash probability measured by an RTCPM crossed a predetermined threshold. Abdel-Aty et al. (2006a) was successful by changing the speed limit

gradually every 10 min at 5 mph rate: an abrupt change in space; a reduction for upstream while an increase for downstream. Abdel-Aty et al. (2006b, 2007b, 2008) found that VSL was effective for medium to high speed regimes and had a limited impact for lower speeds. Later on, Abdel-Aty et al. (2008) suggested a homogeneous speed zone for VSL implementation. Lee and Abdel-Aty (2008) stated that VMS and VSL in tandem could reduce speed variations. Abdel-Aty et al. (2006b) found improved safety in the zone of VSL implementation but the high-crash potential was relocated at the downstream. Carlson et al. (2011) and Lu et al. (2011) applied VSL at the upstream of the bottleneck area to control the outflow of the VSL section. This way, the capacity drop at the bottleneck can be avoided, and the bottleneck capacity can be retrieved. A solution to the shifting of crash risk can be found in Yu and Abdel-Aty (2014) who proposed an optimization algorithm to minimize the overall crash risk for the total VSL corridor. Li et al. (2014) employed VSL close to the freeway recurrent bottlenecks to reduce rear-end collision risks where the control strategy included a start-up threshold, a target speed limit (56.33 km/h), and a speed change rate (16.09 km/h every 30 s). Later on, Li et al. (2016) considered a start-up threshold of 20% as a control strategy to activate the VSL in a large-scale freeway segment and found that the crash risk was reduced by 22.62% and injury severity by 14.67%. Recently, Abdel-Aty and Wang (2017) applied VSL successfully to the congested weaving sections of an expressway for reduction of crash risk.

Although the combination of RTCPM and VSL is promising and also proactive in some aspects, there are still scope of improvements, especially in terms of self-learning or using ‘intelligence’. The aforementioned studies focused only on adjusting the speed limit in respect of control strategy, VSL control zone, time of control, and response time considering the pattern of hazardous traffic state. Accordingly, the objective was limited to bring the traffic back to normal in the best possible way using a predetermined set of VSL-based interventions administered during a specified time interval. Therefore, they lack the embedded ‘intelligent agent’ capable of learning by itself to tackle non-recurrent complex traffic patterns. Reinforcement learning (RL), an artificial intelligence-based semi-supervised machine learning algorithm, can support VSL in this regard. In RL, an agent reacts with the environment through several trial and error to optimize the total reward by choosing a state-action pair for every time step (Watkins, 1989; Sutton and Barto, 1998; Hasselt, 2011). Thus, an RL agent exhibits ability for decision making in respect of proactive speed control (Li et al., 2017; Zhu and Ukkusuri, 2014, Davarynejad et al., 2011). The ability to take

real-time proactive control decisions without the need for a model architecture makes RL appealing in ITS field (Rezaee et al., 2012; El-Tantawy et al., 2010; Abdulhai et al., 2003). A Q-learning based multi-agent RL, which is the most commonly used RL algorithm, was used in a study for motorway ramp-metering control with queuing consideration (Davarynejad et al., 2011). In another study (Rezaee et al., 2012) Q-learning based RL method was employed to a freeway road for ramp-metering to compare its performance with ALINEA controller and found that the Q-learning based controller was able to reduce travel time by 17%. Another study (El-Tantawy et al., 2010) a multi-agent RL was used in conjunction with the game theory to alleviate traffic gridlock. An R-Markov average reward technique (R-MART)-based RL was used to optimize VSL control for reducing travel time and vehicle emission (Zhu and Ukkusuri, 2014). Li et al. (2014, 2016) improved on their genetic algorithm-based VSL optimization strategy by using Q-learning-based RL at a freeway recurrent bottleneck (Li et al. 2017). Isele et al. (2018) learned policies and active sensing behaviors employing RL that exceeded the capabilities of the commonly used heuristic approaches in several categories such as task completion time, goal success rate, and ability to generalize the problem, for navigating occluded intersections with autonomous vehicles. Most of these studies mentioned about the major advantage of RL's model-free property, reduced computational complexity and its ability to accommodate non-recurrent traffic patterns. An RL agent continuously gathers information over different traffic patterns and adapts their control policy on-line, making them suitable for complex traffic network control problem with many non-recurrent patterns.

Previous studies on VSL control can be categorized into two- rule-based and proactive approaches (Khondoker et al., 2015). The former approach used a preset thresholds of traffic parameters to activate fixed VSL values on various locations in order to improve safety (Rama, 1999, Piao and McDonald, 2008). The proactive or model predictive approach (MPC) approach refers to preventing problems before it actually takes place, usually, with a target to reduce travel time (TT) and improve safety (Khondoker, 2015). It is a dynamic process and the VSL values are updated and coordinated with new dataset. For example, a Dutch study (Hoogen et al., 1994) tried to homogenize the traffic flow with two speed limits 70km/h and 90km/h, updated every minute. In France, VSL control activated only when the flow exceeded 3000 veh/h and the maximum VSL limit was set to 110km/h (Rivey, 2010). In other studies (Allaby et al., 2007, Lee et al, 2004) logic tree-based VSL algorithm was used using threshold values of traffic parameters, which showed

improvement in safety but increased TT in all scenarios. A review paper on VSL control (Lu et al., 2014) concluded that improvement of traffic flow was not observed in most VSL approaches targeting safety improvement and that the most VSL strategies are yet immature and ad-hoc. Moreover, in another study, it was shown that even in the proactive approach, with the absence of online traffic prediction models, the VSL controllers cannot perform (Li et al., 2017).

Reinforcement learning (RL)-based VSL control allows agents to automatically determine the ideal actions within a specific context to maximize its performance (Watkins et al, 1992, Sutton and Barto, 1998, Barto, 2003). After getting trained an RL agent can, theoretically, make predictions on system evolution and achieve a proactive control scheme. A study by Zhu and Ukkusuri (2014) used an RL approach for dynamic VSL control in a large roadway network, where, the VSL value in each link was allowed to fluctuate from the real-world VSL value to select the optimal dynamic speed limit scheme. Li et al. (2017) incorporated the RL in VSL control strategies to reduce traffic congestion at recurrent merge bottlenecks on freeways where the bottleneck density control was set as the reward function or objective. The advantage of deep RL is, it can incorporate real-time traffic parameters and can decided appropriate VSL values from a set, and decide when and in what location to trigger depending on the predefined threshold (here, in this thesis a crash risk of 10).

In spite of the success records of Q-learning for VSL-controlled optimization compared to the traditional feedback-based VSL control, there are a few issues, such as adaptability of continuous traffic states, location of VSL control sections, and the reliability of the VSL models in real time in terms of crash risk reduction, which are yet to be resolved. Another new adaptation of RL, namely, dueling DQN (Mihn et al., 2013, 2015; Hasselt et al., 2016), which can accommodate continuous traffic states and identify the correct action quickly during policy evaluation, has emerged recently.

Before jumping into the VSL strategies, deep learning and reinforcement learning is explained in section 5.2 to 5. with theoretical explanations, history and algorithms.

5.3 Artificial intelligence and machine learning:

The terms ‘artificial intelligence (AI)’ and ‘machine learning (ML)’ are often tossed around interchangeably, but they are not exactly the same thing. If anything, ML is a subset of AI. In the Google Trends graph (**Figure 5.1**), it can be seen that AI was the more popular searched term until machine learning passed it for good around August, 2015. ML has become one of – if not the – main applications of artificial intelligence.

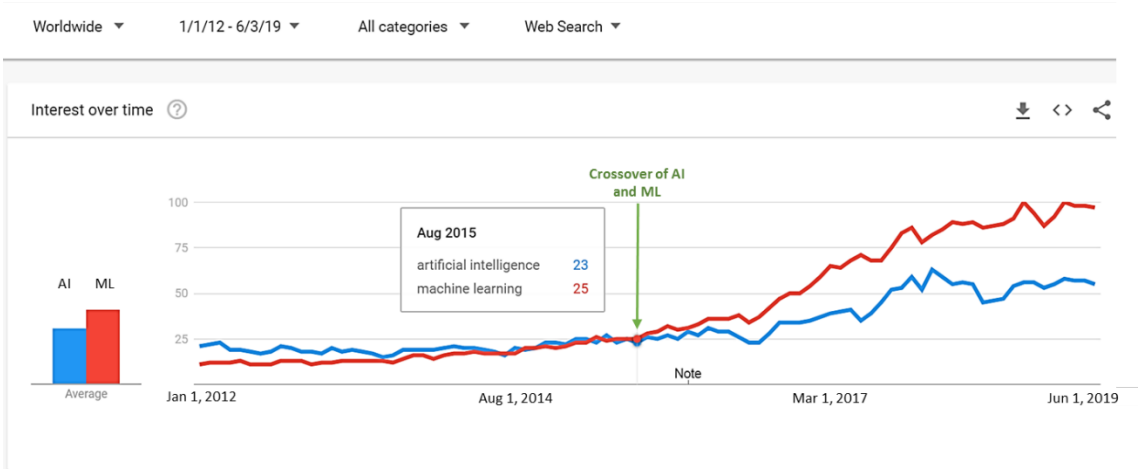


Figure 5. 1 Google trends graph of the term AI and ML (2012- 2019)

Figure-5.2 shows the taxonomy of AI. AI can be considered the all-encompassing umbrella. It refers to computer programs being able to “think,” behave, and do things as a human being might do them. It’s usually classified as either general or applied/narrow (specific to a single area or action).

ML goes beyond AI. It involves providing machines with the data they need to “learn” how to do something without being explicitly programmed to do it. An algorithm such as decision tree learning, inductive logic programming, clustering, reinforcement learning, or Bayesian networks helps them make sense of the inputted data.

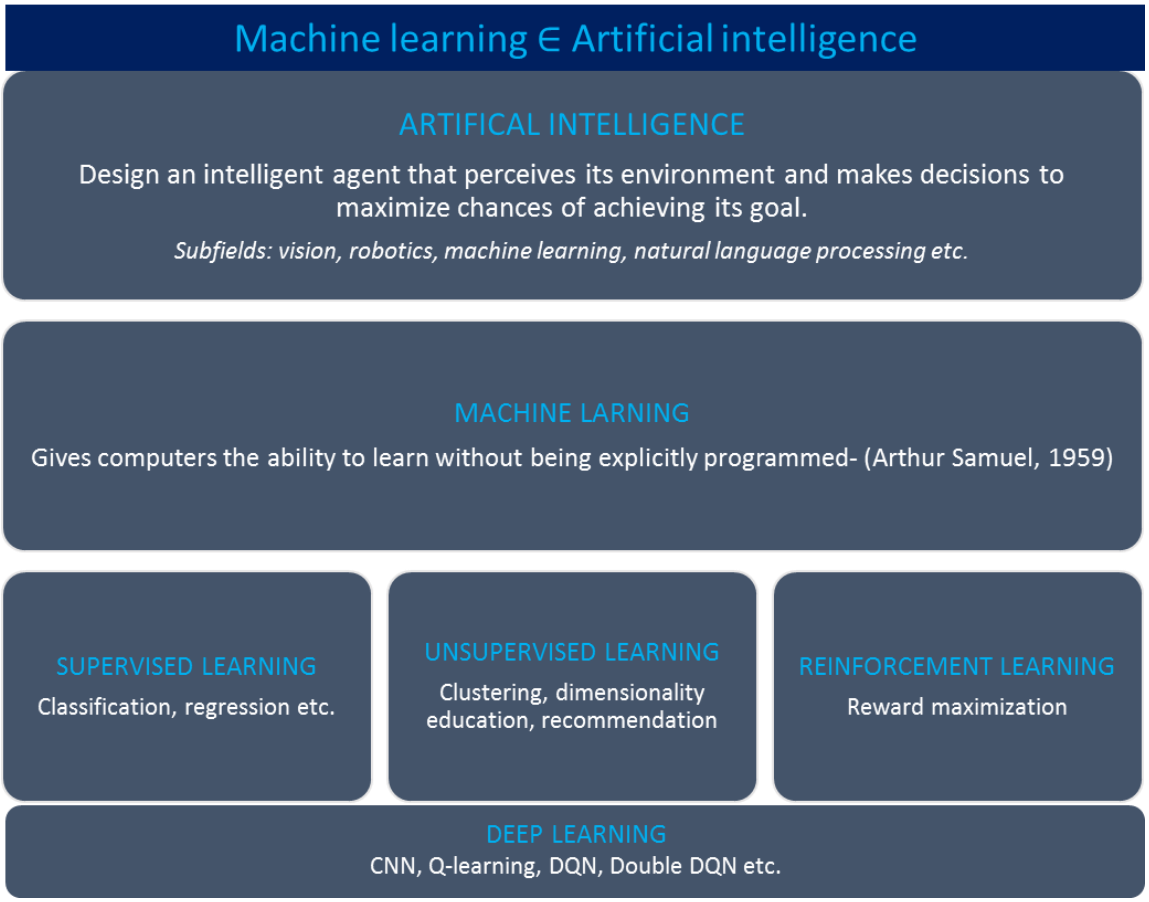


Figure 5. 2 Taxonomy of Artificial intelligence

5.4 Deep Learning

5.4.1 The Definition of *'deep learning'*:

Deep Learning (DL) is a branch of ML- a new take on learning representations from data that puts an emphasis on learning successive layers of increasingly meaningful representations. DL is a new area of ML research, which has been introduced with the objective of moving ML closer to one of its original goals: Artificial Intelligence (AI). DL uses what's called "supervised" learning – where the neural network is trained using labeled data – or "unsupervised" learning – where the network uses unlabeled data and looks for recurring patterns.

The term 'deep learning' was first introduced to the machine learning community by Rina Dechter, a Professor of Computer Science at the University of California in 1986. In her paper, she proposed

a solution to the dead-end situation of intelligent backtracking in PROLOG or Truth-Maintenance-Systems (TMS) in case of search efficiency (Dechter, 1986). Her solution was a constraint recording method as a learning process for making efficient future decisions. She coined the term 'deep learning' while exploring solution to constraint satisfaction problems (CSP) using deep second- order learning.

Geoffrey Hinton is a pioneer in the field of artificial neural networks (ANN) and co-published the first paper ***(***Hinton, 2006***)*** on the backpropagation (BP) algorithm for training multilayer perceptron (MLP) networks which is the basic unit of a DL network. He may have started the introduction of the phrasing "deep" to describe the development of large-scale artificial neural networks (Hinton and Salakhutdinov, 2006, 2009). According to Hinton, Deep Belief Networks were the start of deep learning in 2006 and that the first successful application of this new wave of deep learning was to speech recognition in 2009 (Abdel-Rahman et all, 2010).

One of the major contribution of DL is its ability of working with large number of data or scalability. In addition to scalability, another often cited benefit of DL models is addressed by Peter Norvig (director of research at Google) (Russel and Norvig, 2010)- "it is a kind of learning where the representation formed have several levels of abstraction, rather than a direct input to output". This means DL models have the capability to perform automatic feature extraction from raw data via feature learning (Bengio, 2012). According to Andrew Ng (Andrew Ng's talk in ExtractConf, 2015), chief scientist at Baidu research and former founder of Google Brain, the core of deep learning is the existence of fast enough computers and enough data to actually train large neural networks. He also put emphasis on the scale of the data- the more data are available, the better will be the performance. Jeff Dean from Google Brain project and one of the founder of TensorFlow put emphasis on large-scale neural networks while defining DL (Jeff Dean's talk at Startup Campus Korea, 2016). The 'deep' in 'deep learning' does not refer to any kind of deeper understanding achieved by the approach; rather it stands for this idea of successive layers of representations. It came from the several hierarchal concepts represented by multiple or, deep layers that allows the computer to learn complicated concepts by building them out of simpler ones (GoodFellow and Bengio, 2016). Yann LeCun is the director of Facebook Research has agreed with the above statement. He also, defined DL as the development of large convolutional neural

network (CNN) because, like MLP, this method scales with data and model size and can be trained with backpropagation (LeCunn et al, 1998).

5.4.2 How does DL work?

The way how DL works can be divided into three parts- (a) parameterizing a deep neural network (DNN) by its weights, (b) measuring the quality of the output with loss function, and (c) adjusting of weights using feedback. DNN takes input and maps those to targets via deep sequence of layers by exposure to examples (**Figure 5.3**).

Firstly, the input data is stored as a layer's *weights*. The target is finding a set of values for the weights of all layers in the network so that the network will correctly *learn* to map inputs to their associated targets. Secondly, to control the output, it's important to measure how far away it is from the target and this difference (or, *score*) is measured by the *loss function*. Lastly, the *score* measured is used as a feedback signal (by *backpropagation*) to the network to adjust the value of the weights in a way so that the current *loss function* is minimized. And this adjustment is done by the *optimizer* that carries out the *backpropagation*. Initially, random values are assigned to the weights, which naturally outputs a value far from the target, hence giving a high score of *loss function*. Eventually, with every example the network processes, the *loss score* minimizes leading the network towards the correct direction. This is how a DL program is trained.

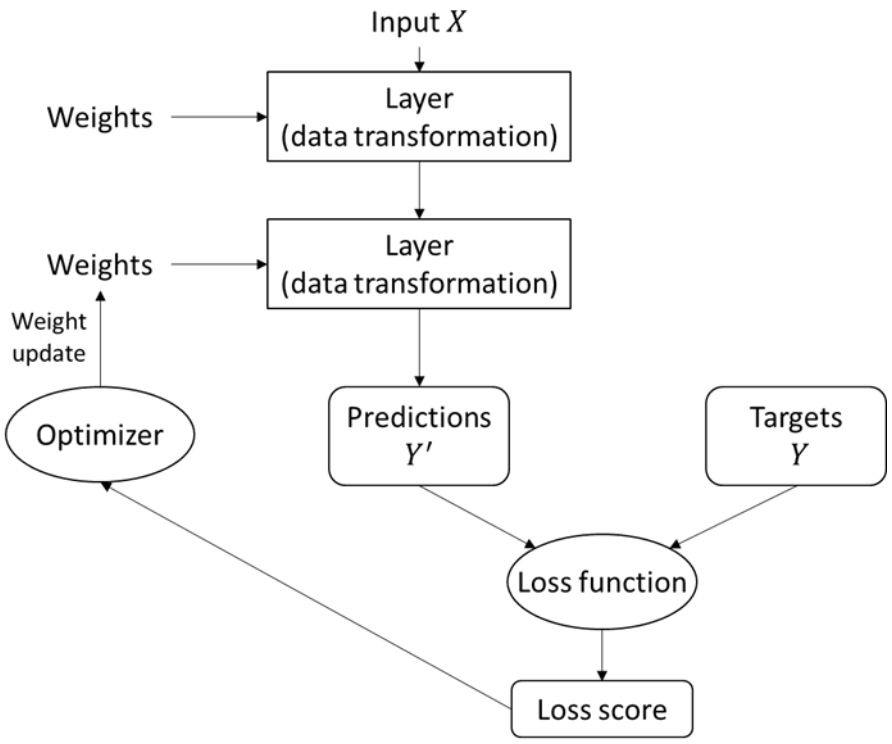


Figure 5. 3 Mechanism of deep learning (François Chollet, 2017)

DL uses layers of algorithms to process data. **Figure 5.4** shows a schematic diagram of an ordinary DNN. Information is passed through each layer, with the output of the previous layer providing input for the next layer. The first layer in a network is called the input layer, while the last layer is called an output layer. All the layers between the two are referred to as hidden layers. Each layer is typically a simple, uniform algorithm containing one kind of activation function.

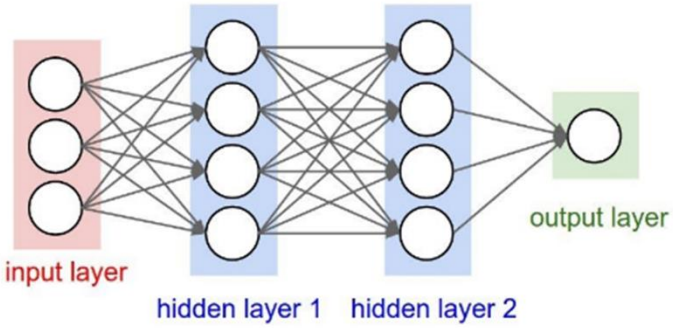


Figure 5. 4 Schematic representation of a typical deep neural network

Feature extraction is another aspect of Deep Learning. Feature extraction uses an algorithm to automatically construct meaningful "features" of the data for purposes of training, learning, and understanding.

5.4.3 Types of deep learning approaches

DL refers to a rather wide class of machine learning techniques and architectures, with the hallmark of using many layers of non-linear information processing that are hierarchical in nature. Depending on how the architectures and techniques are intended for use, e.g., synthesis/generation or recognition/ classification:

Supervised learning

This is by far the most common case. Deep networks for supervised learning are intended to directly provide discriminative power for pattern classification purposes, often by characterizing the posterior distributions of classes conditioned on the visible data. In this method, the model learns to map input data to known targets (annotations). Target label data are always available in direct or indirect forms for such supervised learning.

Example: There are different supervised learning approaches for deep leaning, including Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), including Long Short Term Memory (LSTM), and Gated Recurrent Units (GRU), regression and classification, linear regression, Sequence generation, Syntax tree prediction, Object detection, Image segmentation etc.

Unsupervised learning

This consists of finding interesting transformations of the input data without the help of any targets, for the purpose of better understanding the correlations present in the data at hand. Deep networks for unsupervised or generative learning are intended to capture high-order correlation of the observed or visible data for pattern analysis or synthesis purposes when no information about target class labels is available. In the case of unsupervised learning, the network is provided with some labeled input data and it learns the structure of the data so that it can apply this knowledge to a new input. In this case, the agent learns the internal representation or important features to discover unknown relationships or structure within the input data.

Example: There are several members of the deep learning family that are good at clustering and non-linear dimensionality reduction, including Auto-Encoders (AE), Restricted Boltzmann

Machines (RBM), and the recently developed GAN. In addition, RNNs, such as LSTM and RL, are also used for unsupervised learning in many application domains.

Semi-supervised learning

Semi-supervised learning is learning that occurs based on partially labeled datasets. Hybrid deep networks, where the goal is discrimination which is assisted, often in a significant way, with the outcomes of generative or unsupervised deep networks. This can be accomplished by better optimization or/and regularization of the deep networks by supervised learning. The goal can also be accomplished when discriminative criteria for supervised learning are used to estimate the parameters in any of the deep generative or unsupervised deep networks.

Example: In some cases, DRL and Generative Adversarial Networks (GAN) are used as semi-supervised learning techniques. GAN is discussed in Section 7. Section 8 surveys DRL approaches. Additionally, RNN, including LSTM and GRU, are used for semi-supervised learning as well.

Self-supervised learning

This is a specific instance of supervised learning, but it's different enough to deserve its own category. It is the supervised learning without human-annotated labels. The inputs are still labelled like in supervised learning, but the labels are generated from the input data, typically using a heuristic algorithm. Self-supervised can be interpreted as either supervised or unsupervised depending on whether the focus is on the learning mechanism or on the context of its application.

Example: Autoencoders.

Reinforcement learning

Reinforcement learning has its roots in behavioral psychology. An agent is trained by rewarding it for correct behavior and punishing it for incorrect behavior. In the context of deep reinforcement learning, a network is shown input data and is given a positive or negative reward based on whether it produces the correct output from that input. Thus, in reinforcement learning, we have sparse and time-delayed labels. Over many iterations, the network learns to produce the correct output.

In RL, an agent receives information about its environment and learns to choose actions that will maximize some reward. For example, a neural network looks at a video of a game screen and outputs gaming actions in order to maximize its score, which is trained by RL.

The pioneer in the deep reinforcement learning space was a small British company called DeepMind, which in 2013 published a paper (Mnih, 2013) describing how a convolutional neural network (CNN) could be taught to play Atari 2600 video games by showing it screen pixels and giving it a reward when the score increases. The same architecture was used to learn seven different Atari 2600 games, in six of which the model outperformed all previous approaches, and it outperformed a human expert in three.

5.4.4 A brief history of deep learning:

Until recently, most machine learning and signal processing techniques had exploited shallow-structured architectures. These architectures typically contain at most one or two layers of nonlinear feature transformations. The concept of DL originated from the artificial neural network (ANN). For a long time, training a NN in an efficient way had been a problem until mid-1980s, when several researchers independently rediscovered backpropagation (BP). BP is a way to train chains of parametric operations in NN using gradient descent optimization. But, BP alone didn't work well when it came to large number of hidden layers *(*Bengio, 2009*;* Glorot and Bengio, 2010*)* while optimization- it often got stuck in the local optima, instead of global optima, especially when the network is deep. This is one of the reasons why ML researchers went back to focusing on shallow algorithms like support vector machines (SVMs).

This optimization problem was solved with the introduction of an unsupervised learning algorithm- deep belief net (DBN) (Hinton et al, 2006; Hinton and SalaKhutdinov, 2006). A deep belief net (DBN) is composed of a stack of restricted Boltzmann machines (RBMs). The core component of DBN is a greedy, layer-by-layer learning algorithm which optimizes DBN weights. Initializing the weights of an multilayer perceptron (MLP) with a correspondingly configured deep belief net (DBN), often produces much better results than that with the random weights. So, this type of MLPs with many hidden layers or, in other words, deep neural networks (DNNs), which are learned with unsupervised deep belief net (DBN) pre-training, followed by BP fine-tuning is sometimes called the DBNs by some researchers (Dahl et al, 2012; Mohamed et al, 2012,a,b).

Even sometimes, when DBN is used to initialize the training of a DNN, the resulting network is called DBN-DNN. Recently, researchers are more careful in distinguishing between DBN and DNN.

The very first successful practical application of NN came in 1989 when Yann LeCun combined the ideas of CNN and BP (LeNet) to the problem of hand written zip code classification. After that, in 1995, Vapnik and Cortes came up with a kernel method called 'Support Vector Machine (SVM)'. SVMs aim at solving classification problems by finding good *decision boundaries* between two categories. Decision boundary is a line or surface separating the training data into spaces corresponding to two categories. But SVM had some limitations when it came to large scale datasets and it didn't provide good results for perceptual problems like image classification. A significant amount of researches were conducted around 2000-2010s on neural networks. In 2011, Dan Ciresan from IDSIA (Switzerland) began to win academic image classification competitions with GPU-trained DNNs. Then in 2012, G. Hinton's team lead by Alex Krizhevsky, participated in the ImageNet competitions, which means to classify a dataset of 1.4 million high resolution images into 1000 categories. They were able to achieve an accuracy of 83.6% which was a significant breakthrough. Ever since, the competition has been dominated by deep CNNs (Convnets) every year and by the year 2015, an accuracy of 96.4% were achieved and the problem was considered solved entirely. Since 2012, Convnets have become a popular algorithm for all computer vision tasks. By 2015-2016 there were numerous researches on convnets in terms of computer vision problems and began to flourish in other type of problems like natural language processing (NLP) or voice recognition etc. Convnet has replaces SVM and random forest (RF) completely in wide range of applications.

There have been a lot of developments and advancements in the DL fields over the past 80 years (**Figure 5.6**). The evolution of DL can be summarize with a rough timeline (**Table 5.1**):

Table 5. 1 The history of deep learning (1940- 2019)

1943	McCulloch & Pitts show that neurons can be combined to construct a 'Turing Machine'
1958	Rosenblatt shows that perceptron's will converge if what they are trying to learn can be represented
1965	Alexey Ivakhnenko and V.G. Lapa proposed the first working deep learning networks
1980	Kunihiko Fukushima introduced Neocognitron: a hierarchical neural network capable of visual pattern recognition
1985	The backpropagation algorithm by Geoffrey Hinton et al. revitalizes the field.
1989	CNNs with Backpropagation for document analysis by Yann LeCun
2006	The Hinton lab solves the training problem for DNNs
2015	Generative Adversarial Networks (GAN)
2016-present	A variety of deep learning algorithms are increasingly emerging

5.4.5 Why DL became popular now?

The two key ideas of DL– CNN and BP were already understood in 1989. The long-short term memory (LSTM) algorithm was developed in 1997 and has barely changed since. Hence the question arises why DL is taking over now?

In general, three technical forces are driving advances in ML:

- Hardware
- Datasets and benchmarks
- Algorithmic advances

The true bottlenecks throughout 1990-2000 were the data and hardware. And during this period, the internet took off and high-performance graphic chips were developed due to the boosting of the gaming market.

Hardware- Between 1990 and 2010, the CPUs became about 5000 times faster. But typical DL models for image classification, speech recognition etc. requires more computational power than CPUs. Throughout the 2000s, companies like NVIDIA and AMD have been investing billions of dollars to develop massively parallel chips- GPUs. In 2007, NVIDIA launched CUDA- a programming interface for its line of GPUs. Nowadays, the DL industry is approaching beyond

the GPUs by investing in increasingly specialized and efficient chips for DL. In 216, Google revealed its TPU project, which is 10 times faster and far more energy efficient than GPUs.

Datasets and benchmarks- Over the last 20 years, the rise of the internet made it possible to get access to huge database for ML. The most popular datasets- image, video, natural language etc. are only accessible due to internet (e.g. Flicker, YouTube, Wikipedia etc.).

Algorithmic advances- In addition to hardware and data, a reliable way to train very deep networks was missing until the late 2000s. Early NNs were shallow and were inferior to other ML algorithms like SVMs or RFs. The main issue was that the feedback signal which was used to train a NN, (gradient propagation) would fade away with increasing number of layers. In 2009-2010, there were better activation functions, weight initialization and optimization schemes (e.g. RMSProp, ADAM etc.). Finally, during 2014-2016, more advanced ways of gradient propagation were discovered, such as batch normalization, residual connections and depth-wise separable convolutions.

5.4.6 Properties of DL

DL has several properties that justify its status as an AI revolution. As it is still emerging very fast, in future, there might be more advanced NN, but even that NN would be something inherited from the core concept of today's DL. These core properties can be broadly categorized into four:

1. *Simplicity:* DL replaces the requirement of feature engineering – heavy engineering pipeline with simple end-to-end trainable models.

2. *Scalability:* DL is highly amenable to parallelization on GPUs or TPUs. In addition, DLs iterate over small batches of data while training, which allows them to be trained by arbitrary size of datasets.

3. *Versatility and reusability:* Unlike previous ML approaches, DL models does not require retraining on additional data from scratch. It makes DL viable for continuous online learning. Moreover, trained DL models are repurposable, hence reusable. It means, a DL model trained on image classification problem, can be used for video processing pipeline.

4. *Universal Learning Approach:* The DL approach is sometimes called universal learning because it can be applied to almost any application domain.

5.5 A brief history of RL

The early RL had two major threads- one deals with learning by trial-and-error which basically started in the field of psychology, and the other deals with the problem of optimal control and its solution using dynamic programming and value functions. However, there is a third and less distinct thread which deals with the temporal difference (TD) method. The modern RL is the product of intertwining of these three threads (Sutton and Barto, 2017).

The basic concept of trial-and-error was succinctly expressed by Throndilke (1911) with the 'Law of Effect'. It describes in psychological terms how the animals would grow a tendency to select certain actions based on the events (related to those actions) experienced by them. The term 'reinforcement' appeared in the English translation of Pavlov's (1927) monograph in the context of animal's conditional reflexes. But, Alan Turing (1948) was the first to employ the trial-and-error learning in terms of computer learning calling it the 'pleasure-pain-system'. Several electro-mechanical machines were built using the concept of trial-and-error such as SNARCs (Stochastic Neural-Analog Reinforcement Calculators) introduced by Minsky (1954). This influenced the rise of several other electro-mechanical machines based on the same concept, but many of those shifted their focus from the trial-and-error learning to supervised learning (Clark and Farley, 1955) and created a blurry area where researchers working with supervised learning started to believe they were working on RL. For example, neural network pioneers such as Rosenblatt (1962) and Widrow and Hoff (1960) used the language of rewards and punishments-but the systems they studied were supervised learning systems suitable for pattern recognition and perceptual learning. The confusion could have arose due to the fact that NNs use terms like trial –and –error to update the weights during the process of training. But this is not the same as the RL because, in RL the actions are selected on the basis of evaluative feedbacks (rewards) without any reliance on the information about which action is correct. Influential works were conducted by Donald Michie where he described RL learner called MENACE to play tic-tac-toe (1963). They applied RL learner GLEE and RL controller BOXES to the task of learning to balance a pole hinged to a movable cart on the basis of a failure signal occurring only when the pole fell or the cart reached the end of a track (Michie and Clakson, 1968). This was one of the best example of early RL and several researchers

adopted different versions of this example in their work (Barto, Sutton, and Anderson, 1983; Sutton, 1984). Research on learning automata had a more direct influence on the trial-and-error thread leading to modern reinforcement learning research. Learning automata are simple, low-memory machines for improving the probability of reward in these problems (Narendra and Thathachar, 1974, 1989). John Holland introduced classifier systems, true RL systems including association and value functions (Holland, 1976, 1986). A key component of Holland's classifier systems was the 'bucket-brigade algorithm' for credit assignment.

The individual most responsible for reviving the trial-and-error thread to reinforcement learning within artificial intelligence was Harry Klopf (1972, 1975, 1982). Klopf recognized that essential aspects of adaptive behavior were being lost as learning researchers came to focus almost exclusively on supervised learning. What was missing, according to Klopf, were the hedonic aspects of behavior, the drive to achieve some result from the environment, to control the environment toward desired ends and away from undesired ends. This is the essential idea of trial-and-error learning. Klopf's ideas were especially influential on the authors because our assessment of them (Barto and Sutton, 1981a) led to our appreciation of the distinction between supervised and reinforcement learning, and to our eventual focus on reinforcement learning. Other studies showed how reinforcement learning could address important problems in neural network learning, in particular, how it could produce learning algorithms for multilayer networks (Barto, Anderson, and Sutton, 1982; Barto and Anderson, 1985; Barto and Anandan, 1985; Barto, 1985, 1986; Barto and Jordan, 1987).

Another thread of RL- optimal control and dynamic programming, which describes the problem of designing a controller to minimize a measure of a dynamic system's behavior over time. One of the approaches developed is a functional equation called 'Bellman's equation' uses the concepts of a dynamical system's state and of a value function. The methods for solving optimal control problems by solving this equation came to be known as dynamic programming (Bellman, 1957a). Dynamic programming suffers from what Bellman called 'the curse of dimensionality' meaning that its computational requirements grow exponentially with the number of state variables, but it is still far more efficient and more widely applicable than any other general method. Bellman (1957b) also introduced the discrete stochastic version of the optimal control problem known as Markovian decision processes (MDPs), and Ronald Howard (1960) devised the policy iteration

method for MDPs. All of these are essential elements underlying the theory and algorithms of modern reinforcement learning. Optimal control and dynamic programming was disconnected from learning for a long time possibly because they belong to different disciplines. Also because, the simplest form of dynamic programming is a computation that proceeds backwards in time, making it difficult to see how it could be involved in a learning process that must proceed in a forward direction. The full integration of dynamic programming methods with on-line learning did not occur until the work of Chris Watkins in 1989, whose treatment of reinforcement learning using the MDP formalism has been widely adopted (Watkins, 1989). Since then these relationships have been extensively developed by many researchers, most particularly by Dimitri Bertsekas and John Tsitsiklis (1996).

The third and final thread of RL is the temporal difference (TD) learning. This thread is smaller and less distinct than the other two, but it has played a particularly important role in the field, in part because TD methods seem to be new and unique to RL. TD learning methods are distinctive in being driven by the difference between temporally successive estimates of the same quantity. Arthur Samuel (1959) was the first to propose and implement a learning method that included TD ideas, as part of his celebrated checkers-playing program, which was later studies extensively by Minsky (1961). Later, a method was developed by integrating TD learning with trial-and-error learning, known as the ‘actor critic architecture’, which was applied to Michie and Chambers's pole-balancing problem (Barto, Sutton, and Anderson, 1983). Witten's 1977 paper spanned both major threads of reinforcement learning research: trial-and-error learning and optimal control-while making a distinct early contribution to TD learning. As mentioned earlier, the TD and optimal control threads were fully brought together in 1989 with Chris Watkins's development of Q-learning.

5.5.1 Definition of *Reinforcement learning*

In the field of ML, there are three types of tasks that can be done- supervised learning, unsupervised learning and reinforcement learning. Supervised learning is learning from a training set of labeled examples provided by a knowledgeable external supervisor; whereas unsupervised learning is typically about finding structure hidden in collections of unlabeled data. Reinforcement learning (RL) is the task of learning how agents ought to take sequences of actions in an environment in order to maximize cumulative rewards. RL is different from unsupervised learning

because it tries to maximize a reward signal instead of trying to find hidden structure. Reinforcement learning algorithms study the 'behavior' of subjects in an 'environment' and learn to optimize that behavior. Deep reinforcement learning is the combination of reinforcement learning (RL) and deep learning.

The key aspects of RL are- an agent which learns a 'good behavior' incrementally and through trial-and-error of 'experiences'. Hence, the agent works without requiring any prior knowledge, only acquiring information by interacting with the environment. RL uses 'offline' or 'online' learning methods. In the offline method, the experience is gathered a priori and then is used as a batch for learning. On the other hand, in online learning method, the sequential data is fed to the agent to progressively update its behavior. The basic learning algorithm for both methods are the same except, the online method has to deal with the 'exploration and exploitation dilemma'. Exploitation means to take the same action which leads to a high reward known to the agent by far without trying look for actions which might result in even higher rewards. Exploration enables the agent to explore and try actions which may not lead to a high reward immediately but leading to new actions resulting in higher rewards in the long run at the expense of time spent for exploration. So, a balance between these two is important.

5.5.2 Framework of RL or MDPs

The problem of sequential decision making with RL can be formalized using ideas from dynamical systems theory like Markov decision processes (MDPs). MDPs are a classical formalization of sequential decision making, where actions influence not just immediate rewards, but also subsequent situations, or states, and through those future rewards. The components of MDPs are agent, environment, state action and reward. The agent-environment interface of MDPs are explained below (**Figure 5.5**).

For the sake of simplicity, let us consider the case of Markovian stochastic control processes as (Norris, 1998).

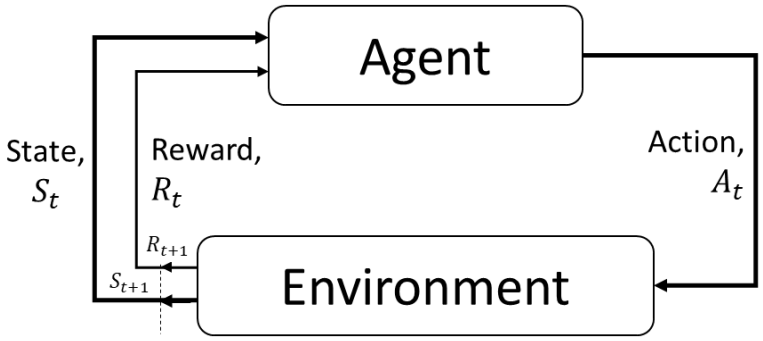


Figure 5. 5 Agent- environment interaction in finite MDPs (Sutton and Barto, 2017)

In MDPs, there is a decision maker, called an 'agent' that interacts with the 'environment' it is placed in. These interactions occur sequentially over time. At each time step, $t = 0, 1, 2, \ldots, n$, the agent will get some representation of the environment's state, $S_t \in \boldsymbol{S}$. Given this state, the agent takes an action, $A_t \in \boldsymbol{A}$. The environment is then transitioned into a new state, $S_{t+1} \in \boldsymbol{S}$ and the agent is given a reward, $R_{t+1} \in \boldsymbol{R}$ as a consequence of the previous action. The algorithm of MDPs is explained in the following steps-

1. At time, t, the environment is in state, S_t.
2. The agent observes the current state, S_t and selects action, A_t.
3. The environment transitions to state, S_{t+1} and grants the agent reward, R_{t+1}.
4. This process then starts over for the next time step, $t + 1$.

Hence, the trajectory representing the sequential process of MDPs can be represented as-

$$S_0, A_0, R_t, S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1}, R_{t+2}, \ldots \ldots \quad (5.1)$$

In a finite MDP, the sets of states, actions, and rewards (S, A, R) all have a finite number of elements and have well defined discrete probability distributions dependent only on the preceding state and action. For example, let's choose a random state, $s' \in \boldsymbol{S}$, action $a' \in \boldsymbol{A}$ and reward, $r' \in \boldsymbol{R}$**.** The probability of receiving reward r' by transitioning to (s', a') at time t, given particular values of the preceding state, $s \in \boldsymbol{S}$ and action $a \in \boldsymbol{A}$**(s)** can be defined as-

$$\Pr(s', r' | s, a) = \Pr(S_t = s', R_t = r' | S_{t-1} = s, A_{t-1} = a) \quad (5.2)$$

This means that the future of the MDP only depends on the current observation, and the agent has no interest in looking at the full history.

An MDP can also be defined in terms of a tuple (S, A, T, R, γ) (Sutton and Barto, 2017, Levet et al, 2018), where

- S is state space.
- A is action space.
- T is transition function: set of conditional transition probabilities between states.
- R is reward function: set of possible rewards (continuous values in a range of [0, R_{max}]).
- $\gamma \in [0, 1]$is the discount factor.

At each time step t, the probability of transitioning to state, S_{t+1} by taking action, A_t is given by the transition function $T(S_t, A_t, S_{t+1})$ and for that the reward received is given by the reward function $R(S_t, A_t, S_{t+1})$ (**Figure 5.6**).

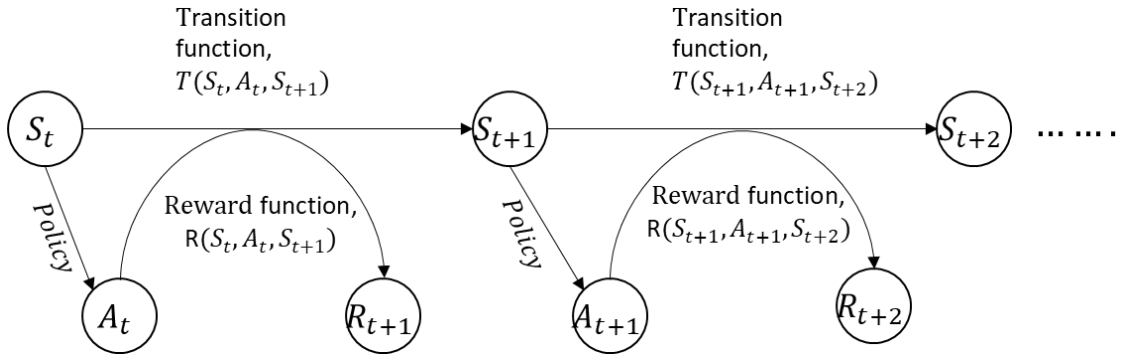


Figure 5. 6 Illustration of MDPs (Levet et al, 2018)

5.5.3 Policy of MDP

Which action, $a \in \boldsymbol{A}$ would the agent choose is defined by the policy, $\pi: S \rightarrow A$. It's the mapping from states to actions. Policies could be either stationary (continuing) or non-stationary (episodic). A non-stationary policy refers to an agent collecting cumulative rewards within a finite number of time steps (Bertsekas et al., 1995), whereas stationary policy deals with cumulative rewards acquired from an infinite horizon. Another criterion to categorize policies is either deterministic, $\pi\ (s)$ or stochastic, $\pi\ (s, a)$. Now the question that arises is how to define the optimality of policies

and which criterion should be optimized when finding policies. This is discussed later in the policy and value function section of this thesis. In this thesis we consider infinite-horizon optimality, which means that all rewards received by the agent are taken into account.

5.5.4 Expected Return of MDP

The agent in MDP follows a reward hypothesis which is to maximize the expected cumulative reward in the long run. It can be formalized mathematically as follow-

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \cdots + R_T \quad (5.3)$$

Where, G_t is the return at time , $R_{t+1} + R_{t+2}$ .... are the possible future rewards calculated at time t, and T is the final time step. In case of non-stationary (episodic) policy, the time step, T is finite and known. In the stationary (continuing) case, $T = \infty$. But, if the final time step is infinite, the return itself could become infinite. To tackle this problem, the concept of discounted return is introduced.

5.5.5 Discounted Return of MDP

In general, the agent seeks to maximize the expected return, where the return, denoted by G_t is defined as some specific function of the reward sequence (equation 5.3). In cases the agent–environment interaction does not break naturally into identifiable episodes, but goes on continually without limit, the concept of discounted return is applied. According to this approach, the agent tries to select actions, A_t so that the sum of the discounted rewards G_t it receives over the future is maximized (equation 5.4).

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots$$
$$= \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \quad (5.4)$$

Where, γ is the discount rate $[0 \leq \gamma \leq 1]$. It ensures that rewards received in the near future are more important than rewards received later. So, when the agent calculates the rewards it expects to receive in the future, the more immediate rewards have more influence when it comes to making a decision about taking a particular action. Equation 5.4 can also be expressed as below-

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots$$
$$= R_{t+1} + \gamma(R_{t+2} + \gamma R_{t+3} + \cdots)$$
$$= R_{t+1} + \gamma G_{t+1} \quad (5.5)$$

Equation 5.5 shows how the successive returns are related to one another. It also shows that even though the return at time t is the sum of infinite future returns, it is actually finite as long as the reward is nonzero and constant, and $\gamma < 1$. For example, if the reward at each time step is a constant 1 and $\gamma < 1$ then the return is-

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \quad = \sum_{k=0}^{\infty} \gamma^k \quad = \frac{1}{1-\gamma} \quad (5.6)$$

Hence, the infinite sum yields a finite result.

5.5.6 Policies, Value Functions and their Optimalities:

5.5.6.1 Policy, π:

Reinforcement learning methods specify how the agent's policy is changed as a result of its experience. There are two questions that arise while discussing the MDPs-

How probable is it for an agent to select any action from a given state?

How good is any given action or any given state for an agent?

The first question is addressed by policies. As mentioned earlier, a policy is a function, $\pi\ (a_t|s_t)$ that maps a given state, s_t to probabilities of selecting each possible action, $a_t \in \boldsymbol{A(s)}$ from that state.

The second question is addressed by value function. Value functions are functions of states, or of state-action pairs, that estimate how good it is for an agent to be in a given state, or how good it is for the agent to perform a given action in a given state. This notion of how good a state or state-action pair is given in terms of expected return. Since the way an agent acts, is influenced by the policy it's following, hence, value functions are defined with respect to policies. There are two kinds of value functions-

5.5.6.2 State Value Function, v_π:

The state value function for policy π, denoted by v_π refers to how good any given state is for an agent following policy π. It gives the value of a state under π. Generally, the value of state, s under policy π is the expected return from starting from s at time t and following policy π thereafter. This can be expressed mathematically as-

$$v_\pi(s) \doteq E_\pi[G_t | S_t = s]$$

$$\doteq E_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s] \quad (5.7)$$

For all $s \in S$.

5.5.6.3 State-action Value Function, q_π:

Similarly, the state-action value function for policy π, denoted by q_π refers to how good is it for an agent to take any given action from a given state while following policy π. It gives the value of an action under π. Generally, the value of an action, a in state s under policy π is the expected return from starting from s at time t, taking action, a and following policy π thereafter. This can be expressed mathematically as-

$$q_\pi(s, a) \doteq E_\pi[G_t | S_t = s, A_t = a]$$

$$\doteq E_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s, A_t = a] \quad (5.8)$$

For all $s \in S$ and a $\in A$. Conventionally, the action-value function $q_\pi(s, a)$ is referred to as the *Q-function*, and the output from the function for any given state-action pair is called a *Q-value*. The letter "Q" is used to represent the quality of taking a given action in a given state.

5.5.6.4 Optimal Policy, π_*:

Solving a reinforcement learning task means, roughly, finding a policy that achieves a lot of reward over the long run. In terms of return, a policy, π is considered to be better than or the same as

policy π'if the expected return of π is greater than or equal to the expected return of π' for all states. It means, $\pi \geq \pi'$, if and only if, $v_\pi(s) \geq v_{\pi'}(s)$ for all $s \in S$. Hence, a policy that is better than or at least the same as all other policies is called the optimal policy, π_*.

5.5.6.5 Optimal State Value Function, v_*:

Although there may be more than one optimal policy, all the optimal policies are denoted by π_*. They share the same state-value function, called the optimal state-value function, denoted by v_*, and defined as

$$v_*(s) = max_\pi v_\pi(s) \tag{5.9}$$

For all $s \in S$. In other words, v_* gives the largest expected return achievable by any policy π for each state.

5.5.6.6 Optimal State-action Value Function, q_*:

Similarly, the optimal policy has an optimal action-value function, or optimal Q-function, which we denote as q_* and define as-

$$q_*(s,a) = max_\pi q_\pi(s,a) \tag{5.10}$$

For all $s \in S$ and $a \in A(s)$. In other words, q_* gives the largest expected return achievable by any policy π for each state-action pair (s,a).

5.5.6.7 Bellman optimality equation for v_* and q_*:

The fundamental property of value function is to satisfy the recursive relationship of the already established value function. So, from the action value function, for any policy π, and any state s, the recursive relationship between values of s and values of its possible successor states-

$$v_\pi(s) \doteq E_\pi[G_t | S_t = s] \quad \text{(by 5.7)}$$

$$= E_\pi[R_{t+1} + \gamma G_{t+1} | S_t = s] \quad \text{(by 5.5)}$$

$$= \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r|s, a) [r + \gamma E_\pi[G_{t+1} | S_t = s]]$$

$$= \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a) \; [r + \gamma v_\pi(s)] \qquad (5.11)$$

For $s' \in S$, $a \in A(s)$ and $r \in R$. The equation 5.11 is called **Bellman equation** for v_π, expressing the relationship between values of a state and values of its successor states. The relationship can be expressed with the backup diagram (**Figure 5.7**). In this figure, the open circles are the states and the solid circles are the state-action pairs. Only three of them are shown here. Starting from state s, the agent could take any of the action sets depending on its policy and dynamics given by the function p and achieve next state s' with a reward r. The Bellman **Equation (5.11)** averages over all the possibilities, weighting each by its probability of occurring. It states that the value of the start state must equal the (discounted) value of the expected next state, plus the reward expected along the way.

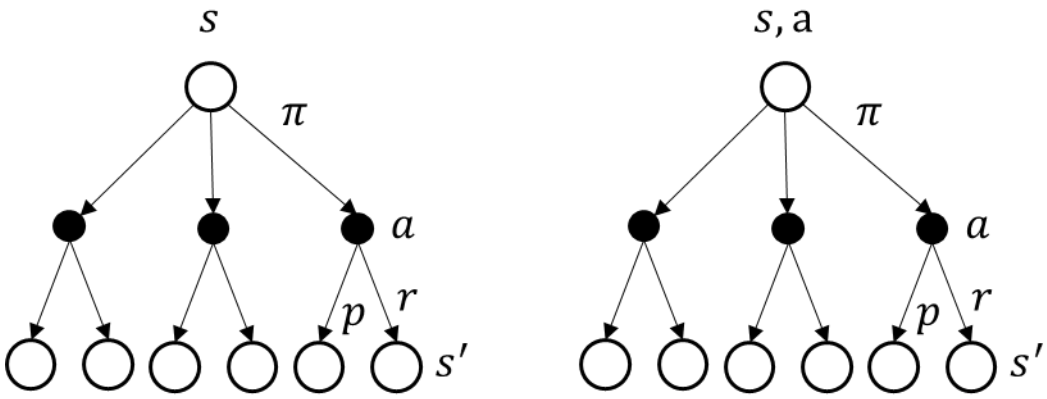


Figure 5. 7 'Backup diagram' for v_π and q_π (Sutton and Barto, 2017)

As mentioned in section 5.5.6.5, v_* is the optimal value function for a policy, so it must satisfy the self-consistency condition given by the Bellman equation for state values (equation 5.11). Intuitively, the Bellman optimality equation expresses the fact that the value of a state under an optimal policy must equal the expected return for the best action from that state:

$$v_*(s) = max_{a \in A(s)} q_{\pi_*}(s, a)$$

$$= max_a E_{\pi_*}[G_t | S_t = s, A_t = a]$$

$$= max_a E_{\pi_*}[R_{t+1} + \gamma G_{t+1} | S_t = s, A_t = a]$$

$$= max_a E[R_{t+1} + \gamma v_* S_{t+1} | S_t = s, A_t = a] \qquad (5.12)$$

$$= max_a \sum_{s',r} p(s', r|s, a) \ [r + \gamma v_*(s')] \quad (5.13)$$

Equation 5.12 and 5.13 are called the **Bellman optimality equations for v_*.**

The similar can be derived for optimal action value function, q_*:

$$q_*(s, a) = max_a E\left[R_{t+1} + \gamma q_*(S_{t+1}, a') \middle| S_t = s, A_t = a\right] \quad (5.14)$$

$$q_*(s, a) = max_a \sum_{s',r} p(s', r|s, a) \ [r + \gamma max_{a'} q_*(s', a')] \quad (5.15)$$

The meaning of the equation is, for any state-action pair (s, a) at time t the expected return from starting in state (s), selecting action (a) and following the optimal policy (q_*) thereafter (also known as the Q-value of this pair) is going to be the expected reward we get from taking action (a) in state (s), which is R_{t+1}, plus the maximum expected discounted return that can be achieved from any possible next state-action pair.

Since the agent is following an optimal policy, the following state s_{t+1} will be the state from which the best possible next action a'can be taken at time $t + 1$. The Bellman equation can be used to find q_* and then the optimal policy can be determined because, with q_*, for any state s, a reinforcement learning algorithm can find the action a that maximizes $q_*(s, a)$.

The optimal policy π_* can be obtained from the q_*:

$$\pi_*(s) = argmax_{a \in A} q_*(s, a) \quad (5.16)$$

5.6 Deep reinforcement learning

The DRL is the combination of DL and reinforcement learning. The general schema of DRL is shown in **Figure 5.8**.

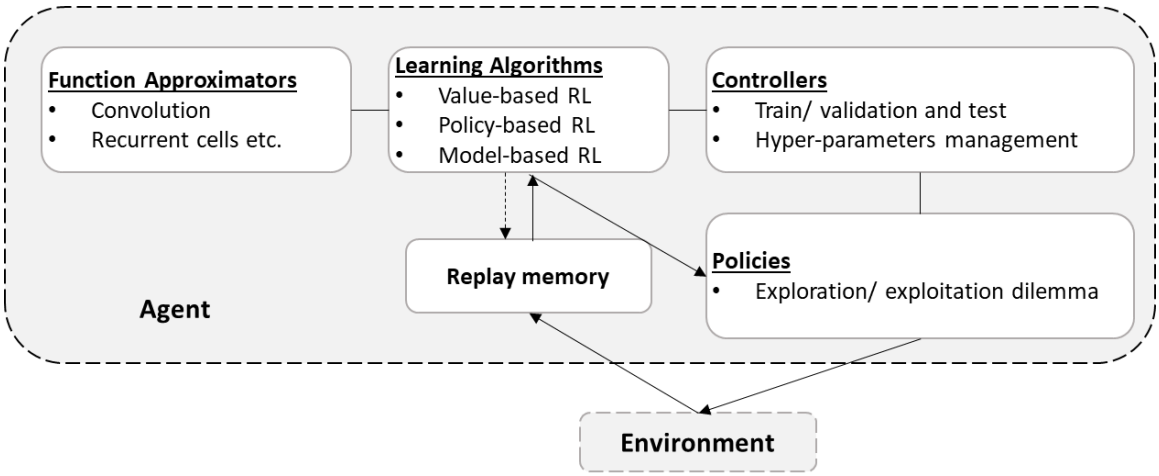


Figure 5. 8 General schema of deep reinforcement learning

5.6.1 Basic Q-learning (QL)

It is a form of simple yet the most popular model-free RL algorithm where an agent learns the best actions suitable for given states by estimating a long-term discounted reward in controlled Markovian domain. According to Watkins et al. (1989), it is an asynchronous dynamic programming that learns via 'temporal difference' method. At time step, t , the agent experiences a state, S_t from a state set under the given environment and selects an action, a_t from the action set. This selection of action by observing a state is the behavioral policy of the agent. Then, the agent updates the state to a new state S_{t+1}, which brings in a reward of r_{t+1} to evaluate the change of state. After repeating this process several times, the agent learns the actions that led to maximum accumulated rewards over time. Each state-action pair receives a Q-value $[Q^t(S_t, a_t)]$ at every time step, which is updated with new training data on a continuous basis using Equation. 5.17.

$$Q^{t+1}(S_t, a_t) \leftarrow (1-\alpha)Q^t(S_t, a_t) + \alpha[r_{t+1} + \gamma \, max \, Q^t(S_{t+1}, a_{t+1})] \qquad (5.17)$$

Here γ is the discount factor ($0 \leq \gamma \leq 1$) that ensures that a higher reward is assigned to the recent actions taken compared to the actions to be taken in the future because the certainty about selecting a state-action pair gets lesser as it goes closer to the future; α is the learning rate that controls the updating rate of the Q-value where the goal of the agent is to maximize the reward over time. Although it is common for an agent to make imprecise estimates while learning at the beginning, QL runs a maximization step that tends to choose overestimated values that caused possibly by erroneous action values irrespective of the source of approximation error (Thrun and Schwartz

1993). The QL agent runs each action for each state multiple times until the convergence of Q-value thereby completing the learning process. As the agent learns the optimum actions for given states provided with the largest Q-values, it can determine the optimal control strategy for the system (Li et al., 2017; Abdulhai et al., 2003).

The Q-learning is conducted with the use of a table which has actions in the horizontal axis and states in the vertical axis, and the table is called Q-table. The table is filled with Q-values that the agent received for taking an action in a particular state. Initially the Q-table is filled with zeroes as the agent doesn't know which action is better. Hence, the agent will start at a random selection of action while maintaining the exploration and exploitation using the $\varepsilon - greedy$ method, where ε $[0, 1]$, the higher the value is, the more will be the exploitation. So, it is wise to start with a lower ε value so that the agent will explore more, meaning that the agent will take different actions from the same state during different episodes. After going through few episodes, the agent will slow down the exploration as it has now experienced all the (s, a) pairs a few times, and will tend to exploit more, that is, the agent will take the actions leading to high rewards more often to gain higher expected returns. All these happens by running several episodes and updating the Q-table with new Q-values with the goal of reducing the loss between the Q-value (q_π) and the optimal Q-value (q_*). So, from equation 5.8 and 5.15-

$$Loss = q_*(s, a) \ - q_\pi(s, a) \tag{5.18}$$

$$Loss = E\left[R_{t+1} + \gamma max_{a'} q_*(s', a')\right] - E[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}] \tag{5.19}$$

Another question that arises while running Q-function is the 'learning rate (α)', which refers to the frequency of updating the Q-table. The agent doesn't just overwrite the old Q-value, but rather, it uses the learning rate as a tool to determine how much information to keep about the previously computed Q-value for the given state-action pair versus the new Q-value calculated for the same state-action pair at a later time step. The higher the learning rate, the more quickly the agent will adopt the new Q-value. For example, if the learning rate is 1, the estimate for the Q-value for a given state-action pair would be the straight up newly calculated Q-value and would not consider previous Q-values that had been calculated for the given state-action pair at previous time steps. The equation to calculate the new Q-value for state-action pair (s, a) at time, t is:

$$q^{new}(s,a) = (1-\alpha)\underbrace{q(s,a)}_{\text{Old Q-value}} + \underbrace{\alpha[R_{t+1} + \gamma\, max_{a'}\, q\,(s',a')]}_{\text{Learned value}} \quad (5.20)$$

The algorithm of Q-learning is as follows-

Algorithm-5.1: Q-learning

Initialize Q-table Q [num_states, num_actions]
Arbitrarily observe initial state s
Repeat
    Select and carryout a section a
    Observe reward r and move to new state s'.
        $q^{new}(s,a) = (1-\alpha)\, q(s,a) + \alpha[R_{t+1} + \gamma\, max_{a'}\, q\,(s',a')]$
        $s = s'$
Until game over

The algorithm is basically doing stochastic gradient descent on the Bellman equation, backpropagating the reward through the episodes and averaging over many trials (or epochs).

5.6.2 Real-time intervention: Variable speed limit with Q-learning

Back to the application of variable speed limit (VSL). The real-time crash prediction and intervention is designed by integrating a real-time crash prediction model with a real-time intervention model. The RTCPMs are built with the BN using the traffic data received from the macro simulation model CTM (chapter 3). In this chapter, the CTM for Shinjuku route 4 of Tokyo metropolitan expressway is employed for traffic data extraction to construct a uniformly distributed virtual detector layout, to which DRL-based VSL intervention will be integrated using Keras- a python-based open source deep learning library (https://keras.io/). Several attempts have been taken over the years by researchers to introduce various VSL strategies depending upon the location, traffic condition and detector locations. Most of the cases, the strategies were set by the researchers themselves based on their expertise and experience. Hence, there is no particular strategy to be singled out which can be claimed to be the 'best' one. In this thesis, DRL is adopted to find out VSL strategies in order to utilize the benefit of DRL's model-free learning ability. Which means, without the assistance of any external force (the model designer), the agent of the model itself will figure out the strategies of VSL from the information it receives from the environment (the traffic data). The procedure is described in the following sub-sections. The algorithm is shown in **Figure 5.9**.

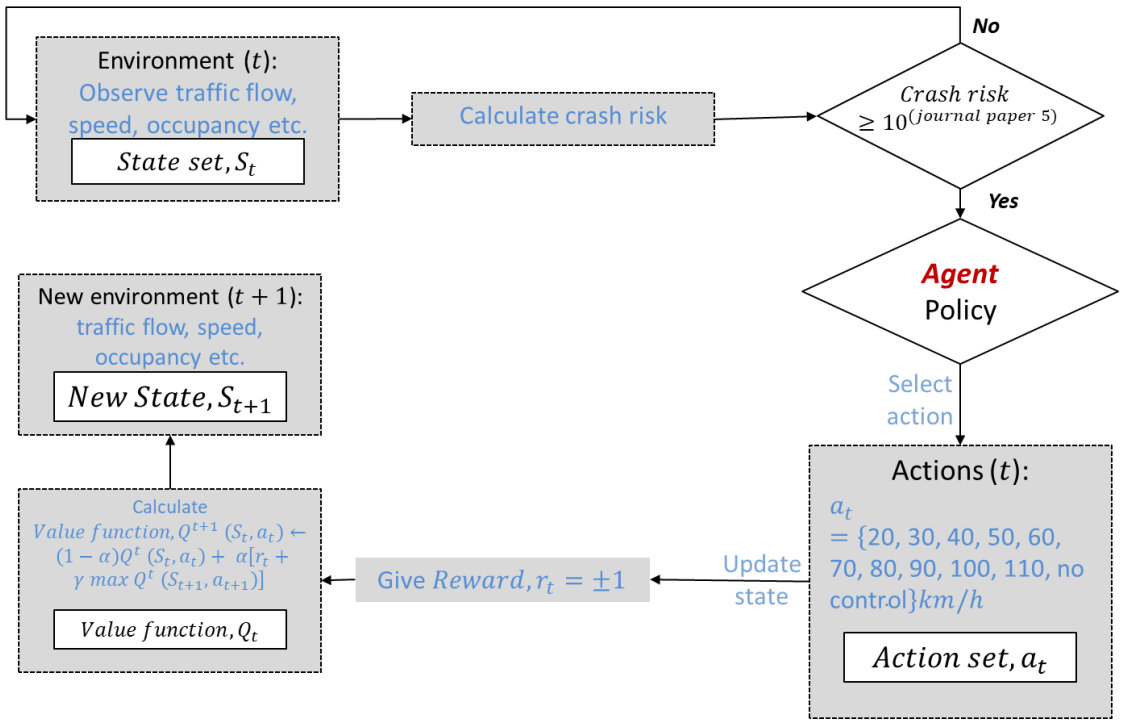


Figure 5. 9 Algorithm for RTCP and DRL-based intervention model.

The detailed explanation is described in the following sections starting from BN-based RTCPM building, CTM-based macrosimulation for traffic data generation, and DRL-based VSL control for intervention model.

5.6.3 The RTCPMs

To construct a DRL-based (Q-learning) intervention, at first a Q-table is required which includes the possible actions and states of the model to be tabulated. The states are the traffic data (flow, speed, occupancy) and the corresponding crash risks. Chapter 4 explains the methodology of BN-based RTCPMs in details. In this chapter, the models and their results are shown directly. A study by Roy et al. (2018 (a)) employed different combinations of six base parameters (flow, speed, occupancy, etc.) and three relative parameters (difference of upstream and downstream flow, etc.) to generate 16 BN- and DBN-based RTCPMs. In their study, some of the RTCPMs had four information parameters and others had three. After judging the overall prediction performances, it was clear that the number of information parameters did not influence the prediction accuracy of the models significantly. The four information parameters found to be the most influential in predicting a crash were: downstream flow (veh/min), speed (km/h), occupancy (%), and difference

values of upstream and downstream flow (veh/min). In this study, four BN-based RTCPMs are built with different combinations of nine information variables: flow (veh/min), speed (km/h), occupancy (%) data from upstream and downstream; and their differences.

After considering the prediction accuracy of crash likelihood and overall accuracy, out of four BN-based RTCPMs model-3 with the information variables downstream speed, downstream occupancy and the difference of up and downstream density was chosen for constructing the Q-table. In order to construct a Q-table the continuous traffic data needs to be discretized into classes. Here, the traffic values are discretized into 5 classes and the corresponding crash risks are also recorded from the BN-based RTCPM (**Table 5.2**).

Table 5. 2 Discretization of traffic parameters

Elements	Class ①	Class ②	Class ③	Class ④	Class ⑤
$V_{down}\,(km/h)$	<49.52	49.52-64.83	64.83-70.69	70.69-76.11	76.11<
$K_{down}\,(veh/km)$	<22.54	22.54-30.66	30.66-40.19	40.19-59.02	59.02<
$\Delta Q\ (veh)$	<−6.50	−6.50-−1.50	−1.50-1.50	1.50-6.50	6.50<

5.6.3 QL-Based VSL Strategy

A Q-table is basically a look up table for the RL agent. The rows of the table represents states and the columns all the possible actions. In this case, with three traffic parameters each categorized into five classes, in total $5^3 = 125$ states are created and all possible actions are either {20, 30, 40, 50, 60, 70, 80, 90, 100, 110, no control} km/h or $\{\pm 10, \pm 0,\ V_{free-flow}\}\ km/h$ depending on the current state of the environment. Hence, there are two Q-tables with (125 x 11) and (125 x 5) sizes. The states and action sets are explained in the following section of this chapter. The Q-table would possibly look like **Table 5.3.** Each cell in the Q-table is filled with zeros initially, afterwards, the table is updated with Q-values that the agent receives over the time after several iterations.

Table 5. 3 Sample of a Q-table

States	Action 1	Action 2	……..	Action 11
State 1	Q-value 1	Q-value 2	……	Q-value 11
State 2				
.				
.				
State 125				Q-value 125

For QL application, the 2.52-km-long segment of route 4 Shinjuku is selected because it qualifies as the longest freeway segment harboring ten detectors (detectors 48 to 39) and has steady state flow during the off-peak hour. The target location where the crash risk needs to be reduced is between cells 12 and 17 (loop detector 42 and 39) (**Figure 5.10**). For the purpose of learning by the QL agent, simulation is run during 12:00-17:59 with a warm-up period of an hour, hence data for 5h simulation was collected. Total 36 iterations were done and the duration of each simulation time step is set to 1 min (total simulation time = 5 h × 60 min × 36 iterations = 10,800 min). Q-value is calculated for each action taken. In the case of the action 'no control', which means at the beginning when no VSL is activated and the crash risk has not exceeded, Q-value is calculated using the simulated data from only one section at cells (14-15), because these two are the central cells of the target section and calculating Q-value with each cell's value would cause computational cost which is not necessary during steady state of traffic. Three major parameters are required to define for Q-learning- states, actions and rewards. These are described below.

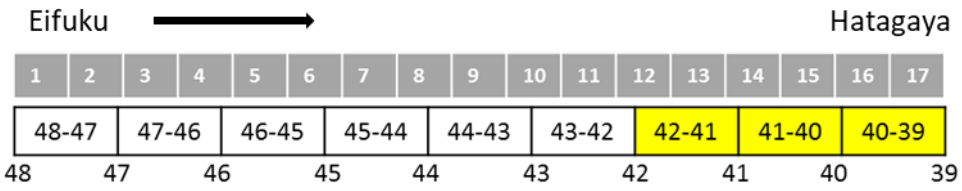


Figure 5. 10 QL-based VSL controlled segment

State, s_t: For QL, states needs to be discretized and defined for learning the speed control strategy by observing simulated traffic flow data from a detector. In this study, the cells 12 to 17 are selected as a section for VSL application, for defining states and is represented by traffic parameters of RTCP Model. The states are discretized into five classes (**Table 5.2**) generating (5^3) = 125 combinations of states.

Rewards, r_t: The target location for calculating the crash risk is between the cells 12 to 17 from which simulated traffic data is collected during 13:01-17:59, and crash risk is calculated based on the 125 states defined in (i) for which 125 crash risks are stored. For this, the crash risk threshold

is set to 10. In case of receiving rewards by the agent, if the crash risk between the target section is lower than 10, a reward of $r_t = +1$ is provided, whereas a reward of $r_t = -1$ is given otherwise.

Action, a_t: A set of 11 values of speed control is selected that includes {20, 30, 40, 50, 60, 70, 80, 90, 100, 110, no control} km/h, where 'no control' suggests that the speed is the same as the free-flow speed for that segment. While executing VSL control, there are two possible situations: (1) VSL control is not running in current state, s_t and (2) VSL control is running in current state, s_t. For the former situation of 'no VSL control', a_t is running in current state, s_t, and the set of actions from which the agent can choose an action is-

$a_{t+1} = \{20, 30, 40, 50, 60, 70, 80, 90, 100, 110, \text{no control}\} km/h$. For the latter case, when VSL control is running at current state, s_t, the choice of actions would be, $a_{t+1} = \{\pm 10, \pm 0, V_{free-flow}\}\ km/h$ for every speed in the VSL speed control set {20, 30, 40, 50, 60, 70, 80, 90, 100, 110, no control} km/h. For example, if the VSL is activated currently and the speed observed at current state, s_t is 70 km/h, then the choices of actions would be, $a_{t+1} = \{80, 60, 70, V_{free-flow}.\}$ km/h.

The learning rate refers to how frequently the RL agent will update its Q-values, which means how much it will remember the old learning and how much it will replace with new experience is decided by the learning rate (α). The learning rate was set to about 0.0001 at first. The discount factor represents the importance given to the predicted possible future rewards compared to the current reward. The value of discount factor (γ) was kept around 0.9. And the final parameter (ε) explains the exploration and exploitation ratios of the agent while training. The value is set to 1 so that the model can explore more and gradually decrease it with every iteration.

5.6.4 Outcome of the QL-Based VSL Strategy

A comparison is performed between the crash risks before and after applying the VSL control between the cells 12 to 17. **Figure 5.11** demonstrates a space-time diagram of the difference of number of events that had crash risk higher than 10, i.e., the number of events with risk higher than 10 are stored before and after VSL was applied. Then the difference of the number of events are plotted against space and time during 13:01-17:59 along the entire segment between cells 12 to 17 (detector 42-39). The red and green colors represent the increase and decrease in number of

events due to application of VSL control (with risk higher than 10), respectively, whilst the white color means no improvement or deterioration in crash risk. Hence, the green color is the desired outcome of VSL control and vice-versa. Additionally, the first row at the bottom of the s-t diagram represents if the VSL was triggered or not. If yes then its colored orange. The second row represents the VSL values taken by the agent at that time, which was selected from the action set (color code of the action set is shown). The color code is shown at the bottom, speed increases gradually from red via orange and yellow to green, and the darkest green color is for 'no control' or free-flow speed. Thus, it can be observed that the QL agent has chosen an action that lowers the speeds in most of the cases. Few other occasions, such as between 14:00-14:10, 14:20-14:30 pm, the agent chose to increase the existing speed limit that led to either improvement or no change of the crash risk most of the times. At one point during 15:10-15:20, after lowering the speed limit, there was a sudden increase of number of events with risk (NER) ≥10, which within next 2 minutes improved the number of crashes significantly. Also, interestingly, few cases can be observed when the VSL control improved NER ≥10 upstream to the target location during 17:30-18:00.

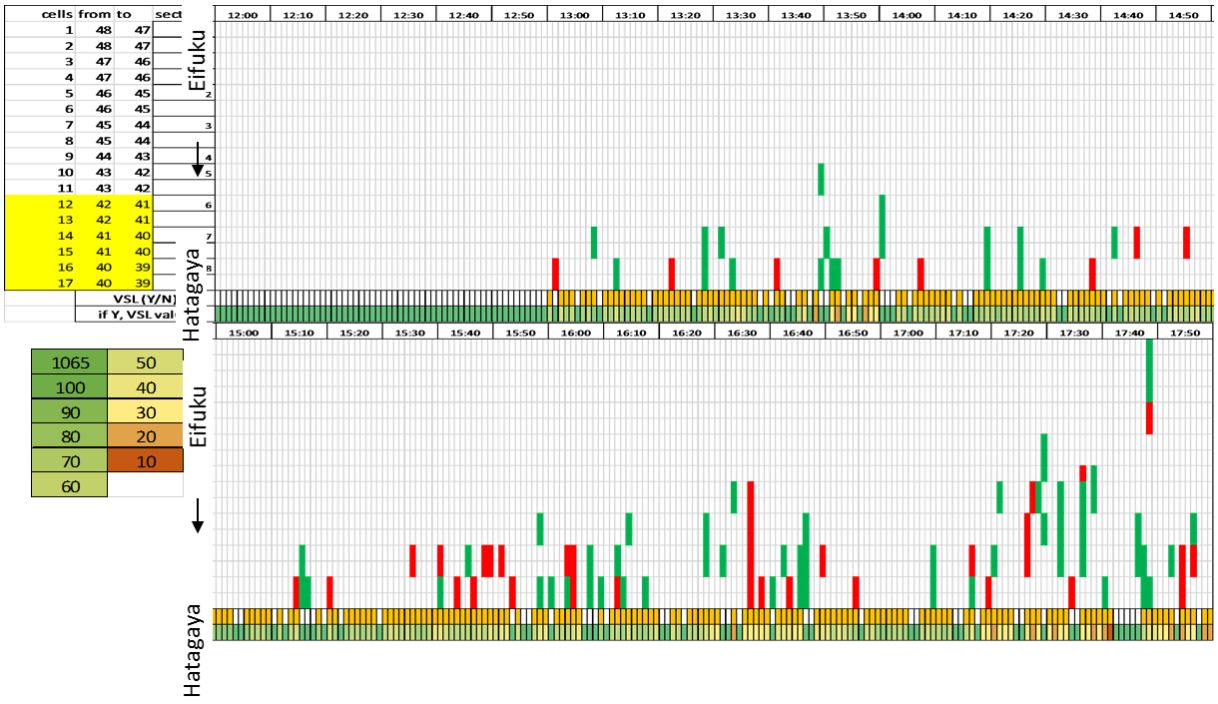


Figure 5. 11 Space-time diagram of change in crash risk before and after the application of VSL control using QL

A bar chart was generated to show if the model was able to lower the number of crashes with risk ≥10 at the target location. **Figure 5.12** compares the NER ≥10 with and without VSL Control. It is verbatim that, VSL control could improve the crash risk at the upstream of the study location (cell 1 to 11) but not significantly, in fact the opposite was observed at cell5 and 6. However, it was able to reduce crash risk at the VSL controlled area (cells 12 to 17) which was the target location to reduce crash risk.

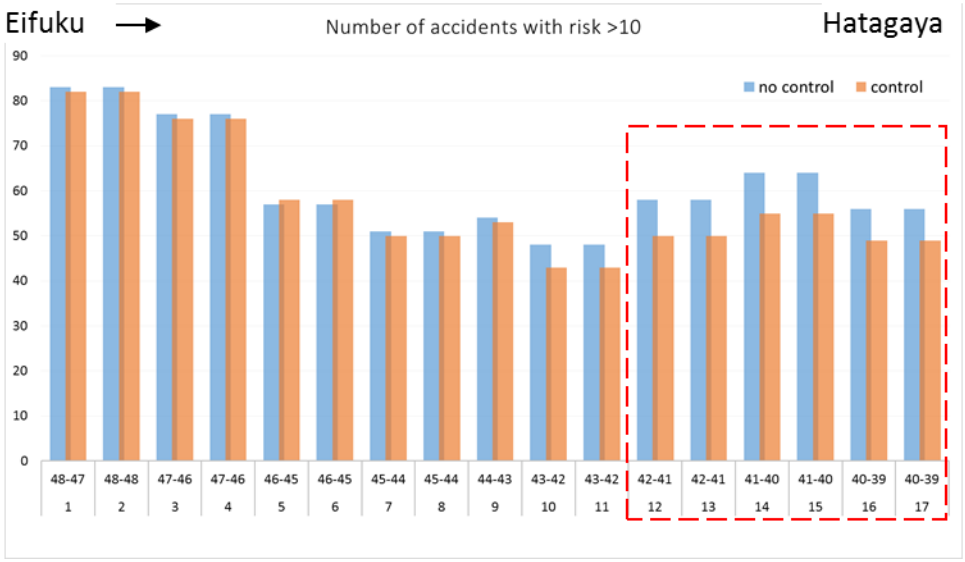


Figure 5. 12 Comparison of NER ≥10 with and without VSL Control (QL)

From **Table 5.4**, improvements can be observed at the targeted location if the risk threshold is set to risk ≥10 whereas deterioration of the risk was also observed at the targeted location if the threshold is set risk = 50. Which means, if the threshold is set to a value as high as 50, the model would not be able to recognize crash prone events. Hence, will take actions only in fewer cases which may in turn cause increase of crash likelihood.

Table 5. 4 Comparison of NER ≥10 and NER = 50 (cells 12 to 17)

Cells	Section	No control	Control	Difference	No control	Control	Difference
		Risk = 50			Risk ≥10		
1	0	9	9	0	83	82	1
2		9	9	0	83	82	1
3	1	0	0	0	77	76	1
4		0	0	0	77	76	1
5	2	8	8	0	57	58	-1
6		8	8	0	57	58	-1
7	3	5	5	0	51	50	1
8		5	5	0	51	50	1
9	4	4	4	0	54	53	1
10	5	7	7	0	48	43	5
11		7	7	0	48	43	5
12	6	9	7	2	58	50	8
13		9	7	2	58	50	8
14	7	7	6	1	64	55	9
15		7	6	1	64	55	9
16	8	3	5	-2	56	49	7
17		3	5	-2	56	49	7

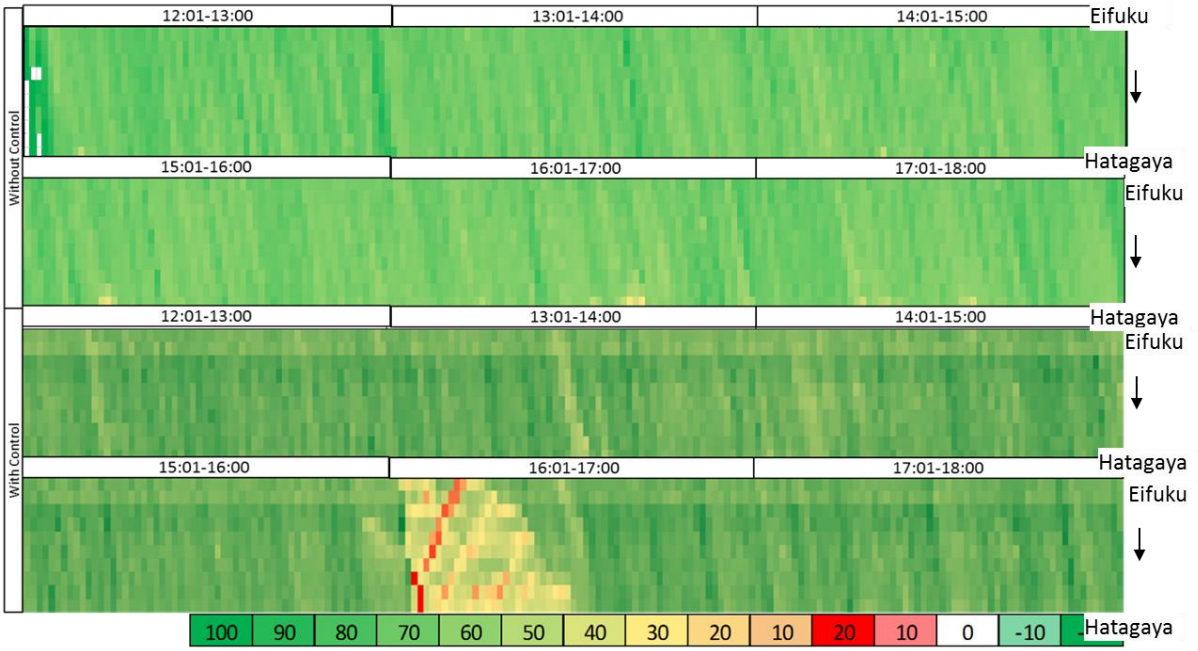


Figure 5. 13 Comparison of NER ≥10 with and without VSL Control (QL)

Figure 5.13 shows a space-time diagram of the entire route during 12:01-17:59 when VSL was applied (bottom of the figure) and there was no speed control. While the goal is to improve crash safety, it is also important to consider the functionability of the route. If a capacity drop appears

due to the VSL control, that might cause not only congestion but also increase the possibility of rear-end crashes. From **Figure 5.13**, it can be seen that the speed control has created a back-wave and a cluster of slow moving vehicles probably due to congestion during 15:30-16:00 which is concerning. This is one of the limitations of the model.

While QL is a useful method for learning policies, it has its other limitations too. As mentioned earlier that the traffic parameters needs to be discretized in order to create the Q-table, which in turn returned with a set of 125 state values. With 125 states and 11 actions there are 1375 state-action combinations that the agent has to monitor at every iteration. But the traffic parameters are continuous and if not discretized, the model would suffer from a huge number of parameters causing slow simulation which is inefficient. Hence, QL method has no scalability and cannot accommodate continuous values, or multiple sections as input parameters. On the other hand, there are neural networks which works better in handling continuous data and are scalable. In the following section, another reinforcement learning method combined with the deep neural network called deep Q-learning (DQL) is introduced to apply VSL.

5.7 Deep Q-Network (DQN)

Deep Q-learning is the is the first deep reinforcement learning (DRL) method proposed by Mihn et al. (2015) that combines RL with DL (a multi-layered deep CNN) (Krizhevsky et al., 2012) to create a single algorithm addressing a range of challenging tasks. The deep CNN consists of hierarchical layers of convolutional filters that enable the algorithm to accept data in abstract form and to categorize the objects directly from the raw sensor data. Similar to QL, deep Q-network (DQN) consists of an agent dealing with the state-action pairs in an environment while receiving rewards with the goal of maximizing the cumulative future rewards. In this case, the deep NN is applied for estimating the Q-value:

$$Q^*(s,a) = max_\pi \mathbb{E}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} \ldots\ldots + \gamma^n r_{t+n} \mid s_t = s, a_t = a, \pi] \qquad (5.21)$$

where $Q^*(s,a)$ is the maximum sum of rewards, r_t discounted by γ at each time step, t, (s) is state, (a) is action, and the behavioral policy is denoted by $\pi = p(a|s)$.

Since NNs have local connectivity (that is, each neuron is connected to only a local region of its input), it avoids these impossible or improbable input combinations. Hence a NN can be used to model a Q-function very effectively.

Experience reply and target network

DQN overcomes unstable learning of RL using two techniques: experience reply and target network. Experience reply randomizes over data, removing the correlations in the state sequences and smoothing the data distribution. In the case of target network, the Q-values are updated only periodically towards the target values thereby reducing the correlations with the target. The agent's strategy would logically be to train the network to predict the best next state's'' given the current state (s, a, r). It turns out that this tends to drive the network into a local minimum. The reason for this is that consecutive training samples tend to be very similar. To counter this, during observation of states, it collects all the previous moves (s, a, r, s') into a large fixed size queue called the reply memory. The reply memory represents the experience of the network. When training the network, it generates random batches from the reply memory instead of the most recent (batch of) transactions. Since the batches are composed of random experience tuples (s, a, r, s') that are out of order, the network trains better and avoids getting stuck in local minima. Yet another approach is to collect experiences by running the network in observation mode for a while in the beginning, when it generates completely random actions (ε = 1) and extracts the reward and next state from the game and collects them into its experience reply queue.

An approximate value function $Q(s, a, \theta_i)$ using the deep CNN is estimated, in which θ_i represents the weight given to the DQN at iteration i. In experience reply, agent's experiences, e_t (s_t, a_t, r_t, s_{t+1}) are stored at each time-step t in a set of data, D_t $\{e_1,\ldots, e_t\}$. The relationship updated by QL as in Eq. (1) is applied on experiences $(s_t, a_t, R, s_{t+1}) \sim$ U (D), which is selected randomly from the samples. The QL update at iteration i uses the following loss function presented in Eq. (3):

$$L_i(\theta_i) = \mathbb{E}_{(s,a,r,s_{t+1}) \sim \text{U (D)}}[(R + \gamma max_{a'} Q(s_{t+1}, a_{t+1}, \theta_i^-) - Q(s_t, a_t, \theta_i))^2] \qquad (5.22)$$

Where, θ_i are the parameters of the DQN at iteration *i* and θ_i^- are the network parameters, used to compute the target at iteration *i*. The target network parameters θ_i^- are only updated with the DQN parameters (θ_i) periodically and are held fixed between individual updates.

The algorithm for DQL is:

Algorithm-1: Deep Q-learning with experience reply

Initialize memory $\mathcal{D}$ to capacity N
Initialize action- value function Q with random weights
For episode= 1, M **do**
Initialize sequence $s_1 = \{x_1\}$ and preprocessed sequenced $\emptyset_1 = \emptyset(s_1)$
For t=1, T **do**
With probability ϵ select a random action a_t
Otherwise select $a_t = max_a Q^*(\emptyset(s_t), a; \theta)$
Execute action a_t in emulator and observe reward r_t and image x_{t+1}
Set $s_{t+1} = s_t, a_t, x_{t+1}$ and preprocess $\emptyset_{t+1} = \emptyset(s_{t+1})$
Store transition $(\emptyset_t, a_t, r_t, \emptyset_{t+1})$ in $\mathcal{D}$
Sample random mini batch of transitions $(\emptyset_t, a_t, r_t, \emptyset_{t+1})$ from $\mathcal{D}$
Set $y_j = \begin{cases} r_j & for\ terminal\ \emptyset_{j+1} \\ r_{j+max_{a'}Q^*(\emptyset_{j+1}),a';\theta)} & for\ non-terminal\ \emptyset_{j+1} \end{cases}$
Perform a gradient descent step on $(y_j - Q(\emptyset_j, a_j; \theta)^2$ according to the equation of loss function
end **for**
end **for**

Algorithm credit: (https://storage.googleapis.com/deepmind-media/dqn/DQNNaturePaper.pdf)

5.7.1 Deep Q-Network (DQN) based VSL strategy

As described in the previous section, DQN consists of neural network, which means, it has an input layer, an output layer and in between these two layers, all the layers are hidden layers. In this scenario, the neurons in the input layer will take states as inputs, which are traffic parameters. The output layer consists of multiple neurons each of those are actions to be taken. For each given state input, the network outputs estimated Q-values for each action that can be taken from that state.

The objective of this network is to approximate the optimal Q-function which will satisfy the Bellman equation-

$$q_*(s,a) = max_a \sum_{s',r} p(s',\ r|s,a)\ [r + \gamma max_{a'} q_*(s',\ a')] \text{ or,} \quad (5.23)$$

$$q_*(s,a) = E\ [r_{t+1} + \gamma max_{a'} q_*(s',\ a')] \quad (5.24)$$

This is because, the Bellman equation can be used to find optimal value function, $q_*(s,a)$. Once $q_*(s,a)$ is found, the optimal policy can be determined, because, knowing $q_*(s,a)$ and with state s as input, the RL agent can find the action a that maximizes $q_*(s,a)$. Then the loss from the network is calculated by comparing the outputted Q-values to the target Q-values (**Equation 5.18** to **5.20**) from the right-hand side of the Bellman equation, and as with any network, the objective here is to minimize this loss. After the loss is calculated, the weights within the network are updated via SGD and backpropagation. This process is done over and over again for each state in the environment until the network sufficiently minimizes the loss and get an approximate optimal Q-function.

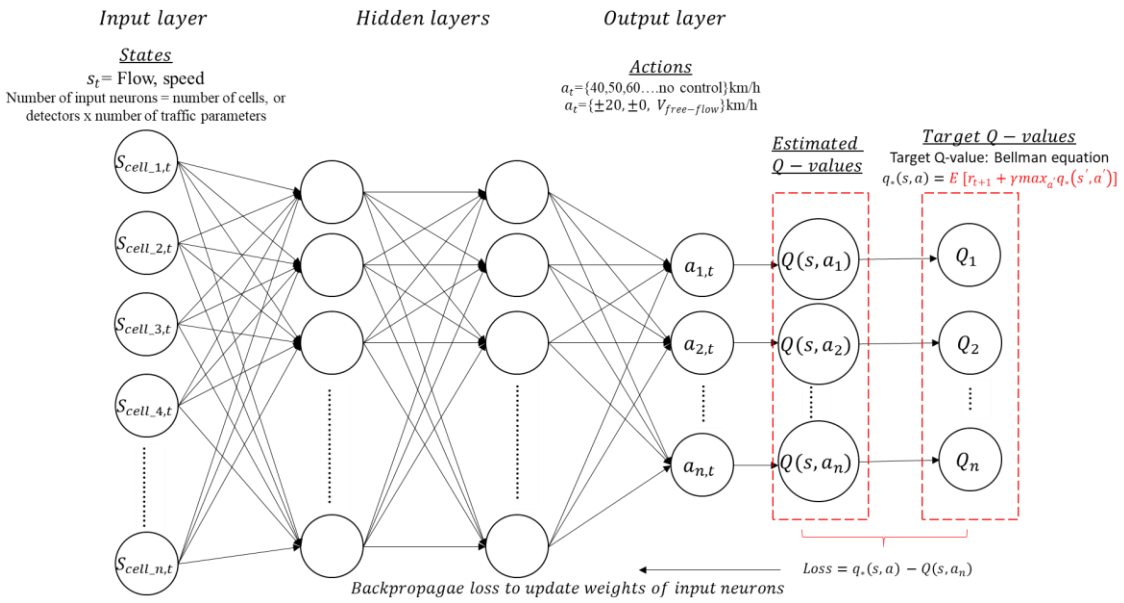


Figure 5. 14 DQN-architecture

Similar to the Q-learning method, the states, rewards and actions needs to be defined for DQL as follows:

State, s_t: States of DQN are the traffic parameters collected from simulation from the similar segment of route 4 consisting cells 1 to 17 (detectors 48-39) (**Figure 5.15**) during 13:01-17:59 (deducting the warm up period), except, there will be two cases with different target locations-

first one consists of cells 9 to 17 (about 1.35 km long) and the second one is longer from cell 5 to 17 (about 1.95 km long).

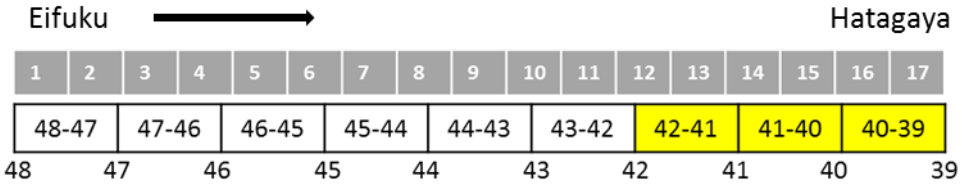


Figure 5. 15 DQN-based VSL controlled segments (cells 12 to 17)

Rewards, r_t: The target location for calculating the crash risk is between cells 9 to 17 (detector 44 and 39). The crash risk threshold for receiving rewards is set to 10. If the crash risk at the target location is lower than 10, a reward of $r_t = +1$ is provided, -1 is given otherwise.

Action, a_t: In case of QL, an action set of 11 actions were used for the case when VSL control was not activated. Now in case of DQN, a set of 8 values of speed control is selected eliminating the lower speed values (10, 20, 30 km/h etc.). This action set includes {40, 50, 60, 70, 80, 90, 100, no control} km/h. The another action set is for the situation when there is an VSL control activated at current state,s_t, and a possible crash risk is observed (i.e. a state leading to crash risk ≥10) the set of actions to be taken is $a_{t+1} = \{\pm 20, \pm 0,\ V_{free-flow}\}$ km/h. For example, if the speed observed at current state, s_t is 70 km/h, then the choices of actions would be, $a_{t+1} = \{90, 70, 50, V_{free-flow}.\}$km/h and the agent will receive a positive or negative reward ($r_t = +1$, -1) based on how well the agent performed by taking that particular action at time t which then changed the current state, s_t to s_{t+1}. Simulation is run during 13:00-17:59 the duration of each simulation time step is 1 min (total simulation time= 5 h × 60 min × 36 iterations = 10,800 min). To evaluate the policy established by the agent, Q-value is calculated for each action taken. For the target section between cell 12 to 17, the input, hidden and output neurons were (12, 36,36,4).

The learning rate refers to how frequently the RL agent will update its Q-values, which means how much it will remember the old learning and how much it will replace with new experience is decided by the learning rate (α). The learning rate was set to about 0.0001 at first. The discount factor represents the importance given to the predicted possible future rewards compared to the current reward. The value of discount factor (γ) was kept around 0.9. And the final parameter (ε)

explains the exploration and exploitation ratios of the agent while training. The value is set to 1 so that the model can explore more and gradually decrease it with every iteration.

5.7.2 Outcome of the DQN- based VSL strategy (cell 12 to 17)

The outcomes of the DQN-based VSL control strategy are presented in **Figures 5.18**. It shows the comparison of NER ≥10 with and without VSL control. Similar as in the case of QL, in the space-time diagram the green bars and the red bars represent improvement and deterioration of the crash risk respectively. It shows the increase or decrease of NER ≥10.

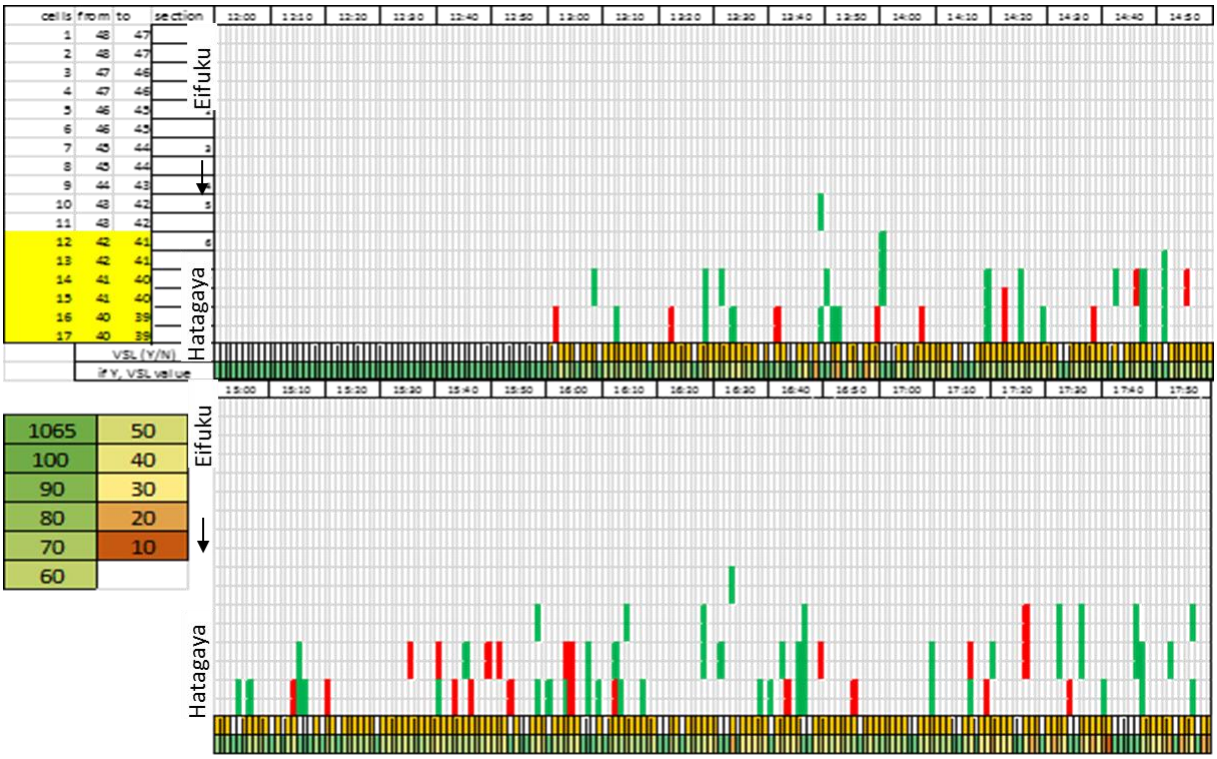


Figure 5. 16 Space-time diagram of change in crash risk before and after the application of VSL control using QL (cell 12 to 17)

From **Figure 5.16** it can be observed that the crash risk has improved at the targeted location (cells 9-17) in few cases, and deteriorated in others. Previously, QL based VSL control showed improvement of risk at the upstream of the targeted location, in case of DQN-based VSL control, improvement in the upstream cells were observed as well when the crash risk threshold was set to

≥10. The risk was improved due to lowering speed limit at most of the times, but increasing the limit caused improvement. This is also notable to mention that the action strategy was made in a way ($\pm 20, \pm 0,\ V_{free-flow}$) km/h so that the consecutive VSL actions taken does not have a huge difference causing a sudden disruption in the traffic flow. Another observation from **Figure 5.17 and Table 5.5** is that there were cases (during 17:20, 17:30, 16:00, 15:30-15:40) when the VSL control worsen the risk at the targeted cells. **Figure 5.17** and **Table 5.5** shows the similar analogy about the reduction of NER ≥10 at the targeted location (cells 12-17) and improvements in the upstream (cells 10 and 11) of it. However, the improvements at the targeted location seems much higher than the one found in the QL-based VSL control. In the table, a comparison between no control and control cases with crash risk = 50 is also shown. It is evident that if the threshold is set to 50, the model barely detects any condition as crash prone and hence VSL control was not triggered and there is almost no changes in crash risk.

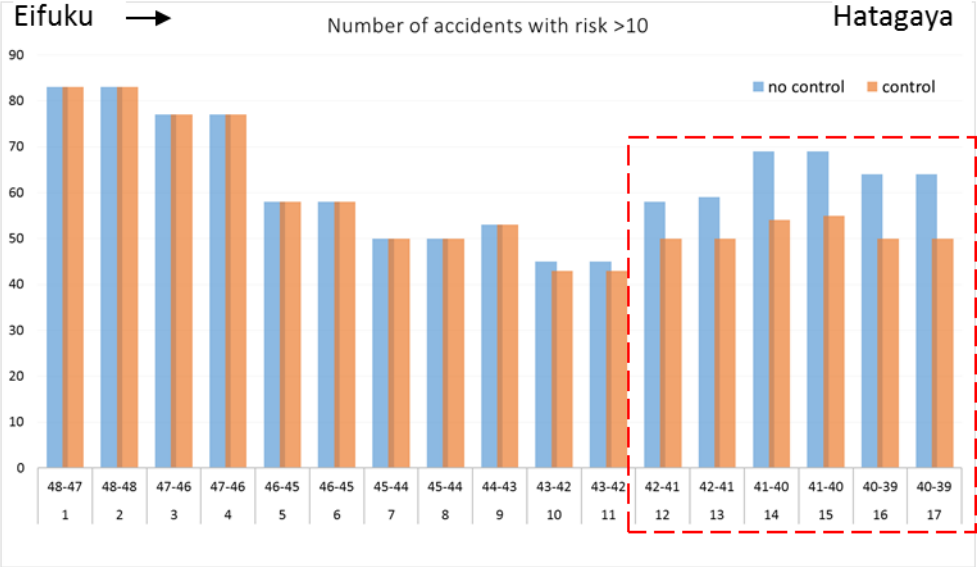


Figure 5. 17 Comparison of NER ≥10 with and without VSL Control (QL) (cell 9 to 17)

Table 5. 5 Comparison of NER ≥10 and NER = 50 (cell 12 to 17)

Cells	Section	No control	Control	Difference	No control	Control	Difference
		Risk = 50			Risk ≥10		
1	0	9	9	0	83	83	0
2		9	9	0	83	83	0
3	1	0	0	0	77	77	0
4		0	0	0	77	77	0
5	2	8	8	0	58	58	0
6		8	8	0	58	58	0
7	3	5	5	0	50	50	0
8		5	5	0	50	50	0
9	4	4	4	0	53	53	0
10	5	7	7	0	45	43	2
11		7	7	0	45	43	2
12	6	9	7	2	58	50	8
13		9	7	2	59	50	9
14	7	7	6	1	69	54	15
15		7	6	1	69	55	14
16	8	3	5	-2	64	50	14
17		3	5	-2	64	50	14

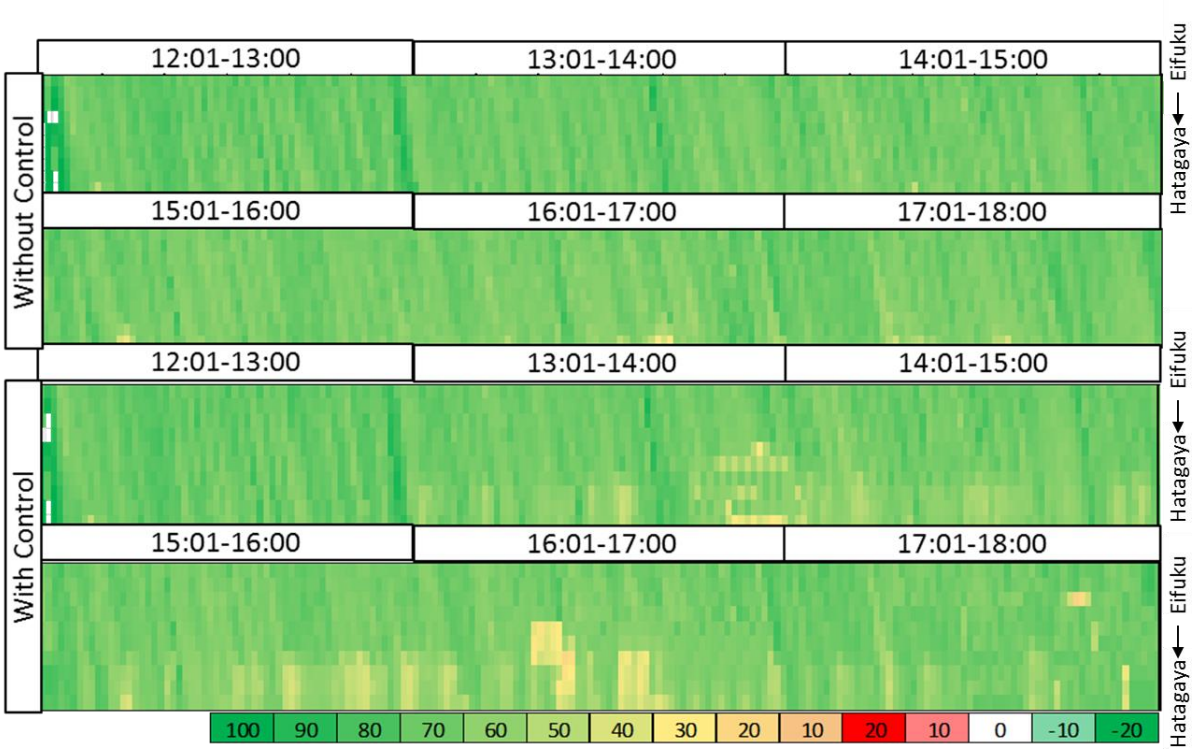


Figure 5. 18 Space-time diagram of speed during 12:01-17:59 at the entire section (cells 1-17) after employing DQN-based VSL control (bottom half) and without control (upper half)

Figure 5.18 shows a space-time diagram of the entire route during 12:01-17:59 when VSL was applied (bottom of the figure) and there was no speed control. While the goal is to improve crash safety, it is also important to consider the functionability of the route. If a capacity drop appears due to the VSL control, that might cause not only congestion but also increase the possibility of rear-end crashes. From diagram 5.20, few changes in the speed profile with the application of VSL control can be observed. The DQN model performed well not only in reducing NER ≥10, but also it did not generate any major fluctuation in the speed of the targeted location except for the slightly low speed occurred during 16:01-17:00. But, the speed remained well distributed overall even after employing the VSL control.

5.8 Chapter conclusion

In this chapter, RL-based VSL was introduced as intervention for the accident risk prevention. The reason for choosing RL-based VSL control is to utilize the model-free property of RL algorithm. As discussed in the introduction of this chapter, several previous studies have employed different

VSL strategies to improve crash risks, by implementing policies such as applying a VSL control value and incorporating functions for gradual increasing or decreasing of the VSL control speed etc. Few researchers even divided the study segment according to bottleneck location, VSL control location and transition from VSL control to bottleneck location to apply different VSL strategies to different segments. There is no concrete solution when it comes to deciding upon a strategy for VSL control. RL has been gaining popularity nowadays, especially among the video gaming community to discover an intelligent agent who can play video games on its own without any outside human player's help. There is no need for constructing a structured model to teach the agent with labeled input data, rather it learns in a model-free way by playing the game and learning to take the best possible actions to win from it experience.

This is an important property and could be utilized in many fields including transportation where decisions and policies needs to be made. Hence, in this thesis the RL method namely Q-learning and deep Q-learning is adopted. First, the classic Q-learning method that uses a Q-table to learn policies was used. It showed promising improvement in reducing NER ≥10. Although it caused a disruption in the speed of the vehicles leading to a congestion on the study location. Whilst this was a useful method for retrieving policy, the method fails when it comes to using continuous data and when the size of the state space increases. In order to overcome the scalability and the discretization issues, DQN-based RL was adopted for policy learning.

Another observation of trying two thresholds- risk = 50 and risk ≥10, showed that the agent was not able to recognize most of the crash prone situations when the threshold was set to 50. This resulted in a drastic deterioration of the safety of the targeted location. Whereas the threshold of risk ≥10 performed well in terms of identifying crash prone situations and in decreasing the number of crashes in the target section.

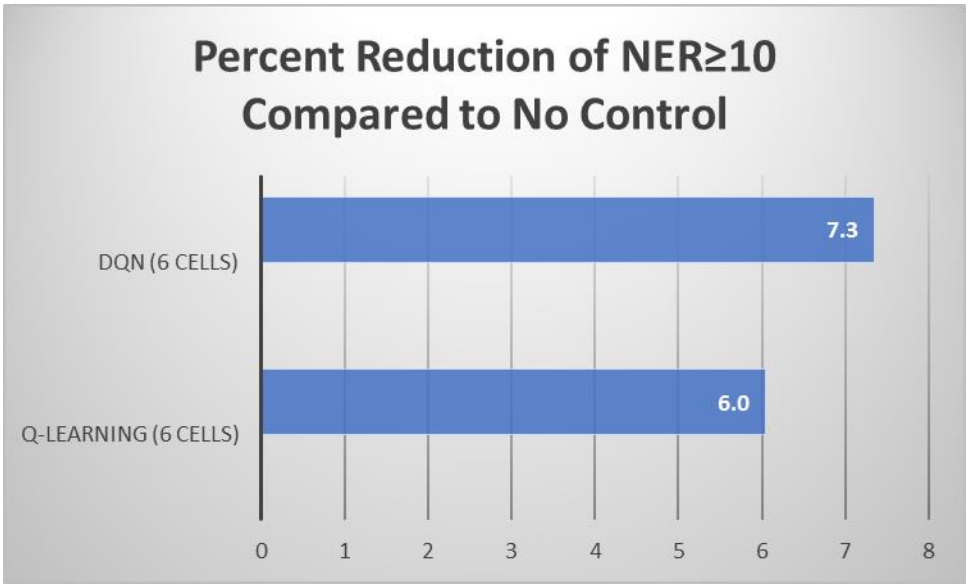


Figure 5. 19 Comparison among Q-learning (0.9 km), DQN (0.9 km)

Figure 5.19 the percent difference of NER ≥10. It is apparent that, between the QL and DQN with the same length (0.9 km), DQN performed better than the QL model. This is because of DQN's ability to take continuous traffic parameters and scalability. Another observation is, that the QL model applied a VSL strategy of speed change according to the action set, $a_t = \{\pm 10, \pm 0, V_{free-flow}\}\, km/h$, whereas DQN agent chose actions from the action set, $a_t = \{\pm 20, \pm 0, V_{free-flow}\}\, km/h$. Hence, the action strategy of incrementing speed at a rate of $20\, km/h$ depending on current states, proves to be better than an increment of $10\ km/h$.

5.9 Chapter References

Abdel-Aty M., Gayah V., 2010. Real-time crash risk reduction on freeways using coordinated and uncoordinated ramp metering approaches. Transportation Engineering 136, 410-423.

Abdel-Aty, M., and Wang, L., 2017. Implementation of variable speed limits to improve safety of congested expressway weaving segments in microsimulation, Transportation Research Procedia 27, 577-584.

Abdel-Aty, M., Cunningham, R.J., Gayah, V.V., Hsia, L., 2008. Dynamic variable speed limit strategies for real-time crash risk reduction on freeways. In: Transportation Research Record: Journal of the Transportation Research Board, No. 2078, Transportation Research Board of the National Academies, Washington, D.C. pp. 108-116.

Abdel-Aty, M., Dilmore, J., Dhindsa, A., 2006a. Evaluation of variable speed limits for real-time freeway safety improvement. Journal of Accident Analysis and Prevention 38(2), 335-345.

Abdel-Aty, M., Dilmore, M.J., Gayah, V.V., 2007a. Considering various ALINEA ramp metering strategies for crash risk mitigation on freeways under congested regime. Journal of Transportation Research, Part C 15(2), 113-134.

Abdel-Aty, M., Dilmore, M.J., Hsia, L., 2006b. Applying variable speed limits and the potential for Crash Migration. In: Transportation Research Record: Journal of the Transportation Research Board, No. 1953, Transportation Research Board of the National Academies, Washington, D.C., pp. 21-30.

Abdel-Aty, M., Pande, A., Lee, C., 2007b, Gayah, V., Santos, C. D., 2007b. Crash risk assessment using intelligent transportation systems data and real-time intervention strategies to improve safety on freeways. Journal of Intelligent Transportation Systems 11(3), 107-120.

Abdel-rahman, M., George, E. D., Hinton, G., E., "Acoustic Modeling using Deep Belief Networks", IEEE trans. On audio, speech, and language processing, 2010

Abdel-rahman, M., George, E. D., Hinton, G., E., "Acoustic Modeling using Deep Belief Networks", IEEE trans. On audio, speech, and language processing, 2010

Abdulhai, B., Pringle, R., Karakoulas, G.J., 2003. Reinforcement learning for true adaptive traffic signal control. Journal of Transportation Engineering, ASCE 129, 278-285. DOI: 10.1061/(ASCE)0733-947X(2003)129:3(278).

Allaby, P., Hellinga, B., Bullock, M., "Variable Speed Limits: Safety and Operational Impacts of a Candidate Control Strategy for Freeway Applications", IEEE Transactions on Intelligent Transportation Systems, Vol. 8, No. 4, pp. 671-680, 2007

Barto, A. G., Reinforcement Learning: An Introduction. Cambridge, MA, USA: MIT Press, 2003.

Barto, A., G., Sutton, R., S., Anderson, C., W., 'Neuronlike elements that can solve difficult learning control problems', IEEE Transactions on Systems, Man, and Cybernetics, 13:835-846. Reprinted in J. A. Anderson and E. Rosenfeld (eds.), Neurocomputing: Foundations of Research, pp. 535-549. MIT Press, Cambridge, MA (1988), 1983,

Bellman, R., E., 'A Markov decision processes', Journal of Mathematical Mechanics, 6:679-684, 1957b.

Bellman, R., E., 'Dynamic Programming', Princeton University Press, Princeton, 1957a.

Bengio, Y. "Deep Learning of Representations for Unsupervised and Transfer learning", JMLR: Workshop and Conference Proceedings 27:17–37, 2012.

Bengio, Y. "Deep Learning of Representations for Unsupervised and Transfer learning", JMLR: Workshop and Conference Proceedings 27:17–37, 2012.

Bengio, Y., "Learning Deep Architectures for AI", Foundations and Trends in Machine Learning. 2 (1): 1–127, 2009, doi:10.1561/2200000006.

Bengio, Y., "Learning Deep Architectures for AI", Foundations and Trends in Machine Learning. 2 (1): 1–127, 2009, doi:10.1561/2200000006.

Bengio, Y., LeCun, Y., Hinton, G., "Deep Learning". Nature. 521: 436–444. doi:10.1038/nature14539, 2015, PMID 26017442.

Bengio, Y., LeCun, Y., Hinton, G., "Deep Learning". Nature. 521: 436–444. doi:10.1038/nature14539, 2015, PMID 26017442.

Bengio, Y.; Courville, A.; Vincent, P. "Representation Learning: A Review and New Perspectives". IEEE Transactions on Pattern Analysis and Machine Intelligence.35 (8): 1798–1828, 2013, arXiv:1206.5538. doi:10.1109/tpami.2013.50.

Bengio, Y.; Courville, A.; Vincent, P. “Representation Learning: A Review and New Perspectives”. IEEE Transactions on Pattern Analysis and Machine Intelligence.35 (8): 1798–1828, 2013, arXiv:1206.5538. doi:10.1109/tpami.2013.50.

Bertsekas, D., P., Tsitsiklis, J., N., ‘Neuro-Dynamic Programming’, Athena Scientific, Belmont, MA, 1996.

Carlson, R.C., Papamichail, I., Papageogiou, M., 2011. Local feedback based mainstream traffic flow control on motorways using variable speed limits. IEEE Trans. Intell. Transp. Syst., 12(4), 1261-1276.

Clark, W., A., Farley, B., G., ‘Generalization of pattern recognition in a self-organizing system’, In Proceedings of the 1955 Western Joint Computer Conference, pp. 86-91, 1955.

Dahl,G., D. Yu, L. Deng, and A. Acero. Context-dependent, pre-trained deep neural networks for large vocabulary speech recognition. IEEE Transactions on Audio, Speech, & Language Processing, 20(1):30–42, January 2012.

Davarynejad, M., Hegyi, A., Vrancken, J., van den Berg, J., 2011. Motorway Ramp-Metering Control with Queuing Consideration using Q-learning, Proc. 14th IEEE Int. Conf. Intell. Transp. Syst., Washington, DC, USA, pp. 1652-1658.

Dechter, R., “Learning while searching in constraint-satisfaction-problems.” Proceedings of the Fifth National Conference on Artificial Intelligence (AAAI-86), Philadelphia, PA, August 1986, pp. 178–183

Dechter, R., “Learning while searching in constraint-satisfaction-problems.” Proceedings of the Fifth National Conference on Artificial Intelligence (AAAI-86), Philadelphia, PA, August 1986, pp. 178–183

Deng, L.; Yu, D. “Deep Learning: Methods and Applications”, Foundations and Trends in Signal Processing. 7 (3–4): 1–199. 2014, doi:10.1561/2000000039.

Deng, L.; Yu, D. “Deep Learning: Methods and Applications”, Foundations and Trends in Signal Processing. 7 (3–4): 1–199. 2014, doi:10.1561/2000000039.

Dynamic Traffic Management, the ASF experience, EasyWay Annual Forum, Lisboa, Nov. 17th, 2010 25.

El-Tantawy S., Abdulhai, B., 2010. Towards multi-agent reinforcement learning for integrated network of optimal traffic controllers. Transp. Letters 2(2), 89-110.

Glauner, P., Deep Convolutional Neural Networks for Smile Recognition (MSc Thesis). Imperial College London, Department of Computing, 2015, arXiv:1508.06535.

Glauner, P., Deep Convolutional Neural Networks for Smile Recognition (MSc Thesis). Imperial College London, Department of Computing, 2015, arXiv:1508.06535.

Glorot, X. and Bengio, Y., "Understanding the difficulty of training deep feed-forward neural networks", In Proceedings of Artificial Intelligence and Statistics (AISTATS). 2010.

Glorot, X. and Bengio, Y., "Understanding the difficulty of training deep feed-forward neural networks", In Proceedings of Artificial Intelligence and Statistics (AISTATS). 2010.

Hasselt, H.V., Guez, A., Silver, D., 2016. Google DeepMind. Deep Reinforcement Learning with Double Q-Learning, Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16), pp. 2094-2100.

Hasselt, H.V., Insights in Reinforcement Learning. PhD thesis, 2011. Utrecht University, the Netherlands.

Hinton, G., E, et al., "A Fast Learning Algorithm for Deep Belief Nets", Neural Computation 18, 1527–1554, 2006

Hinton, G., E, et al., "A Fast Learning Algorithm for Deep Belief Nets", Neural Computation 18, 1527–1554, 2006

Hinton, G., E., and Salakhutdinov, R., R., "Deep Boltzmann Machines", Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS), Clearwater Beach, Florida, USA,Volume 5 of JMLR: W&CP 5, 2009

Hinton, G., E., and Salakhutdinov, R., R., “Deep Boltzmann Machines”, Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS), Clearwater Beach, Florida, USA,Volume 5 of JMLR: W&CP 5, 2009

Hinton, G., E., and Salakhutdinov, R., R., “Reducing the Dimensionality of Data with Neural Networks”, Science, Vol. 313, Pg. 504- 507, 2006;

Hinton, G., E., and Salakhutdinov, R., R., “Reducing the Dimensionality of Data with Neural Networks”, Science, Vol. 313, Pg. 504- 507, 2006;

Hoogen,V., D, Smulders, E. and S., Advies, H., Control by variable speed signs: the results of the Dutch Experiment, IEE Conf. on Road Monitoring & Control, Apr. 26-28, 1994

Hossain, M., Muromachi, Y., 2013. Real-time crash prediction model for the ramp vicinities of urban expressway. IATSS Research 37(1), 68-79. http://doi.org/10.1016/j.iatssr.2013.05.001.

Howard, R., ‘Dynamic Programming and Markov Processes’, MIT Press, Cambridge, MA, 1960.

Ian Goodfellow and Yoshua Bengio and Aaron Courville, “Deep learning”, MIT press book, 2016.

Ian Goodfellow and Yoshua Bengio and Aaron Courville, “Deep learning”, MIT press book, 2016.

Isele, D., Rahimi, R., Cosgum, A., Subramanian, K., Fujimura, K., 2018. Navigating occluded intersections with autonomous vehicles using deep reinforcement learning. https://arxiv.org/pdf/1705.01196.pdf

Katrakazas, C., Quddus, M.A., Chen, W.H., 2017. A simulation study of predicting conflict-prone traffic conditions in real-time. Presented at the Transportation Research Board 96th Annual Meeting, Washington D.C., USA, 8-12 January 8-12 2017.

Klopf, A., H., ‘A comparison of natural and artificial intelligence’, SIGART Newsletter, 53:11-13, 1975.

Klopf, A., H., ‘Brain function and adaptive systems-A heterostatic theory’, Technical Report AFCRL-72-0164, Air Force Cambridge Research Laboratories, Bedford, MA. A summary appears

in Proceedings of the International Conference on Systems, Man, and Cybernetics. IEEE Systems, Man, and Cybernetics Society, Dallas, TX (1974), 1972.

Klopf, A., H., 'The Hedonistic Neuron: A Theory of Memory, Learning, and Intelligence', Hemisphere, Washington, DC, 1982.

Krizhevsky, A., Sutskever, I., Hinton, G., 2012. ImageNet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. 25, 1106-1114.

Large-Scale Deep Learning for Intelligent Computer Systems", talk in Google for Startups Campus Korea, 2016

Large-Scale Deep Learning for Intelligent Computer Systems", talk in Google for Startups Campus Korea, 2016

LeCunn et al., "Gradient-based learning applied to document recognition.", Proceedings of IEEE, 1998

LeCunn et al., "Gradient-based learning applied to document recognition.", Proceedings of IEEE, 1998

Lee, C., Abdel-Aty, M., 2008. Testing effects of warning messages and variable speed limits on driver behavior using driving simulator. In: Transportation Research Record: Journal of the Transportation Research Board, No. 2069, Transportation Research Board of the National Academies, Washington, D.C., pp. 55-64.

Lee, C., Hellinga, B. F., Saccomanno, F. , "Assessing safety benefits of variable speed limits", Transportation Research Record 1897, TRB, National Research Council, Washington, D.C, Vol. 1, No. 1, pp. 183-190, 2004

Lee, C., Hellinga, B., and Ozbay, K., 2006. Quantifying effects of ramp metering on freeway safety. Journal of Accident Analysis and Prevention 38(2), 279-288.

Lee, C., Hellinga, B., Saccomanno, F., 2004. Evaluation of variable speed limits to improve traffic safety. Transport. Res. Part C: Emerg. Technol 14(3), 213-228.

Li, Z., Liu, P., Wei Wang, Chengcheng Xu, 2014. Development of a control strategy of variable speed limits to reduce rear-end collision risks near freeway recurrent bottlenecks, J. Central South Univ., 21(6), 2526-2538.

Li, Z., Liu, P., Xu, C., Duan, H., and Wang, W. Limit Control Strategy to Reduce Traffic Congestion at Freeway Recurrent Bottlenecks, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, VOL. 18, NO. 11, NOVEMBER 2017.

Li, Z., Liu, P., Xu, C., Duan, H., and Wang, W., 2016. Optimal mainline variable speed limit control to improve safety on large-scale freeway segments. Computer-Aided Civil and Infrastructure Engineering 31, 366-380.

Li, Z., Liu, P., Xu, C., Duan, H., and Wang, W., 2017. Reinforcement learning-based variable speed limit control strategy to reduce traffic congestion at freeway recurrent bottlenecks. IEEE Transactions on Intelligent Transportation Systems 18(11).

Liou, Cheng-Yuan; Cheng, Wei-Chen; Liou, Jiun-Wei; Liou, Daw-Ran (2014). "Autoencoder for words". Neurocomputing. 139: 84–96. doi:10.1016/j.neucom.2013.09.055.

Liu, M., Chen, Y., 2017. Predicting real-time crash risk for urban expressways in China. Mathematical Problems in Engineering, Article ID: 6263726.

Lu, X.Y., Varaiya, P., Horowitz, R., Su, D., and Shladover, S., 2011. Novel freeway traffic control with variable speed limit and coordinated ramp metering. Transp. Res. Rec., 2229, 55-65.

Michie, D., 'Experiments on the mechanisation of game learning, characterization of the model and its parameters', Computer Journal, 1:232-263, 1963.

Michie, D., Chambers, R., A., 'BOXES: An experiment in adaptive control', In E. Dale and D. Michie (eds.), Machine Intelligence 2, pp. 137-152. Oliver and Boyd, Edinburgh, 1968.

Minsky, M., L., 'Steps toward artificial intelligence. Proceedings of the Institute of Radio Engineers', 49:8-30. Reprinted in E. A. Feigenbaum and J. Feldman (eds.), Computers and Thought, pp. 406{450. McGraw-Hill, New York, (1963), 1961.

Minsky, M., L., ‘Theory of Neural-Analog Reinforcement Systems and Its Application to the Brain-Model Problem’, Ph.D. thesis, Princeton University, 1954.

Mnih, V., “Playing Atari with Deep Reinforcement Learning”, arXiv:1312.5602, 2013.

Mnih, V., “Playing Atari with Deep Reinforcement Learning”, arXiv:1312.5602, 2013.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M., 2013. Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D., 2015. Human-level control through deep reinforcement learning. Nature 518, 529-533.

Mohamed, A., G. Dahl, and G. Hinton. Acoustic modeling using deep belief networks. IEEE Transactions on Audio, Speech, & Language Processing, 20(1), January 2012.

Mohamed, A., G. Hinton, and G. Penn. Understanding how deep belief networks perform acoustic modelling. In Proceedings of International Conference on Acoustics Speech and Signal Processing (ICASSP). 2012.

Narendra, K., S., Thathachar, M., A., L., ‘Learning Automata: An Introduction’, Prentice-Hall, Englewood Cliffs, NJ, 1989.

Narendra, K., S., Thathachar, M., A., L., ‘Learning automata|A survey’, IEEE Transactions on Systems, Man, and Cybernetics, 4:323-334, 1974.

Park, H., Haghani, A., Samuel, A., Knodler, M.A., 2018. Real-time prediction and avoidance of secondary crashes under unexpected traffic congestion. Accident Analysis and Prevention 112, 39-49.

Pavlov, P., I., ‘Conditioned Reflexes’. Oxford University Press, London, 1927.

Rezaee, K., Abdulhai, B., Abdelgawad, H., 2012. Application of Reinforcement Learning with Continuous State Space to Ramp Metering in Real-World Conditions. Proc. 15th IEEE Int. Conf. Intell. Transp. Syst. (ITSC), Sep 2012, pp. 1590-1595.

Rivey, F., Evaluation of the Dynamic Speed Limit System on the A13 motorway in France, EasyWay Annual Forum, Lisboa, Nov. 17th, 2010

Rosenblatt, F., 'Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms', Spartan Books, Washington, DC, 1962.

Roy, A., Hossain, M., and Muromachi, Y., 2018a. Enhancing the prediction performance of real-time crash prediction model: A Cell Transmission-Dynamic Bayesian Network approach, 97th Annual Meeting of Transportation Research Board, Washington D.C.

Samuel, A., L., 'Some studies in machine learning using the game of checkers', IBM Journal on Research and Development, 3:211-229. Reprinted in E. A. Feigenbaum and J. Feldman (eds.), Computers and Thought, pp. 71-105. McGraw-Hill, New York (1963), 1959.

Schmidhuber, J., "Deep Learning". Scholarpedia. 10 (11): 32832, 2015, doi:10.4249/scholarpedia.32832.

Schmidhuber, J., "Deep Learning". Scholarpedia. 10 (11): 32832, 2015, doi:10.4249/scholarpedia.32832.

Song, H.A.; Lee, S. Y., "Hierarchical Representation Using NMF". Neural Information Processing. Lectures Notes in Computer Sciences. 8226. Springer Berlin Heidelberg. pp. 466–473, 2013, doi:10.1007/978-3- 642-42054-2_58. ISBN 978-3-642-42053-5.

Song, H.A.; Lee, S. Y., "Hierarchical Representation Using NMF". Neural Information Processing. Lectures Notes in Computer Sciences. 8226. Springer Berlin Heidelberg. pp. 466–473, 2013, doi:10.1007/978-3- 642-42054-2_58. ISBN 978-3-642-42053-5.

Stuart Russel and Peter Norvig, "Artificial Intelligence: A Modern Approach", Pearson Publications, 2010.

Stuart Russel and Peter Norvig, “Artificial Intelligence: A Modern Approach”, Pearson Publications, 2010.

Sutton, R. S. and Barto, A. G., Reinforcement Learning: An Introduction. Cambridge, MA, USA: MIT Press, 1998.

Sutton, R., S., ‘Temporal Credit Assignment in Reinforcement Learning’, Ph.D. thesis, University of Massachusetts, Amherst, 1984.

Sutton, R., S., and Barto, A., G., ‘Reinforcement learning: an introduction’, MIT press, 2017.

Sutton, R.,S., and Barto, A.,G. 1998. Introduction to Reinforcement Learning. MIT Press.

Thorndike, E., L., ‘Animal Intelligence’. Hafner, Darien, CT, 1911.

Thrun, S., Shwartz, A., Lee, C.B., 1993. Issues in Using Function Approximation for Reinforcement Learning. Proceedings of the 1993 Connectionist Models Summer School, Lawrence Erlbaum Publisher, Hillsdale, NJ.

Turing, A., M., ‘Intelligent Machinery, A Heretical Theory.’ The Turing Test: Verbal Behavior as the Hallmark of Intelligence, 105, 1948.

Wang, Z., Schaul, T., Hessel, M., Hasselt, H.V., Lanctot, M., Freitas, N., 2016. Google DeepMind London UK. Dueling Network Architectures for Deep Reinforcement Learning, Proceedings of the 33rd International Conference on Machine Learning, JMLR: W & CP 48, New York, NY, USA.

Watkins, C. J. C. H. and Dayan, P., “Q-learning,” Mach. Learn., vol. 8, nos. 3–4, pp. 279–292, 1992.

Watkins, C. J. C. H., 1989. Learning from delayed rewards. PhD thesis, University of Cambridge, England.

Watkins, C., J., C., H., ‘Learning from Delayed Rewards’, Ph.D. thesis, Cambridge University, 1989.

What data scientists should know about deep learning", talk in ExtractConf, 2015.

What data scientists should know about deep learning", talk in ExtractConf, 2015.

Widrow, B., Hoff, M., E., 'Adaptive switching circuits', In 1960 WESCON Convention Record Part IV, pp. 96-104. Institute of Radio Engineers, New York. Reprinted in J. A. Anderson and E. Rosenfeld, Neurocomputing: Foundations of Research, pp. 126-134. MIT Press, Cambridge, MA (1988), 1960.

Witten, I., H., 'An adaptive optimal controller for discrete-time Markov environments', Information and Control, 34:286-295, 1977.

Wu, Y., Abdel-Aty M., 2018. Developing an algorithm to assess the rear-end collision risk under fog conditions using real-time data, Transportation Research C 87, 11-25.

Yang, K., Wang, X., Quddus, M., Yu, R., 2018. Deep learning for real-time crash prediction on urban expressways, 97th Annual Meeting of Transportation Research Board, Washington D.C.

Yu, R., and Abdel-Aty, M., 2014. An optimal variable speed limits system to ameliorate traffic safety risk, Journal of Transportation Research: Part C 46, 235-246. DOI: 10.1016/j.trc.2014.05.016.

Zhu, F. and Ukkusuri, S. V. "Accounting for dynamic speed limit control in a stochastic traffic environment: A reinforcement learning approach," Transp. Res. C, Emerg. Technol., vol. 41, pp. 30–47, Apr. 2014.

Zhu, F., Ukkusuri, S.V., 2014. Accounting for dynamic speed limit control in a stochastic traffic environment: A reinforcement learning approach. Transportation Research Part C 2014, 41, 30-Khandoker, B., Kattan, L., VARIABLE SPEED LIMIT: AN OVERVIEW, Transportation Letters The International Journal of Transportation Research • January 2015.

CHAPTER 6

CONCLUSIONS AND FUTURE SCOPE OF WORK

6.1 The Objectives

Back to chapter 1 where the objectives of this research was established as the following questions-

- How to predict crash risk in real-time?
- In what way the universality of a RTCPM can be improved?
- How to decide on a policy for an intervention to prevent crash?

In order to achieve these objectives, in chapter 3, CTM and modified CTM were applied to the route 3 and 4 to generate uniformly distributed simulated detectors. To answer the first question of the objectives, BN and DBN- based RTCPMs were built with both fixed detector data and CTM generated data, and a comparison was made. For the last objective, in chapter 5 a deep RL-based model-free intervention method was employed to prevent possible crash predicted by the RTCPM. In the next section, the solutions or how the models performed overall is discussed in brief to draw a conclusion.

6.2 Discussion on results

6.2.1 The RTCPMs: with CTM and fixed detectors

In this study it has been attempted to establish a method for using a uniformly and densely distributed simulated detector layout to create a framework for developing a real-time crash prediction model which is transferrable over space. A simple method of the CTM was employed in the study route to generate the simulated detector data. It was found that the CTM-based method could reproduce values of the traffic flow variables with an average error of 13% where the speed data showed higher mean percent error (MPE) as compared to flow and occupancy data. One of the reasons is that the basic FD does not allow us to control speed resulting in simulated speed values unresponsive of the traffic situation. However, with a flexible FD the speed can be controlled which could generate speed data with greater accuracy. Moreover, recent studies

identified (Coifman, 2014; Lu, 2010) speed data to be the most vulnerable data considering the detector type, traffic state, and the quality of other traffic variables.

In order to apply the CTM to an existing road network, the Courant–Friedrichs–Lewy (CFL) condition needs to be met. According to the CFL condition, the cell length must be bound to the product of free-flow speed and the time step. This is to ensure that the vehicle does not pass more than one cell in a single time step. But, in practicality, while simulating, few vehicles might tend to pass multiple cells in a time step. To avoid that, the cell length must be considered longer than the exact calculated value. In this thesis, the cell length was estimated at 150 meters, which was kept constant along the 2.5km long study segment. In order to maintain the CFL condition $\left(\Delta t * \frac{v_f}{\Delta x} \leq 1\right)$(Courant, R.; Lewy, H.; Friedrichs, 1928), the cell length should be kept longer than actual calculated one.

The performance of BN- and DBN-based RTCPMs built with traffic data from both fixed detectors and CTM has been also investigated with 16 RTCPMs. These 16 models with different information variables suggested the most influential variables to be downstream flow, speed, and occupancy, difference of upstream and downstream flow. The comparison results between loop detector and CTM generated BN and DBN models are show below-

1. Comparison between BN and DBN with loop detector data showed that 7 (out of 16) DBN models performed better, which means, BN models dominated with crash prediction accuracy with loop detector data.

2. Comparison between BN and DBN with CTM generated data showed that 6 (out of 16) DBN models performed better, which means, BN models dominated with crash prediction accuracy with CTM generated data.

3. Comparison of BN models with loop detector and CTM generated data showed that (11 out of 16) CTM models performed better, which means, CTM dominated over loop detector data.

4. Comparison of DBN models with loop detector and CTM generated data showed that (9 out of 16) CTM models performed better, which means, CTM dominated over loop detector data.

Although, the results showed that the CTM based RTCPMs performed slightly better than the loop detector based models, it is too early to draw significant conclusions. In any case, it is certain that these preliminary results indicate that CTM is able to generate reliable traffic parameters to overcome the transferability and facilitate a solution to the missing data problem of the future universal RTCPMs.

The comparison between BN and DBN-based RTCPM built with CTM generated data showed that the BN-based model outperforms DBN-based models in most of the cases. This might be because of lack of training data while constructing the RTCPMs. The author believes that the overall prediction could be improved with the inclusion of more training data.

Four RTCPMs were constructed with the most influential traffic parameters found from chapter 3 (downstream speed, downstream occupancy and difference of up and downstream flow) with route 4 data. The models were validated with both route 4 and route 3 data to investigate transferability of the models. The models' performances were tested with four thresholds- 5, 10, 15 and 20%. The results showed that the model's ability of identifying crash likelihood decreases with increasing threshold. In case of model-3, the crash likelihood decreased with increase of threshold, but yet kept over 20%. Hence, this model was selected for the VSL control later on.

After investigating the transferability of the four RTCPMs, model- 3 showed consistency in prediction accuracy of crash likelihood at all thresholds and both validation. For model- 3, although the highest accuracy was found at threshold 5 and 10 (when validated with route 4), the overall accuracy of model-3 at threshold 5 (49%) was less compared to threshold 10 (60%). So, in the later chapters of this thesis, model-3 will be used as the RTCPM and crash risk threshold of 10% will be the threshold for intervention decision making.

6.2.2 The intervention: DRL-based VSL control

In this chapter, RL-based VSL was introduced as intervention for the accident risk prevention. The reason for choosing RL-based VSL control is to utilize the model-free property of RL algorithm.

RL has been gaining popularity nowadays, especially among the video gaming community to discover an intelligent agent who can play video games on its own without any outside human player's help. There is no need for constructing a structured model to teach the agent with labeled input data, rather it learns in a model-free way by playing the game and learning to take the best possible actions to win from it experience.

This is an important property and could be utilized in many fields including transportation where decisions and policies needs to be made. Hence, in this thesis the RL method namely Q-learning and deep Q-learning is adopted. First, the classic Q-learning method that uses a Q-table to learn policies was used. It showed promising improvement in reducing the number of events with risk ≥10 (NER ≥10). Whilst this was a useful method for retrieving policy, the method fails when it comes to using continuous data and when the size of the state space increases. In order to overcome the scalability and the discretization issues, DQN-based RL was adopted for policy learning.

According to our analytical results, it is apparent that, between the QL and DQN with the same length (0.9 km), DQN performed better than the QL model. This is because of DQN's ability to take continuous traffic parameters and scalability

Another observation of trying two thresholds- risk = 50 and risk ≥10, showed that the agent was not able to recognize most of the crash prone situations when the threshold was set to 50. This resulted in a drastic deterioration of the safety of the targeted location. Whereas the threshold of risk ≥10 performed well in terms of identifying crash prone situations and in decreasing the number of crashes in the target section.

The comparison of the performance of QL and DQN-based models can be done from two point of views- comparison with non-RL based VSL control and RL-based VSL control. From the previous studies discussed in section 5.2, it is evident that the application of VSL is an ad-hoc basis. The VSL strategies and control measurement is highly dependent on the traffic on the particular road network. RL-based VSL control has the advantage of adopting various traffic conditions and has the ability to apply VSL accordingly due to its model-free structure. Another way of comparison is between two RL-based VSL methods applied in this thesis: QL and DQN.

QL is the basic reinforcement learning method and DQN is a combination of QL and deep learning (neural network). According to Hasselt et al., (2016), QL algorithm may overestimate action values under certain conditions, and another deep learning method such as Double DQN can help reduce the overestimation resulting in better learning. Another study (Li et al., 2017) showed that an offline QL-based VSL control reduced travel time by 49.34% in stable demand scenario compared to feedback-based VSL control. In this study, the QL and DQN-based VSL control were employed and it was found that the DQN-based VSL control outperformed QL-based VSL control method by 1.3% in terms of safety improvement. Since, the DQN has the ability of including continuous traffic data and has added advantage of random sampling from experience reply, it is expected that DQN would perform better given the learning rate and discount factor etc. remains the same. However, QL uses a set of pre-defined combinations of states and actions to iterate for each time step, hence the number of iterations can be limited by the state-action tuple. On the other hand, the DQN is dependent on the number of layers and the number of neurons and the rate of experience reply. The performance and the execution time (or, iteration number) is dependent on the combinations of the mentioned factors. For example, the deeper the layers are, the better would be the performance but at the cost of longer execution time. It takes several trials to decide upon a set of factors (i.e. neurons, layers, learning rate etc.). Considering the time and simulation complexities, DQN takes longer time and more effort which might give an impression that QL is a better choice. But, QL's lack of scalability or inability to incorporate large scale continuous data is a big issue while handling complicated traffic data. Once established, DQN can be used for feeding more data with less trials to decide the more or less optimum values of the factors. Hence, in the long run, DQN will be a better choice over QL.

No similar study is currently available to compare the significance of the difference of the performances between QL and DQN-based VSL control to reduce crash risk. As learning rate, number of study segment, iteration numbers influence the agent's decision of selecting actions, further study is required to investigate with several scenarios, including additional objective functions such as TTT, and other RL-based algorithms such as Asynchronous Actor-Critic Agents (A3C) or Long-short-term memory (LSTM) methods etc.

6.3 Future scope of work

1. Transferability of an RTCPM can be tested in three ways- terms of detector layout, adaptability and fading properties of the algorithm of the model. One of the properties for transferability is achieved by generating uniformity in detector layout by using CTM. The other properties i.e. adaptability and fading can be attained by employing machine learning algorithms like Bayesian network, K-nearest neighbors (KNN) etc. (Hossain and Muromachi, 2011, Katrakazas et al, 2017). These machine learning algorithms has an inherent ability to modify it to become transferable. However, the KNN was referred as performing in slow speed and BN requires high density data. In this thesis, BN and DBN was employed as traffic data was available for every minute (24h) and for six months. Once the RTCPM is built, the model would run based on the data it was trained on and with the addition of new data, the model updates itself and fades away the historical data. In future this features should be tested while investigating transferability of a model. Additionally, a crash prediction model of an expressway not only influenced by the traffic data, but also affected by the urban design (Hossain et al, 2018) and its surrounding environment, road geometry etc. A study ranked the predictive power of various factors by Granger causality analysis, and established the order of crash predictive power as traffic flow > traffic accident > geographical Position > weather + air quality + holiday + time period, the (Ren et al, 2017). Although, traffic parameters are the most influential ones, road geometry parameters should be considered as information variables while constructing RTCPMs in future to ensure transferability.

2. In order to implement VSL, in thesis, the driving compliance of 100% is assumed. However, the assumption might be far from the reality. In case of the study location, route 4 of Tokyo metropolitan expressway, there is no VSL control available currently. Hence, verification of driver's compliance is not possible for this thesis.

   Driver's compliance or, how drivers respond after activating VSL (or, posted speed limit) is a significant issue which must be confirmed by comparing it with the observed speed value after posted speed limit from the field data. This ensures the effectiveness and real-world applicability of the VSL control.

One way to adopt driver's compliance into VSL control could be done by observing the driver's driving behavior. In real –world, a driver might choose to speed up after observing the posted speed limit in an unsaturated condition (Li et al., 2017). Hence, the difference between posted speed and the observed 'after speed' can be added to the posted speed limit which can be the revised VSL value, $V_{SL}(new) = \min\{V_F, V_{SL} + \Delta V_{overspeed}\}$. After simulating with the revised posted limit, the model's performance can be checked with observed data. Based on the result, some kind of incentives can be introduced, so that drivers responds to the posted VSL value. However, this thesis did not consider driver's compliance as an objective. The study area, route 3 and route 4 of Tokyo metropolitan expressway does not consist VSL control till date. Hence, the driver's actual speeds after posted VSL cannot be measured to incorporate the 'over speed' value into the simulation and validation.

3. With the advancement of hardware capacities, the use of RL models has increased tremendously (Chollet, F, 2016) which makes them capable of incorporating huge database. In the field of transportation RL is being used for active traffic management through congestion analysis (Rezaee et al., 2012), reducing traffic gridlock (El-Tantawy et al., 2010), even for VSL control for reducing travel time and vehicle emission (Zhu and Ukkusuri, 2014), in eliminating traffic congestions at recurrent bottlenecks (Li et al, 2017). In terms of reducing crash risk, few studies were conducted such as support vector machine (SVM) was used for crash prediction (Sun et al, 2017). Deep neural network (DNN) was used for predicting crash-prone traffic conditions (Kui et al., 2018) where they found that DNN can predict 63-65% of the crashes with 5% false alarm rate. Furthermore, it also addressed the class- balancing issue of the training data and concluded that the prediction performance degrades with the increasing size of balanced data. Another study by (Dong et al, 2018) developed a deep learning model which includes an additional regression layer utilizing multivariate negative binomial (MVNB) model. This model showed improved performance in comparison to deep learning model without the additional regression layer and the SVM model. A study by Ren et al, 2018 applied long-short-term memory (LSTM) method to predict crash risk based on frequency of crash risk to capture temporal-spatial patterns of crashes.

   The two basic form of reinforcement learning- QL and DQN was employed in this thesis. The QL utilizes a Q-table which includes all the traffic states values in it. The QL agent's target is

to maximize Q-value by selecting appropriate speed values in certain traffic conditions. On the other hand, DQN utilizes the function approximation capability of neural network by forward-and-back propagating the network parameters (e.g. weights, biases etc.) to adjust the parameters in a way so that appropriate action which has the highest Q-value gets selected.

4. Several studies (Lyles et al., 2004, Soriguera et al., 2013, Li et al., 2017) suggested that the for any VSL control implemented, it would not affect the flow elsewhere in the network, or the overall efficiency of the network. To ensure that, additional objectives such as total time spent (TTS), delays on queues (congestion control) etc. can be formulated in addition to crash risk reduction while implementing the VSL control. Furthermore, the traffic states before and after an event or incident can be observed and considered as a precursor, or state set in the RL-based VSL control.

5. In the past, various RTCPMs have been proposed which can broadly divided into- various types of logistic regression models, neural networks, Bayesian networks, and classifying methods, such as, classification and regression trees, support vector machine (SVM) or simple rule based classifier. From knowledge generation perspective, Bayesian network and classification based methods have advantages over other methods. Both the methods have graphical representations, making the interrelationship among variables easy to comprehend. Classification and regression trees mainly explains the direction of classification of an observation. For example, 'crash' can be a dependent variable and the method can identify combinations of the observations which are more likely to be associated with the crash. SVM models Li et al. (2012) can also evaluate the observation parameters in terms crash or injury severity which proved to have a decent accuracy considering roadway geometry and weather conditions (Qu et al., 2012). These methods can be considered in future RTCPM construction.

   Other DL (Deep Learning) methods such a LSTM, RNN can be used for RTCP, where the AI-based traffic monitoring camera can capture image data of individual vehicle and generate data such as- vehicle type, vehicle capacity, number plate etc. With the help of these data and applying AI, traffic information like trip generation information, traffic dynamics can be understood.

6. In a recent study (Yuan et al., 2019) Bayesian conditional logistic models were developed by incorporating the Bluetooth, adaptive signal control, and weather data in addition to the traditional traffic data to predict crash in real-time, which indicated the most influential parameters to predict crash and for application of integrated active traffic management.

7. The data can be collected using Bluetooth sensors, which can detect Bluetooth enabled vehicles using the fixed censors along roadside. This way, individual driver's behavior, origin-destination, speed, occupancy etc. can be extracted from the Bluetooth sensors with precision. This has been applied in Chuo expressway in Japan, some arterial roads in Australia and Bangkok.

8. This goes without saying that in order to ensure throughput of a road network, the outflow of a bottleneck area needs to be controlled so that the inflow entering the bottleneck remains lower than the outflow from that bottleneck area until the discharge capacity is restored (Kerner, 2007). This can be achieved with traffic control measures such as: ramp metering that limit the number of vehicles entering the bottleneck or/and VSL that reduce the speed of traffic to delay it from entering the bottleneck. Hence, these intervention methods must be considered in the proactive traffic control measurement in the future.

9. One area of future work will be to the investigation of the optimum location of the simulated detectors or cells by experimenting with different cell lengths and different locations (50, 100, and 150 m upstream and downstream). Moreover, other than freeway stretches, the model in future can be upgraded to incorporate ramps rather than considering only the basic freeway segments. The RTCPM is highly dependent on the quantity as well as the quality of the traffic flow data. Therefore, it is recommended that more case studies and different time-steps should be incorporated to ensure an efficient model that can be implemented in real-time.

10.4 Chapter References

D Dong, C., Chunfu, S., Juan, L., Zhihua, X., An Improved Deep Learning Model for Traffic Crash Prediction, Journal of Advanced Transportation, Volume 2018, Article ID 3869106, 2018.

Hado van Hasselt , Arthur Guez,andDavid Silver, Deep Reinforcement Learning with Double Q-Learning, Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)2094, 2016.

"Kerner B. S. (2007). ""Study of freeway speed limit control based on three-phase theory.""

Transportation Research Record No. 1999, pp. 30-39."

Li, X., Liu, P., Wang, W., & Xu, C. (2012). Using Support Vector Machine Models for Crash Injury Severity Analysis. Accident Analysis and Prevention. 45, pp. 478-486

"Lyles, R.W., Taylor, W.C., Lavansiri, D. and Grosskalaus, J. (2004). ""A Field Test and

Evaluation of Variable Speed Limits in Work Zones."" Proceedings of Transportation

Research Board 83rd Annual meeting, Washington, D.C."

Ping Sun ; Guimu Guo ; Rongjie Yu, Traffic crash prediction based on incremental learning algorithm, IEEE 2nd International Conference on Big Data Analysis (ICBDA), 2017.

Qu, X., Wang, W., Wang, W., & Liu, P. (2012). Real-time Freeway Sideswipe Crash Prediction by Support Vector Machine, IET Intelligent Transport Systems , Vo 7, Iss 4, 445–453. http://doi.org/10.1049/iet-its.2011.0230

Ren, H., Song, Y., Wang, J., Hu, Y., Lei, J., A Deep Learning Approach to the Citywide Traffic Accident Risk Prediction, Cornell University, 2018. arXiv:1710.09543.

Ren, H., Song, Y., Liu, J., Hu, Y., and Lei, J., A Deep Learning Approach to the Prediction of Short-term Traffic Accident Risk, 2017. arXiv preprint arXiv:1710.09543.

"Soriguera F., Torné J. M. and Rosas D. (2013). ""Assessment of Dynamic Speed Limit

Management on Metropolitan Freeways."" Journal of Intelligent Transportation Systems:

Technology, Planning, and Operations, vol. 17, no. 1, pp. 78-90."

Yang, K. Wang, X., Quddus, M.,Yu, R., Deep Learning for Real-Time Crash Prediction on Urban Expressways, Transportation Research Board 97th Annual Meeting, 2018.

Yuan, J., Abdel-Aty, M., Wang, L., Lee, J., Yu, R., Wang, X., UTILIZING BLUETOOTH AND ADAPTIVE SIGNAL CONTROL DATA FOR URBAN ARTERIALS SAFETY ANALYSIS. Presented at the 96th TRB Annual meeting, 2017.

Zhibin Li, Pan Liu, Chengcheng Xu, Hui Duan, and Wei Wang, Reinforcement Learning-Based Variable Speed Limit Control Strategy to Reduce Traffic Congestion at Freeway Recurrent Bottlenecks, IEEE Transactions On Intelligent Transportation Systems, Vol. 18, No. 11, November 2017.

APPENDIX I

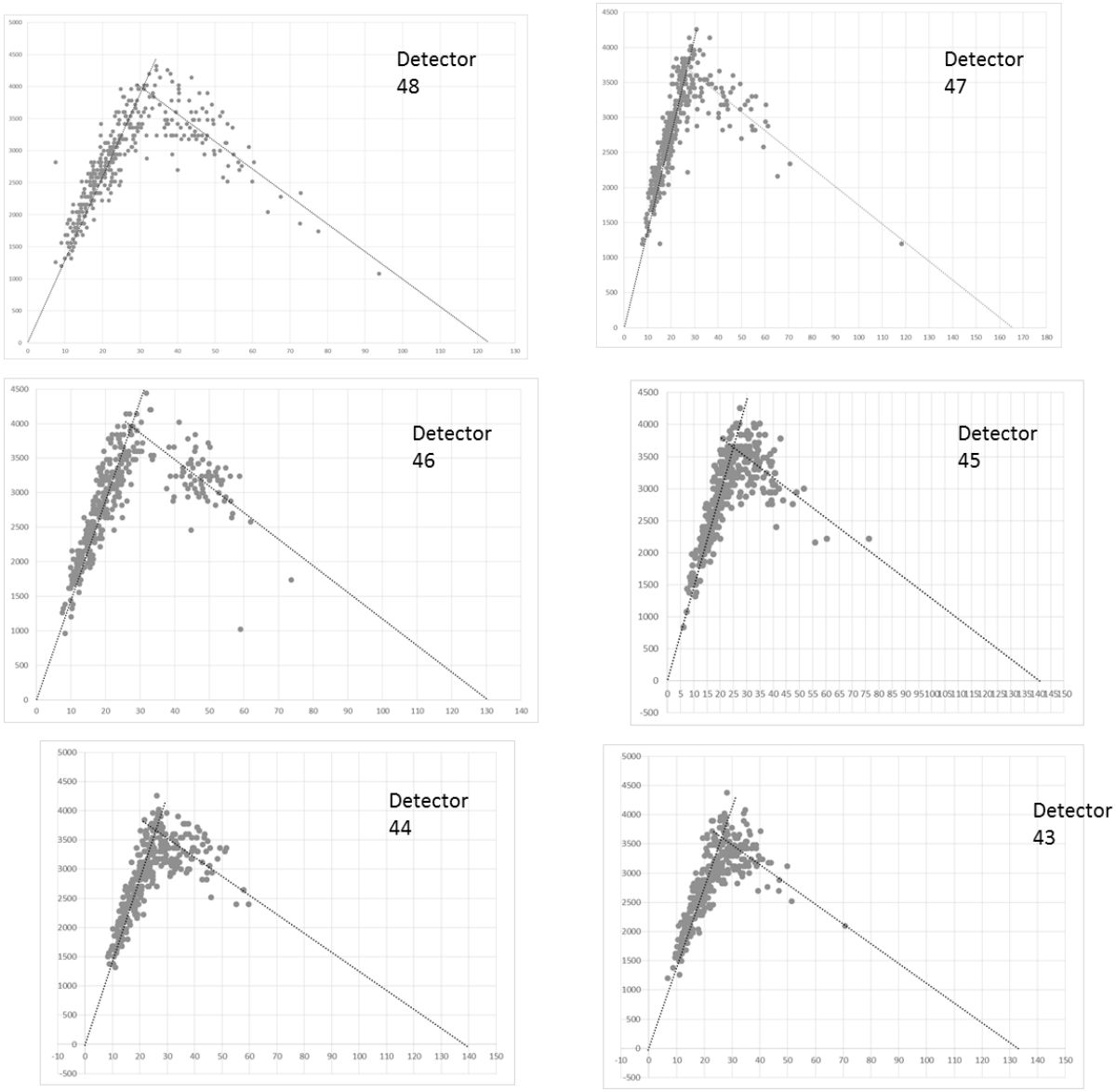


Figure 1. Individual and aggregated fundamental diagram (detector 48-39), June3

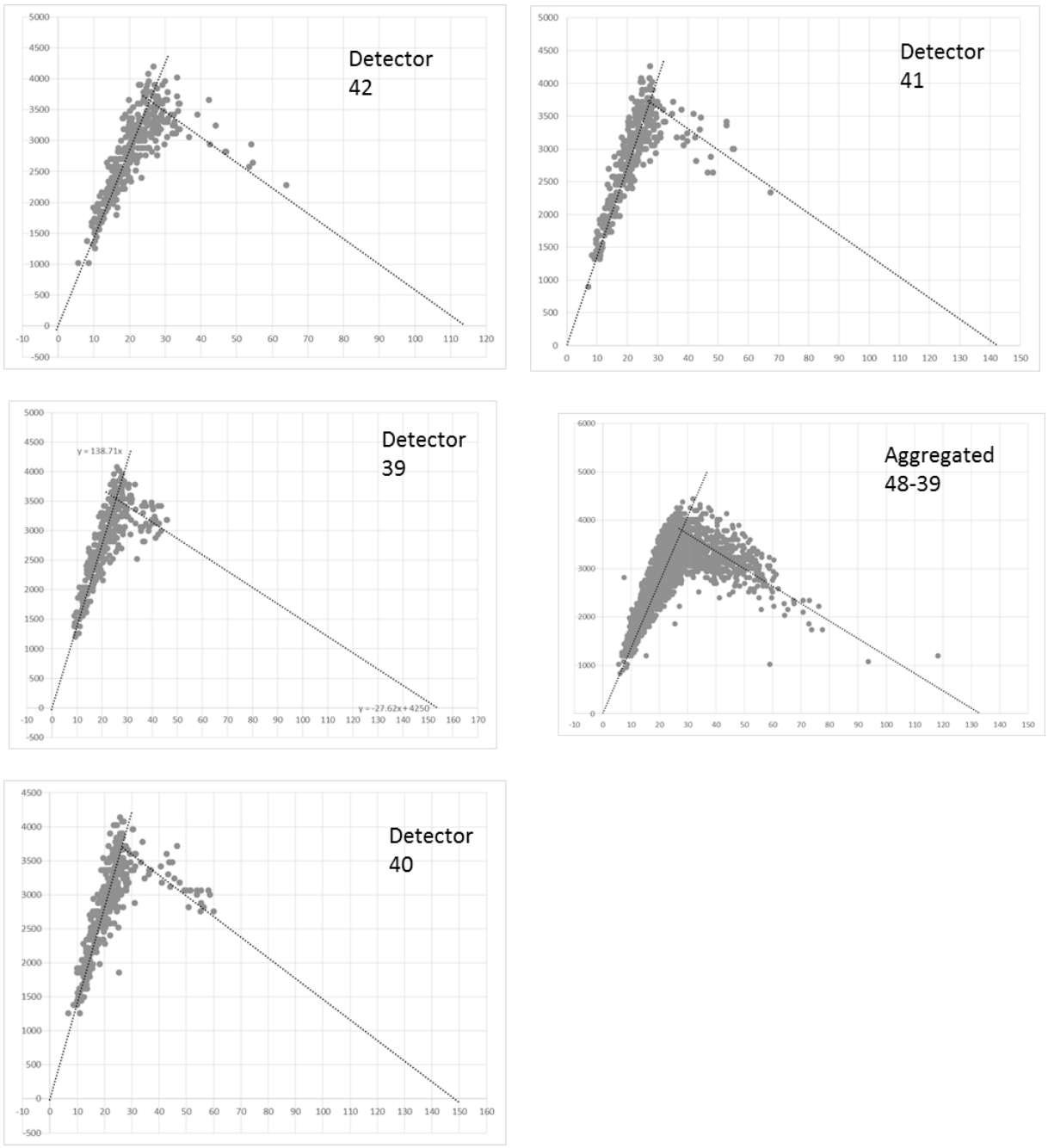


Figure 2. Individual and aggregated fundamental diagram (detector 48-39), June3

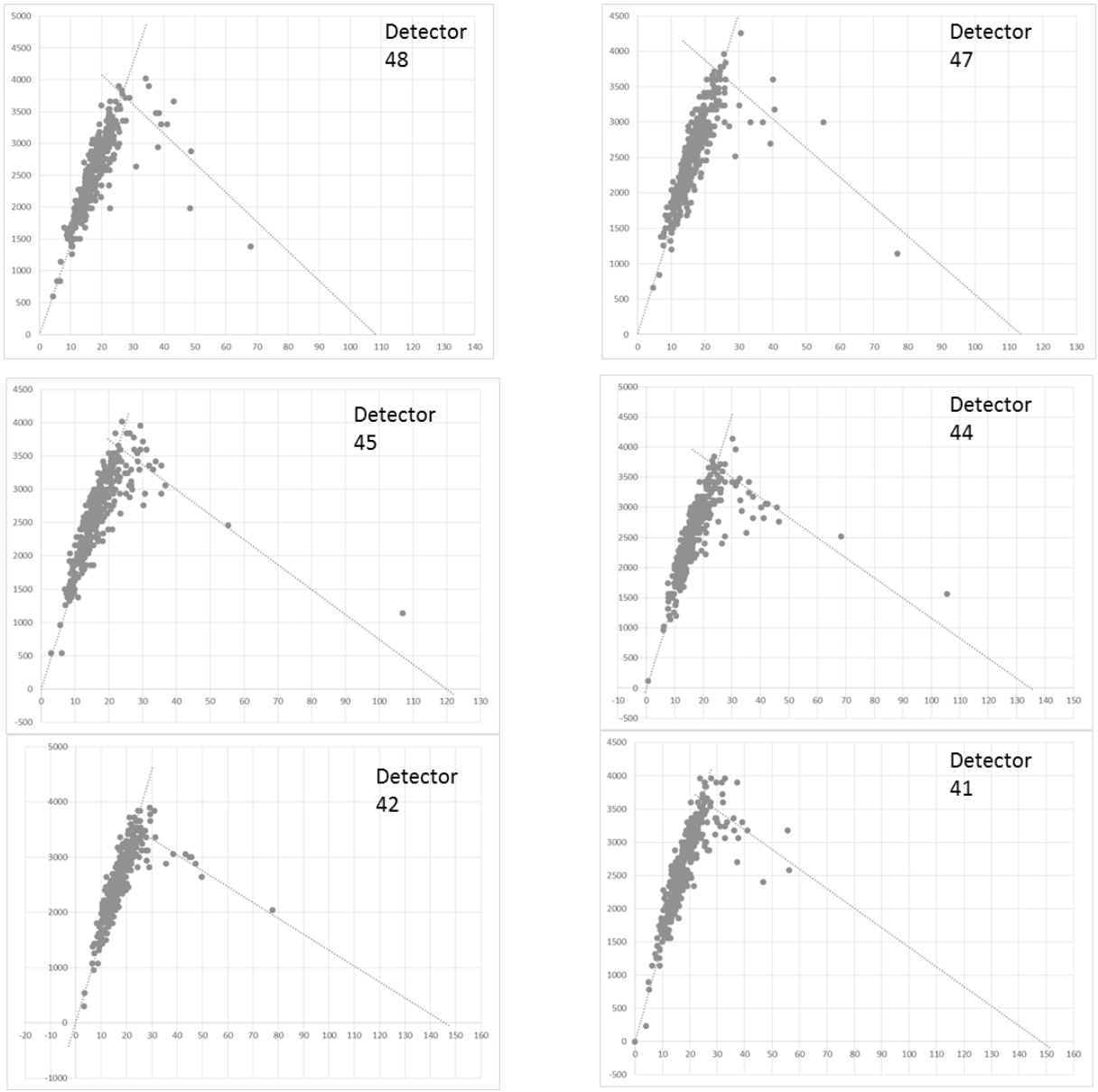


Figure 3. Individual and aggregated fundamental diagram (detector 48-39), April 8

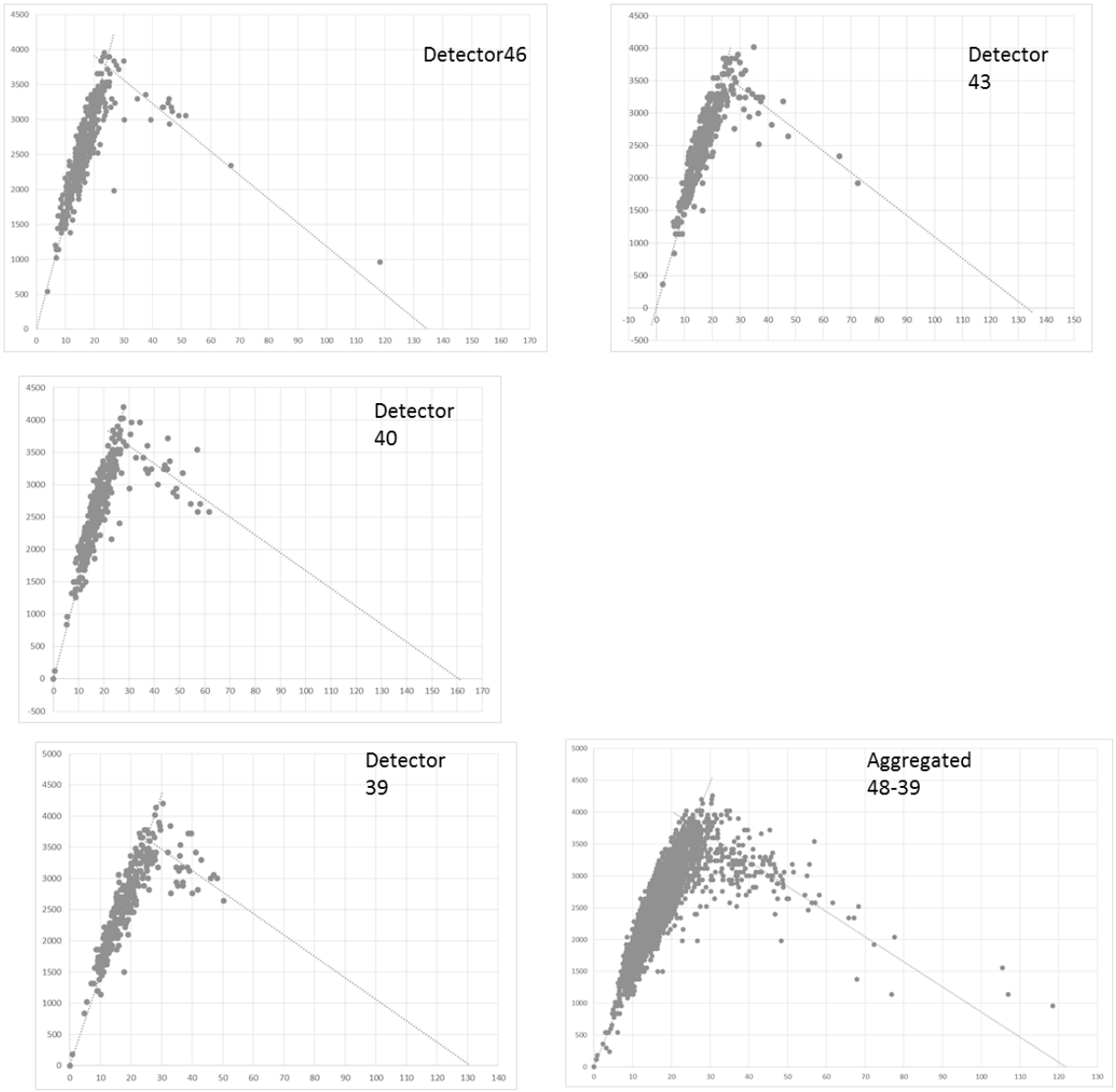


Figure 4. Individual and aggregated fundamental diagram (detector 48-39), April 8

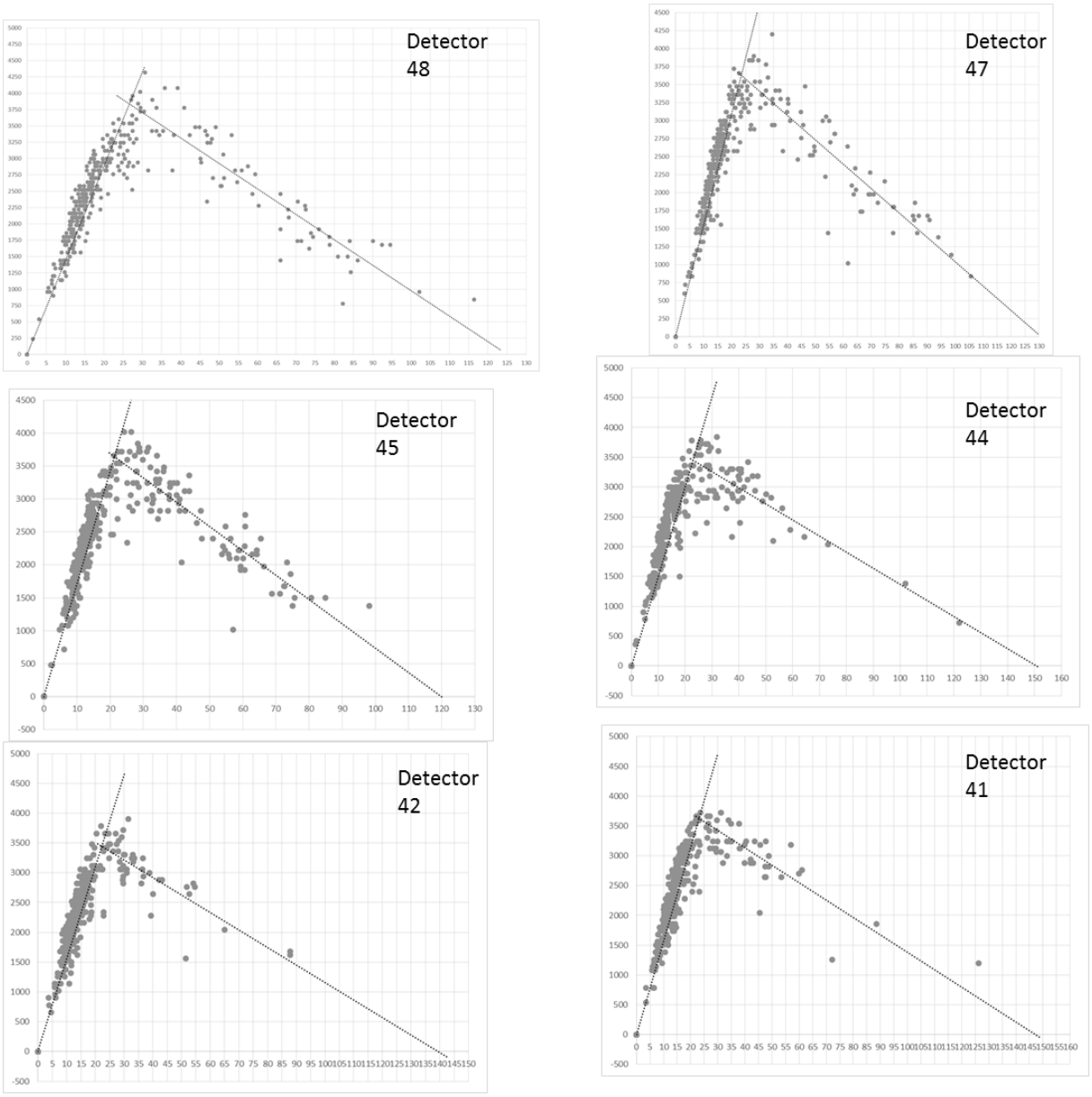


Figure 5. Individual and aggregated fundamental diagram (detector 48-39), March 18

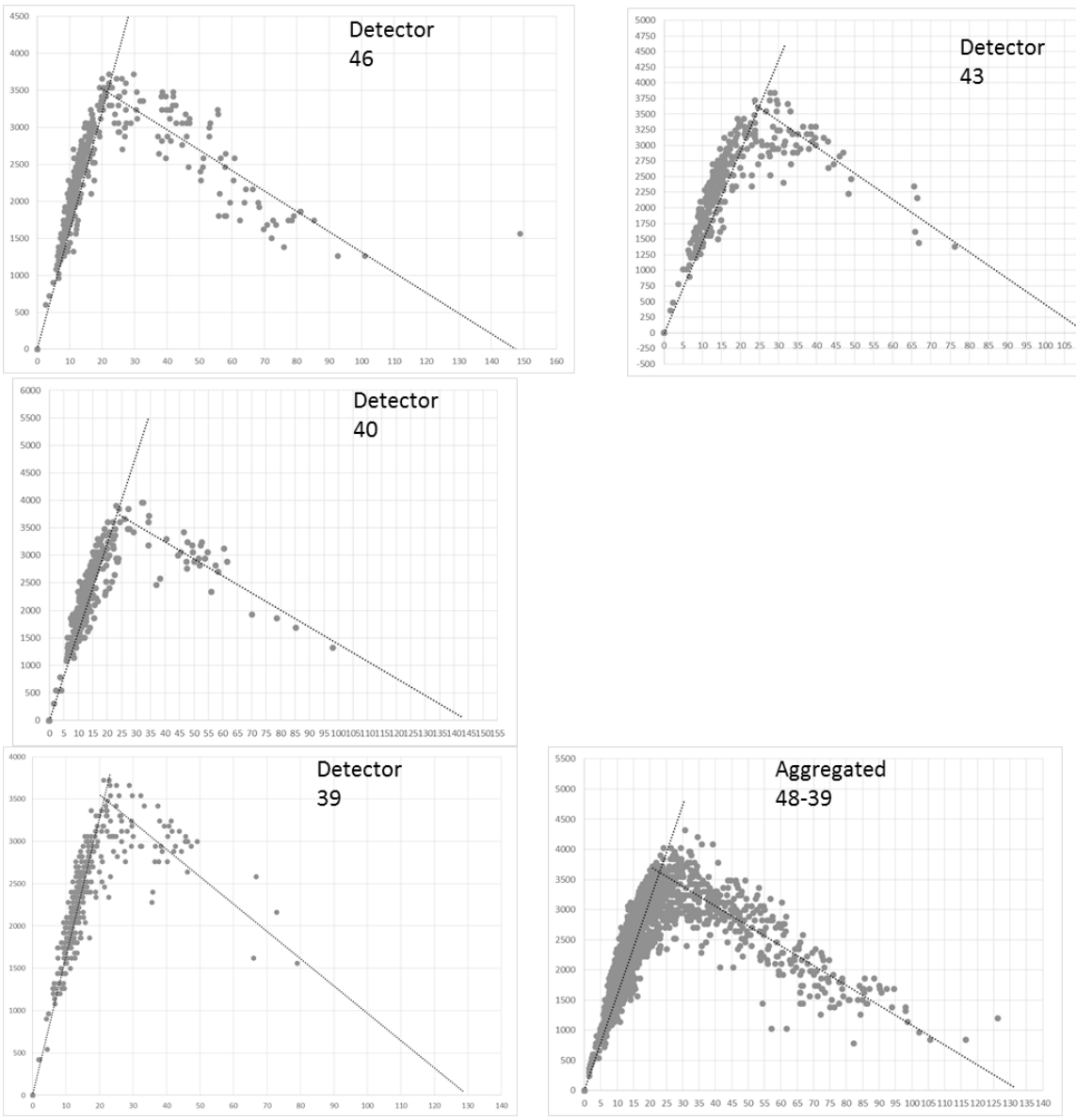


Figure 6. Individual and aggregated fundamental diagram (detector 48-39), March 18

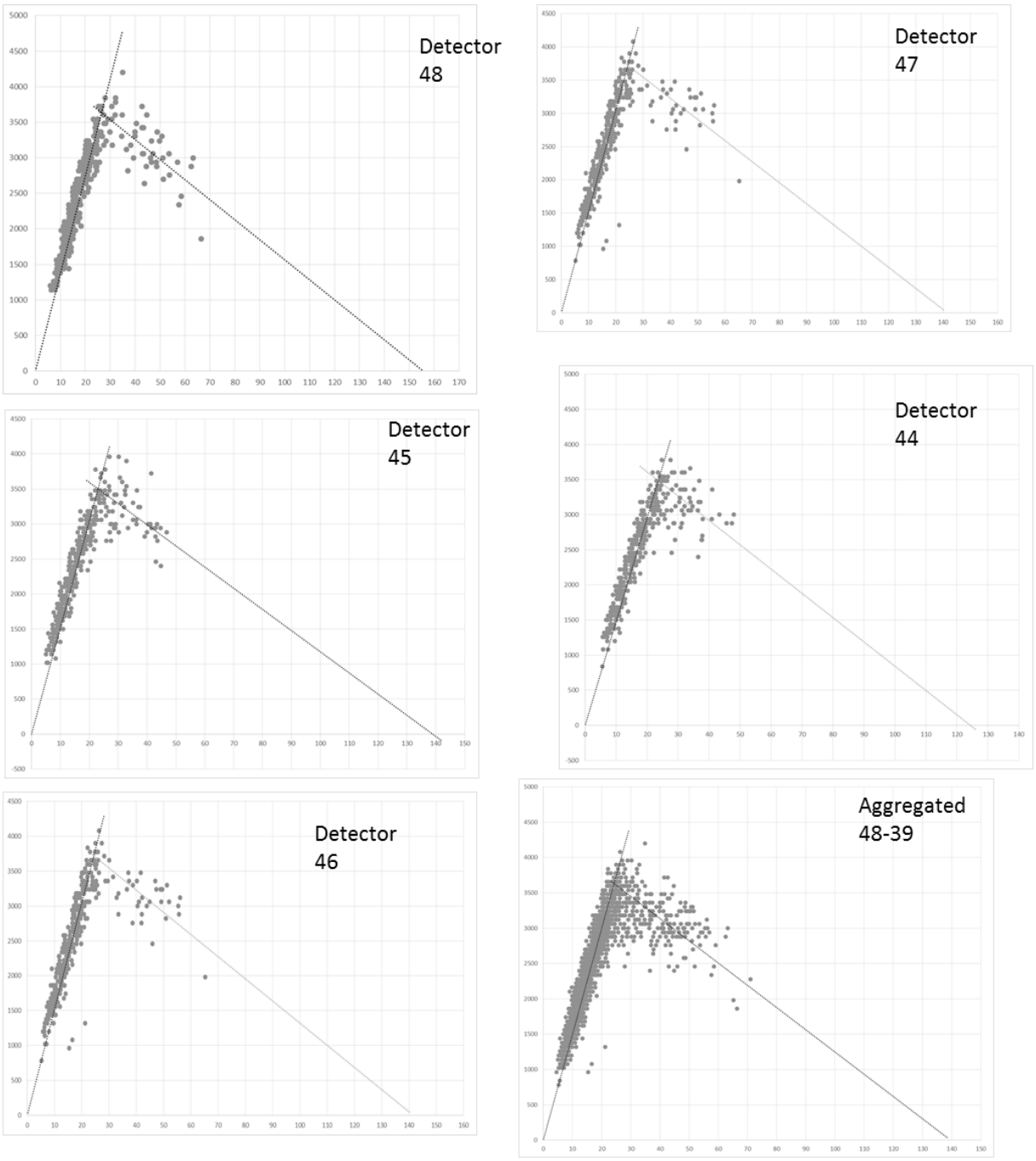


Figure 7. Individual and aggregated fundamental diagram (detector 48-39), July 22

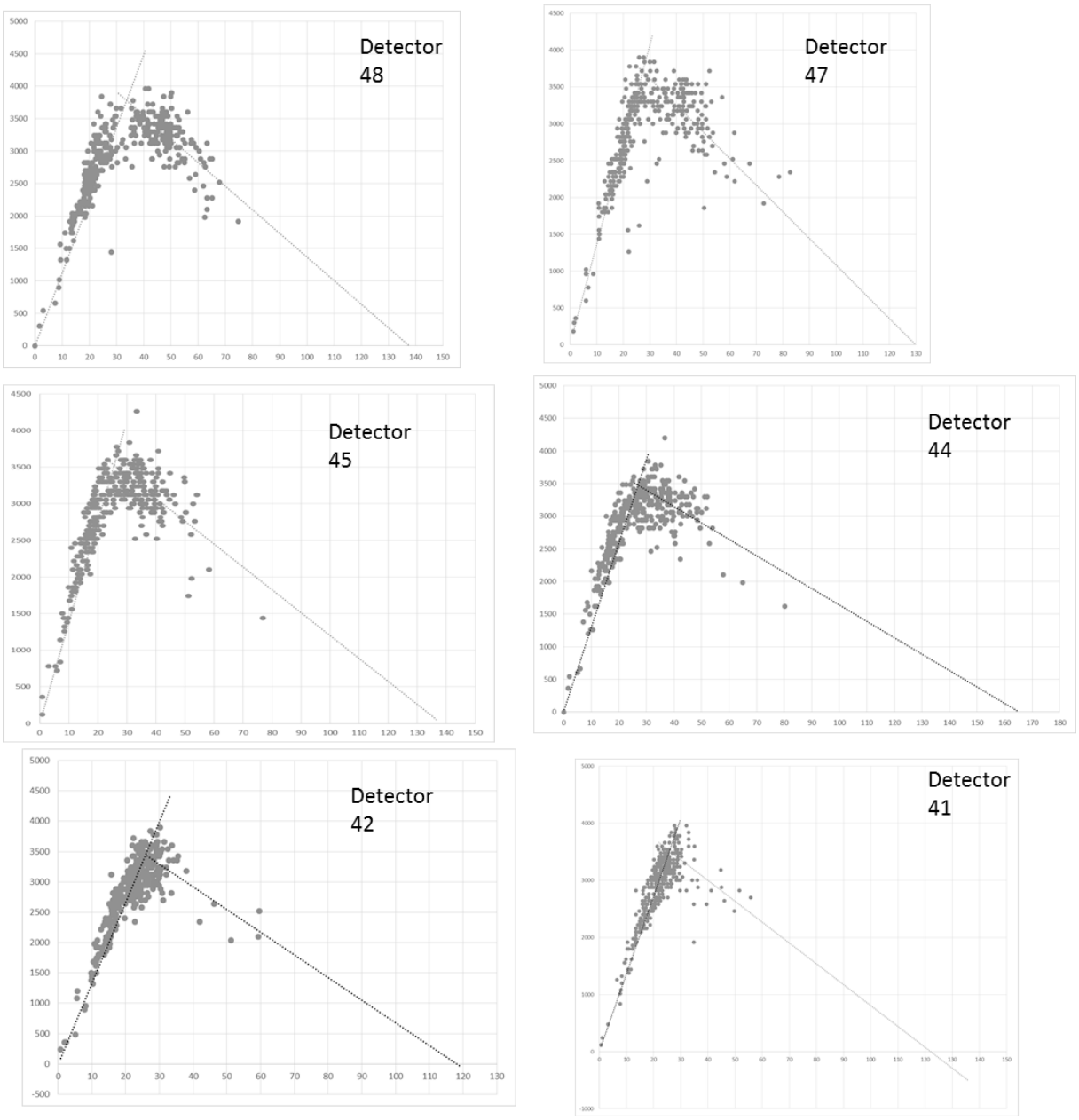


Figure 8. Individual and aggregated fundamental diagram (detector 48-39), July 15

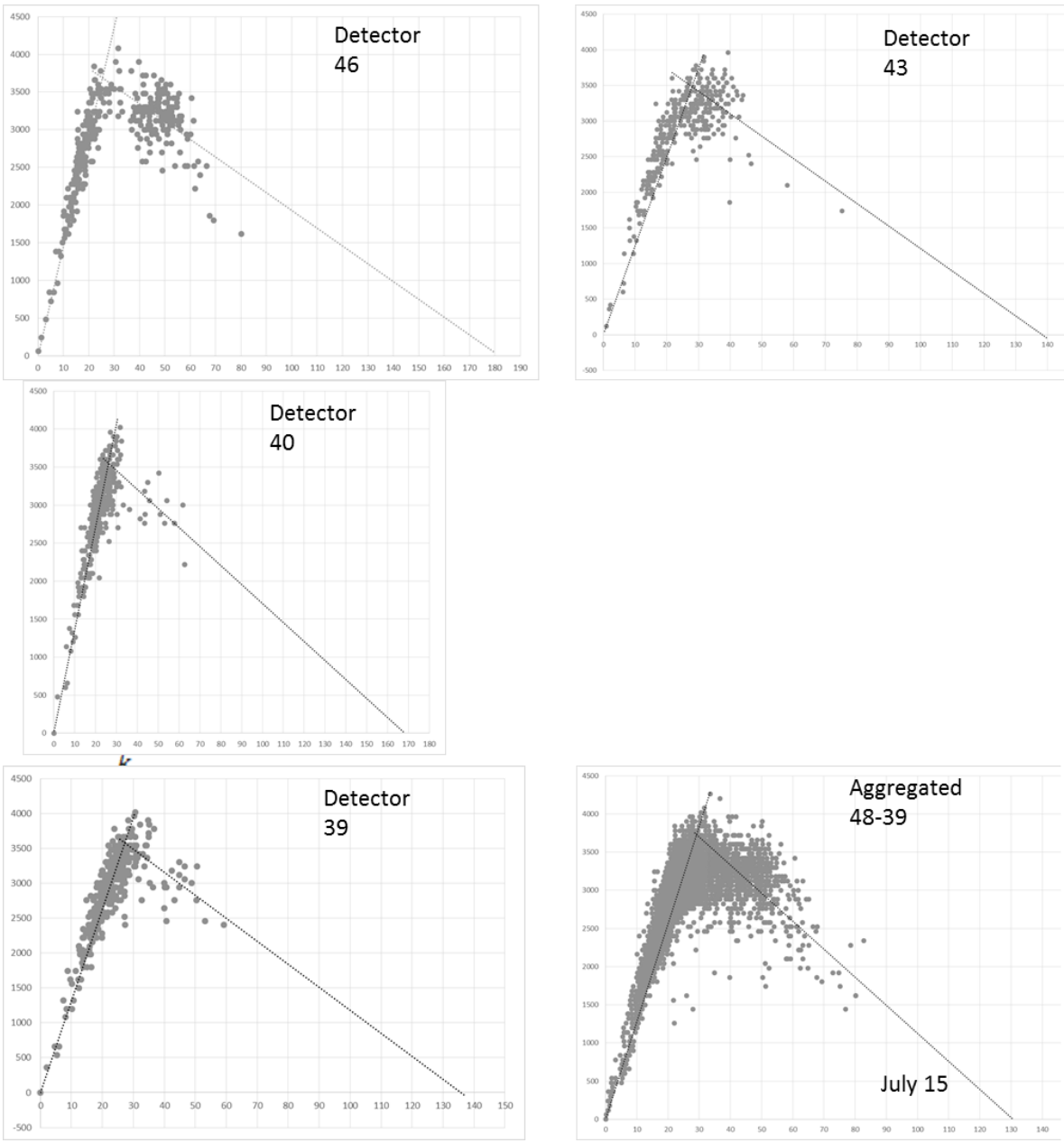


Figure 9. Individual and aggregated fundamental diagram (detector 48-39), July 15

Table 1. Individual and aggregated fundamental diagram (detector 48-39), June 3

Comparison of parameters of the FDs													
Detectors		48	47	46	45	44	43	42	41	40	39	48-39	avg
q_{max}	vehicles/h	4000	3650	3950	4000	3600	3600	3650	3700	3650	3500	3800	3730
v	km/h	131	135	146	146	144	129	140	135	140	140	136	139
k_c	vehicles/km	31	27	27	25	25	28	26	28	26	25	28	27
k_j	vehicles/km	124	165	130	140	139	134	115	142	150	155	133	139
W	km/h	43	26	38	32	32	34	41	32	29	27	36	33

Table 2. Individual and aggregated fundamental diagram (detector 48-39), April 8

Comparison of parameters of the FDs													
Detectors		48	47	46	45	44	43	42	41	40	39	48-39	avg
q_{max}	vehicles/h	4000	3650	3950	4000	3600	3600	3650	3700	3650	3500	3800	3730
v	km/h	131	135	146	146	144	129	140	135	140	140	136	139
k_c	vehicles/km	31	27	27	25	25	28	26	28	26	25	28	27
k_j	vehicles/km	124	165	130	140	139	134	115	142	150	155	133	139
W	km/h	43	26	38	32	32	34	41	32	29	27	36	33

Table 3. Individual and aggregated fundamental diagram (detector 48-39), March 18

Comparison of parameters of the FDs													
Detectors		48	47	46	45	44	43	42	41	40	39	48-39	avg
q_{max}	vehicles/h	3800	3625	3480	3800	3450	3600	3400	3650	3700	3500	3600	3601
v	km/h	143	151	158	166	150	141	148	152	154	152	157	152
k_c	vehicles/km	27	24	22	22	23	26	23	24	24	23	23	24
k_j	vehicles/km	125	130	148	120	150	110	140	147	145	130	133	135
W	km/h	39	34	28	37	27	43	29	30	31	33	33	33

Table 4. Individual and aggregated fundamental diagram (detector 48-39), July 22

Comparison of parameters of the FDs													
Detectors		48	47	46	45	44	43	42	41	40	39	48-39	avg
	vehicles/h	3650	3750	3650	3650	3500	0	0	0	0	0	3600	3640
	km/h	135	150	152	152	146	0	0	0	0	0	150	147
	vehicles/km	27	25	24	23	24	0	0	0	0	0	24	25
	vehicles/km	155	141	166	139	125	0	0	0	0	0	140	145
W	km/h	29	32	26	30	35	0	0	0	0	0	31	30

Table 5. Individual and aggregated fundamental diagram (detector 48-39), July 15

Comparison of parameters of the FDs													
Detectors		48	47	46	45	44	43	42	41	40	39	48-39	avg
	vehicles/h	3650	3750	3650	3650	3500	0	0	0	0	0	3600	3640
	km/h	135	150	152	152	146	0	0	0	0	0	150	147
	vehicles/km	27	25	24	23	24	0	0	0	0	0	24	25
	vehicles/km	155	141	166	139	125	0	0	0	0	0	140	145
W	km/h	29	32	26	30	35	0	0	0	0	0	31	30

Figure 1 to 13. CFL values and speed values

speed (km/h)	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value
80.8	0.6	81.7	0.6	87.4	0.6	82.6	0.6	83.1	0.6	83.9	0.6	87.15	0.6	90.05	0.7	82.5	0.6	83.85	0.6	91.45	0.7	92.4	0.7
81.6	0.6	71	0.5	77.4	0.6	68	0.5	81.7	0.6	83.3	0.6	87.35	0.6	81.65	0.6	81	0.6	83.5	0.6	89.35	0.7	92.85	0.7
82.9	0.6	75.15	0.6	79.85	0.6	67.8	0.5	83.1	0.6	83.5	0.6	89.55	0.7	80.2	0.6	84.05	0.6	86	0.6	86.95	0.6	91.55	0.7
84.5	0.6	83.1	0.6	83.9	0.6	87.15	0.6	80.35	0.6	80.15	0.6	73.3	0.5	81.25	0.6	84.15	0.6	89.05	0.7	81.05	0.6	91.25	0.7
80.5	0.6	81.7	0.6	83.3	0.6	87.35	0.6	77.7	0.6	82.75	0.6	68.75	0.5	81.85	0.6	89.8	0.7	89.55	0.7	80.15	0.6	89.15	0.7
76.95	0.6	83.1	0.6	83.5	0.6	89.55	0.7	77.3	0.6	84.5	0.6	74.1	0.5	83.3	0.6	89.05	0.7	85.65	0.6	80.45	0.6	88.7	0.7
76.75	0.6	80.35	0.6	80.15	0.6	73.3	0.5	81.1	0.6	87.25	0.6	73.9	0.5	86.25	0.6	97.4	0.7	87.65	0.6	90.3	0.7	89.1	0.7
85	0.6	77.7	0.6	82.75	0.6	68.75	0.5	86.9	0.6	87.85	0.7	76.05	0.6	90.5	0.7	89	0.7	88.55	0.7	89.2	0.7	85.9	0.6
75.9	0.6	77.3	0.6	84.5	0.6	74.1	0.5	75.7	0.6	83	0.6	66.9	0.5	86.9	0.6	81.7	0.6	71.5	0.5	79.2	0.6	75.9	0.6
80.5	0.6	81.1	0.6	87.25	0.6	73.9	0.5	77.85	0.6	79.6	0.6	82.3	0.6	82.85	0.6	83.35	0.6	81.85	0.6	82.15	0.6	77.4	0.6
87.6	0.6	86.9	0.6	87.85	0.7	76.05	0.6	75.3	0.6	81.4	0.6	81.25	0.6	79.5	0.6	86.15	0.6	90.8	0.7	83	0.6	89.25	0.7
85.05	0.6	75.7	0.6	83	0.6	66.9	0.5	79.2	0.6	83.8	0.6	76.6	0.6	78.25	0.6	85.25	0.6	84	0.6	83.35	0.6	88.7	0.7
79.85	0.6	77.85	0.6	79.6	0.6	82.3	0.6	78.8	0.6	83.25	0.6	71.7	0.5	78.65	0.6	89.6	0.7	79.85	0.6	86.8	0.6	84.2	0.6
77.35	0.6	75.3	0.6	81.4	0.6	81.25	0.6	77.5	0.6	90.55	0.7	71.5	0.5	90.7	0.7	93.35	0.7	81.25	0.6	90.3	0.7	87.25	0.6
79.8	0.6	79.2	0.6	83.8	0.6	76.6	0.6	76.6	0.6	88.9	0.7	73.25	0.5	87.05	0.6	87.7	0.6	78.6	0.6	92.5	0.7	85.9	0.6
82.25	0.6	78.8	0.6	83.25	0.6	71.7	0.5	83	0.6	88.35	0.7	75.6	0.6	89.05	0.7	87.6	0.6	85.45	0.6	93.05	0.7	82	0.6
87.45	0.6	77.5	0.6	90.55	0.7	71.5	0.5	84.45	0.6	93.3	0.7	84.2	0.6	95.2	0.7	86.85	0.6	88.2	0.7	92.3	0.7	85.8	0.6
90.65	0.7	76.6	0.6	88.9	0.7	73.25	0.5	85.95	0.6	90.85	0.7	86.15	0.6	87.05	0.6	89.45	0.7	91.95	0.7	93.45	0.7	87	0.6
74.2	0.5	83	0.6	88.35	0.7	75.6	0.6	78.4	0.6	81.3	0.6	73.9	0.5	76.7	0.6	81.8	0.6	78	0.6	79.8	0.6	76	0.6
78.55	0.6	84.45	0.6	93.3	0.7	84.2	0.6	82.95	0.6	80.45	0.6	76.9	0.6	74.5	0.6	78.4	0.6	83.95	0.6	79.85	0.6	78.45	0.6
80.25	0.6	85.95	0.6	90.85	0.7	86.15	0.6	80.35	0.6	87.15	0.6	69.9	0.5	86.05	0.6	88.9	0.7	79.75	0.6	89.1	0.7	83.5	0.6
80	0.6	78.4	0.6	81.3	0.6	73.9	0.5	81.35	0.6	87.75	0.7	74.05	0.5	90.2	0.7	85.2	0.6	83.85	0.6	89.35	0.7	83.9	0.6
84.7	0.6	82.95	0.6	80.45	0.6	76.9	0.6	79.25	0.6	86.85	0.6	76.4	0.6	87.15	0.6	86.25	0.6	87.2	0.6	87.15	0.6	81.9	0.6
83.4	0.6	80.35	0.6	87.15	0.6	69.9	0.5	81.15	0.6	83.45	0.6	87.7	0.6	87.3	0.6	85.35	0.6	89.7	0.7	87.75	0.7	82.5	0.6
85.65	0.6	81.35	0.6	87.75	0.7	74.05	0.5	78.7	0.6	88.75	0.7	84.05	0.6	85.35	0.6	88.45	0.7	87.95	0.7	87.45	0.6	83.1	0.6
86.75	0.6	79.25	0.6	86.85	0.6	76.4	0.6	83.15	0.6	87.75	0.7	84.95	0.6	84.2	0.6	86.5	0.6	91.2	0.7	85.45	0.6	84.6	0.6
86.9	0.6	81.15	0.6	83.45	0.6	87.7	0.6	87.1	0.6	93.85	0.7	85.75	0.6	84.65	0.6	83.75	0.6	92.8	0.7	87.45	0.6	83.45	0.6
88.7	0.7	78.7	0.6	88.75	0.7	84.05	0.6	88.3	0.7	92.65	0.7	89.25	0.7	84	0.6	81.8	0.6	92.65	0.7	86	0.6	84.4	0.6
76.6	0.6	83.15	0.6	87.75	0.7	84.95	0.6	76	0.6	78.6	0.6	73.9	0.5	70.6	0.5	71.5	0.5	83	0.6	81.2	0.6	80.3	0.6
77.55	0.6	87.1	0.6	93.85	0.7	85.75	0.6	76.55	0.6	78.25	0.6	84.2	0.6	80.15	0.6	82.35	0.6	80.1	0.6	82.05	0.6	75.6	0.6
91.4	0.7	88.3	0.7	92.65	0.7	89.25	0.7	86.8	0.6	94.1	0.7	86.5	0.6	82.75	0.6	91.7	0.7	95	0.7	88.9	0.7	86.75	0.6
88.35	0.7	76	0.6	78.6	0.6	73.9	0.5	86.75	0.6	89.2	0.7	81.3	0.6	81.05	0.6	93.05	0.7	92.9	0.7	87.85	0.7	86.85	0.6
80.45	0.6	76.55	0.6	78.25	0.6	84.2	0.6	84.45	0.6	89.25	0.7	77	0.6	81.15	0.6	84.15	0.6	91.45	0.7	87.9	0.7	83.05	0.6
81.4	0.6	86.8	0.6	94.1	0.7	86.5	0.6	88.05	0.7	89.1	0.7	84.4	0.6	84.75	0.6	81.75	0.6	87.5	0.6	96.3	0.7	84.55	0.6
87.55	0.6	86.75	0.6	89.2	0.7	81.3	0.6	82.25	0.6	88.75	0.7	89.25	0.7	81.5	0.6	77.6	0.6	89.8	0.7	88.45	0.7	79.8	0.6
80.35	0.6	84.45	0.6	89.25	0.7	77	0.6	80.55	0.6	87.1	0.6	88.65	0.7	83.5	0.6	82.3	0.6	87.55	0.6	85.05	0.6	77.85	0.6
83.85	0.6	88.05	0.7	89.1	0.7	84.4	0.6	86.25	0.6	86.95	0.6	86.3	0.6	82.7	0.6	89.35	0.7	86.55	0.6	86.6	0.6	80.85	0.6
81.7	0.6	82.25	0.6	88.75	0.7	89.25	0.7	87.4	0.6	82.6	0.6	84.75	0.6	81.35	0.6	90.5	0.7	87.75	0.7	94.4	0.7	81.6	0.6
71	0.5	80.55	0.6	87.1	0.6	88.65	0.7	77.4	0.6	68	0.5	76.6	0.6	84.3	0.6	87.7	0.6	82	0.6	76.4	0.6	81.8	0.6
75.15	0.6	86.25	0.6	86.95	0.6	86.3	0.6	79.85	0.6	67.8	0.5	83	0.6	79.2	0.6	81.75	0.6	85.05	0.6	81.8	0.6	80.1	0.6

speed (km/h)	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value
77.05	0.6	76.95	0.6	74.7	0.6	81.45	0.6	86.95	0.6	80.1	0.6	81.95	0.6	88.55	0.7	85.1	0.6	80	0.6	86.15	0.6	77.7	0.6
81.15	0.6	84.5	0.6	77	0.6	76.5	0.6	89.55	0.7	85	0.6	77.85	0.6	87.55	0.6	87	0.6	83.35	0.6	79.45	0.6	87.7	0.6
81.5	0.6	86.7	0.6	88.45	0.7	79.45	0.6	83	0.6	87.5	0.6	87.3	0.6	87.05	0.6	89.25	0.7	84.6	0.6	81.15	0.6	84.5	0.6
78.95	0.6	82.75	0.6	87.55	0.6	75.65	0.6	81.6	0.6	86.3	0.6	85.25	0.6	96.2	0.7	90.1	0.7	80.2	0.6	79.65	0.6	82.75	0.6
81.45	0.6	84.95	0.6	82	0.6	79	0.6	81.9	0.6	85.05	0.6	71.6	0.5	94.25	0.7	94.3	0.7	85.7	0.6	80	0.6	83.45	0.6
85.85	0.6	82.75	0.6	81.5	0.6	77.3	0.6	83.55	0.6	84.1	0.6	79.9	0.6	80.7	0.6	89.1	0.7	87.7	0.6	83.15	0.6	83.55	0.6
91.6	0.7	86.1	0.6	81.75	0.6	83.05	0.6	81.6	0.6	83.55	0.6	79.85	0.6	80.8	0.6	77.6	0.6	85.35	0.6	83.6	0.6	82.6	0.6
84.1	0.6	84.95	0.6	84.95	0.6	86.2	0.6	86.95	0.6	82.25	0.6	79.15	0.6	84.5	0.6	77.85	0.6	79.2	0.6	88.7	0.7	83	0.6
72.6	0.5	87.45	0.6	91	0.7	92.75	0.7	83.7	0.6	81.45	0.6	78.7	0.6	91.35	0.7	93.4	0.7	78.8	0.6	78.9	0.6	83.75	0.6
79.9	0.6	80	0.6	84.6	0.6	88.25	0.7	93.25	0.7	82.55	0.6	79.85	0.6	87.15	0.6	92.05	0.7	82.25	0.6	79.4	0.6	80	0.6
87.55	0.6	82.3	0.6	77.8	0.6	88.3	0.7	93.5	0.7	93.7	0.7	78.9	0.6	86.95	0.6	89.5	0.7	80.65	0.6	78.15	0.6	76.95	0.6
89.8	0.7	83.55	0.6	78.6	0.6	77.2	0.6	87.25	0.6	97	0.7	81.3	0.6	89	0.7	87.7	0.6	82.55	0.6	79.7	0.6	78.4	0.6
87.3	0.6	87.95	0.7	78.3	0.6	79.75	0.6	74.2	0.5	91.55	0.7	84.1	0.6	89.8	0.7	84.7	0.6	85.1	0.6	84.7	0.6	80.8	0.6
85.35	0.6	85.25	0.6	80.85	0.6	84.7	0.6	79.9	0.6	81.5	0.6	87.15	0.6	93.75	0.7	86.1	0.6	85.65	0.6	90.85	0.7	86.8	0.6
85.1	0.6	84.65	0.6	82.25	0.6	86.35	0.6	88.65	0.7	83.2	0.6	84.2	0.6	84.4	0.6	86.5	0.6	86.1	0.6	88.75	0.7	83.2	0.6
89.9	0.7	86.05	0.6	81.05	0.6	87.7	0.6	93.05	0.7	94.3	0.7	80.3	0.6	70.1	0.5	77.3	0.6	84.8	0.6	89.6	0.7	81.9	0.6
89.45	0.7	87.45	0.6	85.65	0.6	85.6	0.6	93	0.7	100.2	0.7	81.4	0.6	75.8	0.6	77	0.6	85.4	0.6	100.2	0.7	84.7	0.6
90.65	0.7	86.55	0.6	90.35	0.7	84.9	0.6	83	0.6	88.95	0.7	81.1	0.6	89	0.7	80.4	0.6	73.6	0.5	97	0.7	84.1	0.6
79.6	0.6	87.45	0.6	89.8	0.7	81.95	0.6	82.85	0.6	86.25	0.6	85.35	0.6	88.9	0.7	81.7	0.6	75.45	0.6	81	0.6	86.2	0.6
75.2	0.6	77.6	0.6	84.6	0.6	81.65	0.6	84.65	0.6	88.15	0.7	89.7	0.7	89.75	0.7	78.15	0.6	87.2	0.6	80	0.6	78.4	0.6
81.7	0.6	84.75	0.6	73.1	0.5	84.95	0.6	81	0.6	84.05	0.6	89.85	0.7	86.6	0.6	81.65	0.6	87.1	0.6	90.15	0.7	78.8	0.6
78.3	0.6	84.2	0.6	70.25	0.5	78.2	0.6	88.7	0.7	90.5	0.7	88.05	0.7	81.5	0.6	78.5	0.6	86.3	0.6	93.85	0.7	84.8	0.6
83.45	0.6	84.4	0.6	85.2	0.6	75.45	0.6	79.6	0.6	88.2	0.7	93.1	0.7	77.05	0.6	77.65	0.6	86.7	0.6	91.45	0.7	87.4	0.6
84.25	0.6	80.85	0.6	88.75	0.7	81.05	0.6	77.4	0.6	81.3	0.6	91.1	0.7	76.4	0.6	74.45	0.6	81.1	0.6	92.7	0.7	83.75	0.6
81.8	0.6	83.3	0.6	89.7	0.7	80.85	0.6	77.35	0.6	77.9	0.6	81.3	0.6	81.1	0.6	79.75	0.6	81.05	0.6	93.4	0.7	81.25	0.6
80.05	0.6	85.1	0.6	82.25	0.6	86.1	0.6	78.55	0.6	87.75	0.7	78.85	0.6	74.2	0.5	84.85	0.6	83.05	0.6	94.75	0.7	80.3	0.6
81.35	0.6	87.15	0.6	81.05	0.6	81.5	0.6	73.85	0.5	81.35	0.6	92.85	0.7	80.45	0.6	78.6	0.6	81.95	0.6	93	0.7	82.35	0.6
85.75	0.6	97.65	0.7	79.65	0.6	80.8	0.6	79.25	0.6	79.1	0.6	93.35	0.7	82.65	0.6	79.4	0.6	79.6	0.6	86.3	0.6	88.6	0.7
76.9	0.6	91.25	0.7	81.75	0.6	81.45	0.6	80.3	0.6	87.95	0.7	89.4	0.7	81.35	0.6	78.35	0.6	77.45	0.6	73.6	0.5	88.7	0.7
77.65	0.6	81	0.6	79.7	0.6	83.8	0.6	79.6	0.6	88.6	0.7	89.25	0.7	80.05	0.6	81.1	0.6	81.1	0.6	84.1	0.6	79.4	0.6
83.05	0.6	81.75	0.6	69.7	0.5	85	0.6	85.35	0.6	87.5	0.6	87.45	0.6	77	0.6	83.15	0.6	82.85	0.6	92.8	0.7	84.1	0.6
80.35	0.6	80.45	0.6	77.5	0.6	76.2	0.6	85.2	0.6	86.15	0.6	83.95	0.6	81.6	0.6	78.25	0.6	80.05	0.6	87.95	0.7	82.4	0.6
78.85	0.6	82.65	0.6	81.8	0.6	78.75	0.6	75.4	0.6	86.65	0.6	88.55	0.7	87.3	0.6	80.3	0.6	81.7	0.6	85.1	0.6	81.8	0.6
81.35	0.6	87.65	0.6	79.7	0.6	87.85	0.7	80.45	0.6	71	0.5	86.65	0.6	90.65	0.7	87.05	0.6	81.9	0.6	87.15	0.6	85.05	0.6
76.35	0.6	86.6	0.6	79.7	0.6	86.45	0.6	85.2	0.6	68.9	0.5	81.7	0.6	88.15	0.7	88	0.7	83.1	0.6	86.15	0.6	81.5	0.6
84.65	0.6	90.1	0.7	74.45	0.6	88.65	0.7	88.7	0.7	87.05	0.6	84.15	0.6	80.4	0.6	83.5	0.6	85.95	0.6	85.05	0.6	87.15	0.6
90.8	0.7	81	0.6	75	0.6	83.85	0.6	81.7	0.6	84.75	0.6	83.35	0.6	82.05	0.6	75.5	0.6	84.5	0.6	89.85	0.7	88.15	0.7
88.7	0.7	84.2	0.6	77.25	0.6	83.25	0.6	81.1	0.6	85.15	0.6	85.3	0.6	82.15	0.6	73.15	0.5	72.7	0.5	86.8	0.6	96.15	0.7
76.9	0.6	81.35	0.6	79.15	0.6	81.3	0.6	79.1	0.6	80.35	0.6	85.6	0.6	80	0.6	79.15	0.6	77.3	0.6	74.3	0.6	91.5	0.7

speed (km/h)	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value
85.4	0.6	85.5	0.6	84.65	0.6	76.85	0.6	87.4	0.6	86.9	0.6	78.6	0.6	85.15	0.6	83	0.6	73.2	0.5	73.1	0.5	87.2	0.6
79.15	0.6	72.9	0.5	87.75	0.7	88.35	0.7	83.5	0.6	86.9	0.6	76.75	0.6	79.8	0.6	82.5	0.6	80.2	0.6	78.4	0.6	75.7	0.6
90.15	0.7	76.4	0.6	84.1	0.6	85.05	0.6	91.35	0.7	87.1	0.6	75.75	0.6	78.35	0.6	80.3	0.6	82	0.6	67.85	0.5	74.2	0.5
89.65	0.7	91.85	0.7	87.05	0.6	84.3	0.6	94.7	0.7	85.85	0.6	77.9	0.6	81.05	0.6	80.1	0.6	81.2	0.6	75.25	0.6	79.55	0.6
98.2	0.7	86.6	0.6	65.45	0.5	81.5	0.6	84.4	0.6	81.65	0.6	78.4	0.6	81.2	0.6	75.65	0.6	75.25	0.6	80.2	0.6	83.2	0.6
90.25	0.7	89.1	0.7	64.15	0.5	66.45	0.5	90	0.7	73.9	0.5	81.4	0.6	82.6	0.6	71.95	0.5	80.8	0.6	83.95	0.6	80	0.6
95	0.7	83.35	0.6	62.6	0.5	72.65	0.5	87.6	0.6	75.85	0.6	77.5	0.6	83.7	0.6	77.35	0.6	79.35	0.6	85.85	0.6	83.2	0.6
91.45	0.7	79.95	0.6	78.95	0.6	76.25	0.6	84.55	0.6	86.65	0.6	79.9	0.6	74.2	0.5	87.6	0.6	81.1	0.6	84	0.6	78.65	0.6
89	0.7	82.4	0.6	85.8	0.6	79.1	0.6	84.95	0.6	86.1	0.6	80.5	0.6	77.45	0.6	83.9	0.6	82.85	0.6	84.55	0.6	78.4	0.6
89.9	0.7	83.7	0.6	91.25	0.7	86.2	0.6	90.15	0.7	86.8	0.6	82	0.6	77.65	0.6	80.65	0.6	77.2	0.6	85.1	0.6	72.05	0.5
77.5	0.6	84.45	0.6	92.4	0.7	87.95	0.7	94.85	0.7	79.8	0.6	80.05	0.6	78.85	0.6	79.45	0.6	80.75	0.6	72.9	0.5	73.9	0.5
76.8	0.6	67.3	0.5	88.35	0.7	93.2	0.7	95.4	0.7	78.9	0.6	77.6	0.6	83.2	0.6	80.85	0.6	79.1	0.6	77.55	0.6	70.5	0.5
94.25	0.7	66.4	0.5	77.6	0.6	91.9	0.7	98.7	0.7	78.4	0.6	73.7	0.5	79.2	0.6	75.75	0.6	80.4	0.6	81.05	0.6	74.4	0.6
90.85	0.7	88.4	0.7	77.1	0.6	73.6	0.5	88.75	0.7	92.3	0.7	72	0.5	82.65	0.6	75.35	0.6	79.9	0.6	83.7	0.6	83.15	0.6
87.9	0.7	85.85	0.6	65.9	0.5	81.5	0.6	74	0.5	89	0.7	84.6	0.6	86.05	0.6	70.45	0.5	80.25	0.6	88.6	0.7	84.7	0.6
90.75	0.7	83.65	0.6	59	0.4	78.2	0.6	79.1	0.6	78.8	0.6	82	0.6	88.4	0.7	82.4	0.6	84.75	0.6	86	0.6	80.15	0.6
87.15	0.6	80.9	0.6	89.45	0.7	87.75	0.7	92.95	0.7	78.25	0.6	71.8	0.5	85.85	0.6	84.6	0.6	91.85	0.7	83.2	0.6	76.05	0.6
84.85	0.6	83.35	0.6	90.45	0.7	88.05	0.7	91	0.7	79.95	0.6	78.85	0.6	73.9	0.5	88.3	0.7	91	0.7	79.4	0.6	69.65	0.5
84	0.6	71.65	0.5	83.5	0.6	89.5	0.7	94	0.7	79.85	0.6	83.45	0.6	76.7	0.6	78	0.6	83.15	0.6	82.65	0.6	71.5	0.5
85.4	0.6	71.05	0.5	85.85	0.6	87.8	0.7	82.5	0.6	74.2	0.5	83.6	0.6	80.35	0.6	79.75	0.6	70.5	0.5	84.4	0.6	72.1	0.5
80.6	0.6	69.65	0.5	85.9	0.6	92.05	0.7	77.6	0.6	82	0.6	76.5	0.6	81.2	0.6	78.4	0.6	76.05	0.6	78.6	0.6	76.9	0.6
85.35	0.6	67.2	0.5	83.7	0.6	89.05	0.7	75.3	0.6	82.75	0.6	76.8	0.6	80.9	0.6	83.6	0.6	83.65	0.6	76.15	0.6	68.4	0.5
86.25	0.6	73.25	0.5	80.2	0.6	86.9	0.6	82.85	0.6	82.9	0.6	79.2	0.6	82.5	0.6	82.2	0.6	89.1	0.7	83.5	0.6	75.6	0.6
85.85	0.6	81.45	0.6	78.65	0.6	72.6	0.5	91.2	0.7	84.35	0.6	77.15	0.6	78.85	0.6	84.25	0.6	94.85	0.7	85.55	0.6	79.6	0.6
86.3	0.6	88	0.7	81.45	0.6	75.85	0.6	87.7	0.6	81.9	0.6	85.7	0.6	82.4	0.6	92.7	0.7	82.35	0.6	80.1	0.6	77.9	0.6
85.25	0.6	81.15	0.6	86.95	0.6	86.25	0.6	82.6	0.6	78.6	0.6	93.1	0.7	83.1	0.6	83.8	0.6	79	0.6	77.75	0.6	73.35	0.5
82.05	0.6	69.75	0.5	84.15	0.6	91.6	0.7	82.3	0.6	78.35	0.6	82.1	0.6	83.9	0.6	94.15	0.7	76.25	0.6	78.5	0.6	71.5	0.5
85.35	0.6	62.3	0.5	83	0.6	86.5	0.6	83.3	0.6	77	0.6	78.6	0.6	70.6	0.5	87.5	0.6	79.6	0.6	82.5	0.6	67.65	0.5
86.2	0.6	65.15	0.5	88.85	0.7	85.2	0.6	79.1	0.6	80.7	0.6	79.55	0.6	75.35	0.6	77.2	0.6	81.3	0.6	85.9	0.6	70.3	0.5
94.35	0.7	66.25	0.5	81.3	0.6	82.95	0.6	79.5	0.6	80.7	0.6	80.05	0.6	83	0.6	80.4	0.6	75.6	0.6	86.55	0.6	74.55	0.6
80.4	0.6	72.4	0.5	80.15	0.6	85.4	0.6	79.35	0.6	77.6	0.6	80.35	0.6	82.5	0.6	90.2	0.7	79.4	0.6	79.9	0.6	78.85	0.6
82.45	0.6	71.7	0.5	83.45	0.6	87.3	0.6	88.75	0.7	77.55	0.6	81.85	0.6	81.55	0.6	89.7	0.7	78	0.6	79.85	0.6	73.5	0.5
81.8	0.6	81.6	0.6	77.8	0.6	85.5	0.6	90.15	0.7	76.35	0.6	76.6	0.6	80.05	0.6	89.15	0.7	76.6	0.6	73.35	0.5	76.2	0.6
83.2	0.6	70.1	0.5	78.4	0.6	70.9	0.5	88.75	0.7	80.8	0.6	84.6	0.6	80.35	0.6	92.05	0.7	74.15	0.5	77.5	0.6	71.6	0.5
85.95	0.6	68.65	0.5	87.15	0.6	77.6	0.6	77.6	0.6	81.3	0.6	82.05	0.6	75.7	0.6	85.3	0.6	74.4	0.6	79.75	0.6	71	0.5
93.65	0.7	66.95	0.5	85.5	0.6	88.5	0.7	74.1	0.5	71.9	0.5	84.1	0.6	77.55	0.6	79.1	0.6	71.9	0.5	83.6	0.6	71.45	0.5
94.55	0.7	70.25	0.5	79.9	0.6	81.7	0.6	88.35	0.7	74.05	0.5	75.7	0.6	77.95	0.6	82.95	0.6	79.15	0.6	79	0.6	67.85	0.5
89.6	0.7	66.5	0.5	73.9	0.5	85.75	0.6	83.85	0.6	78	0.6	79	0.6	76.8	0.6	88.85	0.7	84.6	0.6	83.6	0.6	66.45	0.5
88.7	0.7	75.4	0.6	69.05	0.5	84.45	0.6	85.1	0.6	76.15	0.6	85.35	0.6	78.6	0.6	71.8	0.5	89.4	0.7	87.45	0.6	73.6	0.5

speed (km/h)	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value
80.55	0.6	81.4	0.6	74.2	0.5	86.4	0.6	80.8	0.6	83.7	0.6	92.15	0.7	78.2	0.6	75.2	0.6	88.65	0.7	92.8	0.7	85.25	0.6
85.2	0.6	88.65	0.7	75.4	0.6	84.15	0.6	81.25	0.6	85.15	0.6	87.8	0.7	83	0.6	75.55	0.6	80.3	0.6	83.95	0.6	84.7	0.6
79.8	0.6	89.05	0.7	87.05	0.6	82.85	0.6	77	0.6	83.5	0.6	89.65	0.7	86.25	0.6	74.35	0.6	82.3	0.6	81.5	0.6	82.7	0.6
80.55	0.6	80	0.6	84.3	0.6	87.35	0.6	79.3	0.6	81.75	0.6	88.75	0.7	85.55	0.6	73.6	0.5	84.4	0.6	83.95	0.6	75.5	0.6
70.65	0.5	80.75	0.6	71.5	0.5	85	0.6	78.8	0.6	83.75	0.6	87.5	0.6	85.35	0.6	78.1	0.6	86.7	0.6	87	0.6	83.95	0.6
71.2	0.5	79	0.6	78.55	0.6	74.7	0.6	78.6	0.6	83.95	0.6	88.35	0.7	86.05	0.6	83.65	0.6	84.15	0.6	85.05	0.6	77.95	0.6
73.05	0.5	79.7	0.6	78.5	0.6	76.35	0.6	76.5	0.6	83.3	0.6	87.7	0.6	91.45	0.7	83.5	0.6	89	0.7	92.5	0.7	82.05	0.6
75.1	0.6	82.55	0.6	81.9	0.6	92.25	0.7	85.1	0.6	73.9	0.5	90.35	0.7	85.25	0.6	80.5	0.6	90.85	0.7	87.35	0.6	80.5	0.6
77.1	0.6	84	0.6	79.2	0.6	83.85	0.6	79.85	0.6	79.65	0.6	75.7	0.6	83.4	0.6	80.25	0.6	89.25	0.7	90.3	0.7	82.7	0.6
81.35	0.6	85.95	0.6	79.55	0.6	76.65	0.6	78	0.6	84.4	0.6	79.6	0.6	71	0.5	85.55	0.6	91.9	0.7	91.95	0.7	84.7	0.6
87.6	0.6	82.85	0.6	80.65	0.6	78.05	0.6	79.85	0.6	82.2	0.6	86.85	0.6	82.45	0.6	72.9	0.5	90.35	0.7	97.8	0.7	88.95	0.7
83.5	0.6	88	0.7	85.5	0.6	80.3	0.6	81.55	0.6	82.5	0.6	89.2	0.7	85.2	0.6	82.95	0.6	83	0.6	91.4	0.7	88.55	0.7
68.6	0.5	88.35	0.7	85.55	0.6	78.55	0.6	74.45	0.6	82.15	0.6	89.15	0.7	86.55	0.6	86.05	0.6	84.8	0.6	84.4	0.6	91.3	0.7
69.9	0.5	81.7	0.6	85.85	0.6	84.55	0.6	80.1	0.6	79.05	0.6	86.45	0.6	84.9	0.6	84.9	0.6	87.4	0.6	85.5	0.6	77.5	0.6
73.3	0.5	78.1	0.6	75.4	0.6	87.75	0.7	88.95	0.7	81.35	0.6	88.65	0.7	82.95	0.6	86.05	0.6	89.4	0.7	88.85	0.7	88.25	0.7
77.55	0.6	91.55	0.7	77.2	0.6	76.2	0.6	86.55	0.6	86.85	0.6	87.45	0.6	76.1	0.6	75.35	0.6	89.85	0.7	85.45	0.6	84.75	0.6
81.05	0.6	90.2	0.7	82.1	0.6	82.85	0.6	77	0.6	93.1	0.7	85.4	0.6	79.35	0.6	74.9	0.6	89.25	0.7	84.05	0.6	83.85	0.6
80.1	0.6	89.15	0.7	83.45	0.6	79.2	0.6	77.35	0.6	74.7	0.6	83	0.6	85.1	0.6	70	0.5	88.8	0.7	87.85	0.7	93.25	0.7
73.35	0.5	87.95	0.7	82.1	0.6	83.65	0.6	81.6	0.6	88.3	0.7	78.2	0.6	91.45	0.7	86.8	0.6	89.85	0.7	86.05	0.6	87.55	0.6
71.05	0.5	80.5	0.6	87.7	0.6	85.35	0.6	81.25	0.6	82.45	0.6	73.8	0.5	83	0.6	90.95	0.7	90.25	0.7	88.25	0.7	88.9	0.7
78.9	0.6	81.6	0.6	79.4	0.6	78.05	0.6	82.05	0.6	81.5	0.6	89.5	0.7	78.4	0.6	80	0.6	96.05	0.7	87.4	0.6	89.2	0.7
77.4	0.6	80.65	0.6	80.25	0.6	82.6	0.6	81.05	0.6	87.65	0.6	84.85	0.6	75.55	0.6	77.7	0.6	77.2	0.6	89.15	0.7	92.7	0.7
76	0.6	80.35	0.6	80.75	0.6	83.25	0.6	82.9	0.6	85.15	0.6	81	0.6	79.3	0.6	78.8	0.6	87.3	0.6	79.6	0.6	88.7	0.7
74.9	0.6	76.8	0.6	84.9	0.6	84.7	0.6	85.95	0.6	88.35	0.7	86.3	0.6	82.5	0.6	79.8	0.6	91.7	0.7	79.55	0.6	77.6	0.6
84.2	0.6	74.75	0.6	79.4	0.6	82	0.6	85	0.6	93.5	0.7	80.35	0.6	80.1	0.6	80	0.6	92.9	0.7	95.45	0.7	83.2	0.6
80.45	0.6	89.55	0.7	83.85	0.6	72.5	0.5	87.8	0.7	98	0.7	78.4	0.6	88.45	0.7	79.9	0.6	86.45	0.6	92.45	0.7	90.8	0.7
76.95	0.6	86.65	0.6	86.7	0.6	72.3	0.5	77.9	0.6	92.4	0.7	80.8	0.6	83.9	0.6	82.75	0.6	90.9	0.7	86.3	0.6	86.8	0.6
76.75	0.6	87.55	0.6	86.85	0.6	83.55	0.6	78.9	0.6	81.3	0.6	85.25	0.6	79.85	0.6	90.35	0.7	90.15	0.7	84.1	0.6	83.9	0.6
71.65	0.5	78.85	0.6	81.8	0.6	84.45	0.6	85.35	0.6	88.1	0.7	73.2	0.5	82.6	0.6	91.25	0.7	91.15	0.7	78.65	0.6	88.9	0.7
72.65	0.5	74.3	0.6	77.7	0.6	86.3	0.6	85.6	0.6	89.35	0.7	69.6	0.5	76.5	0.6	89.75	0.7	100.45	0.7	80.5	0.6	88.35	0.7
78.9	0.6	66.55	0.5	89.3	0.7	80.15	0.6	80.95	0.6	95.55	0.7	85	0.6	78.65	0.6	81.2	0.6	96.65	0.7	90.65	0.7	93.3	0.7
80.65	0.6	75.1	0.6	85.5	0.6	79.6	0.6	81.95	0.6	95.45	0.7	83.9	0.6	86.7	0.6	82.45	0.6	79.3	0.6	87.75	0.7	95.4	0.7
79.8	0.6	79.55	0.6	97.2	0.7	77.65	0.6	80.85	0.6	92	0.7	81.7	0.6	85.5	0.6	86.6	0.6	83.2	0.6	72	0.5	105.4	0.8
79.95	0.6	79.3	0.6	93.45	0.7	83	0.6	75.35	0.6	94.75	0.7	81.85	0.6	80.05	0.6	85.4	0.6	89.45	0.7	76.6	0.6	80	0.6
88.05	0.7	77.3	0.6	86.3	0.6	87.55	0.6	89.1	0.7	86.8	0.6	84.45	0.6	75.3	0.6	84.5	0.6	86.3	0.6	80.4	0.6	77.65	0.6
85.95	0.6	79.55	0.6	82.15	0.6	70.9	0.5	88.95	0.7	93.6	0.7	83.2	0.6	77.3	0.6	84.45	0.6	90.95	0.7	79.25	0.6	92.1	0.7
76.2	0.6	79.85	0.6	86.6	0.6	73.3	0.5	80.9	0.6	88.3	0.7	86.1	0.6	79.4	0.6	87.7	0.6	90.9	0.7	82.45	0.6	88.4	0.7
77.15	0.6	76.4	0.6	89.85	0.7	83.2	0.6	81.05	0.6	78.6	0.6	91.05	0.7	81.25	0.6	88.25	0.7	92.15	0.7	81.2	0.6	92.55	0.7
76.9	0.6	72.8	0.5	85.45	0.6	84.45	0.6	80.75	0.6	82.25	0.6	86	0.6	87.8	0.7	90.15	0.7	88.95	0.7	81.05	0.6	98.7	0.7

speed (km/h)	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value
89.85	0.7	76	0.6	86.4	0.6	87.15	0.6	83.25	0.6	78.7	0.6	79.3	0.6	78.35	0.6	82.85	0.6	79.2	0.6	86.6	0.6	75.55	0.6
75.95	0.6	76.75	0.6	88.4	0.7	84.3	0.6	82.55	0.6	76.6	0.6	76.1	0.6	69.7	0.5	81.45	0.6	84.1	0.6	83.6	0.6	79.95	0.6
87.65	0.6	80.4	0.6	92.3	0.7	78.1	0.6	81.65	0.6	83.3	0.6	79.15	0.6	76.25	0.6	73.8	0.5	80	0.6	90.35	0.7	82.85	0.6
80.65	0.6	80.55	0.6	83.7	0.6	77.75	0.6	84.3	0.6	82.9	0.6	81.5	0.6	82.15	0.6	75.95	0.6	73.2	0.5	86.8	0.6	89.55	0.7
72.3	0.5	81.4	0.6	86.45	0.6	76.5	0.6	85.05	0.6	76.95	0.6	82.35	0.6	80.95	0.6	77.3	0.6	75.1	0.6	75	0.6	80.65	0.6
79.2	0.6	78.4	0.6	89.3	0.7	74.05	0.5	86.15	0.6	75.65	0.6	86.5	0.6	80.7	0.6	80.05	0.6	81.35	0.6	77.4	0.6	74.7	0.6
83.25	0.6	83.85	0.6	79.2	0.6	82.7	0.6	84.5	0.6	78.05	0.6	80.55	0.6	78	0.6	79.2	0.6	89.35	0.7	84.25	0.6	80.45	0.6
82.05	0.6	75.85	0.6	72.85	0.5	72.5	0.5	83.3	0.6	83.25	0.6	80.85	0.6	77.3	0.6	78.25	0.6	82.15	0.6	86.95	0.6	76.55	0.6
81.9	0.6	78.7	0.6	95.9	0.7	81	0.6	73.2	0.5	85.25	0.6	82.7	0.6	76.3	0.6	77.25	0.6	75.8	0.6	84.8	0.6	79.95	0.6
81.7	0.6	79.1	0.6	98.85	0.7	82.2	0.6	74.95	0.6	79.6	0.6	82.45	0.6	85.45	0.6	82.85	0.6	63.65	0.5	78.1	0.6	81.75	0.6
77.9	0.6	73.35	0.5	82.7	0.6	83.6	0.6	88.05	0.7	79.35	0.6	72.3	0.5	83.85	0.6	85.2	0.6	72.6	0.5	81.7	0.6	83.6	0.6
85.15	0.6	75.7	0.6	79.45	0.6	77.6	0.6	88.3	0.7	75.9	0.6	77.3	0.6	75.2	0.6	82.85	0.6	77.9	0.6	82.35	0.6	80.1	0.6
88.9	0.7	76.75	0.6	79.05	0.6	76.45	0.6	88.3	0.7	74.25	0.6	84.35	0.6	82.2	0.6	79	0.6	80.9	0.6	86.5	0.6	79.1	0.6
86	0.6	85.4	0.6	85.1	0.6	73.3	0.5	85.05	0.6	77.4	0.6	83.9	0.6	77	0.6	73.9	0.5	73.1	0.5	83.2	0.6	80.6	0.6
78.6	0.6	87.8	0.7	87.35	0.6	73	0.5	82.15	0.6	81.25	0.6	82.15	0.6	77.5	0.6	79.25	0.6	74.8	0.6	80.6	0.6	87.85	0.7
75.1	0.6	81.5	0.6	87.8	0.7	81.8	0.6	81.6	0.6	84.75	0.6	78.05	0.6	76.6	0.6	81.35	0.6	78.1	0.6	74.75	0.6	79.3	0.6
74.1	0.5	80.05	0.6	73.5	0.5	90.1	0.7	79.1	0.6	84.15	0.6	79.5	0.6	79.7	0.6	80.95	0.6	76.65	0.6	78.1	0.6	79.85	0.6
76.4	0.6	75.75	0.6	84.4	0.6	78.9	0.6	89.5	0.7	86.85	0.6	74.7	0.6	80.95	0.6	80.6	0.6	77.7	0.6	79.35	0.6	77.9	0.6
77.45	0.6	79	0.6	81.55	0.6	84.95	0.6	78.2	0.6	85.8	0.6	77.1	0.6	84.75	0.6	81.9	0.6	75.65	0.6	77.05	0.6	79.35	0.6
81.75	0.6	81.15	0.6	80.6	0.6	74.25	0.6	87.45	0.6	77.9	0.6	85.2	0.6	91.35	0.7	85	0.6	79.85	0.6	83.5	0.6	80.45	0.6
83.5	0.6	84.55	0.6	83.85	0.6	73.75	0.5	80.15	0.6	78.3	0.6	83	0.6	86.25	0.6	85.9	0.6	80.65	0.6	84.3	0.6	81.4	0.6
83.25	0.6	87.7	0.6	77.35	0.6	75.95	0.6	79.75	0.6	79.95	0.6	85.1	0.6	72.6	0.5	89	0.7	79.6	0.6	85.65	0.6	83.7	0.6
78	0.6	83.45	0.6	79.65	0.6	79.2	0.6	82.5	0.6	83.7	0.6	77.6	0.6	74.05	0.5	73.9	0.5	81.05	0.6	85.35	0.6	88.9	0.7
78.7	0.6	91.25	0.7	76.6	0.6	78.8	0.6	82.75	0.6	85.8	0.6	78.7	0.6	83.15	0.6	82.15	0.6	74.7	0.6	84.1	0.6	89.4	0.7
68.5	0.5	87.75	0.7	100.05	0.7	86.55	0.6	90.2	0.7	73.55	0.5	77.6	0.6	86.6	0.6	77.5	0.6	76.05	0.6	76.3	0.6	90.9	0.7
72.15	0.5	77	0.6	92.25	0.7	93.5	0.7	92.8	0.7	78.75	0.6	79	0.6	83.55	0.6	77.55	0.6	79.75	0.6	73.75	0.5	74.7	0.6
86.25	0.6	82.2	0.6	73.9	0.5	85.45	0.6	102.7	0.8	79.95	0.6	80.5	0.6	83	0.6	79.25	0.6	78.15	0.6	83.25	0.6	82.55	0.6
81.3	0.6	89.75	0.7	76.05	0.6	72.3	0.5	96.65	0.7	87.85	0.7	87.5	0.6	80.45	0.6	86.3	0.6	80.8	0.6	85.65	0.6	80.9	0.6
77.9	0.6	89.65	0.7	77.3	0.6	75.15	0.6	83	0.6	82	0.6	91.85	0.7	81.9	0.6	83.55	0.6	77.45	0.6	85.55	0.6	85.2	0.6
77.1	0.6	86.5	0.6	79.45	0.6	79.15	0.6	78.45	0.6	72.3	0.5	90.2	0.7	78.4	0.6	82.7	0.6	81.55	0.6	85.35	0.6	85.75	0.6
73.95	0.5	85.7	0.6	78.3	0.6	91.6	0.7	89.25	0.7	84.7	0.6	78.4	0.6	80.85	0.6	87.35	0.6	80.85	0.6	79.25	0.6	85.2	0.6
68.75	0.5	86.35	0.6	90.65	0.7	93.2	0.7	91.35	0.7	80.05	0.6	74.6	0.6	74.3	0.6	91.9	0.7	86.8	0.6	79.4	0.6	86.9	0.6
81.05	0.6	85.3	0.6	93.7	0.7	87.45	0.6	94.85	0.7	81.05	0.6	79.55	0.6	75.15	0.6	79.2	0.6	85.95	0.6	78.55	0.6	86.35	0.6
84	0.6	88	0.7	85.45	0.6	81.4	0.6	94.3	0.7	76.5	0.6	84.5	0.6	80.55	0.6	76.6	0.6	80.2	0.6	85.6	0.6	89	0.7
76.2	0.6	92.35	0.7	82.75	0.6	82.8	0.6	88.1	0.7	78.2	0.6	86.55	0.6	80.75	0.6	82.95	0.6	81.65	0.6	76.8	0.6	87.2	0.6
72.9	0.5	86.3	0.6	81.45	0.6	84.2	0.6	84.7	0.6	79.95	0.6	87.6	0.6	77.55	0.6	81	0.6	77.85	0.6	76.5	0.6	76	0.6
77.25	0.6	76.95	0.6	74.3	0.6	85.65	0.6	83.7	0.6	85.85	0.6	77.15	0.6	77.25	0.6	85.05	0.6	82.3	0.6	82.45	0.6	75.35	0.6
71.6	0.5	78.95	0.6	74.2	0.5	74.7	0.6	82.4	0.6	89.2	0.7	83.4	0.6	76.8	0.6	85.9	0.6	83.7	0.6	84.45	0.6	87.5	0.6
73.25	0.5	82.25	0.6	82.05	0.6	71.5	0.5	73.6	0.5	90.4	0.7	80.45	0.6	74.95	0.6	84.9	0.6	88.15	0.7	77.2	0.6	80.45	0.6

speed (km/h)	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value
84.5	0.6	75.7	0.6	88.35	0.7	75.2	0.6	71	0.5	95.5	0.7	41.4	0.3	22	0.2	61.05	0.5	57.6	0.4	35.05	0.3	73.95	0.5
84.75	0.6	77.65	0.6	88.45	0.7	85.95	0.6	76.9	0.6	80.4	0.6	44.8	0.3	70.1	0.5	61.7	0.5	35.95	0.3	56.85	0.4	74.7	0.6
82.45	0.6	76.75	0.6	85.65	0.6	92.15	0.7	76.15	0.6	48.2	0.4	42.75	0.3	47.4	0.4	44.2	0.3	69.5	0.5	87.55	0.6	76.35	0.6
81.1	0.6	74.7	0.6	81.45	0.6	90.55	0.7	77.9	0.6	79.25	0.6	56.8	0.4	51.05	0.4	43.05	0.3	75.15	0.6	79.7	0.6	78.9	0.6
80.25	0.6	77.75	0.6	86.7	0.6	86.2	0.6	72	0.5	81.15	0.6	39.3	0.3	45.9	0.3	45.45	0.3	50.75	0.4	88.65	0.7	74.7	0.6
79.7	0.6	79.3	0.6	92.05	0.7	78.45	0.6	69.7	0.5	90.75	0.7	16.25	0.1	39.05	0.3	54.4	0.4	47.25	0.4	50.65	0.4	75.25	0.6
73.7	0.5	78.5	0.6	93.35	0.7	79.85	0.6	63.8	0.5	89.75	0.7	28.9	0.2	51.95	0.4	64.45	0.5	30.55	0.2	42.55	0.3	47.2	0.3
76.35	0.6	70.1	0.5	94.05	0.7	81.3	0.6	58.35	0.4	45.5	0.3	58.7	0.4	49.75	0.4	50.15	0.4	60.4	0.4	36.95	0.3	37.8	0.3
82.75	0.6	73.55	0.5	78.6	0.6	78.85	0.6	76.2	0.6	41.35	0.3	40.5	0.3	62.6	0.5	38.25	0.3	51.3	0.4	63.4	0.5	63.4	0.5
84.25	0.6	75.95	0.6	75.9	0.6	71.9	0.5	79.5	0.6	57.55	0.4	45.8	0.3	45.75	0.3	40.65	0.3	51.75	0.4	57.55	0.4	69.5	0.5
79.3	0.6	77.45	0.6	79.8	0.6	66.85	0.5	71	0.5	45	0.3	51.3	0.4	22.2	0.2	50.55	0.4	61.15	0.5	45.75	0.3	69.3	0.5
73.45	0.5	76.05	0.6	81.55	0.6	82.45	0.6	76.75	0.6	56	0.4	35.3	0.3	20.7	0.2	58.7	0.4	39.5	0.3	81.2	0.6	74.3	0.6
70.6	0.5	75.45	0.6	82.25	0.6	78.95	0.6	72.15	0.5	35.35	0.3	36.7	0.3	45.45	0.3	50.1	0.4	64.4	0.5	80.4	0.6	82.1	0.6
73.95	0.5	76.1	0.6	90.15	0.7	76.95	0.6	76.55	0.6	48.9	0.4	62.05	0.5	48.9	0.4	37.95	0.3	72.2	0.5	71.8	0.5	83.35	0.6
82.75	0.6	73.4	0.5	85.85	0.6	72.55	0.5	74.45	0.6	50.85	0.4	43.85	0.3	55.3	0.4	34.3	0.3	50.05	0.4	76.1	0.6	77.2	0.6
84.85	0.6	79.85	0.6	85.15	0.6	71.9	0.5	70.55	0.5	45.5	0.3	43.7	0.3	40.65	0.3	60.05	0.4	52.05	0.4	49.15	0.4	69.55	0.5
76.5	0.6	82.2	0.6	87.7	0.6	72.6	0.5	59.15	0.4	50.6	0.4	19.3	0.1	37.6	0.3	64.6	0.5	50.7	0.4	49.9	0.4	49.75	0.4
73.6	0.5	74.3	0.6	85.55	0.6	60.05	0.4	69.05	0.5	52	0.4	25.8	0.2	63.95	0.5	49.9	0.4	39	0.3	51.5	0.4	36.35	0.3
75	0.6	78.65	0.6	72.3	0.5	77.7	0.6	82.8	0.6	51.6	0.4	45.4	0.3	60.2	0.4	46.3	0.3	47.8	0.4	58.15	0.4	62.7	0.5
71.75	0.5	73.5	0.5	82.05	0.6	69.6	0.5	83.05	0.6	37.75	0.3	63.6	0.5	50.65	0.4	51.1	0.4	45.85	0.3	68.6	0.5	76.05	0.6
70.8	0.5	76.6	0.6	90.05	0.7	71.55	0.5	71.4	0.5	64.5	0.5	78.75	0.6	58.4	0.4	38.85	0.3	64.35	0.5	75.5	0.6	71.65	0.5
77.5	0.6	78.65	0.6	86.9	0.6	78.25	0.6	73.55	0.5	50.55	0.4	38	0.3	28.6	0.2	50.3	0.4	43.5	0.3	82.15	0.6	82.1	0.6
81.5	0.6	75.3	0.6	82	0.6	74.95	0.6	74.45	0.6	41.3	0.3	27.8	0.2	37.1	0.3	67.5	0.5	73.9	0.5	81.9	0.6	85.25	0.6
81.9	0.6	74.35	0.6	80.45	0.6	73.2	0.5	72.5	0.5	38.45	0.3	53.35	0.4	51.15	0.4	35.8	0.3	80.4	0.6	77.4	0.6	81.15	0.6
79.3	0.6	78.7	0.6	82.95	0.6	74.1	0.5	67.7	0.5	49.35	0.4	55.3	0.4	56.95	0.4	33.6	0.2	40.75	0.3	75.95	0.6	63.1	0.5
79.9	0.6	87.2	0.6	79.35	0.6	69.8	0.5	69.7	0.5	53.95	0.4	53	0.4	55.85	0.4	47.2	0.3	44.25	0.3	50	0.4	56.9	0.4
68.5	0.5	89.9	0.7	88.05	0.7	66.45	0.5	64.7	0.5	59.1	0.4	42.05	0.3	33.5	0.2	64.8	0.5	57.25	0.4	48.45	0.4	53.2	0.4
80	0.6	75.7	0.6	86.25	0.6	55.35	0.4	76	0.6	39.4	0.3	37	0.3	45.2	0.3	54.4	0.4	48.65	0.4	62.05	0.5	43.1	0.3
70.75	0.5	70.5	0.5	78.9	0.6	79.9	0.6	81.15	0.6	53.1	0.4	18.65	0.1	57.1	0.4	48	0.4	40.55	0.3	64.2	0.5	65.85	0.5
77	0.6	78.7	0.6	87.5	0.6	73.1	0.5	81.2	0.6	47.35	0.4	53.9	0.4	62.65	0.5	50.65	0.4	37.95	0.3	76.7	0.6	77.55	0.6
78.6	0.6	81.4	0.6	80.2	0.6	72.05	0.5	71.6	0.5	51.65	0.4	72.5	0.5	64.5	0.5	50.05	0.4	64.6	0.5	81	0.6	75.35	0.6
76.35	0.6	80.15	0.6	84.2	0.6	72.55	0.5	70.95	0.5	51.65	0.4	43.5	0.3	43.45	0.3	40.7	0.3	59.6	0.4	80.95	0.6	76.65	0.6
70.05	0.5	86.9	0.6	87.5	0.6	75.1	0.6	71.65	0.5	45.7	0.3	42.25	0.3	45.85	0.3	77.6	0.6	75.1	0.6	82.05	0.6	73.3	0.5
68.1	0.5	86.95	0.6	91.15	0.7	76.55	0.6	73.05	0.5	23.3	0.2	33.3	0.2	27.35	0.2	43.95	0.3	79.2	0.6	73.5	0.5	68.35	0.5
79.1	0.6	81.75	0.6	87.7	0.6	71.4	0.5	73.45	0.5	41.3	0.3	58.95	0.4	58.15	0.4	26.4	0.2	47.35	0.4	77.3	0.6	62.7	0.5
85.45	0.6	84.4	0.6	88.45	0.7	63.7	0.5	67.95	0.5	54.65	0.4	51.9	0.4	55.4	0.4	45.3	0.3	34.15	0.3	47	0.3	72.35	0.5
74.7	0.6	83	0.6	90.95	0.7	58.8	0.4	67.15	0.5	60.95	0.5	44.25	0.3	35.4	0.3	66	0.5	51	0.4	45.3	0.3	63.55	0.5
72.35	0.5	77.8	0.6	88	0.7	64.2	0.5	90.35	0.7	42.9	0.3	59.3	0.4	42.15	0.3	52.3	0.4	58.15	0.4	55	0.4	70.3	0.5
72.1	0.5	84.85	0.6	70.9	0.5	75.8	0.6	110	0.8	28.6	0.2	28.1	0.2	47.4	0.4	49.9	0.4	50.55	0.4	77.9	0.6	73.85	0.5

speed (km/h)	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value
72.8	0.5	88.1	0.7	89.95	0.7	80.85	0.6	80.2	0.6	81.1	0.6	78.75	0.6	86.1	0.6	82.15	0.6	76	0.6	74.3	0.6	86	0.6
70.95	0.5	86.15	0.6	85.7	0.6	83.4	0.6	79.85	0.6	80.5	0.6	75.5	0.6	89.75	0.7	80.65	0.6	81.25	0.6	74.7	0.6	81.45	0.6
69.25	0.5	85.15	0.6	84	0.6	79.8	0.6	81.9	0.6	85.7	0.6	72.65	0.5	77.6	0.6	77.75	0.6	85.75	0.6	79.55	0.6	78.8	0.6
64.4	0.5	84.55	0.6	82.45	0.6	80.7	0.6	83.45	0.6	82.4	0.6	82.05	0.6	80.15	0.6	65.4	0.5	91.6	0.7	92	0.7	80.85	0.6
73.65	0.5	85.7	0.6	82.4	0.6	78.65	0.6	81.85	0.6	79.35	0.6	76.6	0.6	77.95	0.6	73.3	0.5	81.8	0.6	87.95	0.7	86.05	0.6
73	0.5	85	0.6	85.5	0.6	82.7	0.6	81.5	0.6	81.25	0.6	73.65	0.5	80.8	0.6	92.5	0.7	83.8	0.6	82.2	0.6	92.95	0.7
77.05	0.6	72.1	0.5	84	0.6	85.45	0.6	84.05	0.6	80.05	0.6	75.8	0.6	77.35	0.6	88.8	0.7	80.7	0.6	83.6	0.6	86.9	0.6
71.6	0.5	79.95	0.6	77.2	0.6	91.5	0.7	87.85	0.7	78.6	0.6	77.05	0.6	77.4	0.6	81.15	0.6	78.75	0.6	72.7	0.5	85.55	0.6
72.2	0.5	83.5	0.6	80.05	0.6	81.6	0.6	84.7	0.6	90.95	0.7	78.2	0.6	82.1	0.6	76	0.6	78.5	0.6	79.5	0.6	82.85	0.6
72.25	0.5	84.55	0.6	87.9	0.7	83.2	0.6	70.4	0.5	92.15	0.7	77.35	0.6	90.6	0.7	76.65	0.6	83.25	0.6	83.75	0.6	80	0.6
72.6	0.5	83.6	0.6	85.1	0.6	78.9	0.6	75	0.6	77.6	0.6	77.85	0.6	88.75	0.7	73.85	0.5	91.5	0.7	84.5	0.6	83.3	0.6
64.45	0.5	84.8	0.6	80.6	0.6	79	0.6	79.45	0.6	86.3	0.6	72.6	0.5	85.45	0.6	78.05	0.6	90.5	0.7	85.7	0.6	90.3	0.7
63.8	0.5	81.7	0.6	77.4	0.6	83.15	0.6	78.7	0.6	79.8	0.6	78.9	0.6	75.6	0.6	87.85	0.7	95.45	0.7	85.4	0.6	93.5	0.7
68.45	0.5	84.25	0.6	79.7	0.6	91.2	0.7	83.2	0.6	80.2	0.6	71.5	0.5	75.5	0.6	74.7	0.6	96.2	0.7	86.15	0.6	90.95	0.7
86.55	0.6	86.45	0.6	84.35	0.6	92.75	0.7	81.4	0.6	80.4	0.6	74.1	0.5	82.45	0.6	77.4	0.6	76.2	0.6	87.85	0.7	92.65	0.7
82.1	0.6	87.7	0.6	86.75	0.6	95.3	0.7	80.4	0.6	83.75	0.6	76.55	0.6	87.25	0.6	72.2	0.5	74.6	0.6	75.6	0.6	89.25	0.7
81.45	0.6	75.1	0.6	89.95	0.7	94.85	0.7	80.45	0.6	91.8	0.7	78.95	0.6	80.65	0.6	69.7	0.5	83.85	0.6	83.15	0.6	80.4	0.6
71.8	0.5	79.05	0.6	79.3	0.6	90.45	0.7	82.15	0.6	99.25	0.7	74.45	0.6	80.9	0.6	74.2	0.5	87.85	0.7	82.6	0.6	79.55	0.6
72.95	0.5	83.5	0.6	76.35	0.6	76.9	0.6	84.35	0.6	101.15	0.7	75.25	0.6	83.1	0.6	79	0.6	93.25	0.7	83.4	0.6	99.45	0.7
72.95	0.5	84.45	0.6	76.6	0.6	82.1	0.6	72.1	0.5	99.15	0.7	83	0.6	80.95	0.6	79.05	0.6	89.1	0.7	88	0.7	91.5	0.7
72.55	0.5	81.6	0.6	79	0.6	88.4	0.7	81.5	0.6	79.8	0.6	82.85	0.6	86.05	0.6	79.45	0.6	87.3	0.6	87.95	0.7	90.35	0.7
77.15	0.6	84.4	0.6	76.6	0.6	88.85	0.7	87.1	0.6	89.35	0.7	79.6	0.6	80.6	0.6	77.5	0.6	84.1	0.6	83.9	0.6	86.35	0.6
82.35	0.6	90.7	0.7	83.35	0.6	95.35	0.7	86.55	0.6	86	0.6	78.3	0.6	78.9	0.6	81.15	0.6	85.9	0.6	83.25	0.6	83.3	0.6
91.05	0.7	93.95	0.7	82.8	0.6	90.3	0.7	80.05	0.6	95.65	0.7	75.4	0.6	81.4	0.6	77	0.6	83.65	0.6	86.35	0.6	88.85	0.7
94.45	0.7	93.2	0.7	87.45	0.6	87.75	0.7	85.95	0.6	93.05	0.7	79.35	0.6	79.05	0.6	83.6	0.6	70	0.5	85.15	0.6	86.8	0.6
83	0.6	91.8	0.7	89.15	0.7	87.1	0.6	82.75	0.6	87	0.6	75.85	0.6	80.25	0.6	72.95	0.5	70.55	0.5	75	0.6	84.9	0.6
79.65	0.6	81.7	0.6	83	0.6	88.6	0.7	83.6	0.6	84.75	0.6	77.4	0.6	82.1	0.6	75.4	0.6	83.3	0.6	77.5	0.6	77.2	0.6
78.45	0.6	78.85	0.6	77.8	0.6	89.2	0.7	83.9	0.6	87.05	0.6	83.35	0.6	86.5	0.6	77.75	0.6	82.15	0.6	82.8	0.6	79.85	0.6
78.4	0.6	87.95	0.7	73.3	0.5	79.4	0.6	89.35	0.7	88.45	0.7	82.85	0.6	83.9	0.6	73.25	0.5	83.15	0.6	81.6	0.6	80.4	0.6
83.25	0.6	86	0.6	85.05	0.6	76.9	0.6	80.4	0.6	82.05	0.6	84.9	0.6	83.8	0.6	66.45	0.5	83.45	0.6	78.15	0.6	85.55	0.6
87.7	0.6	86.85	0.6	86.6	0.6	82.75	0.6	76.05	0.6	72.9	0.5	80.05	0.6	90.45	0.7	77.5	0.6	81.85	0.6	78.05	0.6	84.4	0.6
91.85	0.7	89.45	0.7	81.45	0.6	85.45	0.6	85.1	0.6	72.2	0.5	71.7	0.5	91.65	0.7	82.9	0.6	79.9	0.6	79.55	0.6	82.65	0.6
94.2	0.7	94.3	0.7	77.85	0.6	82.25	0.6	84.2	0.6	89.9	0.7	75.35	0.6	76.6	0.6	81.95	0.6	75.25	0.6	84.15	0.6	73.7	0.5
91.7	0.7	88.7	0.7	79	0.6	84.55	0.6	83.7	0.6	90.85	0.7	80.15	0.6	70.85	0.5	73.6	0.5	78.45	0.6	86.6	0.6	76.75	0.6
87.45	0.6	93	0.7	82.3	0.6	81.65	0.6	84.9	0.6	86.55	0.6	80.45	0.6	86.4	0.6	74.75	0.6	72.7	0.5	84.4	0.6	80.6	0.6
76	0.6	89.45	0.7	84.05	0.6	82.3	0.6	84.7	0.6	82.95	0.6	79.7	0.6	89.95	0.7	77.25	0.6	84.05	0.6	81.3	0.6	86.75	0.6
80.2	0.6	78.6	0.6	83.65	0.6	85.65	0.6	80.6	0.6	77.2	0.6	76.4	0.6	88.75	0.7	77.8	0.6	81.95	0.6	82.7	0.6	73.9	0.5
87.35	0.6	79.1	0.6	77.2	0.6	85.55	0.6	80.4	0.6	76.7	0.6	75	0.6	88.95	0.7	78.35	0.6	79.15	0.6	75.6	0.6	77.45	0.6
90.6	0.7	91	0.7	78.05	0.6	73.6	0.5	82.2	0.6	77.6	0.6	79.35	0.6	85.95	0.6	73.6	0.5	78.5	0.6	81.9	0.6	72.4	0.5

speed (km/h)	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value
77.1	0.6	79.15	0.6	79.05	0.6	84.5	0.6	78.9	0.6	89.15	0.7	72.85	0.5	88.9	0.7	72.25	0.5	82.85	0.6	81.05	0.6	76.55	0.6
77.15	0.6	81.85	0.6	76.1	0.6	89.9	0.7	76.2	0.6	85.7	0.6	79	0.6	91.15	0.7	74.6	0.6	80.85	0.6	82.1	0.6	88.35	0.7
77.85	0.6	78.8	0.6	80.1	0.6	82	0.6	77.1	0.6	77.2	0.6	91.65	0.7	91.65	0.7	80.1	0.6	84.8	0.6	80.8	0.6	84.55	0.6
83.1	0.6	80.25	0.6	80.05	0.6	83.55	0.6	68.3	0.5	78.1	0.6	83	0.6	84.9	0.6	86.25	0.6	80.1	0.6	84	0.6	83	0.6
85.25	0.6	79.75	0.6	83.85	0.6	88.25	0.7	73.95	0.5	72.95	0.5	84.75	0.6	71.8	0.5	87.05	0.6	76.65	0.6	80.65	0.6	80.4	0.6
82.5	0.6	84.4	0.6	92.1	0.7	95.45	0.7	74.85	0.6	77.8	0.6	77.9	0.6	80.2	0.6	77.9	0.6	73.7	0.5	82.9	0.6	79.65	0.6
82.8	0.6	90.75	0.7	93.25	0.7	87.95	0.7	74.8	0.6	82.9	0.6	81	0.6	87.15	0.6	78.8	0.6	72.3	0.5	87.65	0.6	71.8	0.5
77	0.6	88.25	0.7	88.45	0.7	89.9	0.7	72.5	0.5	87.35	0.6	77.45	0.6	91.9	0.7	71.05	0.5	79.65	0.6	77.2	0.6	79.5	0.6
73.65	0.5	70.8	0.5	85.3	0.6	91.85	0.7	71.7	0.5	86.85	0.6	72.75	0.5	94.85	0.7	73.45	0.5	84.6	0.6	77.1	0.6	81.75	0.6
69.7	0.5	77.65	0.6	73.6	0.5	81.35	0.6	85.3	0.6	92.45	0.7	75.9	0.6	87	0.6	75.55	0.6	92.1	0.7	85.15	0.6	87.5	0.6
77.65	0.6	83.3	0.6	80.35	0.6	67.4	0.5	88.15	0.7	86.6	0.6	79.1	0.6	82.6	0.6	82.9	0.6	83.75	0.6	85.65	0.6	85	0.6
75.95	0.6	83.55	0.6	86.95	0.6	73.9	0.5	78.1	0.6	83.75	0.6	84.8	0.6	82.3	0.6	82.85	0.6	72.95	0.5	83.7	0.6	85.7	0.6
80.4	0.6	85.4	0.6	86.85	0.6	88.3	0.7	75.85	0.6	77.8	0.6	85.85	0.6	85.8	0.6	79.65	0.6	71.15	0.5	81.55	0.6	83.9	0.6
82.15	0.6	81.5	0.6	90.45	0.7	86.4	0.6	71.3	0.5	81.45	0.6	82.3	0.6	87.9	0.7	89.8	0.7	76.15	0.6	81.1	0.6	76.95	0.6
81.65	0.6	78.8	0.6	81.3	0.6	83.55	0.6	75.75	0.6	90.35	0.7	84.2	0.6	77.2	0.6	93.1	0.7	76.25	0.6	82	0.6	74.45	0.6
81.45	0.6	77	0.6	80.85	0.6	81.45	0.6	76.95	0.6	88.05	0.7	76.85	0.6	82.7	0.6	89.8	0.7	82.2	0.6	88.1	0.7	84.75	0.6
79.4	0.6	80.8	0.6	84.5	0.6	77.95	0.6	80.1	0.6	85.5	0.6	78.05	0.6	85.05	0.6	91.3	0.7	78.4	0.6	87.8	0.7	89.75	0.7
72.1	0.5	81	0.6	87.85	0.7	75.4	0.6	79.5	0.6	80.95	0.6	81.15	0.6	85.45	0.6	76.9	0.6	78.55	0.6	75.2	0.6	85.35	0.6
77	0.6	76.7	0.6	91.1	0.7	77.65	0.6	79.7	0.6	80.05	0.6	78.9	0.6	86.75	0.6	82.4	0.6	73.8	0.5	76.35	0.6	86.25	0.6
72.05	0.5	76.6	0.6	77.2	0.6	78.25	0.6	83.15	0.6	77	0.6	69.7	0.5	82.05	0.6	89.05	0.7	76.35	0.6	83.85	0.6	87.1	0.6
75.7	0.6	83.7	0.6	79.85	0.6	73.6	0.5	79.75	0.6	87	0.6	86.35	0.6	84.2	0.6	84.4	0.6	71.45	0.5	84.05	0.6	79	0.6
78.05	0.6	78.7	0.6	78.3	0.6	76.65	0.6	73	0.5	88.15	0.7	100.3	0.7	89.7	0.7	88.65	0.7	75.75	0.6	80.55	0.6	73.6	0.5
74.8	0.6	74.45	0.6	79.7	0.6	80.9	0.6	77.05	0.6	76.4	0.6	93.9	0.7	85.5	0.6	104	0.8	73.3	0.5	89.45	0.7	83.9	0.6
70.45	0.5	75.25	0.6	80.4	0.6	81.9	0.6	78.4	0.6	78.6	0.6	85.9	0.6	85.85	0.6	100.95	0.7	75.85	0.6	88.65	0.7	88.65	0.7
74.85	0.6	73.25	0.5	84.3	0.6	78.55	0.6	78.7	0.6	78.25	0.6	84.7	0.6	76.1	0.6	100	0.7	87.35	0.6	90.95	0.7	84.4	0.6
86.9	0.6	83.5	0.6	85.05	0.6	74	0.5	80.4	0.6	73.4	0.5	82.1	0.6	74.7	0.6	80.2	0.6	88	0.7	95.1	0.7	83.35	0.6
91	0.7	89.7	0.7	83.25	0.6	77.35	0.6	78.15	0.6	78.05	0.6	85.4	0.6	76.9	0.6	82.6	0.6	80.7	0.6	94.7	0.7	87.55	0.6
83	0.6	82.4	0.6	84.2	0.6	75.85	0.6	72.25	0.5	87	0.6	92.45	0.7	82.05	0.6	91	0.7	78.85	0.6	83	0.6	76.55	0.6
75.6	0.6	75	0.6	91.8	0.7	79.75	0.6	77.55	0.6	78.85	0.6	88.55	0.7	82.05	0.6	93.5	0.7	70.1	0.5	82.85	0.6	81.6	0.6
78.75	0.6	77.85	0.6	78.6	0.6	79.05	0.6	78.7	0.6	84.55	0.6	91.45	0.7	82.65	0.6	102.3	0.8	75.6	0.6	80.65	0.6	82.35	0.6
79.35	0.6	66.45	0.5	86.15	0.6	72.5	0.5	83.45	0.6	76.2	0.6	88.15	0.7	76.3	0.6	102.3	0.8	79.15	0.6	86.9	0.6	89.85	0.7
76.55	0.6	67.45	0.5	83.65	0.6	70.85	0.5	80.3	0.6	82.8	0.6	93.85	0.7	78.25	0.6	89.75	0.7	82.05	0.6	91.25	0.7	81.35	0.6
71.3	0.5	73.25	0.5	84.1	0.6	67.5	0.5	80.5	0.6	79.8	0.6	90.45	0.7	85.85	0.6	81.95	0.6	80.45	0.6	86.7	0.6	79.95	0.6
81.95	0.6	82.45	0.6	81	0.6	74.05	0.5	78.05	0.6	76.6	0.6	85.6	0.6	83.3	0.6	84.2	0.6	86.5	0.6	92.85	0.7	83.55	0.6
85.4	0.6	76.55	0.6	89.1	0.7	76.25	0.6	81.75	0.6	89.1	0.7	83.85	0.6	75.9	0.6	89.4	0.7	83.85	0.6	91.2	0.7	75.85	0.6
88.15	0.7	77.55	0.6	84.25	0.6	74.3	0.6	77.7	0.6	89.15	0.7	89.9	0.7	77.55	0.6	76.2	0.6	83.7	0.6	80.9	0.6	75.8	0.6
83.55	0.6	80.3	0.6	92.9	0.7	71.95	0.5	79.35	0.6	83.45	0.6	90.7	0.7	77.5	0.6	73.6	0.5	75.4	0.6	79.45	0.6	78.7	0.6
74.3	0.6	94.3	0.7	100.8	0.7	72	0.5	80.95	0.6	78.4	0.6	89	0.7	81.5	0.6	89.65	0.7	78.9	0.6	79.2	0.6	86	0.6
77.5	0.6	83	0.6	103.8	0.8	79.3	0.6	84.3	0.6	72.8	0.5	92.3	0.7	80.8	0.6	84.45	0.6	77.25	0.6	76.1	0.6	82.1	0.6

speed (km/h)	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value
84.75	0.6	85.3	0.6	92.2	0.7	90.5	0.7	81.1	0.6	81.05	0.6	89.5	0.7	84.4	0.6	86	0.6	83.95	0.6	92.9	0.7	82.6	0.6
89.3	0.7	85.85	0.6	86.6	0.6	86.65	0.6	81.85	0.6	79.5	0.6	85.45	0.6	89.1	0.7	81.9	0.6	70.25	0.5	83.5	0.6	86.2	0.6
79.9	0.6	88.2	0.7	86.55	0.6	91	0.7	84.2	0.6	79.8	0.6	96.65	0.7	85.05	0.6	82.7	0.6	81	0.6	84.8	0.6	88.1	0.7
75.25	0.6	86.1	0.6	86.65	0.6	93.75	0.7	90.1	0.7	77.4	0.6	91	0.7	69.2	0.5	86.45	0.6	80.6	0.6	86.15	0.6	92.7	0.7
81.2	0.6	87.8	0.7	101	0.7	80.1	0.6	84.1	0.6	77.8	0.6	92.25	0.7	90.2	0.7	85.5	0.6	84.55	0.6	80.1	0.6	83.7	0.6
97.35	0.7	81.95	0.6	94.5	0.7	83.45	0.6	87.6	0.6	78.1	0.6	92.05	0.7	99.85	0.7	74.8	0.6	78.85	0.6	86.1	0.6	87.8	0.7
90.65	0.7	85.45	0.6	94.1	0.7	90.15	0.7	85.85	0.6	84.85	0.6	76.6	0.6	94.15	0.7	79.2	0.6	76.15	0.6	87	0.6	88.3	0.7
97.75	0.7	86.3	0.6	89.75	0.7	95.1	0.7	68.85	0.5	87.1	0.6	88.05	0.7	91	0.7	81.55	0.6	79.05	0.6	92.65	0.7	80.1	0.6
93.35	0.7	89.5	0.7	78.7	0.6	91.5	0.7	89.95	0.7	87.75	0.7	90.35	0.7	89.55	0.7	86.85	0.6	79.2	0.6	85.15	0.6	83.65	0.6
84.9	0.6	90.75	0.7	91.15	0.7	90	0.7	87.65	0.6	95.85	0.7	88.45	0.7	77.4	0.6	86.15	0.6	76.25	0.6	85.6	0.6	91.35	0.7
93.95	0.7	85.5	0.6	88.9	0.7	94.25	0.7	87.75	0.7	81.15	0.6	83.05	0.6	92.6	0.7	82.25	0.6	74.45	0.6	85.3	0.6	90.65	0.7
91.65	0.7	89.35	0.7	86.85	0.6	83.2	0.6	85	0.6	79.65	0.6	84.45	0.6	81.65	0.6	85.55	0.6	76.25	0.6	78.8	0.6	86.3	0.6
95.75	0.7	83.3	0.6	86.6	0.6	86.7	0.6	84	0.6	96.95	0.7	84.5	0.6	91.75	0.7	84.3	0.6	82.1	0.6	78.1	0.6	86.4	0.6
86.15	0.6	83.65	0.6	89	0.7	96.9	0.7	81.65	0.6	89.85	0.7	79.55	0.6	86.3	0.6	86.5	0.6	83.2	0.6	86.3	0.6	86	0.6
88.15	0.7	83.85	0.6	88	0.7	88.05	0.7	71.35	0.5	89.9	0.7	83	0.6	85.65	0.6	83.45	0.6	87	0.6	92.5	0.7	74.9	0.6
88	0.7	87.15	0.6	84.7	0.6	91.75	0.7	82.05	0.6	86.5	0.6	83.1	0.6	81.8	0.6	90.35	0.7	80.2	0.6	87.95	0.7	85.45	0.6
76.8	0.6	88.6	0.7	82.15	0.6	82.4	0.6	82	0.6	83.05	0.6	84.5	0.6	75.7	0.6	89.45	0.7	79.6	0.6	86.75	0.6	82.25	0.6
83.7	0.6	88.35	0.7	83.45	0.6	86.55	0.6	83.3	0.6	79.25	0.6	84.8	0.6	78.4	0.6	89.75	0.7	80.4	0.6	87.55	0.6	86.55	0.6
86.8	0.6	86.55	0.6	87.85	0.7	78.5	0.6	81.6	0.6	84.35	0.6	88.1	0.7	78.25	0.6	97.15	0.7	91.4	0.7	80.95	0.6	79.55	0.6
93.25	0.7	79.4	0.6	79.95	0.6	84.55	0.6	81.1	0.6	81.3	0.6	89	0.7	86.05	0.6	89.8	0.7	86.8	0.6	81.35	0.6	79.95	0.6
87.35	0.6	83.25	0.6	77.45	0.6	86.55	0.6	78.35	0.6	87.05	0.6	82.65	0.6	81.75	0.6	70.75	0.5	89.15	0.7	85.5	0.6	81.65	0.6
83.8	0.6	86.65	0.6	84.05	0.6	90.05	0.7	86.9	0.6	86.15	0.6	86.6	0.6	82.15	0.6	79.6	0.6	88.95	0.7	90.2	0.7	79.8	0.6
79.4	0.6	88.9	0.7	75.7	0.6	89.45	0.7	74.15	0.5	82.65	0.6	88.75	0.7	86.4	0.6	94.65	0.7	85.35	0.6	85.5	0.6	75.9	0.6
66.65	0.5	83.4	0.6	71.15	0.5	94.9	0.7	82.8	0.6	91.95	0.7	96.2	0.7	84.55	0.6	96	0.7	85.9	0.6	86.7	0.6	77.25	0.6
81.9	0.6	85.4	0.6	72.15	0.5	87.7	0.6	84.7	0.6	83.65	0.6	89.2	0.7	70.25	0.5	104.6	0.8	87.25	0.6	85.15	0.6	84.8	0.6
76.75	0.6	87.4	0.6	80.95	0.6	84.75	0.6	81.95	0.6	85.2	0.6	82.45	0.6	82.5	0.6	95.45	0.7	93.05	0.7	78.85	0.6	79.1	0.6
79.95	0.6	70.7	0.5	85.65	0.6	89.6	0.7	83.95	0.6	86.2	0.6	82.05	0.6	90.05	0.7	92	0.7	88	0.7	85.25	0.6	86.25	0.6
77.5	0.6	79.1	0.6	86.5	0.6	95.15	0.7	88.7	0.7	87.8	0.7	73.85	0.5	82.7	0.6	82.2	0.6	94.6	0.7	84.85	0.6	87.65	0.6
74.2	0.5	84.35	0.6	90	0.7	90.9	0.7	80.65	0.6	85.65	0.6	82.15	0.6	82.25	0.6	95.35	0.7	94.5	0.7	84.95	0.6	78.45	0.6
76.95	0.6	83.8	0.6	76.75	0.6	91.7	0.7	80.75	0.6	83.7	0.6	77.55	0.6	87.1	0.6	93.85	0.7	91.85	0.7	83.4	0.6	74.5	0.6
83.15	0.6	79.4	0.6	86.4	0.6	90.05	0.7	78.55	0.6	90.6	0.7	82.5	0.6	78.8	0.6	92.1	0.7	90.15	0.7	80.2	0.6	77.25	0.6
74.7	0.6	80.95	0.6	90.85	0.7	94.8	0.7	88.9	0.7	83.4	0.6	84.6	0.6	82.8	0.6	99.65	0.7	83.85	0.6	86.8	0.6	83.2	0.6
79.95	0.6	84.75	0.6	83.75	0.6	82.95	0.6	88.45	0.7	85	0.6	82.7	0.6	83.7	0.6	85.9	0.6	98.2	0.7	80.45	0.6	85.75	0.6
92.4	0.7	82.8	0.6	86.15	0.6	85.75	0.6	90.95	0.7	87.75	0.7	82.15	0.6	86.9	0.6	80.65	0.6	93.35	0.7	82.35	0.6	90.35	0.7
96.75	0.7	82.75	0.6	83.7	0.6	94.45	0.7	92	0.7	91.2	0.7	76.9	0.6	84.05	0.6	87.05	0.6	93.8	0.7	78.1	0.6	85.95	0.6
87.2	0.6	84.4	0.6	83.65	0.6	95.4	0.7	76.45	0.6	84.9	0.6	71.2	0.5	88.65	0.7	84.95	0.6	96.45	0.7	82.55	0.6	80.5	0.6
87	0.6	90	0.7	78.25	0.6	87.15	0.6	81.65	0.6	88.15	0.7	78.35	0.6	86.85	0.6	85.9	0.6	83.45	0.6	85.65	0.6	89.55	0.7
79.2	0.6	88.95	0.7	80.1	0.6	83.95	0.6	85.35	0.6	91.25	0.7	88.4	0.7	87	0.6	86.6	0.6	94.25	0.7	88.9	0.7	88.05	0.7
86.2	0.6	89.25	0.7	81	0.6	85.5	0.6	86	0.6	85.1	0.6	83.4	0.6	77.95	0.6	85.75	0.6	90.1	0.7	84.55	0.6	93.55	0.7

speed (km/h)	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value
87.95	0.7	77.75	0.6	86.55	0.6	84.15	0.6	85.75	0.6	95.6	0.7	77.55	0.6	81.6	0.6	78.85	0.6	77.75	0.6	73.65	0.5	87.35	0.6
89.6	0.7	79.95	0.6	87.15	0.6	91.55	0.7	82.25	0.6	92.15	0.7	85.3	0.6	85.05	0.6	75.4	0.6	80.65	0.6	75.85	0.6	72.95	0.5
86.4	0.6	82.25	0.6	68.9	0.5	83.3	0.6	78.8	0.6	83.85	0.6	83.9	0.6	81.55	0.6	82.55	0.6	69.4	0.5	77.75	0.6	76.7	0.6
83.4	0.6	84.55	0.6	83	0.6	91.7	0.7	82.25	0.6	85.45	0.6	83.6	0.6	76.05	0.6	77.95	0.6	70.65	0.5	74.3	0.6	85.5	0.6
85.15	0.6	86.25	0.6	86.4	0.6	84.75	0.6	100.6	0.7	75.5	0.6	79.45	0.6	76.4	0.6	80.8	0.6	70.3	0.5	73.85	0.5	87.2	0.6
90.2	0.7	90.2	0.7	83.95	0.6	85.35	0.6	86.9	0.6	93	0.7	81.15	0.6	84.1	0.6	82.3	0.6	80.6	0.6	86.5	0.6	82.65	0.6
95.05	0.7	87.2	0.6	81.25	0.6	82.85	0.6	90.45	0.7	81.45	0.6	82.7	0.6	86.3	0.6	75.65	0.6	74.9	0.6	71.95	0.5	82.35	0.6
91.6	0.7	78	0.6	69.3	0.5	86.25	0.6	96.9	0.7	83	0.6	77.6	0.6	83.1	0.6	59.7	0.4	74.75	0.6	72.9	0.5	78.25	0.6
99.05	0.7	90.65	0.7	73	0.5	83.8	0.6	85.6	0.6	76.2	0.6	76.15	0.6	86.8	0.6	81.25	0.6	73.6	0.5	82.15	0.6	77.45	0.6
101.1	0.7	103.4	0.8	73.55	0.5	83.5	0.6	89.05	0.7	75.4	0.6	78.8	0.6	87.75	0.7	91.6	0.7	71.9	0.5	86.35	0.6	76.5	0.6
86.75	0.6	89.45	0.7	82.25	0.6	85.55	0.6	94.8	0.7	82.2	0.6	85.4	0.6	82.25	0.6	91.15	0.7	71.95	0.5	84	0.6	75.6	0.6
94.8	0.7	88.15	0.7	75.05	0.6	85.2	0.6	88.85	0.7	81.05	0.6	84.4	0.6	81.15	0.6	87.85	0.7	72.15	0.5	84.8	0.6	83.4	0.6
95.05	0.7	91.3	0.7	75.65	0.6	75.7	0.6	88.35	0.7	70.95	0.5	84.8	0.6	84.95	0.6	89.35	0.7	74.7	0.6	78.45	0.6	82.9	0.6
91	0.7	78.85	0.6	77.4	0.6	69.9	0.5	88	0.7	81.45	0.6	84.15	0.6	86.4	0.6	83.5	0.6	73.5	0.5	86.4	0.6	83.75	0.6
89.65	0.7	91.05	0.7	77.6	0.6	72.2	0.5	88.7	0.7	82.15	0.6	82.45	0.6	85.7	0.6	66	0.5	69.35	0.5	80.7	0.6	84.75	0.6
86.25	0.6	85.8	0.6	80.25	0.6	85.65	0.6	74.2	0.5	79.4	0.6	75.6	0.6	81.35	0.6	84.25	0.6	72.2	0.5	89.7	0.7	76.75	0.6
90.8	0.7	88.35	0.7	76.75	0.6	84.85	0.6	86.55	0.6	81.65	0.6	84.45	0.6	79.45	0.6	89.9	0.7	71.3	0.5	83.8	0.6	79.05	0.6
90.55	0.7	86	0.6	74.35	0.6	83.9	0.6	81.35	0.6	84.35	0.6	87.8	0.7	80.25	0.6	83.6	0.6	61.9	0.5	85.55	0.6	76.4	0.6
91.2	0.7	91.65	0.7	73.9	0.5	83.55	0.6	87.35	0.6	81.55	0.6	85	0.6	84.05	0.6	81.25	0.6	70.2	0.5	83.45	0.6	83.25	0.6
87.75	0.7	91.95	0.7	75.6	0.6	80.2	0.6	84.6	0.6	54.6	0.4	88.05	0.7	81.75	0.6	83.4	0.6	77.1	0.6	74.3	0.6	80.65	0.6
87.05	0.6	85.8	0.6	80.6	0.6	85.5	0.6	79.35	0.6	84.1	0.6	86.8	0.6	89.75	0.7	72.95	0.5	77.05	0.6	84.4	0.6	81.4	0.6
84.6	0.6	82.45	0.6	85.35	0.6	85.05	0.6	83.75	0.6	86.45	0.6	84.45	0.6	82.9	0.6	74.3	0.6	79.75	0.6	84.35	0.6	84.5	0.6
87	0.6	87.5	0.6	84.9	0.6	93.2	0.7	87.3	0.6	85.65	0.6	83.95	0.6	82.8	0.6	75.3	0.6	83.3	0.6	88.2	0.7	84.5	0.6
83.4	0.6	99.05	0.7	84.75	0.6	91.45	0.7	78.6	0.6	87.1	0.6	82	0.6	86.4	0.6	82.25	0.6	72.35	0.5	84.65	0.6	76.05	0.6
67.55	0.5	88.05	0.7	73.85	0.5	89.1	0.7	88.05	0.7	81.35	0.6	88.15	0.7	75.6	0.6	81.85	0.6	43.7	0.3	87.8	0.7	79.1	0.6
79.05	0.6	86.15	0.6	78	0.6	88.7	0.7	102.15	0.8	78.1	0.6	82.45	0.6	74.4	0.6	79.75	0.6	75.6	0.6	88.4	0.7	86.7	0.6
80.15	0.6	85.65	0.6	88.8	0.7	83.95	0.6	86.15	0.6	67.4	0.5	82.2	0.6	74.35	0.6	80.05	0.6	87.8	0.7	80.7	0.6	83.4	0.6
87.2	0.6	80.45	0.6	89.95	0.7	82.95	0.6	92.6	0.7	78.1	0.6	83.55	0.6	79.9	0.6	80.15	0.6	85.15	0.6	87.8	0.7	86.4	0.6
84.45	0.6	82.7	0.6	86.35	0.6	85.75	0.6	85.7	0.6	87.9	0.7	78.8	0.6	81.45	0.6	69.25	0.5	79.25	0.6	90.2	0.7	85.3	0.6
78.75	0.6	86.8	0.6	86.65	0.6	90.75	0.7	80.4	0.6	81.6	0.6	79.5	0.6	86	0.6	77.3	0.6	81.7	0.6	87.75	0.7	83.3	0.6
81.95	0.6	92.6	0.7	77.15	0.6	84.15	0.6	85.8	0.6	81.3	0.6	79.75	0.6	84.85	0.6	83.55	0.6	79	0.6	87.05	0.6	80.5	0.6
82.15	0.6	93.85	0.7	79.75	0.6	82.55	0.6	86.3	0.6	91.1	0.7	86.1	0.6	75.55	0.6	85.8	0.6	49.05	0.4	84.55	0.6	78.45	0.6
79.15	0.6	87.95	0.7	81.3	0.6	81.2	0.6	84.6	0.6	82.6	0.6	85.5	0.6	83.85	0.6	85.7	0.6	76.05	0.6	81.45	0.6	84.55	0.6
82.45	0.6	96.85	0.7	91	0.7	76.7	0.6	80.4	0.6	80.9	0.6	77.2	0.6	83.5	0.6	77.55	0.6	86.45	0.6	77.1	0.6	84.95	0.6
86.6	0.6	91.5	0.7	86.65	0.6	82.05	0.6	83.65	0.6	84.05	0.6	88.45	0.7	87.8	0.7	69.3	0.5	78.9	0.6	77.4	0.6	82.45	0.6
78.65	0.6	90.55	0.7	91.55	0.7	78.55	0.6	87.5	0.6	87.15	0.6	81	0.6	84.6	0.6	81.5	0.6	80	0.6	79.85	0.6	82	0.6
81.1	0.6	92.55	0.7	90.6	0.7	82.2	0.6	89.55	0.7	81.1	0.6	82.15	0.6	80.65	0.6	79.1	0.6	80.1	0.6	83.6	0.6	69.4	0.5
83.75	0.6	101.15	0.7	81.3	0.6	81.55	0.6	78.85	0.6	86.3	0.6	85.5	0.6	77	0.6	76.95	0.6	72.4	0.5	84.95	0.6	84.55	0.6
78.9	0.6	90.25	0.7	79.8	0.6	84.5	0.6	83.25	0.6	86.15	0.6	88.35	0.7	76.15	0.6	75.1	0.6	64.35	0.5	82.9	0.6	79.25	0.6

speed (km/h)	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value
78.85	0.6	88.9	0.7	89.15	0.7	80.4	0.6	84.65	0.6	89.85	0.7	89.45	0.7	91.05	0.7	79.15	0.6	82.5	0.6	78.05	0.6	82.75	0.6
77.4	0.6	78.05	0.6	86.55	0.6	83	0.6	86.55	0.6	87.5	0.6	79.9	0.6	90.15	0.7	71.35	0.5	79.95	0.6	79.75	0.6	84.8	0.6
77.55	0.6	81.1	0.6	92.6	0.7	90.85	0.7	82.25	0.6	84.35	0.6	87.8	0.7	86.55	0.6	84.9	0.6	85.65	0.6	85.75	0.6	79.9	0.6
85	0.6	90.05	0.7	77.6	0.6	84.8	0.6	86.7	0.6	82.2	0.6	82.55	0.6	89.25	0.7	88.7	0.7	81.3	0.6	79.5	0.6	73.75	0.5
74.45	0.6	84.9	0.6	88.9	0.7	83.8	0.6	87.4	0.6	86.85	0.6	85.3	0.6	84.3	0.6	85.45	0.6	83.3	0.6	80.85	0.6	79.1	0.6
81.2	0.6	92.2	0.7	84.4	0.6	82.35	0.6	92.25	0.7	76.25	0.6	85.4	0.6	84.95	0.6	88.8	0.7	82.5	0.6	81.25	0.6	90	0.7
84.1	0.6	86.95	0.6	93.4	0.7	74.75	0.6	93.65	0.7	82.65	0.6	84.1	0.6	86.25	0.6	89.95	0.7	88.75	0.7	73.5	0.5	85.4	0.6
86.8	0.6	82.85	0.6	85.15	0.6	85.3	0.6	88.25	0.7	79.35	0.6	83.6	0.6	86.1	0.6	82.6	0.6	88.1	0.7	77.65	0.6	84.7	0.6
81.85	0.6	82.9	0.6	82.7	0.6	77.35	0.6	89.65	0.7	84.75	0.6	79.7	0.6	79.7	0.6	82.65	0.6	85.65	0.6	86.4	0.6	91.95	0.7
81	0.6	84.45	0.6	81.75	0.6	80.7	0.6	84.45	0.6	81.9	0.6	83.65	0.6	80.2	0.6	85.6	0.6	87.85	0.7	93.15	0.7	84.75	0.6
77.45	0.6	90.3	0.7	72.35	0.5	80.15	0.6	87	0.6	75.5	0.6	82.5	0.6	81.8	0.6	91.35	0.7	71.75	0.5	88.4	0.7	83.05	0.6
69.5	0.5	83.25	0.6	81.75	0.6	83.5	0.6	87.6	0.6	78	0.6	85.45	0.6	80	0.6	84.6	0.6	80.85	0.6	88.35	0.7	86	0.6
77.35	0.6	83.6	0.6	79.05	0.6	83.15	0.6	94.45	0.7	66.2	0.5	78.1	0.6	81.25	0.6	84.75	0.6	83.7	0.6	84.05	0.6	86.8	0.6
80.15	0.6	87.6	0.6	84.6	0.6	74.6	0.6	88.1	0.7	79.45	0.6	76.2	0.6	79	0.6	81.65	0.6	84.05	0.6	77.5	0.6	86.7	0.6
84.45	0.6	81.65	0.6	81.5	0.6	65.7	0.5	86.95	0.6	77.3	0.6	80.9	0.6	91.1	0.7	71.3	0.5	80.9	0.6	83.4	0.6	83.65	0.6
82.05	0.6	82.15	0.6	80.65	0.6	87.25	0.6	86	0.6	79.45	0.6	74.55	0.6	88.3	0.7	83.9	0.6	76.35	0.6	80.05	0.6	80.8	0.6
85.15	0.6	84.85	0.6	79.6	0.6	88.9	0.7	81.45	0.6	74.75	0.6	78.05	0.6	85.6	0.6	80.05	0.6	78.15	0.6	84.75	0.6	69.85	0.5
91.15	0.7	89.4	0.7	82.05	0.6	84.2	0.6	86.5	0.6	79.85	0.6	77.95	0.6	89.3	0.7	85	0.6	78	0.6	78.1	0.6	82.1	0.6
77.6	0.6	84	0.6	77.4	0.6	85.25	0.6	89.8	0.7	85.3	0.6	82.15	0.6	78.5	0.6	82.4	0.6	89.4	0.7	83.2	0.6	78.85	0.6
78.65	0.6	82.55	0.6	82.7	0.6	86.8	0.6	88.55	0.7	83.95	0.6	79.15	0.6	86.85	0.6	85.05	0.6	70.55	0.5	84.85	0.6	83.45	0.6
79.1	0.6	89.75	0.7	89.8	0.7	78.2	0.6	89.25	0.7	77.85	0.6	74.05	0.5	85.5	0.6	83.25	0.6	47.6	0.4	79.05	0.6	79.95	0.6
83.25	0.6	84.3	0.6	84.3	0.6	74.4	0.6	93.4	0.7	79.3	0.6	80.15	0.6	88	0.7	78.95	0.6	80.45	0.6	71.8	0.5	77.9	0.6
80.65	0.6	90.55	0.7	93.25	0.7	84.05	0.6	92.1	0.7	92.1	0.7	76.8	0.6	83.65	0.6	80.65	0.6	83.4	0.6	75.1	0.6	72.6	0.5
80.4	0.6	83.6	0.6	98.55	0.7	84.45	0.6	85.35	0.6	87.75	0.7	74.6	0.6	77.9	0.6	82.3	0.6	80.55	0.6	86.95	0.6	71.05	0.5
82.25	0.6	87.05	0.6	77.05	0.6	85.5	0.6	90	0.7	86.45	0.6	75.95	0.6	80.2	0.6	92.85	0.7	78.45	0.6	85.35	0.6	76.95	0.6
79.15	0.6	96.15	0.7	85.05	0.6	80.75	0.6	88.95	0.7	89.05	0.7	81	0.6	73.65	0.5	84.95	0.6	77.85	0.6	89.55	0.7	79.75	0.6
78.05	0.6	98.4	0.7	89.6	0.7	86.85	0.6	91.25	0.7	80.5	0.6	78.05	0.6	81.05	0.6	82.6	0.6	64.4	0.5	87.1	0.6	82.15	0.6
76.8	0.6	97	0.7	87.95	0.7	81.3	0.6	91.15	0.7	91.2	0.7	82.9	0.6	78.15	0.6	86.3	0.6	41.2	0.3	84.8	0.6	76.95	0.6
85.5	0.6	85.75	0.6	84.6	0.6	86.8	0.6	93.25	0.7	86.75	0.6	86.25	0.6	73.9	0.5	79.85	0.6	54.95	0.4	75.95	0.6	74	0.5
82.6	0.6	95.3	0.7	80.4	0.6	86.7	0.6	86.1	0.6	91.85	0.7	79.85	0.6	74.8	0.6	84.8	0.6	82.9	0.6	81.85	0.6	81.1	0.6
83.5	0.6	89.3	0.7	76.5	0.6	92.4	0.7	86.4	0.6	90.5	0.7	71.55	0.5	78.35	0.6	79.6	0.6	83.75	0.6	83	0.6	77.45	0.6
87.05	0.6	92.85	0.7	75.95	0.6	84.65	0.6	84.9	0.6	85.4	0.6	85.25	0.6	88.1	0.7	86.75	0.6	81.65	0.6	83.55	0.6	72	0.5
72.5	0.5	88.3	0.7	79.7	0.6	89.65	0.7	85.4	0.6	88	0.7	87.75	0.7	84.75	0.6	86.2	0.6	74.95	0.6	82.1	0.6	76.75	0.6
77.45	0.6	86.6	0.6	79.05	0.6	85.6	0.6	98.4	0.7	84.65	0.6	85.65	0.6	71.95	0.5	86.25	0.6	59.5	0.4	82.8	0.6	87.1	0.6
82.2	0.6	87.75	0.7	82.45	0.6	80.55	0.6	87.05	0.6	88.2	0.7	85.95	0.6	67.1	0.5	85.05	0.6	73.25	0.5	71.15	0.5	82.8	0.6
84.55	0.6	79.75	0.6	85.4	0.6	79	0.6	88	0.7	87.4	0.6	81.5	0.6	86.3	0.6	85.35	0.6	80.7	0.6	81.55	0.6	75.8	0.6
86.5	0.6	86.45	0.6	88.55	0.7	84.7	0.6	84.45	0.6	92.65	0.7	86.6	0.6	89.4	0.7	85.4	0.6	78.3	0.6	80.8	0.6	79.25	0.6
84.3	0.6	85.35	0.6	84.4	0.6	84.35	0.6	75.8	0.6	89.3	0.7	86.1	0.6	90.2	0.7	88.05	0.7	82.35	0.6	80.15	0.6	76.95	0.6
85.65	0.6	86.65	0.6	78.55	0.6	84.8	0.6	83.4	0.6	86.65	0.6	81	0.6	99.15	0.7	94.2	0.7	79.3	0.6	79.95	0.6	81.5	0.6

72.55	0.5	84.2	0.6	88.7	0.7	71	0.5	91.85	0.7	58.8	0.4	62.8	0.5	88.6	0.7	70.15	0.5	81.2	0.6	72.4	0.5	76.35	0.6
78.7	0.6	82.25	0.6	84.5	0.6	69.9	0.5	85.8	0.6	72.05	0.5	53.95	0.4	85.2	0.6	79	0.6	78.6	0.6	69.15	0.5	80.6	0.6
78.55	0.6	76.7	0.6	84.9	0.6	77.45	0.6	89.05	0.7	77.05	0.6	55.1	0.4	88.75	0.7	77.7	0.6	75.65	0.6	79.75	0.6	72.55	0.5
86.8	0.6	80.8	0.6	86.15	0.6	78.95	0.6	84.6	0.6	75.55	0.6	55.8	0.4	87.05	0.6	82.5	0.6	63.6	0.5	78.45	0.6	65.3	0.5
88.9	0.7	87.15	0.6	79.8	0.6	78.35	0.6	82.35	0.6	78.45	0.6	51.95	0.4	81.55	0.6	83.15	0.6	76.05	0.6	67.3	0.5	77.95	0.6
82.15	0.6	85.2	0.6	81.75	0.6	80.5	0.6	88.75	0.7	78	0.6	60.1	0.4	83.15	0.6	81.1	0.6	70.2	0.5	66.1	0.5	86.15	0.6
71.35	0.5	77.95	0.6	83.55	0.6	79.35	0.6	87.35	0.6	68.8	0.5	51.9	0.4	85.6	0.6	81.8	0.6	77.3	0.6	74.3	0.6	84.8	0.6
69.75	0.5	78.3	0.6	90.35	0.7	69.8	0.5	87.05	0.6	46.15	0.3	65.1	0.5	85.2	0.6	82.4	0.6	78.65	0.6	60.05	0.4	87.1	0.6
91.6	0.7	79.9	0.6	84.75	0.6	71.1	0.5	84.35	0.6	66.9	0.5	73.15	0.5	80.65	0.6	63.65	0.5	76.05	0.6	76.45	0.6	83.3	0.6
86.9	0.6	79.55	0.6	85.6	0.6	77.35	0.6	87.35	0.6	78.55	0.6	53.35	0.4	80.6	0.6	79.85	0.6	74.95	0.6	77.65	0.6	77.55	0.6
86.05	0.6	78.25	0.6	89.5	0.7	80.4	0.6	85.55	0.6	75.95	0.6	62.6	0.5	83.9	0.6	88.8	0.7	76.75	0.6	71.65	0.5	66.4	0.5
86.85	0.6	91.35	0.7	85.45	0.6	77.25	0.6	83.8	0.6	78.15	0.6	53.1	0.4	81.25	0.6	83	0.6	57.95	0.4	60	0.4	82.35	0.6
63.85	0.5	84	0.6	81.95	0.6	74.75	0.6	83.45	0.6	80.55	0.6	63.55	0.5	78.65	0.6	87.3	0.6	66.5	0.5	64.05	0.5	93.6	0.7
69.85	0.5	85.95	0.6	86.55	0.6	78.45	0.6	93.45	0.7	76.25	0.6	58.8	0.4	77.8	0.6	86.65	0.6	86.7	0.6	72.55	0.5	85.4	0.6
84.85	0.6	86.7	0.6	102.1	0.8	78.8	0.6	94.6	0.7	52.8	0.4	92.15	0.7	86.3	0.6	86.75	0.6	85.9	0.6	55.55	0.4	82	0.6
83.1	0.6	78	0.6	93.85	0.7	71.1	0.5	93.3	0.7	53.4	0.4	81.6	0.6	85.25	0.6	58.8	0.4	81.75	0.6	74.45	0.6	93.65	0.7
79.55	0.6	80.5	0.6	97.2	0.7	75.35	0.6	92.6	0.7	80.9	0.6	47.05	0.3	85.5	0.6	82.25	0.6	84.2	0.6	76.8	0.6	86	0.6
75.75	0.6	87.85	0.7	86.3	0.6	82.25	0.6	72.3	0.5	80.05	0.6	64.7	0.5	88.4	0.7	88	0.7	76.15	0.6	70.9	0.5	85.45	0.6
76.8	0.6	92.3	0.7	81.9	0.6	83.05	0.6	84.85	0.6	81.1	0.6	65.6	0.5	76.3	0.6	86.8	0.6	55.15	0.4	60.15	0.4	87.3	0.6
66.55	0.5	82.9	0.6	86.75	0.6	84.25	0.6	89.5	0.7	83.6	0.6	84.5	0.6	80.45	0.6	88.4	0.7	79.95	0.6	75.15	0.6	88.55	0.7
74.6	0.6	83.4	0.6	86.05	0.6	83.8	0.6	92.15	0.7	76.4	0.6	81.7	0.6	83.1	0.6	88.2	0.7	87.25	0.6	72.2	0.5	84.1	0.6
78	0.6	80.45	0.6	84.55	0.6	85.75	0.6	79.5	0.6	61.15	0.5	85.9	0.6	88.15	0.7	84.25	0.6	84.35	0.6	45.5	0.3	85.05	0.6
85.4	0.6	74.5	0.6	80.15	0.6	68.8	0.5	75.85	0.6	71.6	0.5	79	0.6	87.85	0.7	88.8	0.7	83.3	0.6	73.1	0.5	82.35	0.6
82.5	0.6	86.7	0.6	79.35	0.6	82.45	0.6	74.25	0.6	84.2	0.6	52.7	0.4	80.95	0.6	92.15	0.7	87.55	0.6	77.15	0.6	70.4	0.5
82.15	0.6	80.5	0.6	83.1	0.6	86.95	0.6	67.95	0.5	82.8	0.6	71.05	0.5	76.6	0.6	91.15	0.7	86.9	0.6	79.65	0.6	81.5	0.6
85.5	0.6	80.8	0.6	73.05	0.5	90.7	0.7	77.2	0.6	82.65	0.6	86.6	0.6	78.45	0.6	83.7	0.6	86.15	0.6	81.75	0.6	77.65	0.6
77.55	0.6	81.6	0.6	72.8	0.5	86.65	0.6	74.6	0.6	81.35	0.6	85.35	0.6	77.35	0.6	79.25	0.6	89.95	0.7	83.8	0.6	85.7	0.6
72.65	0.5	75.5	0.6	75.65	0.6	85.6	0.6	75.3	0.6	72.3	0.5	86.25	0.6	79.95	0.6	76.95	0.6	98.9	0.7	75.1	0.6	75.75	0.6
82.85	0.6	78.7	0.6	83.15	0.6	80.15	0.6	72	0.5	45.2	0.3	86	0.6	84.45	0.6	66.6	0.5	91.25	0.7	41.3	0.3	75.45	0.6
86.4	0.6	77.65	0.6	82.1	0.6	87.2	0.6	74.3	0.6	78.75	0.6	72.5	0.5	77.6	0.6	83.95	0.6	82.2	0.6	77.65	0.6	79.9	0.6
84.65	0.6	74.9	0.6	79.5	0.6	85.55	0.6	76.55	0.6	79.95	0.6	54.75	0.4	72.95	0.5	79.45	0.6	79.55	0.6	90.55	0.7	77.65	0.6
83.6	0.6	74.5	0.6	78.9	0.6	86.15	0.6	71.45	0.5	78.15	0.6	86	0.6	81.65	0.6	76.4	0.6	67.35	0.5	85.05	0.6	68.55	0.5
81.1	0.6	86	0.6	74.65	0.6	87.35	0.6	66.95	0.5	80.2	0.6	87.65	0.6	82.1	0.6	71.7	0.5	84.75	0.6	87.85	0.7	79.05	0.6
75.4	0.6	82.65	0.6	74.3	0.6	89.8	0.7	69.45	0.5	87.5	0.6	90.3	0.7	81.35	0.6	74.25	0.6	79.45	0.6	80.95	0.6	85.2	0.6
76.9	0.6	84.65	0.6	77.4	0.6	89.35	0.7	79	0.6	62.75	0.5	82.05	0.6	80.85	0.6	78.45	0.6	78.2	0.6	74.1	0.5	86.15	0.6
81.05	0.6	82.55	0.6	80.45	0.6	84.8	0.6	74.2	0.5	67.5	0.5	83.5	0.6	84.75	0.6	75.85	0.6	73	0.5	55.4	0.4	85.65	0.6
84.65	0.6	79.55	0.6	77.05	0.6	78.45	0.6	67.4	0.5	77.65	0.6	78.15	0.6	83.25	0.6	67.8	0.5	72.3	0.5	82.4	0.6	81.35	0.6
82.9	0.6	82.75	0.6	77.1	0.6	83.85	0.6	77.5	0.6	90.5	0.7	53.65	0.4	75.75	0.6	71.8	0.5	76.25	0.6	86.7	0.6	83.45	0.6
80.05	0.6	78.2	0.6	74.95	0.6	89.15	0.7	75.1	0.6	95.5	0.7	83	0.6	80.15	0.6	78.95	0.6	62.7	0.5	82.55	0.6	81.1	0.6

speed (km/h)	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value	speed (km	C-value
83.95	0.6	82.35	0.6	95.55	0.7	74.3	0.6	92.15	0.7	73.85	0.5	87.5	0.6	82.85	0.6	82.4	0.6	93	0.7	75.1	0.6	88.5	0.7
90.25	0.7	75.5	0.6	90.6	0.7	86	0.6	88.65	0.7	71.3	0.5	86.45	0.6	84.7	0.6	78.65	0.6	86.95	0.6	58.75	0.4	86.15	0.6
90.3	0.7	77.25	0.6	83.75	0.6	77.5	0.6	86	0.6	84.1	0.6	79.45	0.6	81.6	0.6	61.95	0.5	87.35	0.6	74.2	0.5	82.35	0.6
86.4	0.6	79.3	0.6	83.6	0.6	80.8	0.6	77.25	0.6	82.1	0.6	81.5	0.6	76.8	0.6	81.6	0.6	87.65	0.6	84.05	0.6	75.55	0.6
92.25	0.7	84.7	0.6	73.05	0.5	76.4	0.6	88.5	0.7	84.3	0.6	77.4	0.6	81.45	0.6	95	0.7	76.55	0.6	81.55	0.6	77.4	0.6
88.2	0.7	83.65	0.6	86.2	0.6	73.5	0.5	84.4	0.6	82.8	0.6	85.7	0.6	81.4	0.6	85.95	0.6	75.85	0.6	83.8	0.6	75.25	0.6
86.7	0.6	85.1	0.6	80.6	0.6	76	0.6	87.45	0.6	78.45	0.6	86.45	0.6	69.4	0.5	83.75	0.6	78.8	0.6	85.8	0.6	88.7	0.7
88.75	0.7	86.15	0.6	82.9	0.6	76.35	0.6	87.3	0.6	67.3	0.5	89.6	0.7	76.15	0.6	82.8	0.6	89	0.7	77.45	0.6	86.55	0.6
95.8	0.7	86.1	0.6	83.3	0.6	66.25	0.5	85.65	0.6	75.15	0.6	87.35	0.6	85.5	0.6	73.3	0.5	86.5	0.6	60	0.4	84.05	0.6
92.05	0.7	80.15	0.6	81.9	0.6	71.35	0.5	88.65	0.7	93.25	0.7	74.95	0.6	85.8	0.6	82	0.6	84.65	0.6	68.9	0.5	86.85	0.6
88.15	0.7	78.3	0.6	87.5	0.6	80.5	0.6	91.1	0.7	89.35	0.7	80.4	0.6	86.95	0.6	77.5	0.6	85.5	0.6	89.25	0.7	76.15	0.6
80.8	0.6	89.9	0.7	81.25	0.6	79.5	0.6	81.1	0.6	92.85	0.7	79.55	0.6	88.45	0.7	79.65	0.6	90.35	0.7	86.05	0.6	75	0.6
70.9	0.5	88.9	0.7	64.95	0.5	83.95	0.6	87	0.6	85.3	0.6	80.75	0.6	82.8	0.6	79.95	0.6	66.6	0.5	87.35	0.6	79.5	0.6
91.2	0.7	86.05	0.6	84.85	0.6	79.45	0.6	94.7	0.7	81.15	0.6	77.3	0.6	69.55	0.5	79	0.6	87.1	0.6	89.9	0.7	86.6	0.6
80.4	0.6	95.6	0.7	87.6	0.6	79.6	0.6	89.9	0.7	62.25	0.5	76.8	0.6	82.2	0.6	81.5	0.6	93.5	0.7	75.65	0.6	79.35	0.6
83.95	0.6	76.95	0.6	86.5	0.6	59.15	0.4	86.05	0.6	82.35	0.6	84.5	0.6	87.15	0.6	84.6	0.6	95.45	0.7	54	0.4	83	0.6
78.3	0.6	80.3	0.6	83.4	0.6	74.2	0.5	80.55	0.6	90.25	0.7	78.6	0.6	83.15	0.6	64.25	0.5	93.5	0.7	80.45	0.6	82.45	0.6
80.1	0.6	89.4	0.7	87.4	0.6	85.5	0.6	72.05	0.5	87.3	0.6	76.15	0.6	81.9	0.6	74.5	0.6	95.45	0.7	86.7	0.6	77.95	0.6
81.95	0.6	90.8	0.7	79.7	0.6	86.25	0.6	84.45	0.6	87.3	0.6	71.5	0.5	82	0.6	86.65	0.6	79.55	0.6	82.45	0.6	80.05	0.6
73.3	0.5	84.15	0.6	68.8	0.5	87.15	0.6	81.55	0.6	85.7	0.6	82.25	0.6	81.3	0.6	91.9	0.7	93.85	0.7	81.15	0.6	76.1	0.6
74.75	0.6	85.25	0.6	85.1	0.6	82.35	0.6	80.15	0.6	73.2	0.5	79	0.6	74.6	0.6	88.95	0.7	91	0.7	81.2	0.6	82.75	0.6
76.15	0.6	80.45	0.6	84.05	0.6	81.1	0.6	77.4	0.6	65.35	0.5	81.15	0.6	75.05	0.6	88.75	0.7	96.55	0.7	67.2	0.5	81.75	0.6
82.6	0.6	70.2	0.5	83.5	0.6	54.15	0.4	75.85	0.6	81.3	0.6	82.5	0.6	82.15	0.6	85.5	0.6	88.35	0.7	65.3	0.5	80.05	0.6
84.8	0.6	81.05	0.6	85.3	0.6	84	0.6	78.3	0.6	84.05	0.6	74.95	0.6	82	0.6	75.6	0.6	81.4	0.6	74.55	0.6	89.55	0.7
80.25	0.6	80.1	0.6	87.2	0.6	90.6	0.7	77.6	0.6	80.9	0.6	69.2	0.5	76.35	0.6	86.9	0.6	75.05	0.6	82.25	0.6	90.25	0.7
82.9	0.6	79.95	0.6	92.55	0.7	86.05	0.6	70.8	0.5	79.75	0.6	80.2	0.6	80.25	0.6	90.85	0.7	70.45	0.5	80.7	0.6	60	0.4
83	0.6	81.65	0.6	73.55	0.5	84.4	0.6	68.75	0.5	77.5	0.6	87.75	0.7	74.1	0.5	89.2	0.7	84	0.6	80.75	0.6	78.1	0.6
63.35	0.5	80.1	0.6	81.7	0.6	88	0.7	81.4	0.6	72.5	0.5	85.15	0.6	70.75	0.5	88.35	0.7	70.3	0.5	80.25	0.6	83.95	0.6
79.65	0.6	81.55	0.6	95.45	0.7	78.35	0.6	78.55	0.6	47.85	0.4	92.05	0.7	71.65	0.5	88.35	0.7	74.7	0.6	80.9	0.6	83.4	0.6
85.2	0.6	79.45	0.6	91.15	0.7	66.3	0.5	75.45	0.6	76.35	0.6	94.9	0.7	83.9	0.6	84.6	0.6	75.65	0.6	54.2	0.4	88.35	0.7
89.3	0.7	77.25	0.6	88.35	0.7	81.65	0.6	76.8	0.6	81.95	0.6	91.4	0.7	78.95	0.6	80.5	0.6	75.5	0.6	74.35	0.6	87.55	0.6
88.4	0.7	80.35	0.6	86.3	0.6	86.25	0.6	72.05	0.5	82.85	0.6	65.35	0.5	80.45	0.6	81.2	0.6	82.4	0.6	91.25	0.7	79.6	0.6
88.8	0.7	85.35	0.6	77.55	0.6	84.55	0.6	58.85	0.4	82.8	0.6	95.7	0.7	83.7	0.6	90	0.7	69.85	0.5	85.9	0.6	76.5	0.6
74.6	0.6	83.05	0.6	91.4	0.7	82.25	0.6	79.35	0.6	88.85	0.7	96.45	0.7	87.15	0.6	90.55	0.7	64.3	0.5	91.65	0.7	75.05	0.6
50	0.4	83.8	0.6	81.9	0.6	82.95	0.6	83.1	0.6	87.8	0.7	90.65	0.7	65.75	0.5	88.15	0.7	64.3	0.5	90.9	0.7	84.7	0.6
77.45	0.6	81.1	0.6	91.45	0.7	86.05	0.6	80.15	0.6	60.25	0.4	88.25	0.7	77.2	0.6	90.15	0.7	87.9	0.7	81.9	0.6	89.45	0.7
86.4	0.6	82.8	0.6	89.3	0.7	76.2	0.6	77.2	0.6	82.55	0.6	83.35	0.6	90.5	0.7	76.75	0.6	83.7	0.6	65.7	0.5	92	0.7
83.05	0.6	74.1	0.5	87.15	0.6	83.7	0.6	82.6	0.6	94.5	0.7	70.55	0.5	87.95	0.7	85.8	0.6	78.6	0.6	93.6	0.7	92.1	0.7
79.9	0.6	87.4	0.6	80.65	0.6	93.85	0.7	78.6	0.6	89.65	0.7	82.6	0.6	82.55	0.6	86.7	0.6	79	0.6	92.65	0.7	84.6	0.6